HP LoadRunner

Softwareversion: 12.00

Virtual User Generator-Benutzerhandbuch

Dokument-Releasedatum: März 2014

Software Release-Releasedatum: März 2014

Rechtliche Hinweise

Garantie

Die Garantiebedingungen für Produkte und Services von HP sind in der Garantieerklärung festgelegt, die diesen Produkten und Services beiliegt. Keine der folgenden Aussagen kann als zusätzliche Garantie interpretiert werden. HP haftet nicht für technische oder redaktionelle Fehler oder Auslassungen.

Die hierin enthaltenen Informationen können ohne vorherige Ankündigung geändert werden.

Eingeschränkte Rechte

Vertrauliche Computersoftware. Gültige Lizenz von HP für den Besitz, Gebrauch oder die Anfertigung von Kopien erforderlich. Entspricht FAR 12.211 und 12.212. Kommerzielle Computersoftware, Computersoftwaredokumentation und technische Daten für kommerzielle Komponenten werden an die US-Regierung per Standardlizenz lizenziert.

Copyright-Hinweis

© Copyright 1993-2014 Hewlett-Packard Development Company, L.P.

Markenhinweise

Adobe™ ist eine Marke von Adobe Systems Incorporated.

Microsoft® und Windows® sind eingetragene Marken der Microsoft Corporation in den USA.

Oracle und Java sind eingetragene Marken von Oracle und/oder der zugehörigen Tochtergesellschaften.

UNIX® ist eine eingetragene Marke von The Open Group.

Aktualisierte Dokumentation

Auf der Titelseite dieses Dokuments befinden sich die folgenden identifizierenden Informationen:

- Software-Versionsnummer, die Auskunft über die Version der Software gibt.
- Datum der Dokumentveröffentlichung, das bei jeder Änderung des Dokuments ebenfalls aktualisiert wird.
- Datum des Software-Release, das angibt, wann diese Version der Software veröffentlicht wurde.

Unter der unten angegebenen Internetadresse können Sie überprüfen, ob neue Updates verfügbar sind, und sicherstellen, dass Sie mit der neuesten Version eines Dokuments arbeiten: **http://h20230.www2.hp.com/selfsolve/manuals**

Für die Anmeldung an dieser Website benötigen Sie einen HP Passport. Hier können Sie sich für eine HP Passport-ID registrieren: **http://h20229.www2.hp.com/passport-registration.html**

Alternativ können Sie auf den Link **New user registration** (Neue Benutzer registrieren) auf der HP Passport-Anmeldeseite klicken.

Wenn Sie sich beim Support-Service eines bestimmten Produkts registrieren, erhalten Sie ebenfalls aktualisierte Softwareversionen und überarbeitete Ausgaben der zugehörigen Dokumente. Weitere Informationen erhalten Sie bei Ihrem HP-Kundenbetreuer.

Support

Besuchen Sie die HP Software Support Online-Website von HP unter: **http://www.hp.com/go/hpsoftwaresupport**

Auf dieser Website finden Sie Kontaktinformationen und Details zu Produkten, Services und Support-Leistungen von HP Software.

Der Online-Support von HP Software bietet Kunden mit Hilfe interaktiver technischer Support-Werkzeuge die Möglichkeit, ihre Probleme intern zu lösen. Als Valued Support Customer können Sie die Support-Website für folgende Aufgaben nutzen:

- Suchen nach interessanten Wissensdokumenten
- Absenden und Verfolgen von Support-Fällen und Erweiterungsanforderungen
- Herunterladen von Software-Patches
- Verwalten von Support-Verträgen
- Nachschlagen von HP-Support-Kontakten
- Einsehen von Informationen über verfügbare Services
- Führen von Diskussionen mit anderen Softwarekunden
- Suchen und Registrieren für Softwareschulungen

Für die meisten Support-Bereiche müssen Sie sich als Benutzer mit einem HP Passport registrieren und anmelden. In vielen Fällen ist zudem ein Support-Vertrag erforderlich. Hier können Sie sich für eine HP Passport-ID registrieren:

http://h20229.www2.hp.com/passport-registration.html

Weitere Informationen zu Zugriffsebenen finden Sie unter:

http://h20230.www2.hp.com/new_access_levels.jsp

HP Software Solutions Now greift auf die Website von HPSW Solution and Integration Portal zu. Auf dieser Website finden Sie HP-Produktlösungen für Ihre Unternehmensanforderungen, einschließlich einer Liste aller Integrationsmöglichkeiten zwischen HP-Produkten sowie eine Aufstellung der ITIL-Prozesse. Der URL dieser Website lautet **http://h20230.www2.hp.com/sc/solutions/index.jsp**

Info über die PDF-Version der Online-Hilfe

Bei diesem Dokument handelt es sich um eine PDF-Version der Online-Hilfe. Diese PDF-Datei wird bereitgestellt, um Ihnen das Drucken mehrerer Themen der Hilfe oder das Lesen der Online-Hilfe im PDF-Format zu ermöglichen. Da der Inhalt der Datei ursprünglich zur Anzeige als Online-Hilfe bzw. zur Darstellung in einem Webbrowser erstellt wurde, sind einige Themen möglicherweise nicht ordnungsgemäß formatiert. Einige interaktive Themen sind möglicherweise nicht in der PDF-Version enthalten. Diese Themen können jedoch problemlos direkt aus der Online-Hilfe gedruckt werden.

Inhalt

Virtual User Generator-Benutzerhandbuch 1
Inhalt 6
Willkommen beim Benutzerhandbuch zu Virtual User Generator 33
LoadRunner-Online-Hilfe 33
Weitere Onlineressourcen 33
Neue Funktionen in LoadRunner 12.00 35
VuGen-Konzepte 39
Übersicht über VuGen 39
Übersicht über die Vuser-Technologie 39
Vuser-Typen 40
Vuser-Protokolle 41
Tastenkombinationen 45
IPv6-Unterstützung 48
VuGen-Benutzeroberfläche 50
VuGen-Benutzeroberfläche – Übersicht 50
VuGen-Layouts - Übersicht 56
Lösungs-Explorer - Übersicht 57
Editor - Übersicht 59
Snapshot-Bereich - Übersicht 62
Arbeiten mit Snapshots 65
Hinzufügen einer Textprüfung aus der XML-Ansicht im Snapshot-Bereich 68
Erstellen eines Vuser-Skripts - Workflow 68
Vergleichen von Skripts nebeneinander 69
Erstellen eines Geschäftsprozessberichts 70
Ändern des VuGen-Layouts 71
Optionen - Übersicht 73
Allgemeine Optionen 75
Editor-Optionen 77
Skriptoptionen 83

Dialogfelder "Suchen" und "Ersetzen" 90
Editor 93
Miniaturansicht-Explorer 95
Dialogfeld "Geschäftsprozessbericht" 97
Bereich "Schrittetoolbox" 99
Bereich "Lösungs-Explorer" 101
Ausgabebereich 107
Snapshot-Bereich 109
Bereich "Textmarken" 114
Bereich "Schrittenavigator" 115
Bereich "Fehler" 116
Aufgabenbereich 118
Bereich "Haltepunkte" 119
Überwachungsbereich 122
Bereich "Laufzeitdaten" 123
Bereich "Aufrufliste" 124
Registerkarte "Wiedergabeübersicht" 124
VuGen - Fehlerbehebung und Einschränkungen 132
Übersicht über den Protocol Advisor 133
Verwenden des Protocol Advisor 133
Dialogfeld "Protocol Advisor" 135
Protocol Advisor - Fehlerbeseitigung und Beschränkungen 136
HP Live Network (HPLN)-Integration 137
Herunterladen von Inhalten von HP Live Network (HPLN) in LoadRunner 138
Hochladen von Inhalten aus LoadRunner zu HP Live Network (HPLN) 140
Dialogfeld "Verbindung zu HP Live Network" 142
Dialogfeld "Von HP Live Network herunterladen" 144
Skript-Workflow 148
Erstellen oder Öffnen eines Vuser-Skripts 148
Erstellen von über Vuser-Skripts – Übersicht 148
Erstellen oder Öffnen von Vuser-Skripts 149

Arbeiten mit Application Lifecycle Management 150
Verwalten der Skripts in ALM - Übersicht 150
Versionskontrolle in ALM - Übersicht 150
Verwenden von Skripts in ALM-Projekten 150
Herstellen der Verbindung zu ALM 151
Verwenden von Skripts mit Versionskontrolle in ALM-Projekten 152
Speichern von VuGen-Vuser-Skripts in ALM-Projekten 153
Vergleichen von Vorgängerversionen eines Skripts 154
Dialogfeld "HP ALM-Verbindung" [VuGen] 154
Mehrfachprotokollskripts 156
Skriptverzeichnisdateien 157
Dialogfeld "Neues Skript erstellen" 157
Arbeiten mit .zip-Dateien 159
Vuser-Skriptvorlagen 160
Erstellen oder Öffnen von Vuser-Skriptvorlagen 161
Aufzeichnen eines Vuser-Skripts 162
Aufzeichnen – Übersicht 162
Aufzeichnen eines Vuser-Skripts 162
Aufzeichnungsoptionen 164
Knoten "Citrix > Konfiguration" 165
Citrix > Codegenerierung 165
Knoten "Citrix > Anmelden" 166
Knoten "Citrix > Recorder" 168
Knoten "COM/DCOM > Filter" 169
COM/DCOM > Optionen (Knoten) 173
Korrelationen > Konfiguration 175
Korrelationen > Regeln 177
Bereich "Neue Regel" 178
Dialogfeld "Erweiterte Korrelationseigenschaften" 180
Dialogfeld "Testumgebung für Zeichenfolgenersetzung" 181
Datenbank > Datenbank (Knoten) 182

Datenbank > Dialogfeld "Erweiterte Aufzeichnungsoptionen" 182
Datenformaterweiterung > Kettenkonfiguration (Knoten) 184
Dialogfeld "Präfix/Postfix zu Kette hinzufügen" 186
Datenformaterweiterung hinzufügen 186
Datenformaterweiterung > Codegenerierung (Knoten) 188
Flex > RTMP (Knoten) 189
Flex > Konfiguration (Knoten) (Aufzeichnen) 190
Flex > Externalisierbare Objekte (Knoten) (Aufzeichnen) 191
Allgemein > Codegenerierung 192
Allgemein > Protokolle (Knoten) 193
Allgemein > Aufzeichnen (Knoten) 194
Dialogfeld "Erweiterter URL" 194
Dialogfeld "Erweitertes HTML" 195
Allgemein > Skript (Knoten) 196
GUI-Eigenschaften > Webereigniskonfiguration (Knoten) 199
Dialogfeld "Benutzerdefinierte Konfiguration der Aufzeichnung von Webereignissen" 200
HTTP > Erweitert (Knoten) 202
Dialogfeld "Kopfzeilen" 206
Dialogfeld "Inhaltstypenfilter" 207
Dialogfeld "Nicht-Ressourcen" 208
Flex > Konfiguration (Knoten) (Aufzeichnen) 209
Flex > Externalisierbare Objekte (Knoten) (Aufzeichnen) 209
.NET > Filter (Knoten) 211
Dialogfeld "Neuen Filter erstellen" [.NET Protocol] 212
Filter-Manager [.NET Protocol] 212
Dialogfeld "Verweis hinzufügen" [.NET Protocol] 216
GUI-Eigenschaften > Erweitert (Knoten) 217
.NET > Aufzeichnen - Knoten "Aufzeichnen" 219
Eigenschaft "Remoteobjekte" 221
.NET > Gemeinsam genutzte DLLs 222
Eigenschaften von Mobile TruClient > Knoten "Mobilgerät" 223

Netzwerk > Portzuordnung (Knoten) 224
Dialogfeld "Servereintrag" 226
Dialogfeld "Erweiterte Einstellungen für die Portzuordnung" 228
RDP > Codegenerierung > Erweitert (Knoten) 229
RDP > Codegenerierung > Agent (Knoten) 230
RDP > Codegenerierung > Grundlegend (Knoten) 231
RDP > Clientstart (Knoten) 232
Aufzeichnungseigenschaften > Corba-Optionen (Knoten) 233
Aufzeichnungseigenschaften > Korrelationsoptionen (Knoten) 234
Aufzeichnungseigenschaften > Protokolloptionen (Knoten) 235
Aufzeichnungseigenschaften > Recorder-Optionen (Knoten) 236
Aufzeichnungseigenschaften > Knoten "Serialisierungsoptionen" 238
RTE > Konfiguration (Knoten) 239
RTE > RTE (Knoten) 240
SAPGUI > Automatisch anmelden (Knoten) 241
SAPGUI > Codegenerierung (Knoten) 242
SAPGUI > Knoten "Allgemein" 242
Knoten "Silverlight > Dienste" 243
Dialogfeld "Dienst hinzufügen/bearbeiten" 244
Dialogfeld "Verbindungseinstellungen" 245
Dialogfeld "Daten für das Szenario zu Protokoll und Sicherheit" 245
Datenverkehrsanalyse > Filter für Datenverkehr 247
Knoten "WinSock" 248
Aufzeichnungsoptionen – Sonstige Themen 250
Tabelle für Protokollkompatibilität 250
Übersicht über die Portzuordnung 253
Automatische Erkennung der Portzuordnung 254
EUC-Codierung (nur japanisches Windows) 255
Übersicht über die benutzerdefinierten Einstellungen für die Skriptgenerierung 256
Skriptsprachenoptionen 256
Übersicht über Aufzeichnungsebenen 256

Übersicht über die Serialisierung ... 258
Tipps zum Arbeiten mit der Ereignisabhörung und -aufzeichnung ... 258
Beispiel für Click and Script bei einer Aufzeichnung außerhalb des Kontexts ... 259
Abschnitte des Vuser-Skripts ... 260
Bereitstellen von Authentifizierungsinformationen ... 263
Aufzeichnung über einen Proxy - Übersicht ... 264
Aufzeichnen eines Skripts über einen Proxy ... 266
Importieren von Aktionen in ein Skript ... 270
Erneutes Generieren eines Vuser-Skripts ... 271
Dialogfeld "Aufzeichnung starten" ... 272
Freischwebende Aufzeichnungssymbolleiste ... 276
Bei der Aufzeichnung erstellte Dateien ... 278
Fehlerbehebung und Einschränkungen - Aufzeichnung ... 279
Correlation Studio ... 281
Korrelieren – Übersicht ... 281
Registerkarte "Korrelation" [Design Studio] - Übersicht ... 282
Bestimmen der zu korrelierenden Werte ... 284
Ändern von gespeicherten Parametern ... 285
Korrelation und Parametrierung im Vergleich ... 285
Dienstprogramm für die Wdiff-Korrelation ... 286
Korrelieren von Java-Skripts ... 286
Korrelieren von Skripts mit Design Studio ... 288
Manuelles Korrelieren von Skripts ... 290
Korrelieren von Skripts aus einem Snapshot ... 290
Korrelieren von Winsock-Skripts ... 291
Korrelieren von Skripts - Winsock (Manuell) ... 293
Korrelieren von Skripts - Web (Manuell) ... 294
Korrelieren von Skripts - Siebel-Protokoll ... 296
Korrelieren von Skripts - Oracle NCA ... 301
Korrelieren von Skripts - Microsoft .NET ... 303
Korrelieren von Skripts - Java-Skripts - Serialisierung ... 305

Korrelieren von Skripts - Java 308
Korrelieren von Skripts - XPath-Korrelation in Flex Vuser-Skripts 310
Korrelieren von Skripts - COM-Protokoll 311
Suchen nach Werten, für die eine Korrelation erforderlich ist 312
Ändern von Korrelationsdefinitionen 313
Ausschließen von Inhalten anhand von "Content-Type" 319
Ausschließen von Zeichenfolgen aus der Korrelationsprüfung 320
Korrelationsfunktionen - Datenbank-Vuser-Skripts 321
Korrelationsfunktionen - Java-Vuser-Skripts 321
Web_reg_save_param - Funktionsdetails 322
Korrelationsfunktionen - C-Vuser-Skripts 324
Dialogfeld "Design Studio" [Registerkarte "Korrelation"] 325
Wiedergeben eines Vuser-Skripts 328
Übersicht über die Wiedergabe 329
Wiedergeben eines Vuser-Skripts 329
Laufzeiteinstellungen 330
Laufzeiteinstellungen – Sonstige Themen 331
Inhaltsprüfung – Übersicht 331
Kopieren, Importieren und Exportieren von Laufzeiteinstellungen 331
Multithreading 333
Tabelle für Protokollkompatibilität 334
Browser > Browseremulation (Knoten) 338
Knoten "Citrix > Konfiguration" 341
Knoten "Citrix > Synchronisierung" 342
Knoten "COM/DCOM - Filter" 343
COM/DCOM > Optionen (Knoten) 346
Datenformaterweiterung > Knoten "Konfiguration" 347
Flex > Konfiguration (Knoten) (Laufzeiteinstellungen) 348
Flex > RTMP (Knoten) 348
Flex > Externalisierbar (Knoten) (Laufzeiteinstellungen) 349
Allgemein > Zusätzliche Attribute (Knoten) 350

Allgemein > Protokoll (Knoten) 351
Allgemein > Knoten "Protokoll" (Ajax TruClient) 352
Allgemein > Erweiterte Einstellungen (TruClient) 354
Last > Knoten "Browsereinstellungen" (Ajax TruClient) 355
Last > Sonstige Einstellungen (TruClient) 359
Allgemein > Knoten "Wiedergeben" 360
Allgemein > Verschiedenes (Knoten) 366
Allgemein > Takt (Knoten) 367
Allgemein > Ausführungslogik (Knoten) 368
Allgemein > Latenzzeit (Knoten) 368
Internetprotokoll > ContentCheck (Knoten) 369
Internetprotokoll > Download-Filter (Knoten) 370
Internetprotokoll > Einstellungen (Knoten) 371
Dialogfeld "Erweiterte Einstellungen" 373
Internetprotokoll > Proxy (Knoten) 385
Java-Umgebungseinstellungen > Java - Classpath (Knoten) 386
Java-Umgebungseinstellungen > Java VM (Knoten) 387
JMS > Erweitert (Knoten) 388
MMS > Server und Protokoll (Knoten) 390
Eigenschaften von Mobile TruClient > Knoten "Mobilgerät" 391
.NET > .NET-Umgebung (Knoten) 392
.NET > Gemeinsam genutzte DLLs 393
Netzwerk > Geschwindigkeitssimulation (Knoten) 394
Oracle NCA > Clientemulation (Knoten) 395
RDP > Erweitert (Strukturknoten) 396
RDP > Konfiguration (Knoten) 397
RDP > RDP-Agent (Knoten) 398
RDP > Synchronisierung (Knoten) 399
RTE > RTE (Knoten) 400
SAPGUI > Knoten "Allgemein" 402
Silverlight > Dienste (Knoten) 403

WAP > Gateway (Strukturknoten) 403
WAP > Radius (Strukturknoten) 406
Textmarken - Übersicht 407
Ausführen eines Vuser-Skripts über eine Befehlszeile 407
Ausführen eines Vuser-Skripts über eine Linux-Befehlszeile 408
Verwenden von Textmarken 410
Bei der Wiedergabe erstellte Dateien 411
Debuggen von Vuser-Skripts 413
Debuggen – Übersicht 413
Fehlerbehandlung 416
Zusätzliche Debugginginformationen 417
Arbeiten mit Haltepunkten 418
Beobachten von Ausdrücken und Variablen 419
Debuggen von Web-Vuser-Skripts 420
Debuggen von Skripts mit Haltepunkten 421
Erweitern eines Skripts für Lasttests 422
Erweitern eines Skripts für Lasttests – Übersicht 422
Übersicht über Transaktionen 425
Einfügen von Transaktionen 425
Anzeigen von Transaktionen 427
Rendezvous-Punkte 427
Übersicht über das Hinzufügen von VuGen-Funktionen 428
Allgemeine Vuser-Funktionen 430
Protokollspezifische Vuser-Funktionen 430
Verschlüsseln von Text 431
Codieren von Kennwörtern 431
Verschlüsseln von Text 432
Verschlüsseln/Entschlüsseln von Text 432
Verschlüsseln eines Kennworts 432
Dialogfeld "Password Encoder" 433
Erstellen eines Controller-Szenarios über VuGen 434

Einfügen von Schritten in ein Skript ... 434
Dialogfeld "Controller-Szenario erstellen" ... 435
Parameter ... 436
Parametrierung – Übersicht ... 436
Parametertypen ... 439
Erstellen eines Parameters ... 441
Datenzuweisungsmethoden für die Parametertypen "Datei/Tabelle/XML" ... 442
Datenzuweisungs- und -aktualisierungsmethoden für die Parametertypen "Datei", "Tabelle" und "XML" ... 443
Vuser-Verhalten im Controller (nur LoadRunner) ... 445
Verwenden von bestehenden Parametern ... 446
XML-Parameter ... 446
Erstellen eines XML-Parameters aus einem Webdienstaufruf ... 447
Erstellen von XML-Parametern - Standardmethode ... 448
Definieren von XML-Wertesätzen ... 448
Festlegen einer Zuweisungsmethode ... 452
Ändern der Eigenschaften von XML-Parametern ... 453
Festlegen von AUT-Umgebungsparametern ... 454
Importieren von Parameterdaten aus einer Datenbank ... 454
Dialogfeld "Parameter auswählen oder erstellen" ... 455
Dialogfeld "Parametereigenschaften" ... 456
Dialogfeld "Parametersimulation" ... 466
Dialogfeld "Parameterliste" ... 470
Datenbankabfrage-Assistent ... 471
Dialogfeld "Parameter erstellen" (Winsock) ... 472
Dialogfeld "Ursprünglicher Wert des Parameters" ... 472
Parameter - Fehlerbehebung und Einschränkungen ... 473
Async Studio ... 474
Synchrone und asynchrone Konzepte ... 474
Typen asynchroner Kommunikation ... 475
LoadRunner-Support für asynchrone Kommunikation - Übersicht ... 478
Erstellen eines asynchronen Vuser-Skripts ... 480

API für asynchrone Kommunikation ... 482
Unterschiede zwischen asynchronen Funktionen und synchronen Funktionen ... 483
Änderung eines Vuser-Skripts durch VuGen für die asynchrone Kommunikation ... 484
Definieren des Anfangs einer asynchronen Unterhaltung ... 487
Definieren des Endes einer asynchronen Unterhaltung ... 488
Verwenden von Schwellenwerten für asynchrone Anforderungen ... 490
Feinabstimmung des Endes einer asynchronen Unterhaltung ... 491
Korrelieren asynchroner Vuser-Skripts ... 492
Implementieren von Callbacks ... 492
Ändern von Callbacks ... 495
Parsen von URLs ... 499
Regeln für die asynchrone Kommunikation - Übersicht ... 501
Hinzufügen von Regeln für die asynchrone Kommunikation ... 502
Registerkarte "Asynchron" [Design Studio] ... 504
Dialogfeld "Asynchron-Optionen" ... 506
Beispiel für asynchrone Kommunikation - Poll ... 507
Beispiel für asynchrone Kommunikation - Push ... 510
Beispiel für asynchrone Kommunikation - Long Poll ... 511
Anzeigen von Wiedergabeergebnissen ... 513
Anzeigen von Wiedergabeergebnissen – Übersicht ... 513
Anpassen der Anzeige für die Testergebnisse ... 514
Verbinden zu Application Lifecycle Management über das Fenster "Testergebnisse" ... 515
Senden von benutzerdefinierten Informationen an den Bericht ... 515
Konfigurieren der Anzeige des Fensters "Testergebnisse" ... 515
Öffnen des Fensters "Testergebnisse" einer bestimmten Ausführung ... 516
Auffinden von Schritten in den Testergebnissen ... 517
Fenster "Testergebnisse" ... 517
Dialogfeld "Filter" ... 519
Dialogfeld "Drucken" ... 521
Dialogfeld "Seitenansicht" ... 522
Dialogfeld "In HTML-Datei exportieren" ... 523

Protokolle 525
Protokollunterstützung für asynchrone, IPv6- und 64-Bit-Aufzeichnung 525
Ajax - Click & Script-Protokoll 526
Ajax (Click and Script)-Protokoll - Übersicht 527
Unterstützte Frameworks für das Ajax (Click & Script)-Protokoll 527
Ajax (Click & Script)-Beispielskript 527
Ajax (Click & Script) - Aufzeichnungstipps 528
Ajax (Click & Script) - Wiedergabetipps 530
Ajax (Click & Script) - Verschiedene Tipps 531
Click & Script - Fehlerbeseitigung und Einschränkungen 532
TruClient-Protokoll 537
Was ist TrueClient? 537
Struktur von TruClient-Schritten 540
Grundlegendes zu Schrittereignissen 544
Synchronisieren von TruClient-Skriptschritten 546
Entwickeln von TruClient-Skripts 549
Entwickeln von TruClient-Skripts 549
Aufzeichnen von TruClient-Skripts 551
Berechnen der Anzahl an Lastgeneratoren, die für TruClient-Skripts erforderlich sind 553
TruClient - Debuggen eines TruClient-Skripts 555
Alternative Schritte für TruClient 555
TruClient-Skriptebenen 555
TruClient-Snapshots 557
Debuggen von TruClient-Skripts 558
Weitere Tipps für das Debuggen von Skripts 560
Lösen von Problemen bei der Objekterkennung 560
Fehlerbeseitigung bei der Objekterkennung 568
Fehlerbehandlung von TruClient-Skripts 569
TruClient - Verbessern eines TruClient-Skripts 573
TruClient-Ereignishandler 573
TruClient-Funktionen und -Funktionsbibliotheken 575

Arbeiten mit JavaScript in TruClient-Skripts 575
JavaScript-Unterstützung 575
Weitere Informationen zu JavaScript 576
Verwenden regulärer Ausdrücke 576
Erweitern von TruClient-Skripts 576
Verwenden des Ereignishandlers 579
Einfügen und Ändern von Schleifen 581
Einfügen von benutzerdefiniertem JavaScript- und C-Code in TruClient-Skripts 582
Verwenden von VTS in TruClient 583
Erstellen und Verwenden von Funktionen und Bibliotheken 586
Anzeigen clientseitiger Aufschlüsselungsdaten für TruClient-Skripts 588
TruClient-Dialogfeld "Ereignishandler-Editor" 588
Clientseitige Aufschlüsselungsdaten für TruClient 591
Dialogfeld "Neue Funktion erstellen" 593
Dialogfeld 'Transaktionseditor' (TruClient) 594
TruClient-Seitenleiste 599
TruClient-Startregisterkarte 599
TruClient-Registerkarte "Bearbeiten" 603
Registerkarte "Fenster" 604
TruClient-Toolbox 605
Browser in TruClient 607
Allgemein Browsereinstellungen (TruClient) 607
Privater Modus (TruClient) 608
TruClient-Browser für IE 608
Dialogfeld "TruClient - Allgemeine Einstellungen" 609
TrueClient – Fehlerbehebung und Einschränkungen (Allgemein) 613
Fehlerbehebung bei der Kommunikation zwischen VuGen und TruClient 613
Wortverifizierungsfunktion im Geschäftsprozess 614
Protokollbeschränkungen 614
Konvertieren eines TruClient-Skripts in ein HTTP/HTML-Skript 616
LRFunktionsreferenz 617

TruClient-Funktionen 617
TruClient-VTS-Funktionen 621
TruClient-Eigenschaften 629
Argumente von TruClient-Schritten 630
Citrix-Protokoll 635
Citrix-Protokoll - Übersicht 636
Citrix-Aufzeichnungstipps 636
Citrix-Wiedergabetipps 638
Citrix-Synchronisierung 639
Citrix - Automatische Synchronisierung 640
Citrix - Alternative Synchronisierung 641
Agent für Citrix Presentation Server – Übersicht 642
Fehlerbehebung bei XenApp 5.0 646
Konfigurieren von Citrix-Client und -Server 646
Manuelles Synchronisieren von Citrix-Skripts 647
Installieren bzw. Deinstallieren des Citrix-Agenten 648
Citrix-Funktionen 649
Citrix - Informationen über ICA-Dateien 649
Dialogfeld "Fehler bei der Bitmapsynchronisierung" 651
Citrix - Fehlerbehebung und Einschränkungen 651
Click and Script-Protokolle 656
Click & Script-Protokolle – Überblick 656
Click & Script - Aufzeichnungstipps 658
Click and Script - Wiedergabetipps 659
Click & Script - Sonstige Tipps 661
Click & Script - Erweiterungen 661
Hinweise zur Click & Script-API 664
Ordnungszahlen 665
Leere Zeichenfolgen 665
Click & Script - Fehlerbeseitigung und Einschränkungen 666
COM-Protokoll 670

Übersicht über das COM-Protokoll 670
Übersicht über die COM-Technologie 671
Struktur der COM-Vuser-Skripts 672
COM-Beispiel-Vuser-Skripts 673
Auswählen der aufzuzeichnenden COM-Objekte 677
Datenbankprotokolle 679
Datenbankprotokolle – Übersicht 679
VuGen-Aufzeichnungstechnologie für Datenbanken 680
Datenbanktabellen 680
Protokoll "Flex (RTMP/AMF)" 682
Flex – Übersicht 682
Aufzeichnen von Flex-Skripts 685
AMF 685
RTMP-Funktionen 686
RTMP-Tunneled-Funktionen 687
RTMP/RTMPT-Streaming 688
RTMP Tunneled 696
Aufzeichnen eines Flex-Skripts 697
Festlegen des Flex-Aufzeichnungsmodus 698
Beispiel 698
Codegenerierung im Flex-Protokoll 699
Externalisierbare Objekte in Flex-Skripts 700
Flex-Korrelationen 703
Flex-Snapshots 703
Serialisieren von Flex-Skripts 704
Abfragen einer XML-Struktur 706
Flex - Fehlerbehebung und Einschränkungen 708
GraniteDS (Data Services) 709
Java-Protokolle 710
Übersicht über die Aufzeichnung von Java-Protokollen 710
Übersicht über Java-Vuser-Skripts 711

Übersicht über RMI over IIOP ... 712
Corba-Aufzeichnungsoptionen ... 712
Anbieterklassen für CORBA-Anwendungen ... 712
Aufzeichnen von RMI ... 713
Aufzeichnen eines Jacada-Vusers ... 713
Verwenden von CORBA ... 714
Arbeiten mit RMI ... 715
Verwenden von Jacada ... 716
Übersicht über benutzerdefinierte Java-Filter ... 717
Benutzerdefinierte Java-Filter - Ermitteln der einzuschließenden Elemente ... 718
Aufzeichnen eines Java-Vuser-Skripts ... 719
Aufzeichnen von Java-Skripts bei Verwendung von Windows XP und Windows 2000 Server ... 720
Ausführen eines Skripts als Teil eines Pakets ... 720
Manuelles Einfügen von Java-Methoden ... 721
Manuelles Konfigurieren von Skriptgenerierungseinstellungen ... 723
Erstellen eines benutzerdefinierten Java-Filters ... 725
Struktur der Hook-Datei ... 727
Referenzliste für Java-Symbole ... 729
Java-Vuser-Protokoll - Manuelles Programmieren von Skripts ... 730
Übersicht über das manuelle Programmieren von Java-Skripts ... 730
Tipps zur Programmierung des Java-Protokolls ... 731
Ausführen von Java-Vuser-Skripts ... 732
Öffnen von Java-Vuser-Skripts in Eclipse ... 733
Kompilieren und Ausführen eines Skripts als Teil eines Pakets ... 734
Manuelles Erstellen eines Java-Skripts ... 734
Erweitern eines Java-Skripts ... 737
Fehlerbehebung und Einschränkungen – Java-Protokoll] ... 741
Java über HTTP-Protokoll ... 742
Übersicht über das Java über HTTP-Protokoll ... 742
Anzeigen von Antworten und Anforderungen im XML-Format ... 742
Aufzeichnen mit Java über HTTP ... 743

Debuggen von Java über HTTP-Skripts ... 744
Einfügen von Parametern in Java über HTTP-Skripts ... 745
Java over HTTP - Fehlerbeseitigung und Beschränkungen ... 746
LDAP-Protokoll ... 747
Übersicht über das LDAP-Protokoll ... 747
Beispielskript für ein LDAP-Protokoll ... 747
Definieren von Distinguished Name-Einträgen ... 748
Angeben von LDAP-Verbindungsoptionen ... 749
Fehlerbehebung und Einschränkungen – LDAP ... 751
Mailingdienstprotokolle ... 751
Mailingdienstprotokolle - Übersicht ... 751
IMAP-Protokoll - Übersicht ... 751
MAPI-Protokoll - Übersicht ... 752
POP3-Protokoll - Übersicht ... 753
SMTP-Protokoll - Übersicht ... 753
Protokolle für mobile Anwendungen ... 754
Protokolle für mobile Anwendungen - Übersicht ... 754
Protokolloptionen für Mobilgeräte ... 754
Auswählen einer Aufzeichnungsmethode für mobile Anwendungen ... 755
Netzwerk – Geschwindigkeitssimulation ... 757
Netzwerk > Geschwindigkeitssimulation (Knoten) ... 758
Mobile Application - HTTP/HTML ... 759
Aufzeichnungsmethoden ... 759
Aufzeichnen von Datenverkehr in eine Aufzeichnungsdatei (Sniffer) ... 760
Aufzeichnen von Datenverkehr mit dem mobilen Sniffer-Agenten von VuGen ... 761
Aufzeichnen von Datenverkehr mit einem externen Sniffer-Tool ... 764
Analysieren von Datenverkehr ... 765
Aufzeichnen mit Emulation ... 766
Erstellen eines Skripts durch Analysieren des Datenverkehrs ... 771
Aufzeichnen und Analysieren eines Skripts für mobile Anwendungen ... 774
Aufzeichnen mit Mobile Recorder ... 774

Aufzeichnungs-Assistent ... 776
Dialogfeld "Aufzeichnungsmethode" ... 777
Dialogfeld "Datenverkehr analysieren" ... 778
Dialogfeld "Konfigurieren und Aufzeichnen" ... 779
Dialogfeld "Aufzeichnungsemulation" ... 779
Dialogfeld "Proxy-Aufzeichnung" ... 780
LoadRunner Mobile Recording-Anwendung - Benutzeroberfläche ... 781
Mobile TruClient ... 783
Mobile TruClient-Protokoll ... 783
Aufzeichnen eines Skripts mit Mobile TruClient ... 783
Hinzufügen, Entfernen und Importieren von Mobilgeräteeinstellungen für Mobile TruClient ... 783
Dialogfeld "TruClient-Mobilgeräte-Manager" ... 784
SAP Mobile Platform ... 785
Erstellen eines SMP (SAP Mobile Platform)-Skripts ... 785
.NET-Protokoll ... 786
Übersicht über das .NET-Protokoll ... 786
Hinweise zum Arbeiten mit dem .NET-Protokoll ... 786
Anzeigen von Daten und Datentabellen ... 787
Aufzeichnen von WCF-Duplexkommunikation ... 788
Ersetzen des Callbacks im Skript ... 791
Asynchrone Aufrufe ... 792
Aufzeichnen von dualen HTTP-Bindungen ... 793
Verbindungspooling ... 794
Debuggen von .NET-Vuser-Skripts ... 795
Übersicht über .NET-Filter ... 796
.NET-Filter - Weiterführende Themen ... 797
Richtlinien für das Festlegen von .NET-Filtern ... 798
Konfigurieren von Anwendungssicherheit und Berechtigungen ... 801
.NET - Fehlerbehebung und Einschränkungen ... 803
Oracle NCA-Protokoll ... 804
Übersicht über das Oracle NCA-Protokoll ... 804

Beispielskripts für das Oracle NCA-Protokoll ... 805
Tipps zum Aufzeichnen und Wiedergeben mit Oracle NCA ... 806
Pragma-Modus ... 807
Aktivieren der Aufzeichnung von Objekten nach Name ... 808
Starten von Oracle-Anwendungen über die persönliche Startseite ... 810
Oracle NCA - Fehlerbehebung und Einschränkungen ... 811
RDP-Protokoll ... 813
RDP-Protokoll – Übersicht ... 814
RDP-Aufzeichnungstipps ... 814
Arbeiten mit Daten in der Zwischenablage (RDP-Protokoll) ... 816
Korrelieren von Zwischenablageparametern ... 818
RDP-Snapshots - Übersicht ... 818
Übersicht über die Bildsynchronisierung ... 819
Tipps zur Bildsynchronisierung (RDP-Protokoll) ... 820
Bildsynchronisierung - Geänderte Koordinaten (RDP-Protokoll) ... 821
Festlegen der Sicherheitsebenen in RDP-Vuser-Skripts ... 822
RDP-Agent (Agent für Microsoft Terminal Server) – Übersicht ... 826
Installieren/Deinstallieren des RDP-Agenten ... 828
Hinzufügen von Bildsynchronisierungspunkten zu einem Skript ... 829
Dialogfeld "Fehler bei Bildsynchronisierung" (RDP-Protokoll) ... 829
RDP - Fehlerbehebung und Einschränkungen ... 831
RTE-Protokoll ... 833
Übersicht über das RTE-Protokoll ... 833
Arbeiten mit der Ericom-Terminalemulation ... 834
SSL- und SSH-Unterstützung für Ericom ... 834
Zeicheneingabe in einem Terminalemulator ... 835
Festlegen des Zeitüberschreitungswerts für "TE_type" ... 836
Zulassen von Vorauseingaben ... 836
Generieren von eindeutigen Gerätenamen ... 837
Festlegen der Feldbegrenzungszeichen ... 838
Abrufen von Text vom Terminalbildschirm ... 839

Übersicht über die RTE-Synchronisierung 840
Synchronisieren von Block-Mode-Terminals (IBM) 841
Synchronisieren von Character-Mode-Terminals (VT) 844
Zuordnen von Terminal- und PC-Tastatur 847
Aufzeichnen von RTE-Vuser-Skripts 848
Implementieren der Funktion "Bei Fehler fortfahren" 851
RTE - Terminalemulation - Fehlerbehebung und Einschränkungen 851
SAP-Protokolle 851
Auswählen eines SAP-Protokolltyps 851
SAP GUI-Protokoll 852
SAP-Web-Protokoll 855
SAP (Click & Script)-Protokoll 856
Wiedergabe von optionalen Fenstern mit SAP GUI-Skripts 857
Konfigurieren der SAP-Umgebung 858
Aufzeichnen von SAP GUI-Skripts 863
Wiedergeben von SAP GUI-Skripts 865
Ausführen von SAP GUI-Skripts in einem Szenario 866
Erweitern von SAP GUI-Skripts 867
Weitere SAP-Ressourcen 872
SAP GUI, SAP Web und SAP (Click & Script) - Fehlerbehebung und Einschränkungen 873
Siebel-Webprotokoll 875
Übersicht über das Siebel-Webprotokoll 875
Siebel-Webaufzeichnungsoptionen und -Laufzeiteinstellungen 875
Aufzeichnen von Transaktionsaufschlüsselungsdaten 876
Siebel-Web - Fehlerbeseitigung und Beschränkungen 877
Silverlight-Protokoll 879
Silverlight-Protokoll - Übersicht 879
Importieren von WSDL-Dateien 879
Silverlight - Fehlerbehebung und Einschränkungen 880
Webprotokoll (HTTP/HTML) 881
Webprotokoll (HTTP/HTML) – Übersicht 881

Verwenden des JavaScript-Moduls von LoadRunner 885
JavaScript-Modul: Beispiel für XMLHTTPRequest 891
Konvertieren eines Web (HTTP/HTML)-Vuser-Skripts in ein Java-Vuser-Skript 892
Erstellen eines Skripts für eine REST-API 893
Beispiele 894
Aufzeichnen des SPDY-Protokolls 895
Anwendungen, die mit Smooth Streaming arbeiten 895
Konvertieren eines TruClient-Skript in ein Web HTTP/HTML-Skript 897
Fehlerbehebung und Einschränkungen – Webprotokoll (HTTP/HTML) 898
Webprotokolle (Allgemein) 898
Webprotokolle – Übersicht 898
Web Vuser-Technologie 899
Web-Vuser-Typen 900
Überprüfen von Text und Bildern (Web-Vuser-Skripts) – Übersicht 901
Grundlegendes zu den Funktionen für die Webtextprüfung 902
Hinzufügen von Text- und Bildprüfungen (Web-Vuser-Protokolle) 903
Web-Snapshots - Übersicht 904
Browseremulation - Übersicht 905
Durchführen von Lasttests mit nCipher-HSM 909
Arbeiten mit Daten im Cache 910
Einfügen von Zwischenspeicherfunktionen 912
Datenformaterweiterungen (DFEs) - Übersicht 912
Implementieren der Unterstützung für Datenformaterweiterungen (DFEs) 916
Definieren einer DFE-Kette 916
Aktivieren der DFE-Unterstützung 918
Konfigurieren der DFE-Unterstützung 919
Anwenden von DFE-Ketten auf Abschnitte der HTTP-Nachricht 920
So ändern DFEs ein Vuser-Skript 921
Liste der Datenformaterweiterungen 922
Anwenden von DFEs auf eine Zeichenfolge 923
Übersicht über GWT-DFE (Google Web Toolkit - Datenformaterweiterung) 924

Implementieren der GWT-DFE-Unterstützung ... 928
Fehlerbehebung - Datenformaterweiterung (DFE) ... 930
Webdienste ... 931
Webdienste - Hinzufügen von Skriptinhalt ... 931
Übersicht über Webdiensttests ... 931
Hinzufügen von Webdienst-Skriptinhalt - Übersicht ... 931
Skriptintegration ... 933
Webdienstaufruf - Anhänge ... 933
Spezielle Argumenttypen ... 934
Übersicht über Serververkehrskripts ... 937
Aufzeichnungsdateien ... 938
Filtern von Datenverkehr ... 939
Daten auf sicheren Servern ... 940
Hinzufügen von Inhalt ... 941
Zuweisen von Werten zu XML-Elementen ... 943
Automatisches Erstellen eines Tests ... 944
Erstellen eines Skripts durch Analysieren des Datenverkehrs ... 945
Erstellen von Aufzeichnungsdateien mit "lrtcpdump" ... 947
Bildschirm "Dienste angeben" ... 948
Dialogfeld "Aufzuzeichnende Anwendung angeben" ... 948
Dialogfeld "SOAP importieren" ... 949
Dialogfeld "Neuer Webdienstaufruf" ... 950
Dialogfeld "Eingabeanhang hinzufügen" ... 959
Dialogfeld "Arrayelemente hinzufügen" ... 960
Dialogfeld "Base64-Daten verarbeiten - Einfache Daten" ... 960
Dialogfeld "Base64-Daten verarbeiten - Komplexe Daten" ... 961
Aspektliste ... 963
Bildschirm "Dienste angeben" ... 964
Bildschirm "Informationen zum Datenverkehr angeben" ... 965
Dialogfeld "SSL-Konfiguration" ... 966
Webdienste - Vorbereiten von Skripts für die Wiedergabe ... 967

Übersicht über die Vorbereitung für die Wiedergabe 967
Testen von Transportschichten für Webdienste – Übersicht 967
Senden von Nachrichten über HTTP/HTTPS 967
JMS-Transport – Übersicht 968
JMS-Skriptfunktionen 969
JMS-Nachrichtenstruktur 970
Asynchrone Nachrichten – Übersicht 970
Senden von asynchronen Aufrufen über HTTP/HTTPS 970
WS-Adressierung 971
Datenbankintegration – Übersicht 972
Herstellen einer Verbindung zu einer Datenbank 973
Verwenden von Daten, die aus SQL-Abfragen abgerufen wurden 973
Validieren von Datenbankwerten nach einem Webdienstaufruf 975
Überprüfen von Rückgabewerten anhand einer Datenbank 977
Durchführen von Aktionen in Datasets 978
Anpassung – Übersicht 979
Benutzerhandler 979
Benutzerdefinierte Konfigurationsdateien 981
Beispiele für Benutzerhandler 981
Vorbereiten von Skripts für die Wiedergabe 984
Senden von Nachrichten über JMS 985
Senden von Nachrichten über HTTP/S 987
Definieren einer Testmethode 988
Hinzufügen einer Datenbankverbindung 990
Erstellen eines Benutzerhandlers 991
Anpassen von Konfigurationsdateien 994
Snapshots für Webdienste - Übersicht 994
Dialogfeld "Datenbankverbindung" 998
Dialogfeld "Generator für Verbindungszeichenfolge" 999
Webdienste - Verwalten von Diensten 999
Verwalten von Diensten – Übersicht 999

Importieren von Diensten ... 1002
Vergleichsberichte ... 1003
Webreferenzanalyse ... 1004
Hinzufügen und Verwalten von Diensten ... 1004
Analysieren von Abhängigkeiten in der WSDL-Datei ... 1005
Dialogfeld "Dienste verwalten" ... 1006
Dialogfeld "Verbindungseinstellungen" ... 1009
Dialogfeld "Dienst importieren" ... 1010
Dialogfeld "Nach einem Dienst in UDDI suchen" ... 1011
Dialogfeld "Vergleich von XML/WSDL" ... 1012
Dialogfeld "WSDL-Referenzanalyse" ... 1012
Webdienste - Sicherheit ... 1013
Konfigurieren der Sicherheitseinstellungen – Übersicht ... 1013
Sicherheitstoken und Verschlüsselung ... 1014
SAML-Sicherheitsoptionen ... 1017
Sicherheitsszenarios – Übersicht ... 1018
Auswählen eines Sicherheitsmodells ... 1019
Private, importierte und freigegebene Szenarios ... 1019
Szenariokategorien ... 1020
WCF-Szenarioeinstellungen ... 1022
Das WsHttpBinding-Szenario ... 1023
Das Verbundszenario ... 1024
Die Szenarios "Benutzerdefinierte Bindung" ... 1025
Erweiterte Szenarioeinstellungen ... 1026
WCF-Erweiterungsmöglichkeiten ... 1031
Vorbereiten von Sicherheitsszenarios für die Ausführung ... 1033
Parametrieren von Sicherheitselementen ... 1033
Schützen von benutzerdefinierten Kopfzeilen ... 1033
Emulieren von Benutzern mit Iterationen ... 1034
Hinzufügen von Sicherheitsfunktionen zu einem Webdienstskript ... 1035
Hinzufügen von SAML-Sicherheit ... 1036

Erstellen und Verwalten von Sicherheitsszenarios 1037
Parametrieren von Sicherheitselementen 1039
Dialogfeld "Sicherheitseigenschaften festlegen" 1040
Dialogfeld "Sicherheitsszenario-Editor" 1045
Dialogfeld "Zertifikat auswählen" 1046
Sicherheitsszenarios für Webdienste 1047
Fehlerbehebung und Einschränkungen – Webdienste 1049
Fehlerbehebung und Einschränkungen – Webdienste 1051
Windows-Sockets-Protokoll 1052
Übersicht über das Aufzeichnen von Windows-Sockets 1052
Konvertierungstabellen 1052
Windows-Sockets-Daten 1053
Windows-Sockets-Snapshots - Übersicht 1054
Datennavigationstools 1056
Bearbeiten von Pufferdaten 1056
Aufzeichnen eines Windows-Sockets-Skripts 1057
Anzeigen und Ändern von Windows-Socket-Puffern 1058
Datenpuffer 1061
Dialogfeld "Gehe zu Offset" 1063
MMS-Vuser-Skripts (Multimedia Messaging Service) – Übersicht 1063
Ausführen eines MMS-Szenarios im Controller 1064
Weiterführende Themen 1065
Manuelles Programmieren eines Skripts mithilfe des VuGen-Editors 1065
Übersicht über das manuelle Programmieren von Skripts 1065
C-Vuser-Skripts 1065
Java-Vuser 1067
.NET-Vuser 1067
Erstellen von Skripts in externen IDEs 1068
Erstellen von Vuser-Skripts oder LoadRunner-Tests in Visual Studio oder Eclipse 1068
Erstellen eines Vuser-Skripts in Visual Studio 1069
Erstellen eines Vuser-Skripts in Eclipse 1070

Entwickeln eines Einheitentests mit Visual Studio (NUnit-Test) 1071
Entwickeln eines Einheitentests mit Eclipse (JUnit- oder Selenium-Test) 1072
Verwenden von DLLs und Anpassen von VuGen 1073
Aufrufen von Funktionen in externen DLLs 1073
Lokales Laden einer DLL 1074
Globales Laden einer DLL 1075
Aufzeichnen von OLE-Servern 1076
Verwenden von CmdLine 1076
"CmdLine"-Umgebungsvariablen 1077
Erstellen eines neuen Vuser-Typs 1077
VuGen-Datei- und -Bibliotheksspeicherort 1081
Speichern von Laufzeiteinstellungen in externen Dateien 1082
Erstellen und Ausführen von Skripts unter Linux 1082
Erstellen und Ausführen von Skripts unter Linux - Übersicht 1082
Programmieren von Vuser-Aktionen 1083
Erstellen einer Vorlage 1084
Manuelles Konfigurieren von Laufzeiteinstellungen 1084
Manuelles Definieren von Transaktionen und Einfügen von Rendezvous-Punkten 1087
Manuelles Kompilieren von Skripts unter Linux 1088
Programmieren mit der XML API 1088
Übersicht über das Programmieren mit der XML API 1088
Verwenden von XML-Funktionen 1089
Angeben von XML-Funktionsparametern 1091
XML-Attribute 1093
Strukturieren von XML-Skripts 1093
Erweitern einer aufgezeichneten Sitzung mit XML 1094
Verwenden von Ergebnisparametern 1097
Fehlerbehebung und Einschränkungen - Programmierung 1099
Unterstützung für nicht-englische Sprachen 1101
Übersicht über die Unterstützung für nicht-englische Sprachen 1101
Sprache in der Anforderungskopfzeile 1101

Umwandeln des Codierungsformats einer Zeichenfolge 1101
Umwandeln des Codierungsformats von Parameterdateien 1102
Aufzeichnen von Webseiten mit Fremdsprachen 1104
Fremdsprachen - Fehlerbeseitigung und Beschränkungen 1104
Weitere Komponenten 1108
Komponenten von Drittanbietern 1111
Installieren des Citrix Server-Agenten 1112
Installation von VTS 1113
Installieren des Microsoft Terminal Server-Agenten 1114
Fehlerbehebung und Einschränkungen – Zusätzliche Komponenten 1115
Lernprogramm zu HP LoadRunner 1116
HP LoadRunner-Funktionsreferenz 1117
HP LoadRunner Data Format Extensions Developer Guide 1118

Willkommen beim Benutzerhandbuch zu Virtual User Generator

Willkommen beim HP Virtual User Generator, VuGen, dem HP-Tool zum Erstellen von Vuser-Skripts. Mit VuGen entwickeln Sie ein Vuser-Skript, indem Sie einen Benutzer aufzeichnen, der typische Geschäftsprozesse ausführt. Mit diesen Skripts können Sie reale Situationen emulieren.

Sie verwenden die mit VuGen erstellten Skripts in Verbindung mit anderen Produkten: HP LoadRunner, HP Performance Center und HP Business Service Management.

HP LoadRunner ist ein Tool zum Testen der Leistung von Anwendungen. Es belastet Ihre gesamte Anwendung, um potenzielle Client-, Netzwerk- und Serverengpässe identifizieren und isolieren zu können.

HP Performance Center implementiert die Funktionen von LoadRunner auf Unternehmensebene.

HP Business Service Management unterstützt Sie bei der Optimierung der Verwaltung und Verfügbarkeit von Geschäftsanwendungen und -systemen in der Produktion.

LoadRunner-Online-Hilfe

Bei der LoadRunner-Hilfe handelt es sich um ein Onlinehilfesystem, das die Verwendung von LoadRunner beschreibt. Sie können wie folgt auf die LoadRunner-Hilfe zugreifen:

- Sie können die Startseite der Online-Hilfe von LoadRunner aufrufen, indem Sie in einer der LoadRunner-Komponenten im Menü **Hilfe** HP Virtual User Generator-Hilfe, HP Controller-Hilfe oder HP Analysis-Hilfe auswählen. Dort finden Sie Direktverknüpfungen zu den wichtigsten Hilfethemen.
- Klicken Sie in einem beliebigen Fenster oder Dialogfeld auf F1, um die LoadRunner-Hilfe zu dem Thema zu öffnen, das den gerade angezeigten Bildschirm beschreibt.
- Verwenden Sie das Hilfe-Menü in LoadRunner für den Zugriff auf die Wissensdatenbank und andere nützliche Ressourcen.

Weitere Onlineressourcen

Die folgenden Onlineressourcen bieten weitere Informationen für LoadRunner Benutzer:

Ressource	URL
HP Software-Website	http/: www.hp.com/go/software
HP Software-Unterstützung	http://www.hp.com/go/hpsoftwaresupport
Fehlerbehebung und Wissensdatenbank	http://h20230.www2.hp.com/troubleshooting.jsp.

Ressource	URL
LoadRunner Community-Foren	http://www.hp.com/go/lrpc
LoadRunner-Blog	http://h30499.www3.hp.com/t5/HP-LoadRunner-and-Performance/bg-p/sws-585
HP Live Network (HPLN)	https://hpln.hp.com/group/performance-center-and-loadrunner
LoadRunner-Integrationen	http://support.openview.hp.com/sc/solutions/index.jsp#tab=tab1
LoadRunner auf Twitter	https://twitter.com/hploadrunner
LoadRunner auf Linkedin	http://www.linkedin.com/groups?home=&gid=1879289

Neue Funktionen in LoadRunner 12.00

Möglichkeit zur Ausführung von Lastgeneratoren in der Cloud

- Einfache Bereitstellung von Cloud-Lastgeneratoren über den Controller. Weitere Informationen finden Sie unter "Hinzufügen von cloud-basierten Lastgeneratoren – Überblick".
- Ausführen von Lastgeneratoren über die Amazon-Cloud

Verbesserte Kommunikationsschnittstelle zwischen Controller und Lastgeneratoren

- Möglichkeit zum Konfigurieren des Ports zwischen allen LoadRunner-Komponenten: Controller, Lastgenerator, MI-Listener sowie Monitor over Firewall-Computer
- Möglichkeit, eine Verbindung zwischen Controller und Lastgenerator, MI-Listener und Monitor Over Firewall-Computer per Proxy herzustellen
- Verwendung von Netzwerkverbindungsprofilen, mit denen Sie Verbindungsinformationen, z. B. Port- und Proxy, getrennt für jeden Lastgenerator konfigurieren können. Weitere Informationen finden Sie unter "Dialogfeld "Netzwerkprofil-Manager"".
- Verbesserte Unterstützung und Benutzerfreundlichkeit für SSL-Sicherheit
- Lastgeneratoren können so konfiguriert werden, dass nur Verbindungen von vertrauenswürdigen Controllern mit SSL-Authentifizierung akzeptiert werden

Netzwerk- und Sicherheitsmanager

- Neues Dienstprogramm zur Konfiguration von Netzwerk- und Sicherheitseinstellungen für den LoadRunner-Agent, sowohl für den lokalen als auch den Remotecomputer. Weitere Informationen finden Sie unter "Network and Security Manager - Befehlszeilenwerkzeug".

Erweiterungen zum Testen von mobilen Anwendungen

- Wiedergabe von SMP-Sitzungen (SAP Mobile Platform). Weitere Informationen finden Sie unter "Erstellen eines SMP (SAP Mobile Platform)-Skripts".
- Verbesserte Proxy-Aufzeichnung. Weitere Informationen finden Sie unter "Aufzeichnen eines Skripts über einen Proxy".

Verbesserungen und neue Funktionen in VuGen

- Verbesserte Wiedergabezusammenfassung mit Transaktionsaufschlüsselung und neuen Zeittrenddiagrammen. Weitere Informationen finden Sie unter "Registerkarte "Wiedergabeübersicht"".
- Unterstützung der Bearbeitung geläufiger Dateitypen im VuGen-Editor (nicht auf Vuser-Skripts

beschränkt)

- Mehrere Verbesserungen der Benutzerfreundlichkeit in VuGen für eine höhere Produktivität
- Abbrechen einer Aufzeichnung, ohne dabei das Skript zu überschreiben
- Verbesserte Suchfunktionen
- Debugger-Verbesserungen

Erweiterungen und zusätzliche Integrationen für Protokolle

- **Web HTTP/HTML:**
  - HTML5 WebSocket-Unterstützung Weitere Informationen finden Sie unter den **web_ websocket**-Funktionen in der "LoadRunner-Funktionsreferenz".
  - Konverter für TruClient- in Web HTTP/HTML-Protokollskripts. Weitere Informationen finden Sie unter "Konvertieren eines TruClient-Skripts in ein HTTP/HTML-Skript".
  - SSL-Dienstprogramm mit Benutzeroberfläche zur Ausführung von *OpenSSL*-Befehlen. Weitere Informationen finden Sie unter "SSL-Dienstprogramm".
  - SPDY-Unterstützung. Weitere Informationen finden Sie unter "Aufzeichnen des SPDY-Protokolls".
  - Unterstützung für Internet Explorer 11
  - Unterstützung von TLS 1.1, TLS 1.2 und NPN
  - Erweiterte asynchrone Unterstützung mit ResponseHeader-Callback
  - Fehlerbehebung für die Aufzeichnung: Erkennung von Aufzeichnungsfehlern mitsamt möglicher Lösungen
- **Linux-Unterstützung:** Unterstützung der Wiedergabe auf Linux-basierten Lastgeneratoren für die folgenden Protokolle: FTP, IMAP, LDAP, POP3, SMTP und Windows Sockets
- **TruClient:** Unterstützung für Rendezvous-Punkte, IP-Spoofing, VTS und Shunra-Netzwerkvirtualisierung
- **Silverlight:** Unterstützung für aktuelle Version und IP Spoofing
- **SAP Web:** Correlation Studio-Unterstützung in VuGen
- **LDAP:** Unterstützung für die aktuelle Version des LDAP SDK
- **Flex:** Unterstützung des neusten Apache SDK
- **.NET:** Unterstützung der Version 4.5

- **Citrix:** Unterstützung für die aktuelle client-seitige Receiver-App, Ausführen von mehreren veröffentlichten Anwendungen in derselben Sitzung und Unterstützung des aktuellen XenDesktop
- **SAP:** SAP Application Testing by HP, HANA Edition (LoadRunner und Quality Center/Application Lifecycle Management) hat die Premium Qualification für SAP Vendor Branded Resellers erhalten. Diese Zertifizierung gilt ab LoadRunner 11.50 und Application Lifecycle Management 11.00.

Unterstützung zusätzlicher Technologien und Plattformen

- Windows Server 2012-Unterstützung
- Unterstützung für Aufzeichnung und Wiedergabe in Internet Explorer 11, Chrome, Version 30, und Firefox, Version 23,
- Unterstützung für die aktuellen Versionen von Eclipse, JUnit und Selenium

Externe IDE-Integration/Add-Ins für LoadRunner-Skripts und -Anwendungen

- Möglichkeit, den Controller direkt aus Visual Studio und Eclipse aufzurufen
- Add-in für Visual Studio 2012 für VuGen-Skripts und Entwickler von Komponententests
- Mit der aktuellen Version von Eclipse kompatibel

Weitere Informationen finden Sie unter "Weitere Komponenten".

Verbesserungen beim Virtual Table Server (VTS)

- Erstellen mehrerer ausgeführter VTS-Instanzen
- Ausführen von VTS-Befehlen und -Batchdateien über die Befehlszeile
- VTS-Unterstützung über eine JavaScript-API
- Auffüllen der Datentabellen mit Beispieldaten über das Hauptmenü

Weitere Informationen finden Sie in der Virtual Table Server-Dokumentation.

Integration mit den neuesten HP-Produktversionen

- HP Service Virtualization
- HP Diagnostics
- HP SiteScope
- HP Unified Functional Testing (UFT)

- HP Application Lifecycle Management (ALM)

Verbesserte Dokumentation

- Benutzerhandbuch wurde um Workflow-Abbildungen für Test- und Skripterstellungsprozesse erweitert
- Erweiterung der Dokumentation zu den Flex-, Web-HTTP/HTML- und TruClient-Protokollen durch zusätzliche Aufgaben und Prozeduren
- Effektivere Gliederung der Funktionsreferenz, einschließlich zusätzlicher Codebeispiele und neuen WebSocket-Funktionen

Allgemeine Verbesserungen

- Einfachere und kürzere Produktinstallation
- Unterstützung für Konten ohne Administratorrechte: LoadRunner-Komponenten können nun unter einem Standardbenutzerkonto (ein Administratorkonto ist nicht erforderlich) ausgeführt, und sowohl UAC als auch DEP sind aktiviert. Dies gilt für alle LoadRunner-Komponenten und -Tools, die im Startmenü unter dem Knoten **LoadRunner > Tools** zur Verfügung stehen.
- Verbesserungen bei der Shunra-Integration
- Datenbank-API-Funktionen sind jetzt für alle Protokolle der C-Programmiersprache verfügbar

Community-Lizenz-Bundle

- LoadRunner enthält nun standardmäßig das Community Bundle, welches die Instant-on-Lizenz ersetzt.
  - Das neue Paket enthält permanente Lizenzen für 50 Vuser
  - Enthalten sind alle Protokolle außer für GUI- (UFT), COM/DCOM und Protokolle aus dem Vorlagen-Bundle

Eine vollständige Liste der unterstützten Integrationen und Produktversionen finden Sie auf der Software-Support-Website unter www.hp.com/go/hpsoftwaresupport. Oder wählen Sie **Hilfe > HP Software Support Online**. Klicken Sie auf die Registerkarte für die eigenständige Problemlösung und suchen Sie nach der LoadRunner PAM (Matrix der Produktverfügbarkeit) für diese Version.

VuGen-Konzepte

Der Abschnitt **VuGen-Konzepte** enthält Informationen zu mehreren Themen in Zusammenhang mit der Nutzung von VuGen für die Erstellung von Vuser-Skripts.

Übersicht über VuGen

Wenn Sie eine Umgebung testen oder überwachen, müssen Sie das tatsächliche Verhalten von Benutzern in Ihrem System emulieren. HP-Testtools emulieren eine Umgebung, in der mehrere Benutzer gleichzeitig arbeiten oder auf das System zugreifen. Für diese Emulation wurde der Mensch durch einen virtuellen Benutzer, den so genannten *Vuser*, ersetzt. Die Aktionen, die ein Vuser durchführt, werden in der Regel in einem *Vuser-Skript* aufgezeichnet. Das primäre Tool zum Erstellen von Vuser-Skripts ist Virtual User Generator von HP, auch als VuGen bezeichnet.

VuGen dient nicht nur zur Aufzeichnung von Vuser-Skripts, sondern auch zu deren Ausführung. Das Ausführen von Skripts über VuGen ist beim Debugging nützlich. Mithilfe des Tools können Sie festlegen, wie ein Vuser-Skript im Rahmen eine umfangreicheren Tests ausgeführt wird.

Bei der Aufzeichnung von Vuser-Skripts erzeugt VuGen unterschiedliche Funktionen, welche die während der Aufzeichnungssitzung ausgeführten Aktionen definieren. Diese Funktionen werden von VuGen in den VuGen-Editor eingefügt, um ein Basis-Vuser-Skript zu erstellen.

Die Aufzeichnung von Vuser-Skripts durch VuGen ist nur auf Windows-Plattformen möglich. Aufgezeichnete Vuser-Skripts können jedoch sowohl auf Windows- als auch auf Linux-Plattformen ausgeführt werden.

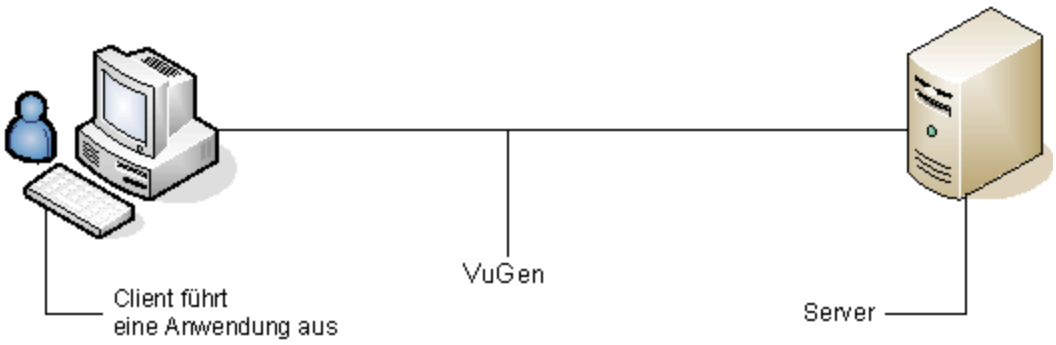
Sie können auch Vuser-Skripts in Ihrer systemeigenen Programmierungsanwendung wie MS Visual Studio programmieren. Für den Zugriff auf die LoadRunner-API installieren Sie das entsprechende auf der LoadRunner-DVD bereitgestellte IDE-Add-in.

Übersicht über die Vuser-Technologie

Vuser emulieren die Aktionen menschlicher Benutzer, indem sie typische Geschäftsprozesse in Ihrer Anwendung durchführen. Die Aktionen, die ein Vuser während der Aufzeichnungssitzung ausführt, werden in einem *Vuser-Skript* beschrieben.

Das HP-Tool zum Erstellen von Vuser-Skripts ist der Virtual User Generator, *VuGen*. Mit VuGen entwickeln Sie ein Vuser-Skript, indem Sie einen Benutzer aufzeichnen, der typische Geschäftsprozesse in einer Clientanwendung ausführt. VuGen zeichnet die Aktionen auf, die Sie bei der Aufzeichnungssitzung durchführen. Dabei werden jedoch nur die Aktivitäten zwischen dem Client und dem Server aufgezeichnet. Anstatt manuell die API-Funktionsaufrufe der Anwendung an den Server zu programmieren, erzeugt VuGen automatisch Funktionen, mit denen wirklichkeitsgetreue Situationen modelliert und emuliert werden.

Während der Aufzeichnung überwacht VuGen den Client und protokolliert alle an den Server gesendeten und vom Server empfangenen Anforderungen.



Bei der Wiedergabe kommunizieren die Vuser-Skripts direkt mit dem Server, indem sie die Aufrufe an die Server-API ausführen. Wenn ein Vuser direkt mit einem Server kommuniziert, sind für die Clientoberfläche keine Systemressourcen erforderlich. Dadurch können Sie eine große Anzahl Vuser gleichzeitig auf einer einzelnen Workstation ausführen und nur wenige Testcomputer zum Emulieren großer Serverlasten verwenden.

Da Vuser-Skripts keine Clientsoftware hinzuziehen, können Sie Vuser darüber hinaus einsetzen, um die Serverleistung zu überprüfen, und zwar noch bevor die Benutzeroberfläche der Clientsoftware vollständig entwickelt wurde.

Mit VuGen können Sie Skripts als eigenständige Tests ausführen. Dies ist nützlich für die Verifizierung der Funktionalität des Skripts. Gleichzeitig können Sie feststellen, wie sich ein Vuser während eines Testlaufs verhält.

Damit Sie die Vuser-Skripts wirksam einsetzen können, fügen Sie diese mit dem LoadRunner-Controller zu einem Szenario hinzu. Beim Ausführen der Vuser sammeln Sie Informationen über die Antwort des Systems. Anschließend können Sie diese Informationen mit dem Analysetool anzeigen. Beispielsweise können Sie beobachten, wie sich ein Server verhält, wenn 100 Vuser gleichzeitig Geld vom Geldautomaten einer Bank abheben.

Die Aufzeichnung von Vuser-Skripts durch VuGen ist nur auf Windows-Plattformen möglich. Aufgezeichnete Vuser-Skripts können jedoch sowohl auf Windows- als auch auf Linux-Plattformen ausgeführt werden.

Vuser-Typen

LoadRunner unterstützt mehrere Vuser-Typen:

Protokollbasierte Vuser

LoadRunner unterstützt verschiedene Vuser-Typen, bei denen die gängigsten Protokolle verwendet werden. Durch jeden Typ werden unterschiedliche Aspekte moderner Systemarchitekturen abgedeckt.

Sie können ein Vuser-Skript mit einem einzelnen oder mit mehreren Protokollen erstellen.

Eine vollständige Liste der verfügbaren Vuser-Protokolle finden Sie unter "Vuser-Protokolle" unten.

Einheitentest-basierte Vuser

LoadRunner unterstützt Einheitentests in Form von **.dll**- oder **.jar**/**.class**-Dateien, die in Microsoft Visual Studio oder Eclipse erstellt wurden.

Um diese Tests zu erstellen, installieren Sie das entsprechende Add-In für die IDE für Entwickler, die sich im Ordner **Additional Components** der LoadRunner-DVD befindet.

GUI-Vuser

LoadRunner bietet die Möglichkeit, Funktionstestskripts in Form von *GUI-Tests* in ein Lasttestszenario zu integrieren. Sie erstellen GUI-Tests mithilfe von HP-Software für Funktionstests -QuickTest oder Unified Functional Testing.

Auf einem Windows-basierten Lastgenerator können Sie nur einen einzigen GUI-Vuser ausführen. Zur Ausführung mehrerer GUI-Vuser verwenden Sie Citrix. Weitere Informationen über Windows-basierte GUI-Vuser finden Sie unter "Verwenden von QuickTest- oder Unified Functional Testing-Tests in LoadRunner".

Vuser-Protokolle

Mit VuGen können Sie eine Vielzahl von Protokollen aufzeichnen, von denen jedes für eine bestimmte Lasttestumgebung oder Topologie geeignet ist und die die Verwendung eines bestimmten Vuser-Skripts bedingen. So können Sie z. B. ein Web (HTTP/HTML)-Vuser-Skript zum Emulieren von Benutzern verwenden, die Webbrowser einsetzen. Zum Emulieren einer FTP-Sitzung können Sie FTP-Vuser verwenden. Die verschiedenen Vuser-Technologien können allein oder gemeinsam verwendet werden, um effektive Lasttests zu erstellen.

In der folgenden Tabelle sind die verfügbaren Vuser-Protokolle und eine kurze Beschreibung der einzelnen Protokolle aufgeführt.

Hinweis: In der LoadRunner-Dokumentation werden die Begriffe *Vuser-Protokolle* und *Vuser-Typen* austauschbar verwendet.

Protokoll	Beschreibung
.NET	Unterstützt die Aufzeichnung von Microsoft .NET-Technologien für Client/Server.

Protokoll	Beschreibung
Ajax (Click & Script)	Eine Abkürzung für Asynchronous JavaScript und XML. Ajax (Click & Script) verwendet asynchrone HTTP-Anfragen und ermöglicht so Webseiten, kleine Informationseinheiten anstelle vollständiger Seiten abzurufen.
C Vuser	Ein generischer virtueller Benutzer, der die standardmäßige C-Bibliothek verwendet.
Citrix ICA	Ein Tool für den Remotezugriff, mit dem Benutzer bestimmte Anwendungen auf externen Computern ausführen können.
COM/DCOM	COM (Component Object Model) - Ein Verfahren zur Entwicklung wieder verwendbarer Softwarekomponenten.
(DNS) Domain Name Resolution	Das DNS-Protokoll ist ein Low-Level-Protokoll, mit dem Sie die Aktionen eines Benutzers auf einem DNS-Server emulieren können. Das DNS-Protokoll emuliert einen Benutzer, der auf einen Domain-Name-Server zugreift, um einen Hostnamen in die entsprechende IP-Adresse aufzulösen. Dieses Protokoll unterstützt lediglich die Wiedergabe, die Funktionen müssen Sie manuell in Ihr Skript einfügen.
Flex	Flex ist eine Anwendungsentwicklungslösung, mit der RIAs (Rich Internet Applications) innerhalb des Unternehmens und über das Web erstellt werden können. Action Message Format (AMF), ein proprietäres Macromedia-Protokoll, mit dem binäre Flash Remoting-Daten zwischen einer Flash-Anwendung und einem Anwendungsserver über HTTP ausgetauscht werden können.
FTP (File Transfer Protocol)	File Transfer Protocol - Ein System, das Dateien über das Netzwerk von einem Speicherort an einen anderen überträgt. Das FTP-Protokoll ist ein Low-Level-Protokoll, mit dem Sie die Aktionen eines Benutzers emulieren können, der mit einem FTP-Server arbeitet.
IMAP (Internet Messaging)	Internet Message Application - Ein Protokoll, über das Clients E-Mails von einem E-Mail-Server lesen können.
Java over HTTP	Ausgerichtet auf die Aufzeichnung Java-basierter Anwendungen und Applets. Es generiert ein Java-Skript, in dem Webfunktionen eingesetzt werden. Dieses Protokoll unterscheidet sich von anderen Java-Protokollen dadurch, dass es Java-Remoteaufrufe über HTTP aufzeichnen und wiedergegeben kann.
Java Record Replay	Allgemeiner Java-Recorder.
Java-Vuser	Java-Programmiersprache mit Unterstützung auf Protokollebene.
LDAP (Listing Directory Service)	Ein Internetprotokoll, mit dessen Hilfe E-Mail-Anwendungen nach Kontaktinformationen auf einem Server suchen können.

Protokoll	Beschreibung
MAPI (Microsoft Exchange)	Messaging Application Programming-Schnittstelle (MAPI), über die Anwendungen E-Mails senden und empfangen können.
MMS (Media Player)	Streaming von Daten von einem Medienserver unter Verwendung des Microsoft-MMS-Protokolls. **Wichtig:** Um Media Player-Funktionen abzuspielen, müssen Sie die Datei **wmload.asf** auf dem Windows Media Server-Computer platzieren. Der VuGen-Computer muss über **mms://<Servername>/wmload.asf** darauf zugreifen können. Bei dieser ASF-Datei kann es sich um eine beliebige Mediendatei handeln, die in **wmload.asf** umbenannt wurde.
MMS (Multimedia Messaging Service)	Ein Messaging-Service, der zum Senden von MMS-Nachrichten zwischen mobilen Geräten verwendet wird.
Mobile Anwendung - HTTP/HTML	Ermöglicht die Aufzeichnung mobiler systemeigener Anwendungen.
ODBC	Open Database Connectivity (ODBC) - Ein Protokoll, das eine gemeinsame Schnittstelle für den Zugriff auf Datenbanken bietet.
Oracle - 2-Tier	Oracle-Datenbank, die eine Standard-2-Tier-Client/Server-Architektur verwendet.
Oracle Web Applications 11i	Die Oracle-Anwendungsschnittstelle zum Durchführen von Aktionen über das Web. Dieser Vuser-Typ erkennt Aktionen der LoadRunner API- und der JavaScript-Ebene.
Oracle NCA	Oracle-Datenbank mit 3-Tier-Architektur, die einen Java-Client, einen Webserver und die Datenbank umfasst.
POP3 (Post Office Protocol)	Ein Protokoll, über das einzelne Computer E-Mails von einem E-Mail-Server abrufen können.
RDP (Remote Desktop Protocol)	Ein Remotesitzungstool, das Microsoft-Remotedesktopverbindungen nutzt, um Anwendungen auf externen Computern auszuführen.
RTE (Remote Terminal Emulator)	Emulation von Benutzern, die Eingaben an zeichenbasierte Anwendungen senden bzw. Ausgaben von diesen Anwendungen empfangen.
SAP (Click & Script)	Emulation von Kommunikation zwischen einem Browser und einem SAP-Server auf Ebene von GUI oder Benutzeraktion.

Protokoll	Beschreibung
SAP GUI	Ein System zur Planung von Unternehmensressourcen (ERP) für die Integration wichtiger Geschäfts- und Managementprozesse mithilfe des SAP GUI-Clients für Windows.
SAP - Web	Ein System zur Planung von Unternehmensressourcen (ERP) für die Integration wichtiger Geschäfts- und Managementprozesse mithilfe von SAP Portal- oder SAP Workplace-Clients.
Siebel - Web	Eine Customer Relationship Management-Anwendung (CRM).
Silverlight	Ein Protokoll für Silverlight-Anwendungen, das Benutzeraktivität auf der Transportebene emuliert. Es ermöglicht das Erstellen abstrakter Skripts, indem die von der Anwendung verwendeten WSDL-Dateien automatisch importiert und konfiguriert werden.
SMP (SAP Mobile Platform)	Ein Protokoll zum Aufzeichnen von Aktionen auf einer mobilen SAP-Anwendung.
SMTP (Simple Mail Protocol)	Simple Mail Transfer Protocol - Ein System zum Verteilen von E-Mail an einen bestimmten Computer.
TruClient - Ajax	Ein erweitertes Protokoll für moderne JavaScript-basierte Anwendungen (einschließlich Ajax), das Benutzeraktivität in einem Webbrowser emuliert. Skripts werden interaktiv in Mozilla Firefox oder Internet Explorer entwickelt.
TruClient - Ajax - Mobile	Ermöglicht die Aufzeichnung mobiler browserbasierter Anwendungen mit der TruClient Ajax-Technologie.
Web (HTTP/HTML)	Emulation von Kommunikation zwischen Browser und Webserver auf HTTP- oder HTML-Ebene.
Webdienste	Webdienste sind eine Programmschnittstelle, über die Anwendungen mit anderen Anwendungen über das Internet kommunizieren können.
Windows-Sockets	Die standardmäßige Netzwerkprogrammierschnittstelle für die Windows-Plattform.

Hinweis: Um Vuser der verschiedenen Protokolle mit dem Controller auszuführen, müssen Sie entweder eine globale Lizenz oder die Lizenzen für die gewünschten Protokolle besitzen. Das Community Bundle enthält 50 Vuser für alle Protokolle, mit Ausnahme von GUI- (UFT), COM/DCOM und Protokollen aus dem Vorlagen-Bundle, wie z. B. C Vuser. Um die Details Ihrer Vuser-Lizenzen zu überprüfen, öffnen Sie das LoadRunner-Lizenzdienstprogramm, indem Sie auf **Start > Alle Programme > HP Software > HP LoadRunner > Lizenz > LoadRunner-Lizenzdienstprogramm** klicken.

Tastenkombinationen

In den folgenden Tabellen sind die verfügbaren Tastenkombinationen für die VuGen-Menüs aufgeführt:

Menü "Datei"

Neues Skript und neue Lösung	Strg+N
Öffnen > Skript/Lösung	Strg+O
Hinzufügen > Neues Skript	Strg+Umschalt+A
Hinzufügen > Vorhandenes Skript	Alt+Umschalt+A
Schließen > Dokument	Strg+F4
Schließen > Lösung	Strg+Umschalt+F4
Skript speichern	Strg+S
Alle Skripts speichern	Strg+Umschalt+S
Datei erneut laden	Strg+Umschalt+U
Drucken	Strg+P
Beenden	Alt+F4

Menü "Bearbeiten"

Rückgängig	Strg+Z
Wiederholen	Strg+Y
Ausschneiden	Strg+X
Kopieren	Strg+C
Einfügen	Strg+V
Löschen	Entf
Alle auswählen	Strg+A
Format > Umgeben mit	Strg+J
Format > Einzug vergrößern	Tab
Format > Einzug verkleinern	Umschalt+Tab
Organisation > Organisation umschalten	Strg+Umschalt+M

Organisation > Alle Organisationen umschalten	Strg+Umschalt+L
Organisation > Nur Definitionen anzeigen	Strg+Umschalt+P
Funktionssyntax anzeigen	Strg+Umschalt+Leertaste
Wort vervollständigen	Strg+Leertaste

Menü "Ansicht"

Lösungs-Explorer	Strg+Alt+L
Suchergebnisse	Strg+Alt+R
Textmarken	Strg+Alt+K
Schrittetoolbox	Strg+Alt+B
Snapshot	Strg+Alt+P
Schrittenavigator	Strg+Alt+S
Miniaturansicht-Explorer	Strg+Alt+T
Eigenschaften	Strg+Alt+F4
Ausgabe	Strg+Alt+O
Vollbild	Alt+Umschalt+Eingabe

Menü "Suchen"

Schnellsuche	Strg+F
Weitersuchen	F3
Nächste Auswahl suchen	Strg+F3
In Dateien suchen	Strg+Umschalt+F
Schnellersetzung	Strg+H
Inkrementelle Suche	Strg+E
Umgekehrte inkrementelle Suche	Strg+Umschalt+E
Textmarken > Textmarke umschalten	Strg+F2
Textmarken > Vorherige Textmarke	Umschalt+F2
Textmarken > Nächste Textmarke	F2
Gehe zu	Strg+G

Menü "Entwurf"

Aktion > Aktion löschen	Entf
Aktion > Aktion umbenennen	F2
In Skript einfügen > Neuer Schritt	Alt+Einfg
In Skript einfügen > Transaktion starten	Strg+T
In Skript einfügen > Transaktion beenden	Strg+Umschalt+T
In Skript einfügen > Kommentar	Strg+Alt+C
Parameter > Parameterliste	Strg+L
Parameter > Neuen Parameter erstellen	Strg+K
Parameter > Parametertrennzeichen konfigurieren	Strg+B
Design Studio	Strg+U

Menü "Aufzeichnen"

Aufzeichnen	Strg+R
Skript neu generieren	Strg+Umschalt+R
Aufzeichnungsoptionen	Strg+F7

Menü "Wiedergeben"

Ausführen	F5
Anhalten	Strg+F5
Kompilieren	Umschalt+F5
Haltepunkt umschalten	F9
Mit Debuggen fortfahren	F5
Schritt für Schritt ausführen	F10
Laufzeiteinstellungen	F4

Menü "ALM"

ALM-Verbindung	Strg+Q

Menü "Fenster"

Nächstes Fenster	Strg+Tab
Vorh. Fenster	Strg+Umschalt+Tab

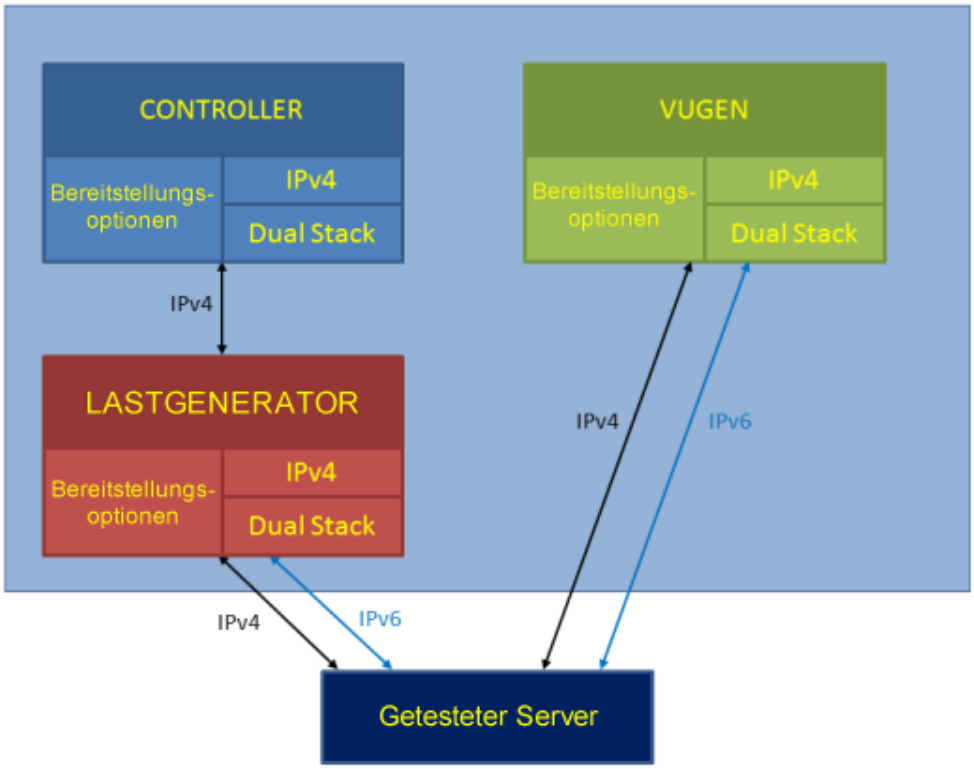
IPv6-Unterstützung

Da die IPv6-Implementierung inzwischen zunehmend verbreitet ist, können Sie mit LoadRunner jetzt zusätzlich zu IPv4-basierten Anwendungen auch IPv6-basierte Anwendungen testen. Die Skriptaufzeichnung unterstützt die gleichzeitige Aufzeichnung für IPv4 und IPv6. Der generierte Code ist nicht IP-spezifisch. Außer bei Web-HTTP-basierten Protokollen wird der Benutzer gar nicht merken, welche IP-Version verwendet wird, wenn das Skript in einem Lasttest wiedergegeben wird. Web-HTTP-Protokolle verfügen über eine Laufzeiteinstellung, bei der Sie für die Wiedergabe zwischen IPv4 und IPv6 wählen können.

IPv6-Bereitstellung

Die interne LoadRunner-Kommunikation zwischen dem Controller und den Lastgeneratoren verwendet weiterhin die IPv4-Kommunikation. Zum Aufzeichnen und Wiedergeben sowohl in IPv4 als auch in IPv6 installieren Sie VuGen und den Lastgenerator auf IPv6-fähigen Computern, wie in der untenstehenden Abbildung gezeigt.

Änderungen in der IPv6-Bereitstellung



Weitere Informationen zu mit IPv6 verbundenen Änderungen finden Sie unter Dialogfeld "Erweiterte Optionen" und im Abschnitt zum IP-Assistenten im *Controller-Benutzerhandbuch*.

Unterstützte Protokolle

Die folgenden Protokolle unterstützen IPv6-Tests:

- Citrix ICA
- DNS
- Flex
- FTP
- IMAP
- Java over HTTP
- Mobile Anwendung - HTTP/HTML
- Windows Sockets – Mehrfachprotokoll
- Oracle NCA
- POP3
- RDP
- Silverlight
- SMTP
- TruClient Ajax
- Click & Script-Protokolle
- Web (HTTP/HTML)
- Webdienste

Protokollunterstützung – Einschränkungen

IPv6-Unterstützung ist für die Protokolle in der obigen Liste mit folgenden Einschränkungen verfügbar:

- **Web-HTTP-Protokoll**
  - FTP vom Web wird nicht unterstützt
  - Webseitenaufschlüsselung wird nicht unterstützt
  - Kerberos wird nicht unterstützt

- Spoofing vom Web wird nicht unterstützt
- PAC-Datei wird nicht unterstützt
- Eine explizite IP (im IPv6-Format) in einem URL-Argument kann nicht verwendet werden. Der folgende Schritt beispielsweise schlägt bei der Wiedergabe fehl:

  ```
  web_url("IPv6",
      "URL=http://[2001:0dc8:8aa3:0000:0b00:8a2e:0370:7334]/",
      "Resource=0",
      "RecContentType=text/html",
      "Referer=",
      "Snapshot=t1.inf",
      "Mode=HTML",
     LAST);
  ```

- Webtrace
  - IPv6 Webtrace wird nicht auf ausgehenden IP6-zu-IP4-Netzwerkschnittstellen unterstützt.
  - IPv6 Webtrace unterstützt den RAW-Socket-Modus nicht.
- **Allgemeine Einschränkungen**
  - Wiedergabefehler können aufgrund eines IPv4/IPv6-Wechsels zwischen Aufzeichnung und Wiedergabe auftreten.

VuGen-Benutzeroberfläche

Im Abschnitt **VuGen-Benutzeroberfläche** werden die Komponenten der Umgebung beschrieben, in der Sie beim Aufzeichnen, Wiedergeben und Debuggen eines Vuser-Skripts arbeiten.

VuGen-Benutzeroberfläche – Übersicht

Die Benutzeroberfläche von VuGen besteht aus mehreren Abschnitten, in denen zahlreiche Bereiche angezeigt werden können. Sie können das Layout dieser Abschnitte und Bereiche ändern, um die Aufzeichnung Ihrer Vuser-Skripts und die Fehlerbeseitigung zu unterstützen. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56.

Die folgende Tabelle enthält eine Beschreibung der einzelnen Bereiche und stellt ein kurzes Anwendungsszenario bereit.

Bereich	Zweck	Anwendungsszenario
Textmarken	Textmarken ermöglichen Ihnen das Festlegen einer Position in einem Skript, sodass Sie diese zur späteren Bearbeitung leicht auffinden können. Weitere Informationen finden Sie unter "Bereich "Textmarken"" auf Seite 114.	
Fehler	Zeigt Skriptfehler und bei der Skriptwiedergabe generierte Warnungen und Meldungen an. Sie können auch benutzerdefinierte Filter für Fehlermeldungen erstellen. Weitere Informationen finden Sie unter "Bereich "Fehler"" auf Seite 116.	**Fehlerfilter aus Ausgabeprotokoll** Nach jedem Testprozess, wie der Codegenerierung und -wiedergabe, können Sie im Fehlerbereich das Fehlerprotokoll überprüfen. Sie können auch andere Typen wie Warnungen und Meldungen anzeigen. Die Community-Suche steht im Kontextmenü für markierte Fehler zur Verfügung. Darüber hinaus können Sie auf die Meldung doppelklicken, um zu der Position im Skript zu springen.
Snapshot	Ein Snapshot zeigt mit einem bestimmten Schritt im Skript verbundene Server- und Clientdaten an. Das Format der Daten hängt von dem Protokoll ab, das für die Skripterstellung verwendet wird. Details zu Snapshots finden Sie unter "Snapshot-Bereich" auf Seite 109.	Sie verwenden den Snapshot-Bereich, um alle im Schritt enthaltenen Daten zu verstehen. Sie können bestimmte Aufgaben durchführen, zum Beispiel Suchen nach Korrelationen, Vergleichen von Aufzeichnungs- und Wiedergabe-Snapshots und Suchen nach bestimmten Werten mit der Standardsuchfunktion.
Datenraster	Eine vereinfachte Ansicht aller mit dem Skript verbundenen Datensätze. Gilt für bestimmte Protokolle wie MSSQL.	Der Datenbereich enthält entweder gesendete oder empfangene Daten. Die Daten werden in Form einer einfach formatierten Tabelle angezeigt. Sie können Operationen wie die Parametrierung und andere Datenmanipulationen durchführen.

Bereich	Zweck	Anwendungsszenario
Lösungs-Explorer	Der Lösungs-Explorer ermöglicht Ihnen das Organisieren und Verwalten mehrerer Skripts in einer benannten Lösung. Dieser Bereich ermöglicht den einfachen Zugriff auf und die Manipulation von Lösungen, wie Skript-Assets, Parameter, Laufzeiteinstellungen und Wiedergabeausführungen. Sie können im Editor-Bereich auf ein Asset doppelklicken, um es zu aktivieren. Sie können mit der rechten Maustaste auf ein Asset klicken, um die für dieses Asset verfügbaren Operationen anzuzeigen. Weitere Informationen finden Sie unter "Bereich "Lösungs-Explorer"" auf Seite 101.	Sie können Skripts jetzt in einer Lösung bündeln. Beispielsweise können Sie Skripts bündeln, die sich auf einen Geschäftsprozess beziehen. Zurzeit ist die Lösungsentität wie folgt beschränkt: • Die Lösungsentität ist auf die lokale Skriptentwicklung beschränkt, wie "Alles Speichern" oder "Öffnen". • Die Lösungsentität kann nicht in vorhandene Management-Tools wie ALM oder Controller importiert werden.

Bereich	Zweck	Anwendungsszenario
Schrittenavigator	Ermöglicht Ihnen das Navigieren zu einem ausgewählten Schritt in Ihrem Skript. Wenn Ihr Skript viele Schritte enthält, können Sie das Suchfeld verwenden, um in den unterschiedlichen Teilen des Schritts nach übereinstimmendem Text zu suchen. Weitere Informationen finden Sie unter "Bereich "Schrittenavigator"" auf Seite 115.	Tabellenansicht aller vorhandenen LoadRunner-API-Schritte. Der Schrittenavigator ersetzt die in früheren VuGen-Versionen vorhandene Strukturansicht. • Dynamischer Filter für verschiedene Schritteigenschaften, zum Beispiel Schrittargumente. • Sie können das Skript im Gültigkeitsbereich für Aktionen oder Skripts anzeigen. • Jeder Schritt mit einem Snapshot ist mit einem Symbol gekennzeichnet. Wenn Sie mit dem Mauszeiger auf einen Schritt zeigen, der über eine verbundene Miniaturansicht verfügt, wird diese als QuickInfo angezeigt. • Durch Doppelklicken auf einen Schritt springen Sie an die entsprechende Stelle im Skript und synchronisieren alle anderen Bereiche. • Aus dem Kontextmenü können Sie verschiedene Aktionen ausführen, wie zum Beispiel das Beenden eines Kopiervorgangs. • Die Schrittenavigation wird basierend auf der Gültigkeit des Skripts synchronisiert. Sie können den Status des Bereichs während der Skriptbearbeitung überprüfen.

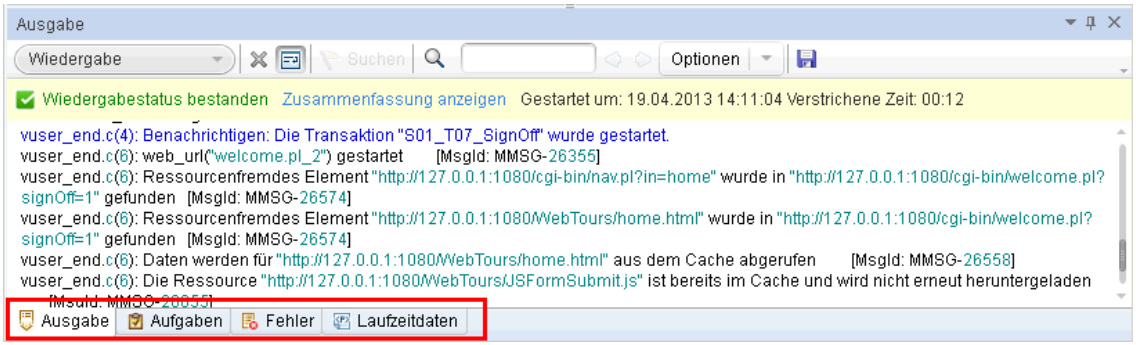
Bereich	Zweck	Anwendungsszenario
Aufgaben	Sie können mit einem Skript oder einer Lösung verbundene Aufgaben bearbeiten, hinzufügen oder nach ihnen suchen. Weitere Informationen finden Sie unter "Aufgabenbereich" auf Seite 118.	**Zentralisierte Aufgabenverwaltung** Während der Skriptentwicklung bieten Kommentaraufgaben die Möglichkeit, Aufgaben in das Skript einzubetten. In einem C-basierten Skript zum Beispiel `//TODO Parameter hinzufügen` Benutzeraufgaben ermöglichen das Hinzufügen von Aufgaben, das Zuweisen des Besitzes und die Verfolgung des Abschlusses von Aufgaben, die mit dem gesamten Skriptprozess verbunden sind. Beispiel **Lasttestszenario mit neu erstelltem Skript hinzufügen**.
Editor	Der Editor-Bereich enthält alle geöffneten Assets, wie Skriptaktionen, zusätzliche Dateien und Berichtsübersicht, und ermöglicht das Bearbeiten von Skriptaktionen. Darüber hinaus können Sie die Community-Suche, die Hilfedokumentation und Browserseiten öffnen. Weitere Informationen finden Sie unter "Editor" auf Seite 93.	Sie können Skriptaktionen aus unterschiedlichen Skripts in einer Lösung gleichzeitig anzeigen. • Sie können Kommentarblöcke einfügen. • Sie können das Farbschema anpassen. • Sie können die Schriften vergrößern. • Optionen für das automatische Vervollständigen mit Strg + Leertaste. • Alle unterstützenden Daten in den Bereichen werden synchronisiert, wenn Sie durch das Skript navigieren. • Die Debug-Funktion des Skripts kann innerhalb des Skripts in Form von Haltepunkten verwaltet werden. • Die Community-Suche steht für Text zur Verfügung, der im Skript markiert ist.

Bereich	Zweck	Anwendungsszenario
Miniaturansicht	Ermöglicht das visuelle Verfolgen des Geschäftsprozesses, den das Skript aufgezeichnet hat. Weitere Informationen finden Sie unter "Miniaturansicht-Explorer" auf Seite 95.	• Die Generierung von Miniaturansichten kann unter **Extras > Optionen** konfiguriert werden. • Sie können die Miniaturansicht durch Doppelklicken in voller Größe anzeigen. • Im Kontextmenü stehen zusätzliche Operationen zur Verfügung, wie **Gehe zu Schritt**.
Ausgabe	Ereignisprotokoll für unterschiedliche VuGen-Operationen, wie Codegenerierung, -wiedergabe und -aufzeichnung. Details finden Sie unter "Ausgabebereich" auf Seite 107.	Im Ausgabebereich werden alle während der Skriptentwicklungsphasen erstellten Protokolle angezeigt, wie Codegenerierung, -wiedergabe und -aufzeichnung. • Sie können aus dem Bereich Schnellsuchen durchführen oder mit Strg + F die vollständige Suche aufrufen. • Sie können die Protokolle speichern.
Haltepunkte	VuGen ermöglicht Ihnen die Verwendung von Haltepunkten in Ihren Vuser-Skripts, um das Debuggen der Skripts zu erleichtern. Weitere Informationen finden Sie unter "Bereich "Haltepunkte"" auf Seite 119.	Ermöglicht das Festlegen und Verwalten von Haltepunkten, damit Sie die Auswirkungen des Skripts an vordefinierten Stellen bei der Ausführung auf Ihre Anwendung besser analysieren können.
Beobachten	Im Überwachungsbereich können Sie Variablen und Ausdrücke bei der Skriptausführung überwachen, wenn es sich im Status **Pausiert** befindet. Weitere Informationen finden Sie unter "Überwachungsbereich" auf Seite 122.	

Bereich	Zweck	Anwendungsszenario
Aufrufliste	In diesem Debug-Bereich können Sie Informationen zu den Methoden und Funktionen anzeigen, die sich aktuell in der Aufrufliste Ihres Skripts befinden, oder den Kontext, in dem die Ausführungssitzung angehalten wurde. Weitere Informationen finden Sie unter "Bereich "Aufrufliste"" auf Seite 124.	

VuGen-Layouts - Übersicht

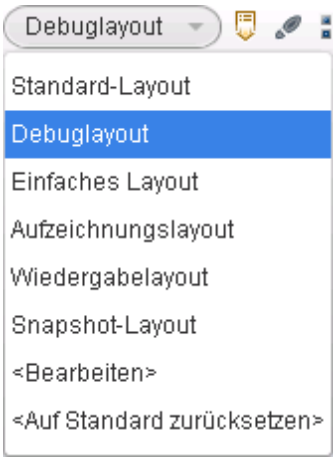
Das VuGen-Fenster besteht aus mehreren Abschnitten. Jeder Abschnitt kann mehrere Bereiche umfassen, wie den Fehlerbereich oder den Snapshot-Bereich. Die Bereiche werden in den Abschnitten als Registerkarten angezeigt. Die folgende Abbildung zeigt einen Abschnitt, der vier Bereiche als Registerkarten enthält: **Ausgabe**, **Aufgaben**, **Fehler** und **Laufzeitdaten**. Aktuell wird der Bereich **Ausgabe** angezeigt.



Die Konfiguration der Abschnitte und Bereiche innerhalb der Abschnitte wird als Layout bezeichnet. VuGen verfügt über verschiedene Standardlayouts: **Standard**, **Debuggen**, **Einfach**, **Aufzeichnen**, **Wiedergabe** und **Snapshot**. Jedes Layout dient dazu, eine bestimmte Phase des Entwicklungsprozesses von Vuser-Skripts zu unterstützen. Beispielsweise umfasst das Wiedergabelayout die Bereiche, die bei der Ausführung eines Vuser-Skripts am nützlichsten sind: **Fehler**, **Aufrufliste**, **Beobachten**, **Haltepunkte**, **Ausgabe** und **Laufzeitdaten**. VuGen verwendet in den jeweiligen Phasen des Skriptentwicklungsprozesses automatisch bestimmte Layouts. Beispielsweise wird das Aufzeichnungslayout bei der Aufzeichnung eines Skripts verwendet, das Wiedergabelayout hingegen bei der Wiedergabe eines Skripts.

In der VuGen-Symbolleiste wird das aktuell verwendete Layout angezeigt:

. Um das Layout zu ändern, klicken Sie auf die Dropdownliste **Layout** und wählen Sie das erforderliche Layout wie unten gezeigt aus.



Ändern eines Layouts

Sie können keine Standardlayouts hinzufügen oder löschen. Sie können jedoch die meisten Standardlayouts von VuGen ändern, um diese an Ihre Anforderungen anzupassen. Wenn Sie ein Layout ändern, können Sie Abschnitte hinzufügen, verschieben, ihre Größe ändern, die Bereiche für jeden Abschnitt auswählen und festlegen, welche dieser Bereiche standardmäßig angezeigt werden. Weitere Informationen zu dieser Aufgabe finden Sie unter "Ändern des VuGen-Layouts" auf Seite 71. Nachdem Sie ein Standardlayout geändert haben, bleibt das Layout in VuGen erhalten, bis Sie dieses erneut ändern oder die Standardlayouts wiederherstellen.

Hinweis: VuGen speichert keine Änderungen, die Sie an dem einfachen Layout durchführen.

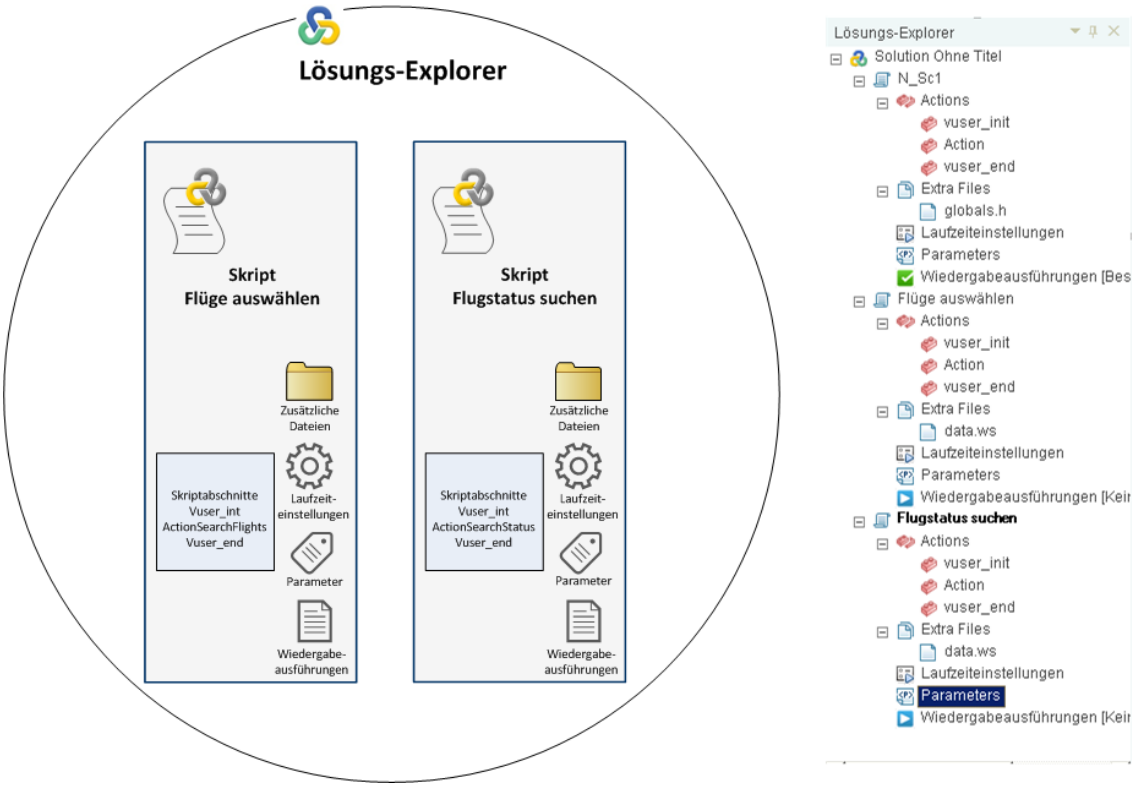
Wiederherstellen der Standardlayouts

Klicken Sie in der VuGen-Symbolleiste auf die Dropdownliste **Layout** und wählen Sie **Auf Standard zurücksetzen** aus. VuGen setzt alle Standardlayouts auf die Standardeinstellungen zurück.

Lösungs-Explorer - Übersicht

Der Lösungs-Explorer erleichtert das Organisieren von und Navigieren durch Skriptentitäten und verbessert den Aufzeichnungs-, Wiedergabe- und Debugprozess. Eine von Ihnen erstellte Lösung kann mehrere Skripts verschiedener Protokolle enthalten, die sich auf den kompletten Geschäftsprozesszyklus beziehen. Jede Skriptentität umfasst zusätzliche Dateien (z. B. Kopfzeilendateien), Laufzeiteinstellungen, Parameter und Wiedergabeausführungen.

Die folgende Abbildung zeigt die Struktur des Lösungs-Explorer:



Lösung

Wenn Sie ein neues Skript erstellen, können Sie einen Namen für die Lösung angeben, die das Skript enthält. Sie können beispielsweise den Geschäftsprozess als Lösungsnamen verwenden. Wenn Sie keinen Namen angeben, wird der Standardname **Ohne Titel** verwendet.

Skripts

Ihre Lösung kann mehrere Skripts enthalten. Klicken Sie auf ein Skript oder auf eines seiner Assets, um das Skript zu fokussieren. Die Fokussierung eines Skripts in einer Lösung hat Auswirkungen auf das Verhalten von VuGen. Wenn Sie beispielsweise die Wiedergabe starten, wird das fokussierte Skript ausgeführt. Darüber hinaus zeigen die Menüoptionen, Schaltflächen der Symbolleisten und Bereiche Funktionen an, die für das Protokoll des Skripts relevant sind. Wenn das fokussierte Skript z. B. in Web HTTP/HTML aufgezeichnet wird, enthält die Symbolleiste die Schaltfläche **Aufzeichnungsoptionen**. Wird das fokussierte Skript dagegen in TruClient aufgezeichnet, enthält die Symbolleiste die Schaltfläche **Skript entwickeln**.

Doppelklicken Sie auf die Aktion des Skripts, um sie im Editor zu öffnen.

Außerdem können Sie Skripts (`<Skriptname>.usr`) aus dem Dateiverzeichnis ziehen und im **Lösung-Explorer** ablegen.

Zusätzliche Dateien

Zusätzliche Dateien, die durch das Skript aufgerufen werden, befinden sich im Knoten **Zusätzliche Dateien** unter **Lösungs-Explorer > <Skriptname>**. Sie können Kopfzeilendateien (`<Kopfzeilendateiname.h>` aus dem Dateiverzeichnis ziehen. Wenn Sie Dateien im Knoten **Zusätzliche Dateien** hinzufügen, werden diese in einem Lasterzeugungsszenario automatisch berücksichtigt.

Zusätzliche Dateien können folgende Informationen enthalten:

- Allgemeine Funktionen von Dienstprogrammen, die von dem Skript verwendet werden (z. B. Code)
- Definitionen von Konstanten und Variablen, die von dem Skript verwendet werden (z. B. Code)
- Spezielle Assets, die während der Skriptausführung verwendet werden (z. B. JPEG-Dateien)
- Datendateien, die während der Skriptausführung durch Skriptcode verändert wurden

Im Folgenden finden Sie Beispiele für Dateitypen, die als zusätzliche Dateien hinzugefügt werden können:

.ws,.h,.c.,dat,.ini,.vbs,.java,.js,.txt,.tux,.rec,.msc,.vdf,.xml,.xsl,.dtd,.html,.htm

Sie können zusätzliche Dateien im Editor bearbeiten, wenn der entsprechende Dateityp unter **Extras > Optionen > Skriptverwaltung in der Liste der Dateitypen, die im Editor bearbeitet werden können** aufgeführt ist. Doppelklicken Sie auf die zusätzliche Datei, um sie im Editor zu öffnen. Informationen zum Ändern der Liste finden Sie unter "Skriptoptionen" auf Seite 83.

Laufzeiteinstellungen

Der Zugriff auf die Laufzeiteinstellungen für ein bestimmtes Skript erfolgt über den Knoten **Laufzeiteinstellungen** in **Lösungs-Explorer > <Skript> > Laufzeiteinstellungen**. Weitere Informationen finden Sie unter "Laufzeiteinstellungen" auf Seite 330.

Parameter

Der Zugriff auf Parameter für ein bestimmtes Skript erfolgt über den Knoten **Parameter** in **Lösungs-Explorer > <Skript> > Parameter**. Weitere Informationen finden Sie unter "Parametrierung – Übersicht" auf Seite 436.

Wiedergabeausführungen

Über diesen Knoten können Sie für jede Iteration in der Wiedergabe auf die **Wiedergabeübersichten** zugreifen. Weitere Informationen finden Sie unter "Bereich "Lösungs-Explorer"" auf Seite 101.

Editor - Übersicht

Der VuGen-Editor ermöglicht Ihnen das Bearbeiten aufgezeichneter Skripts und anderer ergänzender Dateien, zum Beispiel Kopfzeilendateien. Sie können mehrere Dateien gleichzeitig öffnen, indem Sie einfach durch die Registerkarten navigieren.

Der Editor bietet die folgenden Bearbeitungsfunktionen:

Registerkartenansicht

Die Registerkartenansicht des Editors bietet Ihnen die Möglichkeit, mehrere Skript-Assets gleichzeitig zu organisieren und zu bearbeiten. Sie können im Editor folgende Aktionen ausführen:

- Doppelklicken auf ein Skript oder auf zusätzliche Dateien im Lösungs-Explorer, um die Datei im Editor-Bereich als Registerkarte zu öffnen. Der Registerkartenname lautet **<Skriptname>:<Dateiname>**
- Registerkarten ziehen, um diese neu zu sortieren.

Unterstützung mehrerer Programmiersprachen

Aufgezeichnete Skripts werden in der Programmiersprache C generiert, die uneingeschränkte Sprach- und Analyseunterstützung für VuGen bietet. Sie können Ihre Vuser-Skripts durch Hinzufügen von Standard-ANSI C-Funktionen erweitern. ANSI C-Funktionen ermöglichen es Ihnen, Kommentare, Ablaufanweisungen, bedingte Anweisungen usw. für Ihre Vuser-Skripts hinzuzufügen. Sie können Standard-ANSI C-Funktionen zu jedem Typ von Vuser-Skript hinzufügen. Weitere Informationen finden Sie unter "C-Vuser-Skripts" auf Seite 1065.

Darüber hinaus ermöglicht Ihnen der VuGen-Editor, manuelle Skripts mit den folgenden Programmiersprachen zu schreiben:

- Java

  Weitere Informationen finden Sie unter "Java-Vuser" auf Seite 1067.

- C#

  Weitere Informationen finden Sie unter ".NET-Vuser" auf Seite 1067.

- VB.NET

  Weitere Informationen finden Sie unter "Erstellen eines Vuser-Skripts in Visual Studio" auf Seite 1069.

Tipps für den Codeabschluss und QuickInfos für C-Skripts

Der Codeabschluss ermöglicht Ihnen das schnelle und genaue Schreiben von Code, da er eine Liste von Code-Elementen bereitstellt, aus der Sie eine Auswahl treffen können. Drücken Sie die Steuerungstaste und die Leertaste, um den Anweisungsabschluss zu aktivieren, wenn der Cursor sich im Editor befindet. QuickInfos mit Kontextinformationen werden angezeigt, wenn Sie den Mauszeiger auf ein Code-Element bewegen. In der folgenden Tabelle werden die verfügbaren Elemente zum Codeabschluss, Gültigkeitsbereiche, Identifizierungssymbole und QuickInfo-Kontexte beschrieben:

Element für Codeabschluss	Gültigkeitsbereich	Symbol	Quickinfo-Anzeige
ANSI-C-Schlüsselwörter und -Typen	Alle möglichen C-Standardschlüsselwörter.		Funktionstyp, Name und Parameter.
LR-API-Funktionen	Alle im Skript enthaltenen Schritte.		LR-API-Schritt

Element für Codeabschluss	Gültigkeitsbereich	Symbol	Quickinfo-Anzeige
LR-API-Konstanten	Als Trennzeichen für Parametergruppen in Schritten verwendet. Zum Beispiel ENDITEM		LR-API-Konstante
Benutzerfunktionen	Alle Funktionen, die Sie in Aktionsdateien definiert haben.		• Funktionstyp, Name und Parameter. • Bei Verwendung von Funktionen (Insight-Methode) Die erforderlichen Argumente, die bei der Definition jedes Argument markieren und zum nächsten Argument wechseln, wenn Sie das Trennzeichen eingeben.
Variablen	Lokale Variablen – werden nur in der Funktion angezeigt, in der sie definiert wurden. Globale Variablen – außerhalb eines Funktionstextes definiert. Überall im Skript verfügbar.		Typ und Name.
Parameter	Nur in dem Funktionstext verfügbar, in dem sie definiert wurden.		• Parametertyp und -name • Bei Verwendung von Parametern (Insight-Methode) Die erforderlichen Argumente, die bei der Definition jedes Argument markieren und zum nächsten Argument wechseln, wenn Sie das Trennzeichen eingeben.

Standardmäßig ist der Codeabschluss in VuGen global aktiviert. Um den Codeabschluss zu deaktivieren, wählen Sie **Extras > Text-Editor > Codeabschluss**. Deaktivieren Sie das Kontrollkästchen **Codeabschluss aktivieren**.

Skriptcode-Farben für C-Skripts

Um das Schreiben und Debuggen von Skripts zu erleichtern, werden Code-Elementtypen in einem identifizierenden Hinter- und Vordergrund farbig dargestellt. Der farbige Text ermöglicht Ihnen das einfache Lesen von Skripts und das Analysieren von Syntaxfehlern. Die folgende Tabelle enthält Beispiele für Code-Elementtypen und die ihnen zugewiesenen Farben.

Codetyp	Farbbeispiel
Kommentare	`/* Kommentar */ // Kommentar`
Schlüsselwörter	`if (a) { } else { }`
Methodenparametername	`foo( "parameter=wert" )`
LoadRunner-API	`web_url`
Methodenaufruf	`foo( )`
Zeichenfolge	`char * text = "Hallo, Welt!"`

Darüber hinaus können Sie Code-Elementtypen an Ihre Anforderungen anpassen, indem Sie **Extras > Optionen > Editor > Marker und Lineale** auswählen.

Skriptorganisation für C-Skripts

Die Skriptorganisation bietet Ihnen die Möglichkeit, Abschnitte eines Skripts selektiv auszublenden und anzuzeigen, um so die Verwaltung großer Skripts zu erleichtern, indem Sie nur diejenigen Abschnitte anzeigen, die Sie aktuell bearbeiten. Weitere Informationen finden Sie unter " Editor-Optionen" auf Seite 77.

Community-Suche

Sie können über die VuGen-Symbolleiste Web-Suchen durchführen, die im Editor eine Browser-Registerkarte öffnen. Die Standard-Website ist das LoadRunner-Forum, indem Sie nach Themen suchen, Fragen veröffentlichen oder Blogs über Ihre Kenntnisse schreiben können. Sie können zusätzliche Suche-Websites hinzufügen, indem Sie **Extras > Optionen > Allgemein > Community** auswählen. Details zum Hinzufügen zusätzlicher Websites finden Sie unter "Allgemeine Optionen" auf Seite 75.

Snapshot-Bereich - Übersicht

Was ist ein Snapshot?

In Snapshots sind die Daten enthalten, die vom Datenverkehr zwischen Client und Server generiert werden und die aufgezeichnet werden, wenn ein Skript aufgezeichnet und wenn es wiedergegeben wird.

Snapshots werden in verschiedenen Formaten angezeigt und bieten vom Vuser-Protokoll abhängige Funktionen:

Typ	Beschreibung	Snapshot-Funktion
Windows-Sockets	Text- und Hexadezimaldarstellung von gesendeten und empfangenen Datenpuffern.	Korrelation erstellen

Typ	Beschreibung	Snapshot-Funktion
Web HTTP/HTML	Text- und Hexadezimaldarstellung der Anforderung und der Antwort.	• Korrelation erstellen • Korrelationsregel erstellen • Parameter erstellen • Schritt zur Textüberprüfung hinzufügen
XML	XML-, Text- und Hexadezimaldarstellung des Anforderungs- und des Antworttexts.	• Korrelation einfügen • Korrelationsregel erstellen • Parameter erstellen • XML-Prüfung einfügen
Datenbank	Datenraster	• Korrelation erstellen • Raster in Datei speichern
Bildbasiert Protokolle, wie z. B. Click & Script.	Grafiken des Schritts.	

Hinweis: Der Snapshot-Bereich steht nicht für alle Vuser-Protokolle zur Verfügung. Nur bestimmte Vuser-Protokolle bieten Zugriff auf den Snapshot-Bereich.

Snapshot-Bereich

VuGen zeigt Aufzeichnungs- und Wiedergabe-Snapshots im Snapshot-Bereich an. Standardmäßig zeigt der Snapshot-Bereich nur einen Snapshot an. Um Snapshots zu vergleichen, können Sie den Snapshot-Bereich teilen und zwei Snapshots gleichzeitig anzeigen. Der Snapshot-Bereich kann vertikal oder horizontal geteilt werden. In jedem Abschnitt des geteilten Snapshot-Bereichs kann entweder ein Aufzeichnungs-Snapshot oder ein Wiedergabe-Snapshot angezeigt werden. Normalerweise zeigen Sie einen Aufzeichnungs-Snapshot und den zugehörigen Wiedergabe-Snapshot an. Auf diese Weise können Sie den Aufzeichnungs- mit dem Wiedergabe-Snapshot vergleichen.

- Details zum Arbeiten mit dem Snapshot-Bereich finden Sie unter "Arbeiten mit Snapshots" auf der nächsten Seite.
- Informationen zum Snapshot-Bereich finden Sie unter "Snapshot-Bereich" auf Seite 109.

Grundlegende Funktionen des Snapshot-Bereichs

Der für alle Vuser-Protokolle angezeigte Snapshot-Bereich enthält immer dieselben Grundfunktionen. Diese Grundfunktionen bieten folgende Möglichkeiten:

- Anzeigen eines Snapshots oder Teilen des Snapshot-Bereichs, um zwei Snapshots anzuzeigen. Sie können den Snapshot-Bereich vertikal oder horizontal teilen.
- Anzeigen von Aufzeichnungs- und Wiedergabe-Snapshots.

Details zur Verwendung der grundlegenden Funktionen des Snapshot-Bereichs finden Sie unter "Arbeiten mit Snapshots" auf der nächsten Seite.

Synchronisieren von Snapshots

Der für einige Vuser-Protokolle angezeigte Snapshot-Bereich erleichtert das Vergleichen von Snapshots durch "Synchronisieren" der beiden in diesem Bereich angezeigten Snapshots. Wenn Sie beim Synchronisieren von Grafik-Snapshots beispielsweise den Mauszeiger an einer bestimmten Position in einem der Snapshots platzieren, zeigt VuGen eine Markierung an der entsprechenden Position in dem anderen Snapshot an. Wenn Sie einen Snapshot im Snapshot-Bereich vertikal oder horizontal verschieben, verschiebt VuGen auch den anderen Snapshot entsprechend, sodass immer derselbe Abschnitt der Snapshots angezeigt wird.

Hinweis: Die Snapshot-Synchronisierung steht nur für bestimmte Vuser-Protokolle und nur für bestimmte Ansichten in diesen Protokollen zur Verfügung.

Kopieren von Snapshots in die Zwischenablage

Sie können einen bildbasierten Snapshot in die Zwischenablage kopieren. Auf diese Weise können Sie das Bild in eine Grafikanwendung importieren und dort analysieren und ändern.

Details zum Kopieren eines Snapshots in die Zwischenablage finden Sie unter "Arbeiten mit Snapshots" auf der nächsten Seite.

Hinweis: Die Funktion zum Kopieren von Snapshots in die Zwischenablage steht nur für RDP-, Citrix- und SAP GUI-Protokolle zur Verfügung.

Kopieren von Snapshot-Text in die Zwischenablage

Sie können Text aus einem Snapshot in die Zwischenablage kopieren. Anschließend können Sie den Text aus der Zwischenablage in eine andere Anwendung einfügen.

Details zum Kopieren von Snapshot-Text in die Zwischenablage finden Sie unter "Arbeiten mit Snapshots" auf der nächsten Seite.

Hinweis: Die Funktion zum Kopieren von Snapshot-Text in die Zwischenablage steht nur für Ajax Click & Script) und SAP Click & Script-Protokolle zur Verfügung.

Angepasste Funktionen des Snapshot-Bereichs

Neben den grundlegenden Funktionen des Snapshot-Bereichs bieten die Snapshot-Bereiche für einige Vuser-Protokolle angepasste Funktionen. Beispielsweise ermöglicht Ihnen der Snapshot-Bereich für RDP-Vuser-Skripts das Anzeigen von Snapshots in den Modi **Vollständig** oder **Bild**, während Sie im Snapshot-Bereich für Winsock-Vuser-Skripts Snapshots in den Modi **Text** oder **Hex** anzeigen können. Die Steuerungen für die angepassten Funktionen finden Sie in den Symbolleisten des Snapshot-Bereichs.

Snapshot bei Fehler

Neben der Anzeige von Aufzeichnungs- und Wiedergabe-Snapshots können Sie im Snapshot-Bereich auch Snapshots von Fehlern anzeigen, die während der Wiedergabe eines Skripts aufgetreten sind. Die Funktion zum Erstellen von Snapshots bei Fehlern ist nur für bestimmte Vuser-Protokolle verfügbar.

Sie generieren und zeigen Snapshots von Fehlern nur an, wenn die Funktion zum Erstellen von Snapshots bei Fehlern aktiviert ist. Details zum Aktivieren der Funktion zum Aufzeichnen von Snapshots bei Fehlern finden Sie unter "Arbeiten mit Snapshots" unten.

Vergleichen von Snapshots

Mit der Schaltfläche **Vergleichen** im Snapshot-Bereich können Sie zwei Snapshots vergleichen. Um die Funktion **Vergleichen** zu aktivieren, müssen Sie zunächst den Snapshot-Bereich teilen, um zwei Snapshots anzuzeigen. Standardmäßig verwendet VuGen das Dienstprogramm *WDiff*, um Snapshots zu vergleichen. Sie können ein alternatives Vergleichstool angeben, wie unter "Skriptoptionen" auf Seite 83 beschrieben.

Hinweis: Die Snapshot-Vergleichsfunktion steht nur für Web (HTTP/HTML)- und Web Services-Protokolle zur Verfügung.

Festlegen der Snapshot-Optionen

In VuGen können Sie verschiedene Optionen festlegen, um zu definieren, wie Snapshots im Snapshot-Bereich angezeigt werden. Details zu diesen Snapshot-Optionen finden Sie unter "Skriptoptionen" auf Seite 83.

Arbeiten mit Snapshots

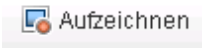
In diesem Abschnitt wird die Verwendung der grundlegenden Funktionen des Snapshot-Bereichs beschrieben. Eine Übersicht zu den Snapshot-Funktionen finden Sie unter "Snapshot-Bereich - Übersicht" auf Seite 62.

Anzeigen des Snapshot-Bereichs

Führen Sie eine der folgenden Aktionen aus, um den Snapshot-Bereich anzuzeigen:

- Wählen Sie **Ansicht > Snapshot** aus.
- Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Snapshot** .
- Klicken Sie im Editor in einen Schritt, der eine Referenz zu einem Snapshot enthält.
- Doppelklicken Sie im Schrittenavigator auf einen Schritt, der eine Referenz zu einem Snapshot enthält. Beachten Sie, dass im Schrittenavigator jeder Schritt, der einen Snapshot enthält, ein Snapshot-Symbol aufweist. Sie können Ihren Mauszeiger auf dem Snapshot-Symbol platzieren, um eine Miniaturansicht des Snapshots anzuzeigen.

Anzeigen eines Aufzeichnungs-Snapshots im Snapshot-Bereich

Klicken Sie in der Symbolleiste des Snapshot-Bereichs auf die Schaltfläche **Aufzeichnungs-Snapshot** 

.

Anzeigen eines Wiedergabe-Snapshots im Snapshot-Bereich

Klicken Sie in der Symbolleiste des Snapshot-Bereichs auf die Schaltfläche **Wiedergabe-Snapshot**

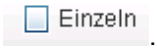
. Wenn für den Schritt mehr als ein Wiedergabe-Snapshot vorhanden ist, klicken Sie auf **Wiederholung** und wählen Sie die erforderliche Iterationsnummer aus.

Teilen des Snapshot-Bereichs, um zwei Snapshots anzuzeigen

Wenn im Snapshot-Bereich nur ein Snapshot angezeigt wird, klicken Sie in der Symbolleiste des Snapshot-Bereichs auf die Schaltfläche zum Teilen des Snapshot-Bereichs

.

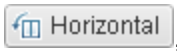
Anzeigen nur eines Snapshots im Snapshot-Bereich

Wenn im Snapshot-Bereich zwei Snapshots angezeigt werden, klicken Sie in der Symbolleiste des Snapshot-Bereichs auf die Schaltfläche **Einzeln** 

.

Wechseln zwischen einer horizontalen und vertikalen Teilung des Snapshot-Bereichs

Wenn im Snapshot-Bereich zwei Snapshots angezeigt werden:

- Klicken Sie in der Symbolleiste des Snapshot-Bereichs auf die Schaltfläche für die horizontale Teilung 

, um zwei Snapshots nebeneinander anzuzeigen.
- Klicken Sie in der Symbolleiste des Snapshot-Bereichs auf die Schaltfläche für die vertikale Teilung

, um zwei Snapshots übereinander anzuzeigen.

Synchronisieren der Anzeige zweier Snapshots, wenn der Snapshot-Bereich geteilt ist

1. Vergewissern Sie sich, dass der Snapshot-Bereich geteilt ist und zwei Snapshots anzeigt.
2. Klicken Sie in der Symbolleiste des Snapshot-Bereichs auf die Schaltfläche **Synchronisierung** .

Hinweis: Die Snapshot-Synchronisierung steht nur für bestimmte Vuser-Protokolle und nur für bestimmte Ansichten in diesen Protokollen zur Verfügung.

Kopieren eines Snapshots in die Zwischenablage

1. Zeigen Sie den Snapshot im Snapshot-Bereich an.
2. Klicken Sie mit der rechten Maustaste auf den Snapshot und wählen Sie **Bild in Zwischenablage kopieren** aus.

Hinweis: Die Funktion zum Kopieren von Snapshots in die Zwischenablage steht nur für RDP-, Citrix- und SAP-Vuser-Skripts zur Verfügung.

Kopieren von Snapshot-Text in die Zwischenablage

1. Zeigen Sie den Snapshot im Snapshot-Bereich an.
2. Markieren Sie den Text, den Sie kopieren möchten.
3. Klicken Sie mit der rechten Maustaste in den markierten Text und wählen Sie **Auswahl kopieren** aus.

Aktivieren der Funktion zur Snapshot-Erstellung bei Fehlern

1. Klicken Sie auf **Wiedergeben > Laufzeiteinstellungen**. Das Dialogfeld **Laufzeiteinstellungen** wird geöffnet.
2. Klicken Sie unter **Allgemein** auf **Verschiedenes**.
3. Aktivieren Sie im Bereich **Fehlerbehandlung** das Kontrollkästchen **Bei Fehler Snapshot erzeugen**.

Festlegen der Snapshot-Optionen

1. Klicken Sie auf **Extras > Optionen**. Das Dialogfeld **Optionen** wird geöffnet.
2. Klicken Sie auf **Skripterstellung** und dann auf **Snapshot**. Die Snapshot-Optionen werden auf der rechten Seite des Dialogfelds angezeigt.

Hinzufügen einer Textprüfung aus der XML-Ansicht im Snapshot-Bereich

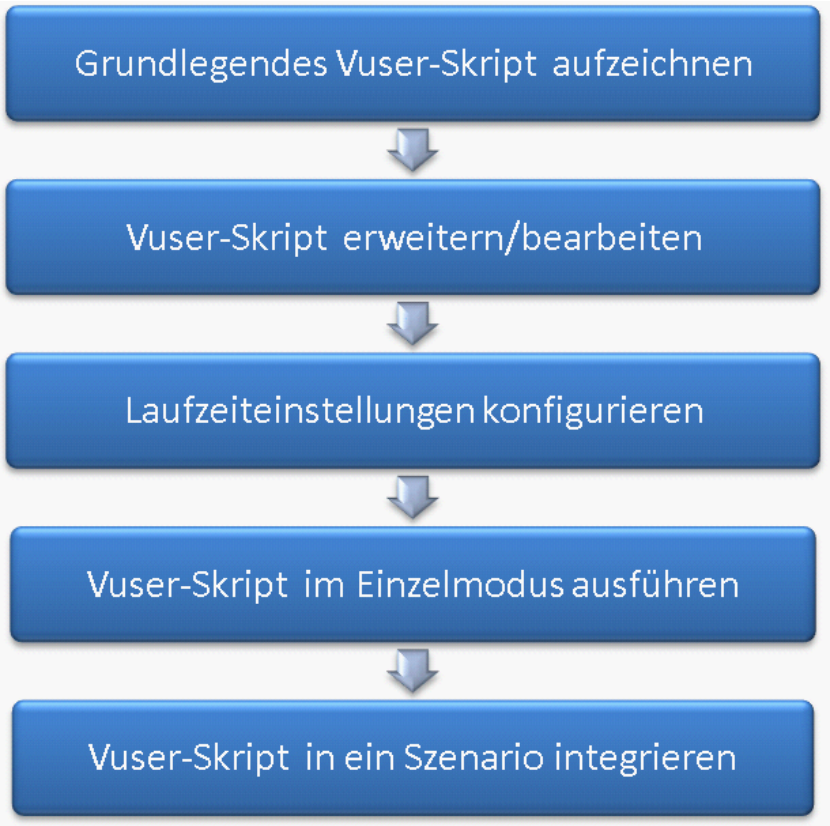
Nachdem Sie ein Vuser-Skript aufgezeichnet haben, können Sie aus der XML-Ansicht im Snapshot-Bereich eine Textprüfung hinzufügen. Details zur XML-Ansicht im Snapshot-Bereich finden Sie unter Snapshots That Have an XML View.

So fügen Sie eine Textprüfung aus der XML-Ansicht im Snapshot-Bereich hinzu:

1. Klicken Sie auf **Ansicht > Snapshot** oder klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Snapshot-Bereich anzeigen** .
2. Zeigen Sie im Snapshot-Bereich einen Snapshot an, der den zu verifizierenden Text enthält.
3. Klicken Sie auf der rechten Seite des Snapshot-Bereichs auf die Registerkarte **XML-Ansicht**.
4. Klicken Sie im Snapshot-Bereich auf die Registerkarte **Antworttext**.
5. Navigieren Sie in der Strukturansicht oder in der Tabellenansicht zu der zu verifizierenden Textzeichenfolge und markieren Sie diese.
6. Klicken Sie mit der rechten Maustaste in die Auswahl und wählen Sie **Schritt zur Textüberprüfung hinzufügen** aus. Das Dialogfeld **Text suchen** wird geöffnet.
7. Ändern Sie die Optionen im Dialogfeld **Text suchen**. Details zu den Optionen des Dialogfelds erhalten Sie, indem Sie auf F1 drücken, um die Function Reference zu öffnen.
8. Klicken Sie auf **OK**, um einen **web_reg_find**-Schritt in das Vuser-Skript einzufügen.

Erstellen eines Vuser-Skripts - Workflow

In der folgenden Abbildung wird der Entwicklungsprozess für ein Vuser-Skript dargestellt:



Ein Vuser-Skript wird folgendermaßen erstellt:

1. Zeichnen Sie ein Basisskript mit VuGen auf.
2. Erweitern Sie das Basisskript, indem Sie Ablaufsteuerungsanweisungen und andere LoadRunner-API-Funktionen zum Skript hinzufügen.
3. Konfigurieren Sie die Laufzeiteinstellungen. Diese Einstellungen beinhalten Iteration, Protokoll und Timing und definieren das Verhalten der Vuser bei der Skriptausführung.
4. Überprüfen Sie die Funktionalität des Skripts, indem Sie es im Einzelmodus ausführen.
5. Nachdem Sie die Funktionalität des Skripts geprüft haben, können Sie es in Ihre Umgebung integrieren: ein LoadRunner-Szenario, ein Performance Center-Lasttest oder eine Business Process Monitor-Konfiguration.

Vergleichen von Skripts nebeneinander

Vuser-Skripts können mithilfe des Vergleichstools verglichen und nebeneinander dargestellt werden.

So vergleichen Sie Vuser-Skripts:

1. Klicken Sie mit der rechten Maustaste im **Lösungs-Explorer** auf das primäre Skript und wählen Sie **Als erstes zu vergleichendes Objekt festlegen** aus.
2. Klicken Sie mit der rechten Maustaste im **Lösungs-Explorer** auf das sekundäre Skript und wählen Sie **Vergleichen** aus, um die Vergleichsfunktion auszuführen.

   oder
3. Klicken Sie mit der rechten Maustaste und wählen Sie **Mit externem Objekt vergleichen**.

   Sie können ein Asset mit einer Datei außerhalb der Lösung vergleichen. Diese Option legt das markierte Asset als primäres Asset fest und öffnet Windows Explorer, damit Sie das sekundäre Asset auswählen können.

Hinweis: Sie können das Vergleichstool unter **Optionen > Skripterstellung > Vergleich** ändern. Weitere Informationen finden Sie unter "Skriptoptionen" auf Seite 83.

Erstellen eines Geschäftsprozessberichts

In der letzten Phase der Skripterstellung können Sie einen Bericht erstellen, der Ihren Geschäftsprozess beschreibt. VuGen exportiert die Skriptinformationen in eines der folgenden Formate:

- Microsoft Word
- Acrobat PDF
- HTML

Sie können eine vordefinierte oder eine mit VuGen bereitgestellte Vorlage verwenden, um Berichte mit Zusammenfassungsinformationen über Ihren Testlauf zu erstellen. Die VuGen-Vorlage liegt im Docx-Format von Microsoft Word 2007 vor. Sie können sie bearbeiten und an Ihre Erfordernisse anpassen.

Über VuGen können Sie den Inhalt des Berichts anpassen, indem Sie angeben, welche Art von Informationen Sie einfügen möchten.

Hinweis: Geschäftsprozessberichte sind für die folgenden Protokolle verfügbar: Ajax (Click & Script), TruClient, Citrix ICA, Oracle NCA, Oracle Web Applications 11i, RDP, SAP (Click & Script), SAP GUI, SAP - Web, Web (HTTP/HTML) und Web Services-Protokolle.

1. ### Erstellen eines Geschäftsprozessberichts

   Wählen Sie **Extras > Geschäftsprozessbericht** und nehmen Sie die Einstellungen im Dialogfeld vor. Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld

"Geschäftsprozessbericht"" auf Seite 97.

2. **Konfigurieren von erweiterten Optionen**

   Klicken Sie zum Ändern der erweiterten Berichtsoptionen wie dem Inhaltsverzeichnis, Snapshots und der Dokumentvorlage auf die Schaltfläche **Mehr**. Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Geschäftsprozessbericht"" auf Seite 97.

Ändern des VuGen-Layouts

Das VuGen-Fenster besteht aus mehreren Abschnitten. Jeder Abschnitt kann mehrere Bereiche umfassen, wie den Fehlerbereich und den Snapshot-Bereich. Wenn ein Abschnitt mehr als einen Bereich enthält, werden die Bereiche innerhalb des Abschnitts als Registerkarten angezeigt. In diesem Abschnitt wird beschrieben, wie Sie die im VuGen-Fenster angezeigten Fensterabschnitte und Bereiche anpassen und ändern.

Verschieben eines Bereichs in einen neuen Fensterabschnitt

Sie können beliebige VuGen-Bereiche in einen neuen Abschnitt verschieben. Der neue Abschnitt kann ein Teil eines vorhandenen Abschnitts sein oder die gesamte linke, rechte, obere oder untere Fläche des VuGen-Fensters einnehmen.

Ziehen Sie im VuGen-Fenster die Titelleiste oder Registerkarte des zu verschiebenden Bereichs an eine neue Position. (Wenn der gewünschte Bereich im VuGen-Fenster nicht angezeigt wird, können Sie diesen im Menü **Ansicht** auswählen.) Wenn Sie den Bereich über die Abschnitte des VuGen-Fensters ziehen, wird in der Mitte des aktiven Abschnitts eine komplexe Markierung und an jeder Kante des VuGen-Fensters eine einfache Markierung angezeigt.

Hinweis: Wenn Sie die Titelleiste eines Abschnitts mit mehreren Bereichen in Registerkarten ziehen, werden alle Bereiche in diesem Abschnitt in den neuen Abschnitt verschoben.

Markierungstyp	Markierung	Beschreibung
Komplexe Markierung - Aktueller Abschnitt		Positionen im ausgewählten Bereich eines neuen Abschnitts. Der neue Abschnitt wird über oder unter dem **aktiven Abschnitt** bzw. links oder rechts von ihm erstellt, abhängig von der Pfeilmarkierung, die Sie beim Loslassen der Maustaste auswählen.

Markierungstyp	Markierung	Beschreibung
Einfache Markierung - VuGen-Fenster		Positionen im ausgewählten Bereich eines neuen Abschnitts. Der neue Abschnitt wird über oder unter dem **VuGen-Fenster** bzw. links oder rechts von ihm erstellt, abhängig von der Pfeilmarkierung, die Sie beim Loslassen der Maustaste auswählen.

Verschieben eines Bereichs in einen vorhandenen Fensterabschnitt

Sie können beliebige VuGen-Bereiche aus einem Abschnitt in einen neuen Abschnitt verschieben. Wenn ein Abschnitt mehr als einen Bereich enthält, werden die Bereiche innerhalb des Abschnitts als Registerkarten angezeigt.

1. Ziehen Sie im VuGen-Fenster die Titelleiste oder Registerkarte des zu verschiebenden Bereichs an eine neue Position. (Wenn der gewünschte Bereich im VuGen-Fenster nicht angezeigt wird, können Sie diesen im Menü **Ansicht** auswählen.) Wenn Sie den Bereich über die Abschnitte im VuGen-Fenster ziehen, wird in der Mitte des aktiven Abschnitts eine komplexe Markierung angezeigt.

2. Bewegen Sie den Cursor über die mittlere Schaltfläche der komplexen Markierung. Wenn Sie die Maustaste loslassen, wird der ausgewählte Bereich in Form einer Registerkarte zu dem ausgewählten Abschnitt hinzugefügt.

3. Wiederholen Sie diesen Vorgang für jeden Bereich, den Sie verschieben möchten.

Hinweis: Wenn Sie die Titelleiste eines Abschnitts mit mehreren Bereichen in Registerkarten ziehen, werden alle Bereiche in diesem Abschnitt in den ausgewählten Abschnitt verschoben.

Freischwebende Bereiche und Andocken von Bereichen

Angedockte Bereiche sind an einer Position innerhalb des VuGen-Fensters fest verankert. Wenn Sie beispielsweise einen Bereich an eine Position verschieben, die durch eine Markierung gekennzeichnet ist, wird dieser Bereich in dieser Position angedockt.

Freischwebende bzw. unverankerte Bereiche werden über allen anderen Fenstern angezeigt. Freischwebende Bereiche können an eine beliebige Position auf Ihrem Bildschirm gezogen werden, auch außerhalb des VuGen-Fensters. Freischwebende Bereiche besitzen eine eigene Titelleiste.

- Um einen Bereich schweben zu lassen, klicken Sie mit der rechten Maustaste auf die Titelleiste und wählen die entsprechende Option, um die Verankerung des Bereichs aufzuheben. Der Bereich wird über allen anderen Fenstern und Bereichen mit einer eigenen Titelleiste angezeigt.
- Um einen Bereich anzudocken, doppelklicken Sie auf seine Titelleiste oder klicken mit der rechten Maustaste auf seine Titelleiste und wählen **Als Dokument im Registerkartenformat andocken** aus. Der Bereich wird wieder an seiner vorherigen Position im VuGen-Fenster angezeigt.

Optionen - Übersicht

Im Dialogfeld **Optionen** sind die VuGen-Anwendungseinstellungen zusammengefasst. Die Einstellungen gelten für alle in VuGen verfügbaren Protokolle.

Die Einstellungen werden bei der entsprechenden Kategorie ausführlich beschrieben.

In der folgenden Tabelle sind die Kategorien und zugehörigen Bereiche des Dialogfelds **Optionen** aufgelistet.

Allgemein	• In der **Aufgabenliste** können Sie Kommentartags hinzufügen, ändern und löschen. • Im Bereich **Skripts und Lösungen** können Sie die Projekteinstellungen konfigurieren. • Im Bereich **Community** können Sie Community-Suchwebsites hinzufügen, ändern und löschen. Weitere Informationen finden Sie unter "Allgemeine Optionen" auf Seite 75.

Editor	• Unter **Allgemein** können Sie die Schriftart und andere allgemeine Optionen konfigurieren. • Unter **Marker und Lineale** können Sie die Marker und Lineale für den im Skript enthaltenen Text konfigurieren. • Im Bereich **Verhalten** konfigurieren Sie die Einstellungen, die sich auf Registerkarten, Tastatur und Maus beziehen. • Unter **Codefarbe** können Sie die Schriftattribute für Überschrifts-, Steuer- und Texttyp jeder Sprache konfigurieren. • Unter **Codeabschluss** können Sie die Einstellungen für den Codeabschluss konfigurieren. Weitere Informationen finden Sie unter " Editor-Optionen" auf Seite 77.
Skripterstellung	• Im Bereich **UI-Aufzeichnung** können Sie die Einstellungen konfigurieren, die sich auf die Aufzeichnungsoberfläche beziehen, z. B. die beim Aufzeichnen verwendete freischwebende Symbolleiste. • Im Bereich **Wiedergabe** konfigurieren Sie die Einstellungen für die Skriptwiedergabe. • Im Bereich **Skriptverwaltung** können Sie die Liste der Dateierweiterungen bearbeiten. • Im Bereich **Vergleich** geben Sie den Speicherort der Vergleichsanwendung ein. • Im Bereich **Schrittenavigator** können Sie Hintergrund- und Vordergrundfarbe des Bereichs **Schrittenavigator** auswählen. • Im Bereich **Miniaturansichten** konfigurieren Sie den **Miniaturansicht-Explorer**. • Unter **Ausgabebereich** wird die Darstellung des **Ausgabebereichs** konfiguriert. • Unter **Snapshot** konfigurieren Sie die Darstellung der Snapshots in Ihrem Skript. • Im Bereich **Parameter** können Sie das Trennzeichen definieren, das für Parameterklammern im Skript verwendet wird. • Unter **Parser** wird der Parser für die Programmiersprache C aktiviert. Weitere Informationen finden Sie unter "Skriptoptionen" auf Seite 83.

Allgemeine Optionen

In diesem Bereich können Sie allgemeine Optionen der Benutzeroberfläche konfigurieren.

Aufgabenliste

Zugriff	**VuGen > Extras > Optionen > Allgemein > Aufgabenliste**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Kommentartags	In diesem Bereich können Sie Namen von Tags hinzufügen, löschen oder ändern, die Sie zum Benennen von Kommentaraufgaben in Ihren Skripts verwenden. **Markierungen.** Liste der verfügbaren Tags. **Name.** Zeigt den Namen des hervorgehobenen Tags in der Token-Liste an. In diesem Bereich können Sie die aktuelle Markierung ändern, hinzufügen oder löschen. **Hinzufügen.** Ermöglicht das Hinzufügen eines Tags zur Token-Liste. **Bearbeiten.** Ermöglicht Ihnen, den Namen des Tags in der Token-Liste zu ändern. **Löschen.** Ermöglicht Ihnen, ein Tag aus der Token-Liste zu entfernen.

Skripts und Lösungen

Zugriff	**VuGen > Extras > Optionen > Allgemein > Skripts und Lösungen**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Einstellungen	**Standardprojektspeicherort** Ermöglicht Ihnen, einen Pfad zu Ihren gespeicherten Projekten festzulegen. Standardspeicherort = `C:\Users\<Benutzername>\Documents\SharpDevelop Projects` **Beim Start vorherige Lösung laden** Ermöglicht Ihnen, die vorherige Lösung automatisch zu laden. Diese Option ist standardmäßig aktiviert.

Startseiteneinstellungen	**Startseite beim Start anzeigen** Diese Option ist standardmäßig aktiviert. **Startseite nach dem Laden des Skripts schließen** Diese Option ist standardmäßig aktiviert.

Menü "ALM"

Zugriff	**VuGen > Extras > Optionen > Allgemein > ALM**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Verbindung mit ALM im CAC-Modus herstellen	Stellt die Verbindung mit HP Application Lifecycle Management mithilfe einer CAC (Common Access Card) her. Hierdurch können Sie sich anmelden, ohne einen Benutzernamen oder ein Kennwort angeben zu müssen.

Community

Zugriff	**VuGen > Extras > Optionen > Allgemein > Community**
Siehe auch	Details finden Sie im Abschnitt zur Community-Suche unter "Editor - Übersicht" auf Seite 59.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Community-Suchwebsites	

	Fügt eine neue Suchwebsite zur Liste der Community-Suchwebsites hinzu. • **Name**: Ermöglicht Ihnen das Festlegen eines Namens der Suchwebsite, der in der VuGen-Symbolleiste angezeigt wird. • **URL**: Ermöglicht Ihnen, den URL der Suchwebsite anzugeben. Beispiel-URLs http://www.bing.com/search?q=%QUERY% http://www.google.com/search?q=%QUERY% http://www.google.de/search?q=%QUERY% (lokalisierte Google-Website, z. B. de für Deutschland) http://en.wikipedia.org/wiki/%QUERY%
	Ermöglicht Ihnen, die Eigenschaften der benutzerdefinierten Suchwebsite zu bearbeiten.
	Löscht die Suchwebsite aus der Liste der verfügbaren Websites.
	Verschiebt die Suchwebsite in der Liste der verfügbaren Websites nach unten.
	Verschiebt die Suchwebsite in der Liste der verfügbaren Websites nach oben.

Editor-Optionen

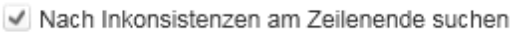
In diesem Bereich können Sie die Optionen des Texteditors konfigurieren.

Allgemein

Zugriff	**VuGen > Extras > Optionen > Editor > Allgemein**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Schriftart	
Textschriftart	Ermöglicht das Auswählen der Schriftart.
Größe	Ermöglicht das Auswählen der Schriftgröße.

Allgemeine Optionen	
Zeilenumbruch	Bricht den Text automatisch in die nächste Zeile um. Diese Option ist standardmäßig deaktiviert.
Klassen-/Funktionsbrowser anzeigen	Klassen-/Funktionsbrowser anzeigen oder ausblenden. Bei Aktivierung dieser Option können Sie schnell zu einer bestimmten Klasse oder Funktion im Skript navigieren, indem Sie diese aus der Dropdownliste im Browser auswählen. Diese Option ist standardmäßig aktiviert.
Zeilennummern anzeigen	Aktiviert die Zeilennummerierung des Skripts im Editor. Diese Option ist standardmäßig aktiviert.
Nach Inkonsistenzen am Zeilenende suchen	Aktivieren Sie den Editor, um nach Inkonsistenzen am Zeilenende Ihres Skripts zu suchen. Wenn diese Option aktiviert ist, wird im Skripteditor ein Dienstprogramm angezeigt, mit dem Sie die Zeilenenden entweder nach einem Windows-Standard (CRLF) oder nach einem Linux-Standard (LF) normalisieren können.  Diese Option ist standardmäßig aktiviert.
URL-Hypertextlinks im Editor aktivieren	Aktiviert URLs in Skripts, sodass diese wie Hypertext-Links funktionieren. Durch das Deaktivieren dieser Option kann die Leistung gesteigert werden. Diese Option ist standardmäßig aktiviert.

Marker und Lineale

Zugriff	**VuGen > Extras > Optionen > Editor > Marker und Lineale**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Marker und Lineale	
Leerzeichen anzeigen	Aktiviert Markierungen, die vorhandene Tabstopps anzeigen. Diese Option ist standardmäßig deaktiviert.
Tabstopps anzeigen	Aktiviert Markierungen, die das Zeilenende anzeigen. Diese Option ist standardmäßig deaktiviert.

Zeilenendemarker anzeigen	Aktiviert Markierungen, die das Zeilenende anzeigen. Diese Option ist standardmäßig deaktiviert.
Fehler unterstreichen	Aktiviert das Unterstreichen von Fehlern. Diese Option ist standardmäßig aktiviert.
Übereinstimmende Klammer markieren	Aktiviert das Markieren übereinstimmender Klammern. Diese Option ist standardmäßig aktiviert.
Übereinstimmende Vorkommen von Nicht-Schlüsselwörtern markieren	Wenn diese Option aktiviert ist, können Sie ein Nicht-Schlüsselwort in Ihrem Skript auswählen, und der Editor markiert alle anderen Vorkommen in Ihrem Skript. `class MyClass { void foo() { int i; for (i = 0; i < 10; i++) { } int index; for(index = 0; index < 10; index++) while(true) { } while(true) { } } }` Diese Option ist standardmäßig aktiviert.

Verhalten

Zugriff	**VuGen > Extras > Optionen > Editor > Verhalten**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Tabstopps	
Einzug	Ermöglicht das Festlegen des Einzugs von Tabstopps. Der Standardeinzug beträgt vier Leerzeichen.

Tabstopps in Leerzeichen konvertieren	Konvertiert Tabstopps in Leerzeichen. Diese Option ist standardmäßig deaktiviert.
Intelligenten Einzug verwenden	Wendet das Einzugsformat aus der vorherigen Zeile automatisch an. Diese Option ist standardmäßig aktiviert.
Verhalten	
Zoomen mit Mausrad aktivieren	Ermöglicht die Verwendung des Mausrades für die Zoom-Funktion. Diese Option ist standardmäßig aktiviert.
Ganze Zeile ausschneiden oder kopieren, wenn keine Markierung festgelegt wurde	Ermöglicht Ihnen, eine ganze Zeile im Editor auszuschneiden oder zu kopieren, vorausgesetzt, der Cursor befindet sich in der betreffenden Zeile. Diese Option ist standardmäßig aktiviert.
STRG+Klick-Tastenkombination für "Zur Definition wechseln" aktivieren	Aktiviert die Tastenkombination STRG+Klick für "Zur Definition wechseln". Dies ermöglicht Ihnen, den Cursor zur Definition einer Funktion zu bewegen, die Sie im Editor markiert haben. Diese Option ist standardmäßig aktiviert.

Codefarbe

Zugriff	**VuGen > Extras > Optionen > Editor > Markieren**
Wichtige Informationen	Markieren-Optionen bieten die Möglichkeit, die Farbe von Skriptelementen anzupassen.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

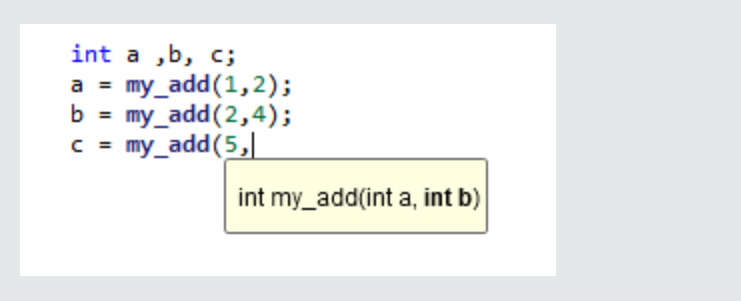
Element der Oberfläche	Beschreibung
<Sprachauswahl>	Dropdownliste mit Skriptsprachen, deren Darstellung Sie anpassen können: • C# • HTML • VuGen Script • XML
<Elementauswahl>	Liste von Code-Elementen, deren Darstellung Sie anpassen können.

Vordergrundfarbe	Ermöglicht das Auswählen einer Farbe aus der Farbpalette. Die ausgewählte Farbe wird auf den Vordergrund des Code-Elements angewendet.
Hintergrundfarbe	Ermöglicht das Auswählen einer Farbe aus der Farbpalette. Die ausgewählte Farbe wird auf den Hintergrund des Code-Elements angewendet.

Codeabschluss

Zugriff	**VuGen > Extras > Optionen > Editor > Codeabschluss**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Codeabschlussfunktionen aktivieren	Die Funktionen für den Codeabschluss sind aktiviert. Weitere Informationen zum Codeabschluss finden Sie unter "Editor - Übersicht" auf Seite 59. Diese Option ist standardmäßig aktiviert.
Syntax-QuickInfo aktivieren	Aktiviert QuickInfos, die Funktionsargumente anzeigen. Jedes Argument wird beim Definieren markiert, wenn das Trennzeichen eingegeben wird, verschiebt sich die Markierung auf das nächste Argument.  Diese Option ist standardmäßig aktiviert.
Beim Zeigen mit dem Mauszeiger auf eine ID QuickInfo anzeigen	Wenn diese Option aktiviert ist, wird eine Beschreibung der Code-Elemente angezeigt, wenn Sie den Mauszeiger über die ID bewegen. Diese Option ist deaktiviert, wenn **QuickInfo nur im Debugmodus anzeigen** aktiviert ist. Diese Option ist standardmäßig aktiviert.
QuickInfo nur im Debugmodus anzeigen	QuickInfos werden nur im Debugmodus angezeigt. Diese Option ist standardmäßig deaktiviert.
In die Codeabschlussliste einbeziehen	

ANSI-C-Schlüsselwörter	Ermöglicht das Einschließen von ANSI-C-Schlüsselwörtern in die Codeabschlussliste. Diese Option ist standardmäßig aktiviert.
LoadRunner-API-Schritte	Ermöglicht das Einschließen von LoadRunner-API-Schritten in die Codeabschlussliste. Diese Option ist standardmäßig aktiviert.
LoadRunner-API-Konstanten	Ermöglicht das Einschließen von LoadRunner-API-Konstanten in die Codeabschlussliste. Diese Option ist standardmäßig deaktiviert.
Benutzerdefinierte Funktionen	Ermöglicht das Einschließen benutzerdefinierter Funktionen in die Codeabschlussliste. Diese Option ist standardmäßig aktiviert.
Funktionsparameter, lokale und globale Variablen	Ermöglicht das Einschließen von Funktionsparametern sowie lokaler und globaler Variablen in die Codeabschlussliste. Diese Option ist standardmäßig aktiviert.

Organisation

Zugriff	**VuGen > Extras > Optionen > Editor > Organisation**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Organisationsfunktionen aktivieren	Diese Option ermöglicht das Erweitern und Ausblenden von Skriptabschnitten. Diese Option ist standardmäßig aktiviert.
Organisation von Schritten aktivieren	Diese Option ermöglicht das Erweitern und Ausblenden von Skriptschritten. Diese Option ist standardmäßig deaktiviert.
Wenn Schritt länger ist als [ ] Zeichen	Ermöglicht das Definieren der Zahl von Zeichen in einem Schritt, bevor das Organisieren implementiert wird. Diese Option ist standardmäßig deaktiviert.
Wenn Schritt aus mehr besteht als [ ] Zeilen	Ermöglicht das Definieren der Zahl von Zeilen in einem Schritt, bevor das Organisieren implementiert wird. Diese Option ist standardmäßig deaktiviert.

Skriptoptionen

In diesem Bereich können Sie die mit dem Aufzeichnen, Wiedergeben und Debuggen von Skripts verbundenen Optionen konfigurieren.

UI-Aufzeichnung

Zugriff	**VuGen > Extras > Optionen > Skripterstellung > UI-Aufzeichnung**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Transparenzmodus für unverankerte Symbolleiste "Aufzeichnen" aktivieren	Zeigt eine transparente, unverankerte Aufzeichnungssymbolleiste an. Wenn Sie auf die Symbolleiste klicken oder den Mauszeiger auf die Symbolleiste bewegen, wird diese deckend. Diese Option ist standardmäßig deaktiviert.
Schaltfläche 'Aufzeichnung abbrechen' aktivieren	Aktiviert die Schaltfläche **Aufzeichnung abbrechen** auf der freischwebenden Aufzeichnungs-Symbolleiste. **Hinweis:** Wenn die Schaltfläche **Aufzeichnung abbrechen** aktiviert ist, kann es beim Start einer Aufzeichnung in ein vorhandenes Skript zu einer Verzögerung kommen. Die Verzögerung tritt auf, während VuGen eine Kopie des vorhandenen Skripts erstellt. Diese Option ist standardmäßig aktiviert.
Dialogfeld "Aufzeichnung starten" öffnen, wenn neues Skript erstellt wurde	Öffnet das Dialogfeld **Aufzeichnung starten** automatisch, wenn ein neues Skript erstellt wurde. Diese Option ist standardmäßig deaktiviert.
Transaktionen automatisch schließen	Aktivieren Sie diese Funktion, wenn für geöffnete Transaktionen vor dem Aufzeichnen einer weiteren Aktion ein Schritt zum Beenden der Transaktion eingefügt werden soll. Diese Option ist standardmäßig deaktiviert.

Wiedergabe

Zugriff	**VuGen > Extras > Optionen > Skripterstellung > Wiedergeben**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Layout	**Layout während der Wiedergabe nicht ändern**. Standardmäßig wechselt VuGen bei der Wiedergabe eines Skripts automatisch das UI-Layout in das Layout **Debuggen**. Mit dieser Option können Sie Ihr ausgewähltes Layout während der Wiedergabe beibehalten. Diese Option ist standardmäßig deaktiviert.
Animierte Ausführung	**Animierte Ausführung** Sie können ein Vuser-Skript im animierten oder nicht animierten Modus ausführen. Wenn Sie das Skript im animierten Modus ausführen, markiert VuGen die im Vuser-Skript ausgeführte Zeile. Beim Ausführen im nicht animierten Modus führt VuGen das Vuser-Skript aus, ohne jedoch die ausgeführte Zeile zu markieren. Diese Option ist standardmäßig aktiviert. **Verzögerung bei animierter Ausführung** Sie können eine Verzögerung für die Markierung in der animierten Ausführung festlegen, damit Sie die Auswirkungen jedes Schrittes besser nachvollziehen können. Sie legen die Verzögerung in Millisekunden fest. Die Standardverzögerung beträgt 1. **Funktionen nur in den Aktionsabschnitten animieren** Animiert nur den Inhalt der Aktionsabschnitte, jedoch nicht die init- oder end-Abschnitte. Diese Option ist standardmäßig aktiviert.

Ergebnisse	**Speichern des Ergebnisses der Wiedergabeübersicht nach jeder Skriptausführung in einem benannten Ordner aktivieren** Wenn diese Option aktiviert ist, fordert das Dialogfeld Sie auf, eine Ergebnisdatei zu benennen, bevor Sie in VuGen ein Skript ausführen. Ist diese Option nicht aktiviert, benennt VuGen das Verzeichnis automatisch mit 'result1'. Nachfolgende Skriptausführungen überschreiben automatisch die vorherigen Ergebnisdateien, sofern Sie keinen anderen Namen angeben. **Hinweis:** Ergebnisse werden in einem Unterverzeichnis des Skripts gespeichert. Diese Option ist standardmäßig deaktiviert. **Bericht während der Skriptausführung generieren** Diese Option ist standardmäßig deaktiviert.
Während der Wiedergabe	• **Laufzeit-Viewer während der Wiedergabe anzeigen** Aktiviert den Laufzeit-Viewer. • **Fenster automatisch anordnen** Wählen Sie diese Option aus, um die beiden Viewer nebeneinander anzuordnen. Diese Option ist standardmäßig deaktiviert. • **Wiedergabestatistik aufzeichnen** Wählen Sie diese Option aus, um die Aufzeichnung von Wiedergabestatistiken zu aktivieren. Die aufgezeichneten Daten werden im Zusammenfassungsbericht für die Wiedergabe angezeigt. Diese Option ist standardmäßig aktiviert. Weitere Informationen finden Sie unter "Registerkarte "Wiedergabeübersicht"" auf Seite 124.
Nach Wiedergabe	**Nach Wiedergabe anzeigen** Legt fest, wie VuGen nach der Wiedergabe vorgehen soll: • **Skript.** Das Skript im Editor anzeigen. • **Wiedergabeübersicht.** Direkt zum Fenster **Wiedergabeübersicht** im Editor wechseln. (Standardeinstellung)

Skriptverwaltung

Zugriff	**VuGen > Extras > Optionen > Skripterstellung > Skriptverwaltung**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Auflisten der Dateitypen, die im Editor bearbeitet werden können (nach Erweiterung)	Ermöglicht Ihnen, die Liste der gültigen Dateierweiterungen zu ändern, die im Editor bearbeitet werden können.

Vergleich

Zugriff	VuGen > Extras > Optionen > Skripterstellung > Vergleich

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Pfad zum Vergleichstool	Sie können ein Vergleichstool auswählen, das beim Vergleichen von zwei Skripts verwendet werden soll. VuGen stellt ein Standardvergleichstool (Wdiff) bereit. Klicken Sie auf die Durchsuchen-Schaltfläche, um nach einem Vergleichstool eines Drittanbieters zu suchen, das auf Ihrem lokalen Computer installiert ist.
Befehlszeilenargumente	Diese Option enthält die erforderlichen Standardargumente %1 und %2 für die beiden zu vergleichenden Dateien. Sie dürfen diese Argumente nicht ändern. Sie können bei Bedarf zusätzliche Argumente für Ihr Vergleichstool hinzufügen.

Schrittenavigator

Zugriff	VuGen > Extras > Optionen > Skripterstellung > Schrittenavigator

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Markieren im Editor aktivieren	Diese Option bietet Ihnen die Möglichkeit, gefilterte Schritte in Ihrem Skript zu markieren.
Hintergrundfarbe	Ermöglicht Ihnen das Auswählen einer Hintergrundfarbe für die in Ihrem Skript gefilterten Schritte.
Rahmenfarbe	Ermöglicht Ihnen das Auswählen einer Rahmenfarbe für die in Ihrem Skript gefilterten Schritte.

Miniaturansichten

Zugriff	VuGen > Extras > Optionen > Skripterstellung > Miniaturansichten

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Miniaturansicht-Explorer aktivieren	Aktiviert die Diaansicht der generierten Miniaturansichten im Miniaturansicht-Explorer. Durch Doppelklicken auf eine Miniaturansicht wird der Cursor an dem zugeordneten Schritt im Skript platziert.
Die einem Schritt zugeordnete Miniaturansicht markieren	Synchronisiert die Anzeige des Miniaturansicht-Explorers beim Durchführen eines Bildlaufs durch die Schritte im Editor.
Wichtige Miniaturansichten standardmäßig anzeigen	VuGen zeigt nur Miniaturansichten an, die unmittelbar mit dem Geschäftsprozess in Zusammenhang stehen. Andere Miniaturansichten werden standardmäßig herausgefiltert.
Automatisches Erstellen aktivieren	Ermöglicht die automatische Erstellung von Miniaturansichten während der Leerlaufzeit der Anwendung.
Miniaturansichten im Skriptordner zwischenspeichern	Optimiert die VuGen-Leistung durch das Speichern der gerenderten Miniaturansichten in einer Cachedatei. Miniaturansichten werden nach der Generierung aus der Cachedatei geladen.

Ausgabebereich

Zugriff	**VuGen > Extras > Optionen > Skripterstellung > Ausgabe**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Format	**Zeilenumbruch** Ermöglicht das Umbrechen von Wörtern im Ausgabebereich.
Schriftart	**Textschriftart** Ermöglicht das Auswählen einer Schriftart. **Größe** Ermöglicht das Auswählen einer Schriftgröße.

Java

Zugriff	**VuGen > Extras > Optionen > Skripterstellung > Java**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Speicherort von Eclipse-IDE	**Durchsuchen** Ermöglicht das Festlegen des Speicherorts des Eclipse-Programms, **eclipse.exe**.

Citrix

Zugriff	VuGen > Extras >Optionen > Skripterstellung > Citrix

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Client während der Wiedergabe anzeigen	Zeigt den Citrix-Client während der Skriptwiedergabe.
Popupfenster für Bitmap-Auswahl anzeigen	Gibt eine Popup-Meldung beim Einfügen einer "Get Text"- oder "Sync Bitmap"-Funktion aus, bevor eine Bitmap oder ein Text im Snapshot ausgewählt wird.

Snapshots

Zugriff	VuGen > Extras > Optionen > Skripterstellung > Snapshots

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Snapshot-Anzeige aktivieren	Ermöglicht die Anzeige von Snapshots, wenn Sie im Editor oder Schrittenavigator auf einen Schritt klicken. Mit dieser Option können Sie die Leistung von VuGen bei der Arbeit mit umfangreichen Skripts verbessern.
Erweiterte XML-Ansicht aktivieren	Aktiviert folgende Darstellungsfunktionen für den XML-Viewer: • XML-Struktur • Farbliche Darstellung Deaktivieren Sie diese Option, um den Speicherplatzbedarf der XML-Ansicht zu verringern.

Snapshot-Zwischenspeicherung aktivieren	Ermöglicht die Zwischenspeicherung von Snapshots. Deaktivieren Sie diese Option, wenn Sie mit großen Snapshots arbeiten und nicht genügend Speicherplatz zur Verfügung haben.
Snapshots mit Textdaten nicht laden, wenn größer als	Text-basierte Snapshots werden nicht geladen, wenn sie die angegebene Größe überschreiten.
Snapshots mit Binärdaten nicht laden, wenn größer als	Binäre [hexadezimale] Snapshots werden nicht geladen, wenn sie die angegebene Größe überschreiten.
Snapshot mit XML-Daten nicht laden, wenn größer als	XML-basierte Snapshots werden nicht geladen, wenn sie die angegebene Größe überschreiten.

Parameter

Zugriff	**VuGen > Extras > Optionen > Skripterstellung > Parameter**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Linkes Trennzeichen des Parameters	Ermöglicht das Festlegen eines linken Trennzeichens für Parameter. Die folgenden Zeichen sind zulässig: !, #, $, %, &, (,), [,], {, }, |, ~, `, <, >, ? **Hinweis:** Wenn Sie das linke Trennzeichen ändern, wird das neue Trennzeichen nur auf neue Vuser-Skript und nicht auf bereits vorhandene angewendet.
Rechtes Trennzeichen des Parameters	Ermöglicht das Festlegen eines rechten Trennzeichens für Parameter. Die folgenden Zeichen sind zulässig: !, #, $, %, &, (,), [,], {, }, |, ~, `, <, >, ? **Hinweis:** Wenn Sie das rechte Trennzeichen ändern, wird das neue Trennzeichen nur auf neue Vuser-Skript und nicht auf bereits vorhandene angewendet.
Hintergrundfarbe für Parameter	Hier können Sie die Hintergrundfarbe von Parametern in einem Vuser-Skript festlegen.

Rahmenfarbe für Parameter	Hier können Sie die Rahmenfarbe für die Parameter in einem Vuser-Skript festlegen.
Standardtrennzeichen verwenden	Stellt die Standardwerte für die linken und rechten Trennzeichen für Parameter wieder her.

Parser

Zugriff	**VuGen > Extras > Optionen > Skripterstellung > Parser**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
C-Sprachparser aktivieren	Wenn Sie den C-Sprachparser deaktivieren, kann bei der Arbeit mit sehr großen Skripts die Anwendungsleistung verbessert werden. Die folgenden Funktionen werden jedoch deaktiviert: • Bearbeiten von Schrittargumenten im **Editor** • Vervollständigung von Anweisungen • Snapshots • Aufgaben • Miniaturansichten • Zusätzliche mit Schritten verbundene Funktionen Diese Option ist standardmäßig aktiviert.

Dialogfelder "Suchen" und "Ersetzen"

Über diese Dialogfelder können Sie Textzeichenfolgen in Vuser-Skripts und -Lösungen suchen und ersetzen.

Dialogfeld "Suchen"

Zugriff	• **VuGen > Suchen > Schnellsuche** • **VuGen > Suchen > In Dateien suchen**
Hinweis	Einige in der folgenden Tabelle aufgeführten Elemente der Oberfläche werden nur im Dialogfeld **Schnellsuche** angezeigt, andere nur im Dialogfeld **In Dateien suchen**.

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Dropdownliste "Suchtyp">	Ermöglicht Ihnen, den Typ der durchzuführenden Suche anzugeben. **Schnellsuche** oder **In Dateien suchen**. Beachten Sie, dass die Elemente der Oberfläche im Dialogfeld **Suchen** abhängig von der Auswahl unterschiedlich sind.
Text suchen	Geben Sie den Text ein, nach dem gesucht werden soll.
	Zeigt den Generator für reguläre Ausdrücke an. Dieser ermöglicht Ihnen, im Feld **Text suchen** einen regulären Ausdruck zu erstellen.
Regulärer Ausdruck	Wählen Sie **Regulärer Ausdruck** aus, um anzugeben, dass die **Text suchen**-Zeichenfolge ein regulärer Ausdruck ist.
Bereich	Geben Sie die Dateien an, in denen gesucht werden soll.
Groß-/Kleinschreibung beachten	Unterscheidet bei der Suche zwischen Groß- und Kleinschreibung.
Nur ganzes Wort	Sucht nach dem ganzen Wort, nicht nach dem Suchbegriff als Teil eines längeren Worts.
Suchrichtung nach oben	Führt die Suche aufwärts aus.
In Suche einschließen	Ermöglicht Ihnen das Auswählen von Entitäten, die in die Suche einbezogen werden.
Verzeichnis	Geben Sie den Ordner an, der die in einer **In Dateien suchen**-Suche durchsuchten Dateien enthält.
Optionen	Ermöglicht Ihnen, Optionen für eine **In Dateien suchen**-Suche festzulegen.
Weitersuchen	Ermittelt das nächste Vorkommen des Texts oder regulären Ausdrucks im Feld **Text suchen**.

Alle suchen	Ermittelt alle Vorkommen des Texts oder regulären Ausdrucks, der im Feld **Text suchen** angezeigt wird. Die Ergebnisse werden im Bereich mit den Suchergebnissen angezeigt. Im Bereich mit den Suchergebnissen können Sie die folgenden Aktionen ausführen: • Klicken Sie mit der rechten Maustaste auf einen Eintrag, und wählen Sie **Suchen** aus, um den zugehörigen Text im Vuser-Skript oder an anderer Stelle anzuzeigen. • Klicken Sie mit der rechten Maustaste auf einen Eintrag, und wählen Sie **Kopieren** aus, um das ausgewählte Suchergebnis in die Zwischenablage zu kopieren. • Klicken Sie mit der rechten Maustaste auf einen Eintrag, und wählen Sie **Alles kopieren** aus, um alle Suchergebnisse in die Zwischenablage zu kopieren.

Dialogfeld "Ersetzen"

Zugriff	**VuGen > Suchen > Schnellersetzung**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Text suchen	Geben Sie den Text ein, nach dem gesucht werden soll.
	Zeigt den Generator für reguläre Ausdrücke an. Dieser ermöglicht Ihnen, im Feld **Text suchen** einen regulären Ausdruck zu erstellen.
Regulärer Ausdruck	Wählen Sie **Regulärer Ausdruck** aus, um anzugeben, dass die **Text suchen**-Zeichenfolge ein regulärer Ausdruck ist.
Ersetzen durch	Geben Sie den Text ein, mit dem Sie den in Ihrer Suche gesuchten und gefundenen Text ersetzen möchten.
Bereich	Geben Sie die Dateien an, in denen gesucht werden soll.
Optionen	Ermöglicht Ihnen, Optionen für die Suche festzulegen.
Groß-/Kleinschreibung beachten	Unterscheidet bei der Suche zwischen Groß- und Kleinschreibung.
Nur ganzes Wort	Sucht nach dem ganzen Wort, nicht nach dem Suchbegriff als Teil eines längeren Worts.
Suchrichtung nach oben	Führt die Suche aufwärts aus.

Forts.

Weitersuchen	Ermittelt das nächste Vorkommen des Texts oder regulären Ausdrucks im Feld **Text suchen**.
Ersetzen	Ersetzt den ausgewählten Text durch den Text im Feld **Ersetzen durch**.
Alle ersetzen	Ersetzt alle gefunden Vorkommen des Texts oder regulären Ausdrucks im Feld **Text suchen** durch den Text im Feld **Ersetzen durch**.

Editor

In diesem Bereich können Sie Skripts und andere verknüpfte Skriptdateien bearbeiten. Darüber hinaus können Sie eine Browsersitzung öffnen, um Websites zu durchsuchen, z. B. das LoadRunner-Forum.

Zugriff	Dieser Editor wird beim Laden von VuGen geöffnet.
Wichtige Informationen	• Der Editor wird automatisch als Teil des Standard-Layouts angezeigt. • Drücken Sie **STRG+TAB**, um eine Liste der Registerkarten und Bereiche anzuzeigen. Markieren Sie Registerkarten und klicken Sie darauf, um im Editor zwischen Registerkarten zu wechseln. • Drücken Sie **STRG + G**, oder wählen Sie **Suchen > Gehe zu...**, um zu einer bestimmten Zeile im Skript zu wechseln. • Andere Bereiche der Hauptbenutzeroberfläche, wie Ausgabe, Fehler und Snapshot, synchronisieren ihre Anzeige basierend auf Ihrer Position im Editor. • Sie können im Lösungs-Explorer auf ein Asset klicken, um es im Editor zu öffnen. • Skriptänderungen werden als markierter Text im Editor angezeigt. Es folgen einige Beispiele: ▪ Das Skript wurde geändert, jedoch nicht gespeichert. ▪ Das Skript wurde geändert und gespeichert. ▪ Es wurde ein Haltepunkt eingefügt.
Siehe auch	• "Editor - Übersicht" auf Seite 59 • " Editor-Optionen" auf Seite 77

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Bundsteg>	Im Bundsteg des Editors können Sie u. a. folgende Umschaltfunktionalität hinzufügen: • Verwenden von Haltepunkten Weitere Informationen finden Sie unter "Arbeiten mit Haltepunkten" auf Seite 418 oder "Bereich "Haltepunkte"" auf Seite 119. • Verwenden von Textmarken Weitere Informationen finden Sie unter "Verwenden von Textmarken" auf Seite 410.
Klassen-/Funktionsbrowser	Bei Aktivierung dieser Option können Sie schnell zu einer bestimmten Klasse oder Funktion im Skript navigieren, indem Sie diese aus der Dropdownliste im Browser auswählen.
Zeilennummern	Zeigt die Zeilennummern im Skript an.
Kontextmenü	
Bereich kommentieren	Ermöglicht das Auskommentieren markierter Skriptzeilen.
Einzug	Fügt einen Einzug für die ausgewählten Zeilen im Skript hinzu.
Snapshot anzeigen	Zeigt den Snapshot an, der dem markierten Skriptschritt zugeordnet ist.
Auswahl korrelieren	Öffnet Design Studio und sucht nach dynamischen Werten, die im ausgewählten Skriptabschnitt korreliert werden sollen.

Element der Oberfläche	Beschreibung	
Einfügen		Einfügen eines Schrittes **Transaktion starten** in Ihr Skript. Weitere Informationen finden Sie unter "Übersicht über Transaktionen" auf Seite 425.
		Fügt einen Schritt **Transaktion beenden** in Ihr Skript ein. Weitere Informationen finden Sie unter "Übersicht über Transaktionen" auf Seite 425.
		Einfügen eines Schrittes **Rendezvous-Punkt** in Ihr Skript. Weitere Informationen finden Sie unter "Rendezvous-Punkte " auf Seite 427.
		Fügt Kommentare in Ihr Skript ein.
	Meldung protokollieren	Fügt eine Meldung `lr_log_message("Meldung hier eingeben");` in Ihr Skript ein.
Haltepunkte	Schaltet einen Haltepunkt um.	
In Community suchen	Öffnet den standardmäßigen Community-Suchbrowser.	

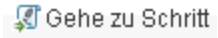
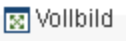
Miniaturansicht-Explorer

In diesem Bereich können Sie durch Miniaturansichten Ihres Geschäftsprozesses blättern und so anhand einer grafischen Darstellung eines Schrittes leichter zu den jeweiligen Positionen im **Editor** navigieren. Umgekehrt können Sie einen Bildlauf durch den **Editor** durchführen und den visuellen Kontext Ihres Skripts im **Miniaturansicht-Explorer** anzeigen.

Zugriff	Führen Sie einen der folgenden Schritte durch: • **VuGen > Ansicht > Miniaturansicht-Explorer** • Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche .

Wichtige Informationen	• Sie können diesen Ausschnitt in andere Bereiche der Hauptbenutzeroberfläche verschieben. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56. • Sie können den **Miniaturansicht-Explorer** unter **Optionen > Skriptoptionen** konfigurieren. Weitere Informationen finden Sie unter "Skriptoptionen" auf Seite 83. • Miniaturansichten werden in derselben Reihenfolge wie die Aktionen im Lösungs-Explorer erstellt und nicht über die Einstellungen unter **Laufzeiteinstellungen > Allgemein > Ausführungslogik** gesteuert. • Wenn Sie die automatische Erstellung von Miniaturansichten unter **Optionen > Skripterstellung > Miniaturansichten** aktivieren, kann VuGen Miniaturansichten während der Leerlaufzeit der Anwendung erstellen. • Wenn das Windows Aero-Design aktiviert ist, zeichnen Miniaturansichten realistischere Bilder der Anwendung auf.
Relevante Aufgaben	"Ändern des VuGen-Layouts" auf Seite 71

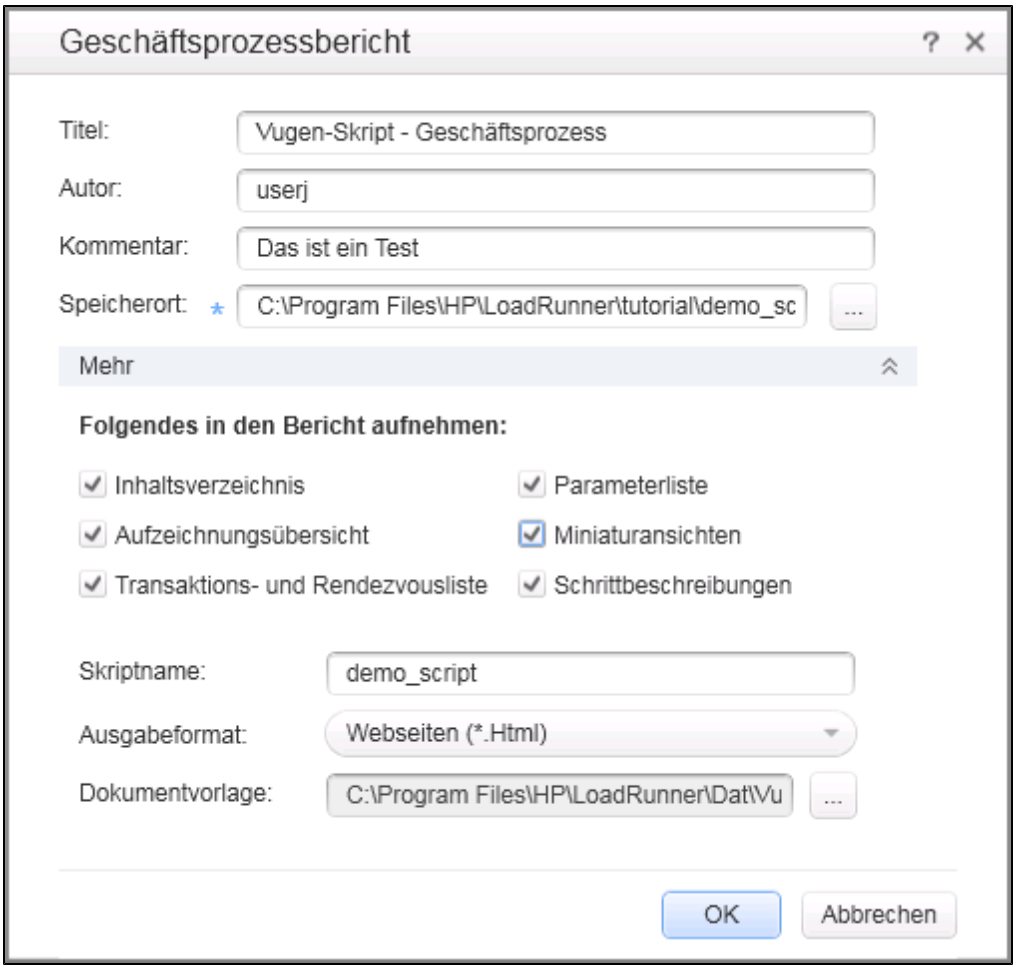
Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
	Verschiebt den Cursor zu dem Schritt im **Editor**, der mit der markierten Miniaturansicht im **Miniaturansicht-Explorer** verknüpft ist.
	Ermöglicht eine Vollbildansicht der Miniaturansicht.
	Synchronisiert den Bildlauf im **Editor** mit der zugehörigen Miniaturansicht im **Miniaturansicht-Explorer** und dem zugehörigen Schritt im **Schrittenavigator**.
	Filtert kleinere Miniaturansichten heraus, die nicht direkt im Zusammenhang mit dem aufgezeichneten Geschäftsprozess stehen.
	Aktualisiert die generierten Miniaturansichten.
	Eine Seite im **Miniaturansicht-Explorer** nach links scrollen.
	Zur vorherigen Miniaturansicht im **Miniaturansicht-Explorer** wechseln.
	Zur nächsten Miniaturansicht im **Miniaturansicht-Explorer** wechseln.

Element der Oberfläche	Beschreibung
	Eine Seite im **Miniaturansicht-Explorer** nach rechts scrollen.

Dialogfeld "Geschäftsprozessbericht"

In diesem Dialogfeld können Sie einen Geschäftsprozessbericht erstellen.



Zugriff	**VuGen > Extras > Geschäftsprozessbericht erstellen**.
Relevante Aufgaben	"Erstellen eines Geschäftsprozessberichts" auf Seite 70

Nachfolgend werden die Elemente der Oberfläche beschrieben:

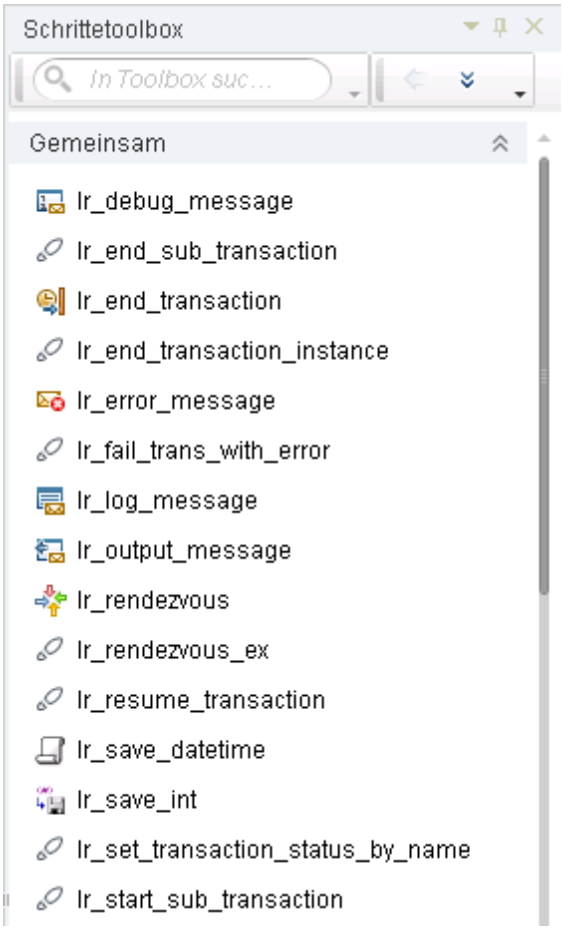
Element der Oberfläche	Beschreibung
Titel	Der Titel des Berichts.

Element der Oberfläche	Beschreibung
Autor	Ihr Name.
Kommentar	Weitere Kommentare, die in dem Bericht angezeigt werden sollen.
Speicherort	Der Speicherpfad für den Bericht. **Standardwert:** Skriptordner.
Mehr	Erweitert das Dialogfeld **Geschäftsprozessbericht**, sodass weitere Optionen angezeigt werden.
Inhaltsverzeichnis	Für Ihren Bericht wird ein Inhaltsverzeichnis erstellt. Wenn Sie eine Option deaktivieren, wird diese im Inhaltsverzeichnis nicht angezeigt. **Standardwert:** Aktiviert.
Aufzeichnungsübersicht	Eine Zusammenfassung der Aufzeichnungssitzung, wie sie beim Klicken auf den Link **Aufzeichnungsübersicht** in der Aufgabenliste dargestellt wird. **Standardwert:** Aktiviert.
Transaktions- und Rendezvousliste	Eine Liste aller Transaktionen und Rendezvous-Punkte, die im Skript definiert wurden. **Standardwert:** Aktiviert.
Parameterliste	Eine Liste aller Parameter, die im Skript definiert wurden. Diese Liste entspricht den Parametern, die im Dialogfeld **Parameterliste** aufgeführt sind (**Entwurf > Parameterliste**). **Standardwert:** Aktiviert.
Miniaturansichten	Ein aktueller Snapshot des aufgezeichneten Schritts, der neben dem Namen und der Beschreibung des Schritts angezeigt wird. **Standardwert:** Aktiviert.
Schrittbeschreibungen	Eine kurze Beschreibung jedes Schritts. **Standardwert:** Aktiviert.
Skriptname	Der .usr-Dateiname des Skripts.

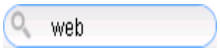
Element der Oberfläche	Beschreibung
Ausgabeformat	Erstellt den Bericht im ausgewählten Format. Folgende Formate stehen zur Verfügung: • Microsoft Word • Adobe PDF • HTML
Dokumentvorlage	Der Pfad und Dateiname der Vorlage, die für den Bericht verwendet werden soll. Die Standardvorlage ist im **dat**-Ordner von LoadRunner gespeichert. Um die Berichtsvorlage zu ändern, klicken Sie auf die Schaltfläche **Durchsuchen** und geben eine neue Vorlage mit einer .docx-Erweiterung an. Wenn Sie eine neue Vorlage erstellen möchten, empfehlen wir Ihnen, eine vorhandene Vorlage als Grundlage für die neue zu verwenden. Dadurch wird sichergestellt, dass die erforderlichen Textmarken und Formate in der neuen Schablone erhalten bleiben.

Bereich "Schrittetoolbox"

In diesem Bereich können Sie API-Funktionen in Ihr Skript ziehen und dort ablegen.

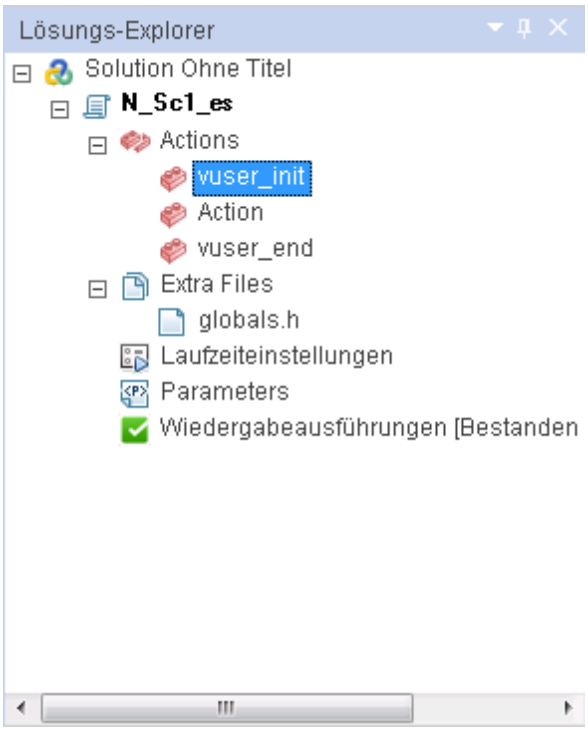
Beispiel für die Benutzeroberfläche	
Zugriff	Führen Sie einen der folgenden Schritte durch: • **Entwurf > In Skript einfügen > Neuer Schritt** • Klicken Sie im Skript mit der rechten Maustaste in den **Editor und wählen Sie Einfügen > Neuer Schritt** • Drücken Sie STRG+ALT+B
Wichtige Informationen	• Das Parameter-Dialogfeld eines Schrittes wird beim Hinzufügen des Schrittes zum Skript geöffnet. • Sie können Schritte in Ihr Skript ziehen und dort ablegen. • Es ist nicht möglich, einen Schritt in einen Schritt aus der **Schrittetoolbox** zu ziehen und dort abzulegen, Sie können jedoch manuell einen Schrittparameter in einem Schritt hinzufügen. • Wenn Sie einen Schritt an der falschen Stelle im Skript einfügen, kann das Skript möglicherweise nicht ordnungsgemäß ausgeführt werden.
Relevante Aufgaben	"Einfügen von Schritten in ein Skript" auf Seite 434.

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Funktionsliste>	Zeigt eine Liste der verfügbaren Funktionen an, die in die folgenden Kategorien geteilt ist: • Gemeinsam • Webprüfungen • Services • XML • Web-JS • Asynchron
	**Suchen.** Ermöglicht eine inkrementelle Suche in der Funktionsliste durch die Eingabe von Text. Geben Sie z. B. "web" im Suchfeld ein, werden nur die Funktionen in der Funktionsliste angezeigt, die die Buchstaben "web" enthalten.
	Schritt aktueller Position hinzufügen. Fügt den markierten Schritt an der aktuellen Stelle in das Skript ein.
	Alle erweitern/Alle reduzieren. Erweitert die Schrittkategorien oder blendet sie aus.

Bereich "Lösungs-Explorer"

Dieser Bereich ermöglicht das Navigieren durch eine Lösung, die Skript-Assets, Parameter, Laufzeiteinstellungen und Wiedergabeausführungen enthält.



Zugriff	Verwenden Sie eine der folgenden Methoden: • **Ansicht > Lösungs-Explorer** • Drücken Sie **Strg + Alt + L**
Wichtige Informationen	• Der Lösungs-Explorer wird automatisch als Teil des Standard-Layouts angezeigt. • Sie können diesen Ausschnitt in andere Bereiche der Hauptbenutzeroberfläche verschieben. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56. • Andere Bereiche der Hauptbenutzeroberfläche, wie Ausgabe, Fehler und Snapshot, synchronisieren ihre Anzeige basierend auf Ihrer Position im Lösungs-Explorer. • Sie können im Editor-Bereich auf ein Asset doppelklicken, um es zu aktivieren. Sie können mit der rechten Maustaste auf ein Asset klicken, um die für dieses Asset verfügbaren Operationen anzuzeigen.
Siehe auch	• "VuGen-Benutzeroberfläche – Übersicht" auf Seite 50 • "Lösungs-Explorer - Übersicht" auf Seite 57 • "VuGen-Layouts - Übersicht" auf Seite 56 • "Importieren von Aktionen in ein Skript" auf Seite 270

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Lösung>	Container für Skripts. • Der Standardlösungsname ist "Ohne Titel". **Optionen des Kontextmenüs** • Neues Skript hinzufügen Weitere Informationen finden Sie unter "Dialogfeld "Neues Skript erstellen"" auf Seite 157. • Vorhandenes Skript hinzufügen • Alle Skripts speichern • Lösung schließen • Lösung speichern unter...

<Skript>	Container für Skript-Assets, einschließlich Skriptaktionen, zusätzliche Dateien, Laufzeiteinstellungen und Parameter. **Optionen des Kontextmenüs** • **Skript speichern** Speichern des Skripts. • **Skript speichern unter...** Speichern des Skripts mit einem neuen Namen oder an einem neuen Speicherort, zum Beispiel ALM. • **In Vorlage exportieren...** Sie können Skripts als Vorlagen speichern. Weitere Informationen finden Sie unter "Erstellen oder Öffnen von Vuser-Skriptvorlagen" auf Seite 161. • **Skript entfernen** Ermöglicht Ihnen das Löschen eines Skripts aus der Lösung oder das Aktivieren eines Kontrollkästchens, um ein Skript aus dem Dateiverzeichnis zu löschen. • **Zu vergleichende Dateien auswählen** Sie können ein Asset mit einem anderen vergleichen. Diese Option legt das markierte Asset als primäres Asset fest. • **Vergleichen mit <Dateiname>** Führt einen Vergleich zwischen dem ausgewählten und dem primären Asset durch. • **Mit externer Datei vergleichen...** Sie können ein Asset mit einer Datei außerhalb der Lösung vergleichen. Diese Option legt das markierte Asset als primäres Asset fest und öffnet Windows Explorer, damit Sie das sekundäre Asset auswählen können. • **Zu vergleichenden Ordner auswählen** Sie können einen Ordner mit einem anderen vergleichen. Diese Option legt den markierten Ordner als primären Ordner fest. • **Mit externem Ordner vergleichen...** Führt einen Vergleich zwischen dem ausgewählten und dem

	primären Ordner durch. • **Skriptordner öffnen** Während der Aufzeichnung erstellt VuGen eine Reihe von Konfigurations-, Daten- und Quellcodedateien. Diese Dateien enthalten Vuser-Laufzeit- und Setup-Informationen. VuGen speichert diese Dateien zusammen mit dem Skript. Sie können den Ordner auf dem lokalen Datenträger öffnen, in dem das Skript gespeichert ist. Für Skripts, die an einem anderen Speicherort gespeichert sind, zum Beispiel ALM, öffnet diese Option den temporären Ordner auf dem lokalen Datenträger.
<Aktionen>	Container für individuelle Aktionen, einschließlich der Standardaktionen: • vuser_init • action • vuser_end **Optionen des Kontextmenüs** • Neue Aktion erstellen Ermöglicht das Hinzufügen eines neuen Skriptaktionsblocks zu Ihrem Skript. • Aktion importieren Ermöglicht das Importieren einer Aktion aus einem vorhandenen Skript.

Zusätzliche Dateien	Container für zusätzliche Dateien, die mit Ihrem Skript verbunden sind. Sie können direkt von VuGen aus auf diese zusätzlichen Daten zugreifen. **Optionen des Kontextmenüs** • Neue Datei erstellen Ermöglicht das Erstellen einer neuen Datei, die dann zum Knoten für zusätzliche Dateien Ihres Skripts hinzugefügt wird. • Dateien zum Skript hinzufügen Ermöglicht das Hinzufügen von Dateien zum Knoten für zusätzliche Dateien Ihres Skripts. Die Dateien, die Sie hinzufügen, müssen bereits vorhanden sein. • Von HPLN heruntergeladene Daten hinzufügen Ermöglicht das Hinzufügen von Dateien zum Knoten für zusätzliche Dateien Ihres Skripts. Die Dateien müssen zuvor bereits von HPLN heruntergeladen worden sein. • Dateien aus Ordner und Unterordnern hinzufügen Ermöglicht das Hinzufügen des Inhalts von Ordnern oder Unterordnern zum Knoten für zusätzliche Dateien Ihres Skripts. Details zu zusätzlichen Dateien finden Sie unter "Lösungs-Explorer - Übersicht" auf Seite 57.
Laufzeiteinstellungen	Öffnet das Dialogfeld "Laufzeiteinstellungen". Weitere Informationen finden Sie unter "Übersicht über Laufzeiteinstellungen".
Parameter	Ermöglicht das Erstellen, Bearbeiten und Auflisten der Parameter, die mit Ihrem Skript verbunden sind. Weitere Informationen finden Sie unter "Parameter" auf Seite 436. **Optionen des Kontextmenüs** • Parameterliste • Neuen Parameter erstellen • Parameter bearbeiten • Parametertrennzeichen konfigurieren

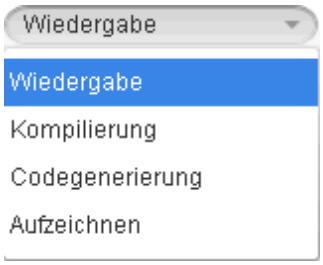
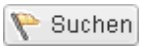
Wiedergabeausführungen	Ermöglicht das Anzeigen des Zusammenfassungsberichts für die Wiedergabe. **Optionen des Kontextmenüs** • Wiedergabeübersicht öffnen Ermöglicht das Öffnen der Wiedergabeübersicht für eine ausgewählte Iteration im Editor. • Testergebnisse öffnen Ermöglicht das Öffnen der Testergebnisse für eine ausgewählte Iteration im Editor.

Ausgabebereich

Im Ausgabebereich werden Meldungen angezeigt, die während der Aufzeichnung, Kompilierung und Wiedergabe des Skripts generiert wurden.

Zugriff	Wählen Sie **Ansicht > Ausgabe** aus oder klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Ausgabebereich anzeigen** .
Wichtige Informationen	• Sie können diesen Ausschnitt in andere Bereiche der Hauptbenutzeroberfläche verschieben. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56. • Wenn Sie ein vorhandenes Skript öffnen, stammen die angezeigten Elemente im **Ausgabebereich** von der letzten Wiedergabe oder Kompilierung.
Relevante Aufgaben	"Anpassen des Layouts von VuGen"
Siehe auch	"VuGen-Benutzeroberfläche – Übersicht" auf Seite 50

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Der Typ der anzuzeigenden Ausgabe. Die folgenden Typen sind verfügbar: • **Wiedergabe.** Zeigt die bei der Skriptwiedergabe generierten Meldungen an. **Hinweis:** Wenn Sie auf einen Eintrag im Wiedergabeprotokoll doppelklicken, verschiebt VuGen den Cursor in die entsprechende Zeile im Editor. • **Kompilierung.** Zeigt die Kompilierungsmeldungen an. • **Codegenerierung.** Zeigt den bei der Aufzeichnung generierten Code an. • **Aufzeichnen.** Zeigt die bei der Aufzeichnung generierten Meldungen an. • **Aufgezeichnete Ereignisse.** Zeigt Ereignisse an, die während der Aufzeichnung aufgetreten sind.
	Löscht alle Meldungen aus der Meldungsliste.
	Zeilenumbruch umschalten. Wenn diese Option ausgewählt ist, wird der Text jeder Nachricht nach Bedarf in die nächste Zeile umgebrochen.
	Springt zu dem Speicherort im Quelldokument, der für die ausgewählte Ausgabemeldung relevant ist.
	<Suchfeld>. Die Textzeichenfolge, die Sie suchen möchten. Sie können Ihre Suche optimieren, indem Sie eine der nachfolgend beschriebenen **Optionen** auswählen. Drücken Sie die `EINGABETASTE`, um die Suche zu starten.
	Vorherige suchen/Weitersuchen. Markiert die nächste oder vorherige Zeichenfolge, die dem im Feld **Suchen** eingegebenen Text entspricht. Diese Schaltflächen sind erst nach Eingabe von Text im Feld **Suchen** verfügbar.

Element der Oberfläche	Beschreibung
Optionen	Mit den folgenden Optionen können Sie Ihre Suche optimieren: • **Groß-/Kleinschreibung.** Unterscheidet bei der Suche zwischen groß und klein geschriebenen Buchstaben. • **Nur ganzes Wort.** Sucht nur nach dem Auftreten von ganzen Wörtern, nicht nach Teilen von längeren Wörtern. • **Regulären Ausdruck verwenden.** Behandelt die angegebene Textzeichenfolge als regulären Ausdruck. **Hinweis:** Erweiterte reguläre Ausdrücke und die mehrzeilige Suche werden nicht unterstützt.
	Öffnet das Dialogfeld **Speichern unter**, damit Sie den Inhalt der Meldungsliste als Textdatei speichern können.
Zusammenfassung anzeigen [Nur im Wiedergabeprotokoll verfügbar]	Öffnet die Registerkarte **Wiedergabeübersicht**. Weitere Informationen finden Sie unter "Registerkarte "Wiedergabeübersicht"" auf Seite 124.

Snapshot-Bereich

Zugriff	Wählen Sie **Ansicht > Snapshot** aus oder klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Snapshot-Bereich anzeigen** .

Wichtige Informationen	• Darstellung und Funktionalität des Snapshot-Bereichs hängen vom Protokoll des aktuellen Vuser-Skripts ab. Neben diesen Standardsteuerelementen kann der Snapshot-Bereich weitere Steuerelemente enthalten, die vom aktuellen Vuser-Protokoll abhängen. • Das neue Web-Snapshot-Modell ist mit Vorgängerversionen von LoadRunner kompatibel, es kann jedoch vorkommen, dass Snapshot-Daten fehlen. Generieren Sie in diesem Fall das Skript neu. • Wenn Sie unter Windows 2008 R2 einen Snapshot aus dem Schrittenavigator in SAP GUI und Webprotokollen öffnen, wird der Snapshot möglicherweise nicht automatisch geöffnet. **Lösung:** Die verstärkte Sicherheitskonfiguration für Internet Explorer muss deaktiviert werden, damit Hilfeinhalt angezeigt wird. Diese ist standardmäßig aktiviert. (**Systemsteuerung > Verwaltung > Server-Manager > Verstärkte Sicherheitskonfiguration für IE konfigurieren**). • Sie können diesen Ausschnitt in andere Bereiche der Hauptbenutzeroberfläche verschieben. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56.
Siehe auch	"Snapshot-Bereich - Übersicht" auf Seite 62 "Arbeiten mit Snapshots" auf Seite 65

Standardsteuerelemente im Snapshot-Bereich

Im Snapshot-Bereich werden die folgenden Standardsteuerelemente angezeigt:

Element der Oberfläche	Beschreibung
Einzeln	Anzeigen eines einzelnen Snapshots im Snapshot-Bereich.
Teilen	Teilen des Snapshot-Bereichs, um zwei Snapshots anzuzeigen
Horizontal	Anzeigen zweier Snapshots im Snapshot-Bereich nebeneinander.
Vertikal	Anzeigen zweier Snapshots im Snapshot-Bereich untereinander.
Aufzeichnen	Anzeigen des Aufzeichnungs-Snapshots im Snapshot-Bereich.
Wiedergabe	Anzeigen des Wiedergabe-Snapshots im Snapshot-Bereich.
Wiederholung: 1	Auswählen der Iterationsnummer des Wiedergabe-Snapshots, der angezeigt werden soll.

	Synchronisieren der Anzeige zweier Snapshots, wenn der Snapshot-Bereich geteilt ist Beachten Sie, dass die Snapshot-Synchronisierung nur für bestimmte Vuser-Protokolle und nur für bestimmte Ansichten in diesen Protokollen zur Verfügung steht.
Vergleichen	Vergleicht die beiden Snapshots, die aktuell im Snapshot-Bereich angezeigt werden. Um die Funktion **Vergleichen** zu aktivieren, müssen Sie zunächst den Snapshot-Bereich teilen, um zwei Snapshots anzuzeigen. Standardmäßig verwendet VuGen das Dienstprogramm *WDiff*, um Snapshots zu vergleichen. Sie können ein alternatives Vergleichstool angeben, wie unter "Skriptoptionen" auf Seite 83 beschrieben. Beachten Sie, dass die Snapshot-Vergleichsfunktion nur für Web-HTTP/HTML- und Web Services-Protokolle zur Verfügung steht.

Citrix-, RDP- und SAP-Protokolle

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Anzeigen des nächsten Aufzeichnungs-Snapshots, der bei der Skriptwiedergabe angefügt wurde.
	Anzeigen des vorherigen Aufzeichnungs-Snapshots, der bei der Skriptwiedergabe angefügt wurde.
Bild	Anzeigen einer grafischen Darstellung des Snapshots. Sie können die Anzeige zweier Snapshots im Snapshot-Bereich synchronisieren. Snapshots werden in der Bildansicht schneller angezeigt als in der vollständigen Ansicht.
Vollständig	Anzeigen einer grafischen Darstellung des Snapshots. Sie können die Anzeige zweier Snapshots im Snapshot-Bereich nicht synchronisieren. Snapshots werden in der vollständigen Ansicht langsamer angezeigt als in der Bildansicht.

Windows-Sockets-Protokoll

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Hex	Anzeigen der Pufferdaten im Hexadezimalformat.

Text	Anzeigen der Pufferdaten als Text.
Gehe zu	Öffnen des Dialogfelds **Gehe zu Offset**, das zum Navigieren in den Datenpuffern dient.

Ajax (Click and Script) und Web (Click and Script)-Protokolle

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Seitenansicht	Anzeigen einer grafischen Darstellung der Webseite.
Seitenquelle	Anzeigen des HTTP-Quellcodes der Webseite.

Webprotokoll (HTTP/HTML)

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Seitenansicht	Zeigt Schrittdaten im HTML-Format an.
HTTP-Daten	Zeigt Schrittdaten im HTTP-Format an. Dies ermöglicht es dem Benutzer, ausführliche Informationen zum Schritt einschließlich Anforderungsdaten, Antwortdaten, Cookies und Kopfzeilen anzuzeigen. Weitere Informationen finden Sie unter "Web-Snapshots - Übersicht" auf Seite 904.
Raster	(Nur HTTP-Ansicht). Zeigt die HTTP-Flussdaten in einem Listenformat an.
Struktur	(Nur HTTP-Ansicht). Zeigt die HTTP-Flussdaten in einer Strukturansicht an.

Protokolle mit einer XML-Anforderung/Antwort (z. B. das Web Services-Protokoll)

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Antwort	Zeigt die Serverantwort im XML-Format an.
Anforderung	Zeigt die Anforderung im XML-Format an.
	Öffnet das Dialogfeld für die XPath-Suche, indem Sie mithilfe eines XPath-Standardausdrucks nach dem XML-Code suchen können.

<Kontextmenü>	• **Auswahl kopieren.** Kopiert den in der Textansicht markierten Text in die Zwischenablage. • **In Community suchen.** Führt mit dem in der Textansicht als Suchzeichenfolge markierten Text eine Community-Suche aus. Weitere Informationen zum Ausführen einer Community-Suche finden Sie unter "Editor - Übersicht" auf Seite 59. • **XPath kopieren.** Kopiert in der Strukturansicht den XPath des ausgewählten Knotens in die Zwischenablage. Kopiert in der Textansicht den XPath des XML-Elements, in dem sich der Cursor befindet, in die Zwischenablage. • **Vollständigen Wert kopieren.** Kopiert in der Strukturansicht den vollständigen XML-Code des ausgewählten Knotens in die Zwischenablage. Kopiert in der Textansicht den vollständigen XML-Code des XML-Elements, in dem sich der Cursor befindet. • **XML-Prüfung einfügen.** Öffnet das Dialogfeld **XML-Prüfung einfügen**, in dem Sie einen **XML suchen**-Schritt in das Vuser-Skript einfügen können. Diese Option ist nur in der Antwortansicht verfügbar. Diese Option ist nur für Knoten vom Typ **Attribut** und **Wert** verfügbar. • **Wert in Parameter speichern.** Öffnet das Dialogfeld **Wert in Parameter speichern**, in dem Sie den ausgewählten Wert in einem einfachen Parameter speichern können. Diese Option ist nur in der Antwortansicht verfügbar. Diese Option ist nur für Knoten vom Typ **Attribut** und **Wert** verfügbar. • **XML in Parameter speichern.** Öffnet das Dialogfeld **Wert in Parameter speichern**, in dem Sie den ausgewählten Wert in einem XML-Parameter speichern können. Diese Option ist nur in der Antwortansicht verfügbar. • **Korrelation erstellen.** Öffnet die Registerkarte **Korrelation** in Design Studio. Der im Snapshot-Bereich markierte Text wird als manueller Korrelationseintrag in Design Studio angezeigt. Weitere Informationen finden Sie unter "Manuelles Korrelieren von Skripts" auf Seite 290.

	Diese Option ist nur in der Antwortansicht verfügbar. Diese Option ist in der Strukturansicht für Knoten vom Typ **Attribut** und **Wert** verfügbar und in der Textansicht, wenn der Text markiert ist. • **Korrelationsregel erstellen.** Öffnet das Dialogfeld **Als Regel hinzufügen**, in dem Sie den markierten Text als Teil einer Korrelationsregel hinzufügen können. Weitere Informationen finden Sie unter "Registerkarte "Korrelation" [Design Studio] - Übersicht" auf Seite 282. Diese Option ist nur in der Antwortansicht verfügbar. Diese Option ist in der Strukturansicht für Knoten vom Typ **Attribut** und **Wert** verfügbar und in der Textansicht, wenn der Text markiert ist.

Datenbankprotokolle

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Aufzeichnen	Anzeigen des Aufzeichnungs-Snapshots im Snapshot-Bereich.
Datenraster	Zeigt die Daten in einem Raster an.

Bereich "Textmarken"

Im Bereich **Textmarken** wird eine Liste der Textmarken im Vuser-Skript angezeigt. Sie können zwischen den Textmarken navigieren, um Ihren Code zu analysieren und zu debuggen.

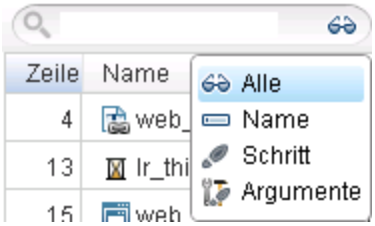
Zugriff	**VuGen > Ansicht > Textmarken**
Wichtige Informationen	• Alle zu einem Vuser-Skript hinzugefügten Textmarken werden beibehalten, wenn Sie den Bereich **Textmarken** schließen und erneut öffnen. • Sie können diesen Ausschnitt in andere Bereiche der Hauptbenutzeroberfläche verschieben. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56.
Relevante Aufgaben	"Verwenden von Textmarken" auf Seite 410 "Anpassen des Layouts von VuGen"
Siehe auch	"VuGen-Benutzeroberfläche – Übersicht" auf Seite 50

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Textmarkenliste>	Zeigt eine Liste aller Textmarken, die im Vuser-Skript definiert sind. Sie können auf eine beliebige Textmarkenzeile doppelklicken, um direkt zur entsprechenden Zeile im Vuser-Skript zu navigieren.
	Wechselt den Status der ausgewählten Textmarke.
	Navigiert zur vorherigen Textmarke im Bereich.
	Navigiert zur nächsten Textmarke im Bereich.
	Löscht die ausgewählte Textmarke.
	Löscht alle Textmarken.

Bereich "Schrittenavigator"

Im Bereich **Schrittenavigator** können Sie zu einem Schritt navigieren, der in Ihrem Skript ausgewählt ist. Wenn Ihr Skript viele Schritte enthält, können Sie das Suchfeld verwenden, um in den unterschiedlichen Teilen des Schritts nach übereinstimmendem Text zu suchen.



Zugriff	VuGen > Ansicht > Schritte
Wichtige Informationen	• Sie können in einer Aktion oder einem Schritt suchen, jedoch nicht in mehreren Skripts in einer Lösung. • Sie können diesen Ausschnitt in andere Bereiche der Hauptbenutzeroberfläche verschieben. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56. • Möglicherweise kommt es während der Wiedergabe zu einer Leistungsminderung, wenn der Bereich des Schrittenavigators angezeigt wird. **Problemumgehung:** Schließen Sie den Bereich des Schrittenavigators, wenn Sie ein Script wiedergeben. Alternativ können Sie das Wiedergabelayout so konfigurieren, dass der Bereich des Schrittenavigators ausgeschlossen wird.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Feld "Suchen"	Sie können unterschiedliche Teile der Schritte nach übereinstimmendem Text durchsuchen. Sie können in den folgenden Teilen der Schritte suchen: • **Alle**. (Standard) Alle Teile des Schritts (Name, Schritt, Argumente). • **Name**. Der Name des Schritts. • **Schritt**. Die Beschreibung des Schritts (zum Beispiel web_url, web_custom_request) • **Argumente**. Die Argumente für den Schritt. Geben Sie in das Suchfeld den Text ein, nach dem Sie in Ihren Schritten suchen möchten, und wählen Sie den Teil der Schritte aus, in dem die Suche erfolgen soll. Im Bereich **Schritte** werden nur diejenigen Schritte angezeigt, die Ihren Suchkriterien entsprechen.
Zeile	Die Nummer des Schritts im Skript.
Name	Ein Schrittname.
Schritt	Der Typ des Schritts.
	Datei-Parser-Indikator. Ein grünes Symbol zeigt an, dass das Parsing erfolgreich war. Ein rotes Symbol zeigt an, dass das Parsing fehlgeschlagen ist.
Aktion	Die Aktion, in der der Schritt erstellt wurde.
{0} Schritte werden angezeigt	Zeigt die Gesamtzahl der Schritte im Skript oder in der Aktion an.

Bereich "Fehler"

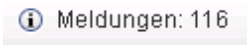
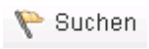
Im Bereich **Fehler** werden die im Skript gefundenen Wiedergabe- und Syntaxfehler aufgelistet. Sie können die einzelnen Fehler suchen und beheben.

Zugriff	VuGen > Ansicht > Fehler

Forts.

Wichtige Informationen	• Sie können diesen Ausschnitt in andere Bereiche der Hauptbenutzeroberfläche verschieben. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56. • Wenn Sie ein vorhandenes Skript öffnen, stammen die angezeigten Elemente im Bereich **Fehler** von der letzten Wiedergabe oder Kompilierung.
Relevante Aufgaben	"Anpassen des Layouts von VuGen"
Siehe auch	"VuGen-Benutzeroberfläche – Übersicht" auf Seite 50

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Verfügbare Kategorien definieren>	Filtert die Fehlerliste nach Fehlerquelle.
	Blendet Syntaxfehler ein oder aus.
	Blendet die während der Ausführung erkannten Warnungen ein oder aus.
	Blendet die während der Ausführung erkannten Informationsmeldungen ein oder aus.
	Springt zur Zeile im Editor, die den Fehler enthält.
! <Ausrufezeichen>	Meldungstyp: • Fehler • Warnung • Informationsmeldung
Zeile	Die Zeile, die den Fehler enthält.

Forts.

Beschreibung	Eine Beschreibung des Fehlers, der Warnung oder Meldung sowie Tipps zur Problemlösung. Beispiel: Ein Syntaxfehler wird angezeigt, wenn Sie einen Bedingungsblock mit einer If-Anweisung geöffnet, aber nicht mit einer End If-Anweisung geschlossen haben. Die Beschreibung lautet "Erwarteter Ausdruck". Hinweis: Wenn die Beschreibung für die vorgesehene Spalte zu lang ist, wird eine QuickInfo mit der vollständigen Beschreibung angezeigt, wenn der Cursor über die Spalte bewegt wird. In bestimmten Fällen kann VuGen den Fehler nicht genau identifizieren und zeigt eine Reihe möglicher Fehlerbedingungen an, z. B.: 'End Sub' oder 'End Function' oder 'End Property' erwartet. Überprüfen Sie die Anweisung in der angegebenen Zeile, um festzustellen, welcher Fehler für Sie relevant ist.
Datei	Der Name der Datei, die die problematische Anweisung enthält.
Pfad	Der vollständige Pfad der Datei, die den Fehler generiert hat.
Testen	Der Name des relevanten Skripts.

Aufgabenbereich

In diesem Bereich können Sie Aufgaben hinzufügen, bearbeiten und überwachen, die mit einem Skript oder den Gesamtzielen des Projekts verbunden sind.

Zugriff	**VuGen > Ansicht > Aufgabenbereich**
Wichtige Informationen	Sie können diesen Ausschnitt in andere Bereiche der Hauptbenutzeroberfläche verschieben. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56.
Relevante Aufgaben	"Anpassen des Layouts von VuGen"
Siehe auch	"VuGen-Benutzeroberfläche – Übersicht" auf Seite 50

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

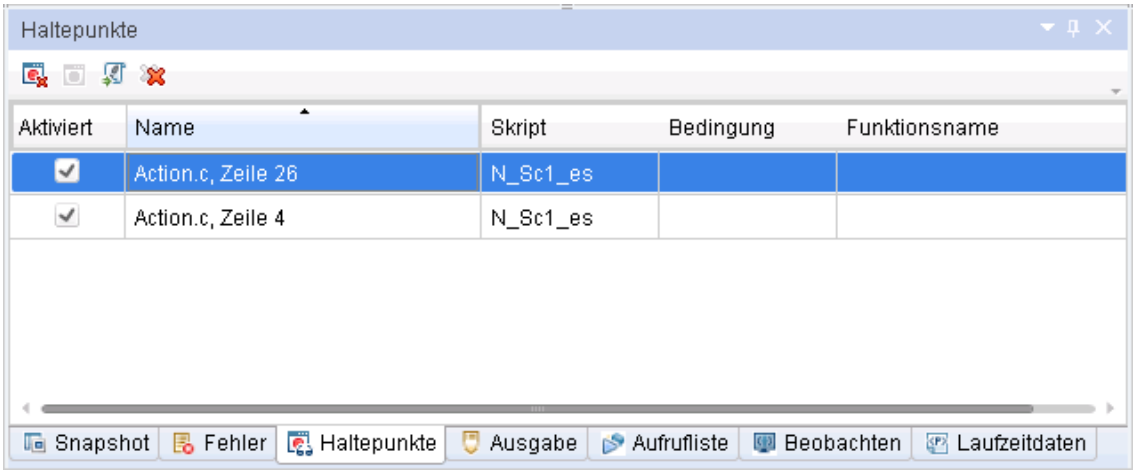
Element der Oberfläche	Beschreibung

Forts.

Registerkarte **Kommentare**	Kommentaraufgaben werden dem Skript direkt hinzugefügt, wenn Sie die Kommentarsyntax Ihrer Skriptsprache verwenden und Schlüsselwörter wie TODO oder FIXME einschließen. In C würde ein Kommentarskript beispielsweise so aussehen: `//TODO Add Parameter` Im Bereich für Aufgaben werden folgende Informationen über die einzelnen Aufgaben angezeigt: • !: Aufgabentyp • Zeile: In welcher Zeile sich die Aufgabe befindet. Durch Doppelklicken auf die Aufgabe gelangen Sie zu dieser Position im Skript. • Beschreibung: Die Inhalte des Schlüsselworts und der Aufgabe. • Datei: Aktion • Pfad: Der Speicherort der Aktion.
Registerkarte "Benutzeraufgaben"	Sie können Benutzeraufgaben hinzufügen, bearbeiten und löschen: Klicken Sie auf , um eine Aufgabe hinzuzufügen. Klicken Sie auf , um eine Aufgabe zu bearbeiten. Klicken Sie auf , um eine Aufgabe zu löschen.
<Aufgabenfilter>	Sie können Aufgaben filtern, die mit einem bestimmten Skript verknüpft sind, oder alle mit der Lösung verknüpften Aufgaben anzeigen.

Bereich "Haltepunkte"

Im Bereich **Haltepunkte** können Sie Haltepunkte festlegen und verwalten und damit besser überprüfen, welche Auswirkungen das Skript an vordefinierten Stellen bei der Ausführung hat.



Zugriff	**VuGen > Ansicht > Debuggen > Haltepunkte**
Wichtige Informationen	Sie können diesen Ausschnitt in andere Bereiche der Hauptbenutzeroberfläche verschieben. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56.
Relevante Aufgaben	"Debuggen von Skripts mit Haltepunkten" auf Seite 421

Bereich "Haltepunkte"

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

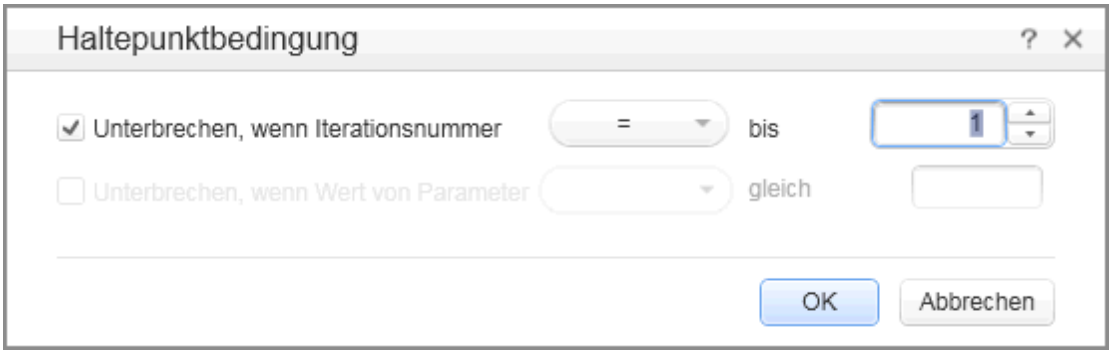
Element der Oberfläche	Beschreibung
	Entfernt den ausgewählten Haltepunkt.
	Entfernt alle Haltepunkte.
	Platziert den Cursor im Vuser-Skript in der Zeile, die den ausgewählten Haltepunkt enthält.
	Deaktiviert den ausgewählten Haltepunkt, wenn dieser aktiviert ist, und aktiviert ihn, wenn er deaktiviert ist.
	Ermöglicht Ihnen die Eingabe von Bedingungen für den ausgewählten Haltepunkt. Weitere Informationen finden Sie unter "Dialogfeld "Haltepunktbedingung"" auf der nächsten Seite.

Forts.

<Haltepunkttabelle>	Eine Liste der Haltepunkte und deren Platzierungen im Skript. Aktivieren Sie das Kontrollkästchen **Aktiviert** neben einem Haltepunkt, um den entsprechenden Haltepunkt zu aktivieren. Deaktivieren Sie das entsprechende Kontrollkästchen **Aktiviert**, um einen Haltepunkt zu deaktivieren.
Aktiviert	Ein Kontrollkästchen, das angibt, ob der Haltepunkt aktiviert oder deaktiviert ist und über das Sie den benachbarten Haltepunkt aktivieren oder deaktivieren können.
Name	Der Name der Datei, die den Haltepunkt enthält, sowie die Zeilennummer in der Datei, die den Haltepunkt enthält.
Skript	Der Name des Vuser-Skripts, das den Haltepunkt enthält.
Bedingung	Die Bedingung, die für diesen Haltepunkt gilt. Wenn keine Bedingung vorhanden ist, wird die Wiedergabe immer am nächsten Haltepunkt angehalten.
Funktionsname	Der Name der Funktion im Vuser-Skript, das den Haltepunkt enthält.

Dialogfeld "Haltepunktbedingung"

Im Dialogfeld **Haltepunktbedingung** können Sie Haltepunktbedingungen nach der Iterationsnummer, nach dem Wert eines oder mehrerer Parameter, oder nach einer Kombination aus Parameterwerten und Iterationsnummer angeben.



Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
Unterbrechen, wenn Iterationsnummer ist	Ein Kontrollkästchen, mit dem Sie einen Haltepunkt abhängig von der Iterationsnummer angeben können.

Forts.

<Operand>	Wählen Sie einen der folgenden Operanden: • = (Gleich) • <= (Kleiner als oder gleich) • => (Gleich oder größer als) • < (Kleiner als) • > (Größer als)
in	Geben Sie eine Iterationsnummer ein.
Unterbrechen, wenn Parameter	Ein Kontrollkästchen, mit dem Sie einen Haltepunkt abhängig vom Wert eines Parameters angeben können.
<Parametername>	Wählen Sie einen Parameter aus der Dropdownliste aus.
Wert ist	Geben Sie den Parameterwert ein, für den Sie einen Haltepunkt wünschen.
OK	Wenden Sie die Bedingungen auf den ausgewählten Haltepunkt an. Alle zukünftigen Wiedergaben werden nur an diesem Haltepunkt angehalten, wenn diese Bedingungen erfüllt sind.

Überwachungsbereich

Im Überwachungsbereich können Sie Variablen überwachen, während ein Skript ausgeführt wird.

Zugriff	**VuGen > Ansicht > Debuggen > Beobachten**
Wichtige Informationen	• Dieser Bereich ist nur relevant, wenn eine ausgeführte Sitzung angehalten wird. • Sie können diesen Ausschnitt in andere Bereiche der Hauptbenutzeroberfläche verschieben. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Ermöglicht Ihnen das Hinzufügen einer Variable zur Überwachungsliste.

Forts.

	Ermöglicht das Bearbeiten der Variable, die in der Überwachungsliste ausgewählt ist.
	Löscht die ausgewählte Variable aus der Überwachungsliste.
	Löscht alle ausgewählten Variablen aus der Überwachungsliste.
Ausdruck	Die Variable, deren Wert Sie beobachten möchten.
Wert	Der aktuelle Wert der Variable. Der ausgewertete Wert wird nur angezeigt, wenn eine ausgeführte Sitzung angehalten wird.
Typname	Der Typ des Variablenwerts nach der Auswertung (zum Beispiel **Ganzzahl** oder **Zeichen**). Wenn eine Variable im aktuellen Kontext nicht ausgewertet werden kann, ist der angezeigte Typ **Falscher Ausdruck**.

Bereich "Laufzeitdaten"

Im Bereich **Laufzeitdaten** werden Informationen über die aktuelle Skriptausführung angezeigt.

Zugriff	**VuGen > Ansicht > Debuggen > Laufzeitdaten**
Wichtige Informationen	• Sie können nur während der Skriptwiedergabe auf den Bereich **Laufzeitdaten** zugreifen. • Sie können diesen Ausschnitt in andere Bereiche der Hauptbenutzeroberfläche verschieben. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56.
Relevante Aufgaben	"Anpassen des Layouts von VuGen"
Siehe auch	"VuGen-Benutzeroberfläche – Übersicht" auf Seite 50 "Wiedergeben eines Vuser-Skripts" auf Seite 329

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
Iteration	Zeigt die aktuelle Iterationsnummer an.
Aktion	Zeigt den Aktionsnamen des aktuell wiedergegebenen Schritts an.
Zeilennummer	Zeigt die Zeilennummer des aktuell wiedergegebenen Schritts an.

Forts.

Verstrichene Zeit	Zeigt die verstrichene Zeit seit dem Beginn der Wiedergabe an.
<Parameter>	Zeigt alle im Skript definierten Parameter und ihre Substitutionswerte auf der Basis der ausgewählten Aktualisierungsmethode an (sequentiell, einmalig usw.). VuGen zeigt diese Informationen auch dann an, wenn der Parameter im Skript nicht verwendet wird.

Bereich "Aufrufliste"

In diesem Debug-Bereich können Sie Informationen zu den Funktionen anzeigen, die sich aktuell in der Aufrufliste Ihres Skripts befinden.

Zugriff	**VuGen > Ansicht > Debuggen > Aufrufliste**
Wichtige Informationen	• Dieser Bereich ist nur relevant, wenn eine ausgeführte Sitzung angehalten wird. • Sie können auf ein beliebiges Element im Bereich der Aufrufliste doppelklicken, um zum Anfang der entsprechenden Funktion/Aktion zu navigieren. • Dieser Bereich ist schreibgeschützt. • Sie können diesen Ausschnitt in andere Bereiche der Hauptbenutzeroberfläche verschieben. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56.
Relevante Aufgaben	"Anpassen des Layouts von VuGen"
Siehe auch	"VuGen-Benutzeroberfläche – Übersicht" auf Seite 50

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Funktionsname	Die Bezeichnung der aktuell aufgerufenen Funktion.
Dateiname	Der Name der Datei, die die aufgerufene Funktion enthält.
Zeilennummer	Die Nummer der Zeile, in der die Funktionsdefinition beginnt.

Registerkarte "Wiedergabeübersicht"

Auf dieser Registerkarte stehen eine Zusammenfassung der Wiedergabeergebnisse und Hyperlinks zu Details zur Skriptwiedergabe zur Verfügung.

Zugriff	Führen Sie einen der folgenden Schritte durch: • **Lösungs-Explorer >** Rechtsklick auf **Wiedergabeausführungen >** Auswahl der **Registerkarte "Wiedergabeübersicht öffnen"** • Auswahl des Hyperlinks Zusammenfassung anzeigen im **Ausgabebereich**
Wichtige Informationen	Um Transaktionsaufschlüsselungsdaten zu aktivieren, wählen Sie **Optionen > Skripterstellung > Wiedergeben > Wiedergabestatistik aufzeichnen**. **Achtung:** Das Aktivieren der Funktion **Wiedergabestatistik aufzeichnen** beeinflusst die Wiedergabeleistung.
Siehe auch	• "Ausgabebereich" auf Seite 107 • "Wiedergeben eines Vuser-Skripts" auf Seite 329

Element der Oberfläche	Beschreibung
Ergebnis-Dashboard	Zeigt grundlegende Skriptinformationen an, u. a.: • **Skriptname** • Wiedergabestatus Zeigt als Wiedergabestatus des Skripts entweder Bestanden oder Fehler an. • Verstrichene Zeit Verstrichene Gesamtzeit der Skriptwiedergabe. • Startzeit Startzeit der Skriptwiedergabe • Endzeit Endzeit der Skriptwiedergabe • Latenzzeit Gesamtdauer der **Latenzzeit**[1], die während der Skriptwiedergabe verstrichen ist. • Verstrichene Zeit Gesamtdauer der **Verstrichenen Zeit**, die während der Skriptwiedergabe verstrichen ist. • **Ergebnisbereich auswählen** Ermöglicht die Auswahl eines bestimmten Iterationsergebnisdatensatzes aus einer Dropdownliste oder aus einem Durchschnitt aller Iterationen.

[1]Zeitdauer, die in ein Skript eingefügt wird, um die Bedenkzeit zu simulieren, die entsteht, bevor der Benutzer den nächsten Schritt ausführt.

Element der Oberfläche	Beschreibung
Skripttransaktionen	Zeigt eine Liste aller Transaktionen in Ihrem Skript mit folgender Statistik an: • Dauer: Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht die angezeigte Zeit der Dauer der Transaktion. Wenn der Bereich ein Durchschnittswert ist, entspricht die angezeigte Zeit der durchschnittlichen Dauer aller Iterationen. • Trend: Sparkline-Darstellung der Werte für alle Iterationen. • Status Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, wird als Status entweder **Bestanden** oder **Fehler** angezeigt. Wenn der Bereich als Durchschnittswert angegeben wird, entspricht der angezeigte Status der Anzahl der Transaktionen mit dem Status Bestanden/Iterationen gesamt. • Größe Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht die angezeigte Größe der Größe der an den Client zurückgegebenen Daten. Wenn es sich bei dem Bereich um einen Durchschnittswert handelt, entspricht die angezeigte Größe der durchschnittlichen Größe der an den Client zurückgegebenen Daten.

Element der Oberfläche	Beschreibung
Verbindungen pro Domäne	Zeigt Statistiken nach Domäne pro Transaktion an. Hinweis: Die Statistik ist nur für Web HTTP/HTML-Protokollskripts verfügbar. • Domänenname Name der Domäne oder die IP-Adresse der Domäne, falls der Name nicht gefunden wird. • **Anzahl** Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht die angezeigte Anzahl der Anzahl der Verbindungen zum Server pro Domäne. Wenn es sich bei dem Bereich um einen Durchschnittswert handelt, entspricht die angezeigte Anzahl dem Durchschnitt der Verbindungen zum Server pro Domäne. • **Anzahl %** Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht die Anzahl % dem Prozentsatz der Verbindungen zum Server pro Domäne. Wenn es sich bei dem Bereich um einen Durchschnittswert handelt, entspricht die Anzahl % dem durchschnittlichen Prozentsatz der Verbindungen zum Server pro Domäne. • **Trefferzahl** Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht die angezeigte Trefferzahl der Anzahl der vom Server angeforderten Dateien. Wenn es sich bei dem Bereich um einen Durchschnittswert handelt, entspricht die angezeigte Trefferzahl der durchschnittlichen Anzahl der vom Server angeforderten Dateien. • **Größe** Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht die angezeigte Größe der Größe der vom Server pro Domäne zurückgegebenen Daten. Wenn es sich bei dem Bereich um einen Durchschnittswert handelt, entspricht die angezeigte Größe der durchschnittlichen Größe der vom Server pro Domäne zurückgegebenen Daten.

Element der Oberfläche	Beschreibung
	• Größe % Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht Größe % dem Prozentsatz der vom Server pro Domäne zurückgegebenen Daten in Relation zu den insgesamt zurückgegebenen Daten. Wenn es sich bei dem Bereich um einen Durchschnittswert handelt, entspricht Größe % dem durchschnittlichen Prozentsatz der vom Server pro Domäne zurückgegebenen Daten in Relation zu den insgesamt zurückgegebenen Daten. • Trend: Sparkline-Darstellung der Werte für alle Iterationen. • DNS-Zeit Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht die angezeigte DNS-Zeit der zeitlichen Dauer, die der Webbrowser benötigt, um einen Hostnamen in die entsprechende IP-Adresse zu übersetzen. Wenn es sich bei dem Bereich um einen Durchschnittswert handelt, entspricht die angezeigte DNS-Zeit der durchschnittlichen zeitlichen Dauer, die der Webbrowser benötigt, um einen Hostnamen in die entsprechende IP-Adresse zu übersetzen.

Element der Oberfläche	Beschreibung
Verbindungen pro Inhaltstyp	Zeigt Statistiken nach Inhaltstyp pro Transaktion an. Hinweis: Die Statistik ist nur für Web HTTP/HTML-Protokollskripts verfügbar. • **Inhaltstyp** Liste der vom Server zurückgegebenen Inhaltstypen, beispielsweise ein Bild, ein JavaScript, ein CSS. • **Anzahl** Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht die angezeigte Anzahl der Anzahl der Verbindungen pro Inhaltstyp. Wenn es sich bei dem Bereich um einen Durchschnittswert handelt, entspricht die angezeigte Anzahl der durchschnittlichen Anzahl der Verbindungen pro Inhaltstyp. • **Anzahl %** Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht die Anzahl % dem Prozentsatz der Verbindungen zum Inhaltstyp des Servers. Wenn es sich bei dem Bereich um einen Durchschnittswert handelt, entspricht der angezeigte Status dem durchschnittlichen Prozentsatz der Verbindungen zum Inhaltstyp des Server in Relation zur Gesamtzahl der Verbindungen. • **Größe** Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht die angezeigte Größe der Größe der vom Server für jeden Inhaltstyp zurückgegebenen Daten. Wenn es sich bei dem Bereich um einen Durchschnittswert handelt, entspricht die angezeigte Größe der durchschnittlichen Größe der vom Server für jeden Inhaltstyp zurückgegebenen Daten. • Größe % Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht Größe % dem Prozentsatz der vom Server für jeden Inhaltstyp zurückgegebenen Daten in Relation zu den insgesamt zurückgegebenen Daten. Wenn es sich bei dem Bereich um einen Durchschnittswert

Element der Oberfläche	Beschreibung
	handelt, entspricht Größe % dem durchschnittlichen Prozentsatz der vom Server für jeden Inhaltstyp zurückgegebenen Daten in Relation zu den insgesamt zurückgegebenen Daten. • Trend: Sparkline-Darstellung der Werte für alle Iterationen.
Verbindungen pro HTTP-Status	Zeigt Statistiken nach dem HTTP-Status pro Transaktion an. Hinweis: Die Statistik ist nur für Web HTTP/HTML-Protokollskripts verfügbar. • **Status** Liste mit von der Transaktion zurückgegebenen HTTP-Statuscodes. • **Anzahl** Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht die angezeigte Anzahl der Anzahl der Verbindungen mit HTTP-Statuscode pro Transaktion. Wenn es sich bei dem Bereich um einen Durchschnittswert handelt, entspricht die angezeigte Anzahl der durchschnittlichen Anzahl der Verbindungen mit HTTP-Statuscode pro Transaktion. • Anzahl % Wenn es sich bei dem Bereich um eine einzelne Iteration handelt, entspricht die angezeigte Anzahl dem Prozentsatz der Verbindungen mit HTTP-Statuscode pro Transaktion. Wenn es sich bei dem Bereich um einen Durchschnittswert handelt, entspricht die angezeigte Anzahl dem durchschnittlichen Prozentsatz der Verbindungen mit HTTP-Statuscode pro Transaktion. • Trend: Sparkline-Darstellung der Werte für alle Iterationen.
Wiedergabestatistik – Übersicht	Eine Liste der Skriptstatistiken und deren Werte nach der Wiedergabe.

Element der Oberfläche	Beschreibung
Weitere Informationen finden Sie unter:	• **Testergebnisse** Weitere Informationen finden Sie unter "Fenster "Testergebnisse"" auf Seite 517. • **Wiedergabeprotokoll**
<Ändern der Laufzeiteinstellungen>	Ermöglicht Ihnen den Zugriff auf **Laufzeiteinstellungen** für das aktive Skript.
<Ändern der Regeln für die Inhaltsprüfung>	Der **ContentCheck**-Mechanismus von VuGen ermöglicht Ihnen die Ermittlung aller Typen von Fehlern, die vom Webserver gesendet wurden. Weitere Informationen finden Sie unter "Internetprotokoll > ContentCheck (Knoten)" auf Seite 369.

VuGen - Fehlerbehebung und Einschränkungen

In diesem Abschnitt werden die Fehlerbehebung und die für VuGen geltenden Einschränkungen beschrieben.

Fehlerbehebung bei Snapshots

Wenn Sie auf einen Schritt ohne Snapshot treffen, können Sie anhand der folgenden Hinweise herausfinden, warum kein Snapshot verfügbar ist. Beachten Sie, dass nicht allen Schritten Snapshots zugeordnet sind: Nur Schritte, die Bildschirmoperationen enthalten oder für das Web gedacht sind und Inhalte in einem Browserfenster darstellen, weisen Snapshots auf.

Mehrere Protokolle ermöglichen eine Deaktivierung der Snapshot-Aufzeichnung mithilfe der Aufzeichnungsoptionen.

Wenn für den ausgewählten Schritt kein Snapshot der **Aufzeichnung** anzeigt wird, kann dies unterschiedliche Ursachen haben:

- Das Skript wurde mit der VuGen-Version 6.02 oder einer früheren Version aufgezeichnet.
- Für bestimmte Schritt-Typen werden keine Snapshots erzeugt.
- Die importierten Aktionen enthalten keine Snapshots.

Wenn für den ausgewählten Schritt kein **Wiedergabe**-Snapshot angezeigt wird, kann dies unterschiedliche Ursachen haben:

- Das Skript wurde mit der VuGen-Version 6.02 oder einer früheren Version aufgezeichnet.
- Die importierten Aktionen enthalten keine Snapshots.
- Die Vuser-Dateien sind in schreibgeschützten Verzeichnissen gespeichert, und VuGen konnte

die Wiedergabe-Snapshots nicht speichern.

- Der Schritt stellt die Navigation zu einer Ressource dar.

Internet Explorer und Windows Server-Computer

Bei Verwendung von Internet Explorer auf Windows Server-Computern wie Windows Server 2008 und 2012 blockt die verstärkte Sicherheitskonfiguration des Browsers bestimmte Aktionen. Folgende Meldung wird ausgegeben: "Inhalt von der unten aufgeführten Website wird durch die verstärkte Sicherheitskonfiguration für Internet Explorer geblockt." Auf diese Weise wird der automatische Download von Dateien verhindert, die für Ihren Workflow erforderlich sind. **Problemumgehung:** Fügen Sie in Internet Explorer unter "Vertrauenswürdige Sites" den Eintrag "about:internet" hinzu.

Speicherprobleme

Wenn das Skript sehr groß ist, treten möglicherweise Probleme beim Öffnen des Skripts auf, wenn der Speicher auf den Computer nicht ausreicht. Um die VuGen-Speicherauslastung zu reduzieren, wählen Sie **Extras > Optionen**und deaktivieren Sie die Option **C-Sprachparser aktivieren**, um den C-Parser zu deaktivieren. Beachten Sie, dass sich dies auf einige der Grafikfunktionen, z. B. die Schrittenavigation und die Miniaturansichten auswirkt.

Übersicht über den Protocol Advisor

Der Protocol Advisor unterstützt Sie dabei, ein geeignetes Protokoll zum Aufzeichnen eines Vuser-Skriptes festzulegen. Er durchsucht Ihre Anwendung nach Elementen von verschiedenen Protokollen und zeigt eine Liste der ermittelten Protokolle an. Diese Protokolle müssen als Richtlinien und als Ausgangspunkt verwendet werden, um das optimale Protokoll für Ihre Anwendung zu finden.

In den meisten Fällen schlägt der Protocol Advisor mehrere Protokolle zusammen mit Kombinationen von Protokollen vor. Richtlinien zur Verwendung der vorgeschlagenen Protokolle finden Sie unter "Verwenden des Protocol Advisor" unten.

Verwenden des Protocol Advisor

Diese Aufgabe beschreibt einen typischen Workflow, um das optimale Protokoll zum Aufzeichnen Ihrer Anwendung mit dem Protocol Advisor zu finden.

1. **Starten des Protocol Advisor**

   Wählen Sie in VuGen **Aufzeichnen > Protocol Advisor > Anwendung analysieren** aus.

   Wenn Sie eine Webanwendung analysieren, müssen Sie die folgenden Angaben bereitstellen:

- Zu analysierendes Programm
- URL-Adresse
- Arbeitsverzeichnis

Wenn Sie eine Windows-Anwendung analysieren, müssen Sie die folgenden Angaben bereitstellen:

- Zu analysierendes Programm
- Programmargumente soweit vorhanden
- Arbeitsverzeichnis

Weitere Informationen finden Sie unter "Dialogfeld "Protocol Advisor"" auf der nächsten Seite.

2. **Ausführen typischer Geschäftsprozesse**

   Führen Sie typische Geschäftsprozesse in Ihrer Anwendung aus. Versuchen Sie, durch eine Reihe von Geschäftsprozessen zu navigieren, um sicherzustellen, dass Ihre Ergebnisse verständlich sind. Klicken Sie auf **Analyse fertig stellen**, um die Analyse zu beenden und die Ergebnisse anzuzeigen.

3. **Speichern der Ergebnisse**

   Wählen Sie **Aufzeichnen > Protocol Advisor > Ergebnis speichern** oder klicken Sie in der Symbolleiste oben im Bericht auf **Ergebnis speichern**. Geben Sie einen Namen ein und wählen Sie den Ordner aus.

   Um einen vorherigen Bericht anzuzeigen, wählen Sie **Aufzeichnen > Protocol Advisor > Letzte Berichte**. VuGen speichert die Historie der letzten sechzehn erstellten Protocol Advisor-Berichte. Um die Historie zu löschen, wählen Sie **Aufzeichnen > Protocol Advisor > Letzte Berichte löschen**.

4. **Auswählen eines Protokolls und Erstellen eines neuen Vuser-Skripts**

   Wählen Sie eines der Protokolle unter Berücksichtigung der folgenden Prioritätenfolge aus und erstellen Sie ein Vuser-Skript mit diesem Protokoll:

   - Multi-/Kombinationsprotokoll
   - Das Anwendungsprotokoll der obersten Ebene
   - Das erste Protokoll in der Liste

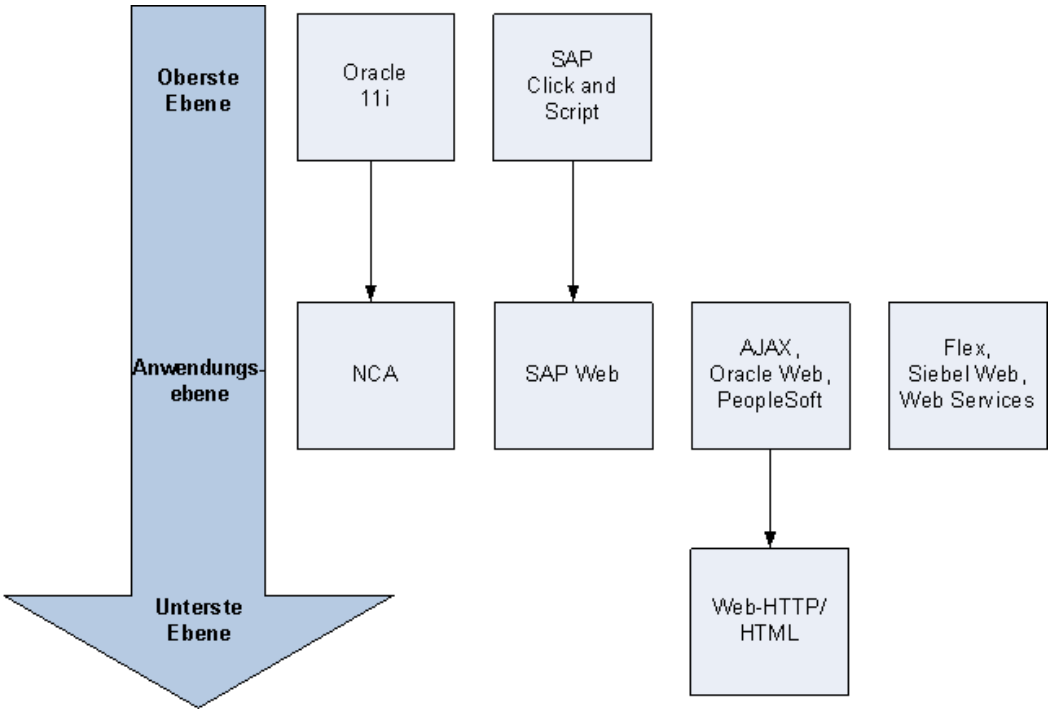
5. **Erweitern, Debuggen und Prüfen der Wiedergabe**

   Erweitern und debuggen Sie Ihr Skript, bis Sie es erfolgreich wiedergeben können. Fahren Sie, wenn Sie ein Vuser-Skript nach der Erweiterung und dem Debuggen nicht erfolgreich wiedergeben können, mit dem nächsten Schritt fort.

6. **Auswählen eines anderen Protokolls bei fehlgeschlagener Wiedergabe und Wiederholen des Vorgangs**

   - **Für alle nicht webbasierten Protokolle:** Kehren Sie zur Seite **Protocol Advisor-Ergebnis** zurück, wählen Sie das nächste Protokoll in der Liste aus und wiederholen Sie die zuvor beschriebenen Schritte. Befinden sich keine weiteren Protokolle mehr in der Liste, wenden Sie sich an den HP-Kundendienst.

   - **Für webbasierte Protokolle:** Das folgende Diagramm zeigt mehrere webbasierte Protokolle nach ihrer Anwendungsebene angeordnet. Die Pfeile geben an, in welcher Reihenfolge Sie ein neues Protokoll versuchen sollten. Wenn beispielsweise das erste Protokoll, das Sie verwendet haben, Oracle Web Applications 11i war und es nicht erfolgreich wiedergegeben werden konnte, versuchen Sie anschließend das Protokoll Oracle NCA.



Dialogfeld "Protocol Advisor"

In diesem Dialogfeld können Sie Ihre Anwendung analysieren, um das geeignete VuGen-Protokoll für die Aufzeichnung zu ermitteln.

Zugriff	Verwenden Sie eine der folgenden Methoden: • **VuGen > Aufzeichnen > Protocol Advisor > Anwendung analysieren** • **VuGen > Datei > Neues Skript und neue Lösung > Protocol Advisor**
Wichtige Informationen	• Die in diesem Dialogfeld angezeigten Optionen hängen von Ihrer Auswahl im Feld **Anwendungstyp** ab.
Relevante Aufgaben	"Verwenden des Protocol Advisor" auf Seite 133

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Anwendungstyp	Windows- oder Webanwendung
Programmargumente (nur Win32-Anwendungen)	Geben Sie die Befehlszeilenargumente für die oben angegebene ausführbare Datei an. Wenn Sie beispielsweise *plus32.exe* mit den Befehlszeilenoptionen `peter@neptune` angeben, wird der Benutzer *Peter* beim Starten von *plus32.exe* mit dem Server *Neptune* verbunden. Diese Option wird nur für Windows-Anwendungen angezeigt.
Zu analysierendes Programm	Wählen Sie den Browser, die Webanwendung oder die Windows-Anwendung aus, die Sie analysieren möchten.
URL-Adresse (nur Internet-Anwendungen)	Die Anfangs-URL-Adresse. Diese Option wird nur für Webanwendungen angezeigt.
Arbeitsverzeichnis	Das Arbeitsverzeichnis für die Anwendung, soweit erforderlich.

Protocol Advisor - Fehlerbeseitigung und Beschränkungen

In diesem Abschnitt werden die Fehlerbehebung und die für den Protocol Advisor geltenden Einschränkungen beschrieben.

- Diese Funktion ermittelt nur von LoadRunner unterstützte Protokolle.
- Manche Webprotokolle werden auf der Basis des URL ermittelt. Der URL kann beispielsweise Schlüsselwörter enthalten, wie z. B. SAP. Wenn Sie also einen anderen URL oder eine andere Anwendung mit demselben URL verwenden, können die Ergebnisse abweichen.

- Die folgenden Protokolle werden häufig gefunden, eignen sich jedoch nicht immer zur Verwendung. Sie sollten sie nur verwenden, wenn keine anderen ermittelten Protokolle vorliegen.
  - COM/DCOM
  - Java
  - .NET
  - WinSock
  - LDAP
- Wenn Sie den Protocol Advisor unmittelbar nach dem Starten des Erkennungsprozesses beenden, hinterlässt er möglicherweise einen nicht zugeordneten Prozess in der erkannten Anwendung, wodurch alle nachfolgenden Aufzeichnungen und Erkennungssitzungen fehlschlagen.

Lösung: Beenden Sie den nicht zugeordneten Prozess manuell.

HP Live Network (HPLN)-Integration

HP Live Network (HPLN) bietet Ihnen zusätzliche Softwareinhalte und Informationen über Ihre HP-Softwareprodukte. Mit der Integrationsfunktion für LoadRunner und HPLN können Sie Inhalte herunter- und hochladen, die dann von anderen LoadRunner-Benutzern gemeinsam genutzt werden können.

Sie können folgende Inhaltstypen von HPLN herunterladen und zu HPLN hochladen:

- Aktions-/Funktionsdateien mit den Erweiterungen .c, .java, .js und .cs.

  Hinweis: Aus Sicherheitsgründen müssen .js-Dateien vor dem Hochladen zu HPLN komprimiert und nach dem Herunterladen von HPLN dekomprimiert werden.

- Korrelationsdateien mit der Erweiterung .cor
- DFE-Dateien (Datenformaterweiterung) mit der Erweiterung .vucsx
- VuGen-Add-In-Dateien mit der Erweiterung .zip

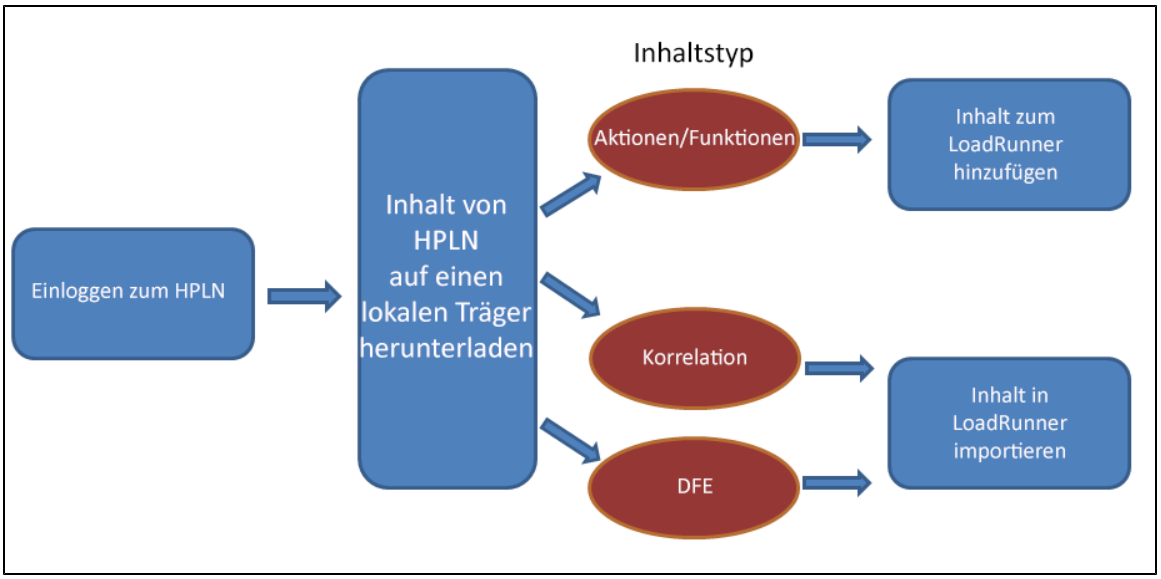
Weitere Informationen über die Verwendung von HPLN finden Sie in der folgenden Tabelle:

Vorgehensweise ...	Beschreibungen der Oberfläche
• "Herunterladen von Inhalten von HP Live Network (HPLN) in LoadRunner" unten • "Hochladen von Inhalten aus LoadRunner zu HP Live Network (HPLN)" auf Seite 140	• "Dialogfeld "Verbindung zu HP Live Network"" auf Seite 142 • "Dialogfeld "Von HP Live Network herunterladen"" auf Seite 144

Herunterladen von Inhalten von HP Live Network (HPLN) in LoadRunner

Die folgende Aufgabe beschreibt das Herunterladen von Inhalten von HPLN und das Importieren dieser Inhalte in Ihr LoadRunner-Projekt.

Der Ablauf für das Herunterladen von Inhalt nach LoadRunner lautet wie folgt:



1. Klicken Sie in der VuGen-Hauptsymbolleiste auf HPLN und melden Sie sich bei HPLN an.

   Wenn Sie bereits angemeldet sind, werden Sie automatisch auf die Seite zum Herunterladen von HPLN weitergeleitet.

   Hinweis: Die erste Anmeldung bei HPLN in einer VuGen-Sitzung kann mehrere Minuten in Anspruch nehmen, während die Seite mit dynamischen Inhalten erstellt wird. Nachfolgende Ladevorgänge der Inhaltsseite sind wesentlich schneller.

   Weitere Informationen zur Anmeldung bei HPLN finden Sie unter "Dialogfeld "Verbindung zu HP Live Network"" auf Seite 142.

2. Wählen Sie im Bildschirm **Von HP Live Network herunterladen** den relevanten HPLN-Inhaltstyp aus, und klicken Sie auf dem relevanten Inhalt auf **Download**, um den Inhalt herunterzuladen.

   Wenn Sie auf **Download** klicken, wird der Status des Downloadvorgangs angezeigt, und wenn die ausgewählte Datei heruntergeladen wurde, wird der Status entsprechend geändert.

   Hinweis: Heruntergeladener Inhalt wird im entsprechenden Unterordner von **%programdata%\Hewlett-Packard\LoadRunner\HPLN** gespeichert.

   Weitere Informationen zum Herunterladen von Inhalten finden Sie unter "Dialogfeld "Von HP Live Network herunterladen"" auf Seite 144.

3. Führen Sie je nach Typ des heruntergeladenen Inhalts folgende Schritte aus, wenn Sie den Inhalt in LoadRunner verwenden möchten:

   a. **Aktions-/Funktionsinhalte**

      i. Wenn Sie eine ZIP-Datei mit einer JS-Datei heruntergeladen haben, müssen Sie die JS-Datei entpacken, bevor Sie die Datei als zusätzliche Datei hinzufügen können.

      ii. Klicken Sie im Lösungs-Explorer für das Skript, das Sie entwickeln, mit der rechten Maustaste auf **Zusätzliche Dateien**, und wählen Sie **Von HPLN heruntergeladene Dateien hinzufügen** aus.

      iii. Wählen Sie dann im Ordner **%programdata%\Hewlett-Packard\LoadRunner\HPLN\function** die relevante Funktionsdatei aus, und klicken Sie auf **Öffnen**.

   b. **Korrelationsinhalte**

      i. Wählen Sie im VuGen-Hauptmenü **Aufzeichnen** > **Aufzeichnungsoptionen** > **Korrelationen - Regeln** aus, und klicken Sie auf **Importieren**.

      ii. Wählen Sie im Ordner **%programdata%\Hewlett-Packard\LoadRunner\HPLN\cor** die Korrelationsdatei aus, und klicken Sie auf **Öffnen**.

   c. **DFE-Inhalte**

      i. Wählen Sie im VuGen-Hauptmenü **Aufzeichnen** > **Aufzeichnungsoptionen** > **Datenformaterweiterung - Codegenerierung** aus.

      ii. Wählen Sie **Datenformaterweiterung aktivieren** aus, und klicken Sie auf **Importieren**.

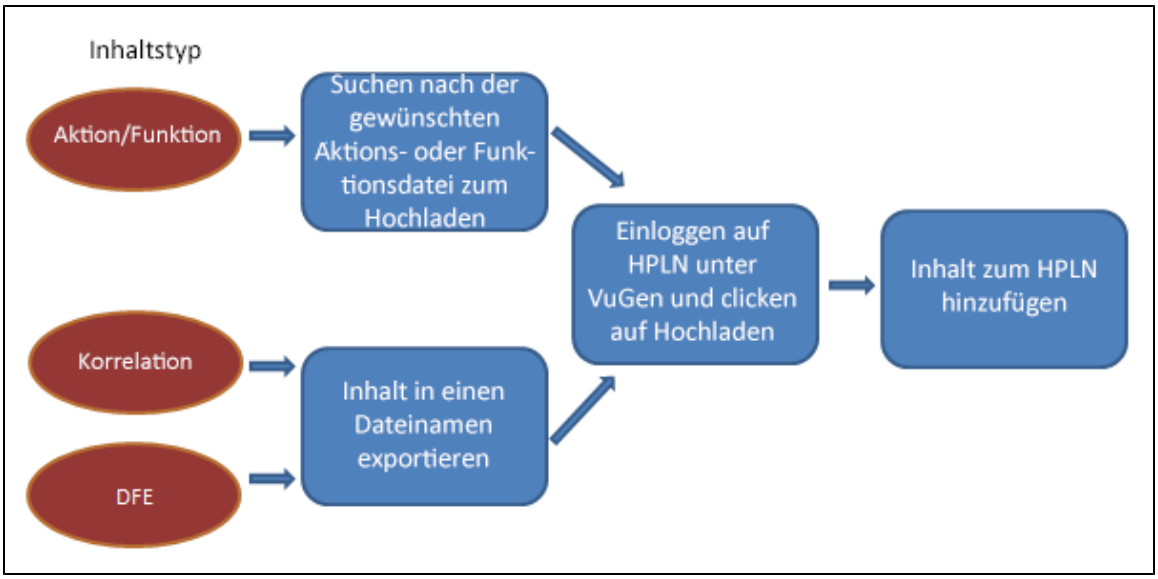
      Wenn die Meldung "Durch diesen Vorgang werden die aktuellen Einstellungen überschrieben" angezeigt wird und die aktuellen Einstellungen überschrieben werden sollen, klicken Sie auf **Ja**.

iii. Wählen Sie im Ordner **%programdata%\Hewlett-Packard\LoadRunner\HPLN\dfe** die DFE-Datei aus, und klicken Sie auf **Öffnen**.

Hochladen von Inhalten aus LoadRunner zu HP Live Network (HPLN)

In der folgenden Aufgabe wird beschrieben, wie Sie Inhalte aus LoadRunner zu HPLN hochladen können.

Der Ablauf für das Hochladen von Inhalten aus LoadRunner zu HPLN lautet wie folgt:



1. Exportieren Sie die Inhalte, die Sie hochladen möchten, wie folgt in eine Datei:

Aktions-/Funktionsinhalte

Suchen Sie als Vorbereitung für das Hochladen zu HPLN nach der gewünschten Aktions- oder Funktionsdatei.

> **Hinweis:** Wenn Sie eine JS-Datei hochladen möchten, müssen Sie diese aus Sicherheitsgründen komprimieren und dann die komprimierte Datei hochladen.

Korrelationsinhalte

a. Wählen Sie im VuGen-Hauptmenü **Aufzeichnen** > **Aufzeichnungsoptionen** > **Korrelationen - Regeln**.

b. Klicken Sie auf **Exportieren**, wählen Sie eine **Zu exportierende Anwendung** aus und klicken Sie erneut auf **Exportieren**.

c. Geben Sie einen Dateinamen und Speicherort ein, und klicken Sie dann auf **Speichern**.

> **Hinweis:** Die Datei wird standardmäßig im Ordner "%programdata%\Hewlett-Packard\LoadRunner\HPLN\cor" gespeichert.

DFE-Inhalte

a. Wählen Sie im VuGen-Hauptmenü **Aufzeichnen** > **Aufzeichnungsoptionen** > **Datenformaterweiterung - Codegenerierung**.

b. Wählen Sie **Datenformaterweiterung aktivieren**, und klicken Sie dann auf **Exportieren**.

c. Geben Sie einen Dateinamen und Speicherort ein, und klicken Sie dann auf **Speichern**.

> **Hinweis:** Die Datei wird standardmäßig im Ordner "%programdata%\Hewlett-Packard\LoadRunner\HPLN\dfe" gespeichert.

2. ## Anmelden bei HPLN

Klicken Sie in der VuGen-Hauptsymbolleiste auf HPLN und melden Sie sich bei HPLN an.

> **Hinweis:** Die erste Anmeldung bei HPLN in einer VuGen-Sitzung kann mehrere Minuten in Anspruch nehmen, während die Seite mit dynamischen Inhalten erstellt wird. Nachfolgende Ladevorgänge der Inhaltsseite sind wesentlich schneller.

> **Hinweis:** Wenn Sie bereits angemeldet sind, werden Sie automatisch auf die Seite zum Herunterladen von HPLN weitergeleitet.

3. Klicken Sie im Bildschirm **Von HP Live Network herunterladen** auf **Inhalt auf HPLN hochladen** .

4. Klicken Sie im HPLN-Hauptbildschirm auf die Registerkarte **Inhalt** und anschließend auf **Inhalt hinzufügen**.

5. ## Starten des Verfahrens zum Hochladen

Geben Sie die folgenden Daten ein:

a. Den Namen und eine Beschreibung für den Inhalt.

> **Hinweis:** Name und Beschreibung des Inhalts werden verwendet, wenn Sie den Inhalt auf dem Bildschirm **Von HP Live Network herunterladen** durchsuchen.

b. Die unterstützten Produktversionen.

> **Hinweis:** Wenn das System die Inhaltsliste für den Bildschirm **Von HP Live Network herunterladen** vorbereitet, werden die Inhalte angezeigt, die für die von Ihnen verwendete Version von VuGen relevant sind.

c. Die Version.

Geben Sie eine Versionsnummer ein. Eine Versionsnummer wird beim ersten Hinzufügen von Inhalten automatisch generiert. Die Versionsnummer für den Inhalt ist pro Aktualisierung eindeutig. Wenn Sie später den angegebenen Inhalt aktualisieren, müssen Sie eine andere Versionsnummer eingeben.

d. Der Inhaltstyp.

Wählen Sie den relevanten Inhaltstyp.

> **Hinweis:** Der Inhalt wird entsprechend dem Inhaltstyp bereitgestellt und ausgewählt: "lr_cr" für Korrelationsinhalte, "lr_dfe" für DFE-Inhalte und "lr_ff" für Aktions-/Funktionsinhalte.

6. **Anhängen von Inhalten**

Klicken Sie auf **Inhaltsanhänge hinzufügen**.

Im Bildschirm zum Erstellen von Inhaltsanhängen:

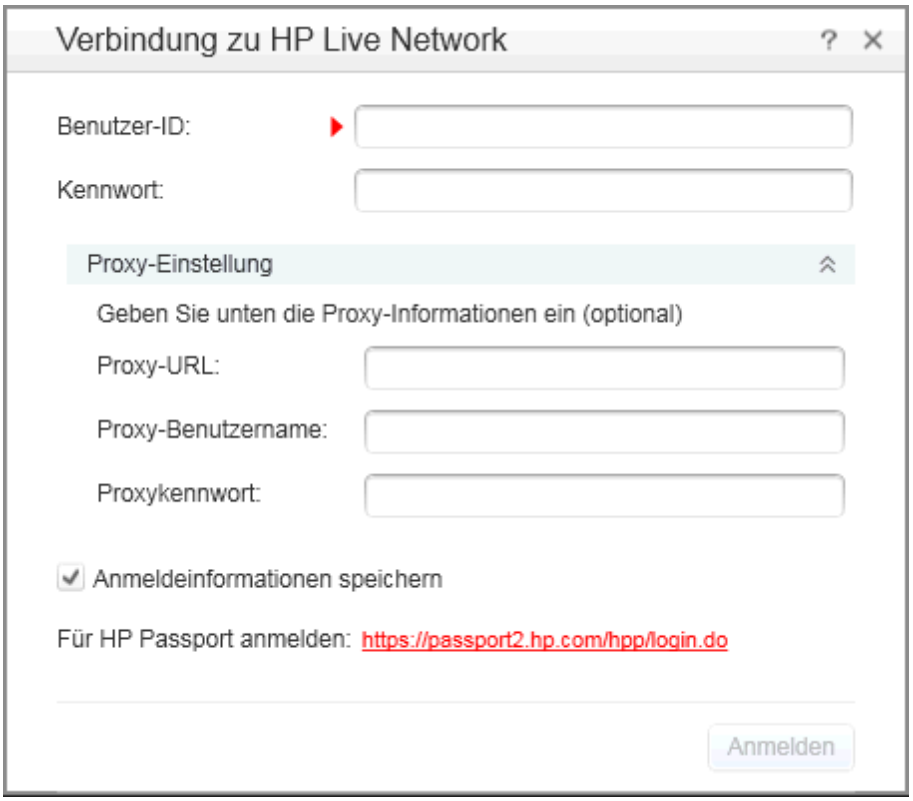
a. Geben Sie den Namen und die Version ein, und klicken Sie dann auf die Registerkarte **Dateianhänge**.

b. Klicken Sie auf **Datei auswählen**, wählen Sie die gewünschte Inhaltsdatei zum Hochladen aus, und klicken Sie auf **Fertig stellen**.

7. Klicken Sie nach dem Überprüfen der Zusammenfassung auf **Veröffentlichen** und in dem sich daraufhin öffnenden Bildschirm auf **Absenden**.

Dialogfeld "Verbindung zu HP Live Network"

Über dieses Dialogfeld können Sie sich bei HP Live Network (HPLN) anmelden und erhalten dann Zugriff auf die HPLN-Inhaltsdateien.



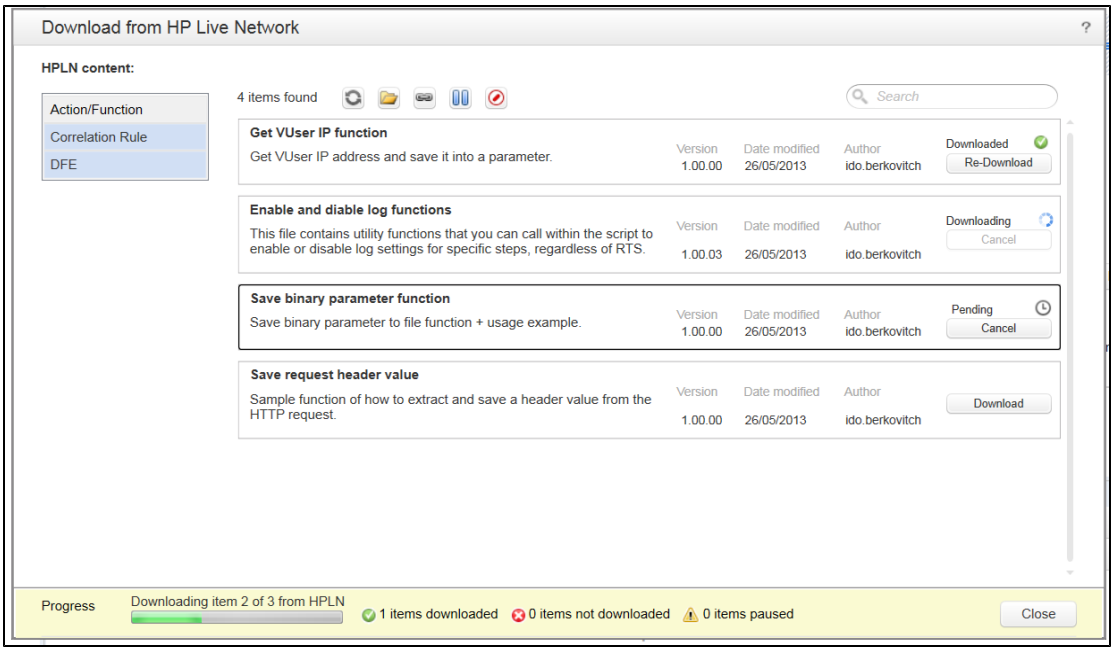
Zugriff	Führen Sie einen der folgenden Schritte durch: • Klicken Sie auf der VuGen-Hauptsymbolleiste auf HPLN. • Wählen Sie im VuGen-Hauptmenü **Extras > HPLN-Anmeldeinformationen zurücksetzen** aus.
Wichtige Informationen	Wenn Sie **Anmeldeinformationen speichern** auswählen, wird das Dialogfeld **Verbindung zu HP Live Network** nur angezeigt, wenn Sie die Anmeldeinformationen zurücksetzen. **Hinweis:** Wenn Sie zum ersten Mal in einer Sitzung auf HPLN zugreifen, kann es mehrere Minuten dauern, bis die Funktion das Fenster **Von HP Live Network herunterladen** vollständig anzeigt.
Relevante Aufgaben	"Herunterladen von Inhalten von HP Live Network (HPLN) in LoadRunner" auf Seite 138 "Hochladen von Inhalten aus LoadRunner zu HP Live Network (HPLN)" auf Seite 140

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
Benutzer-ID	Geben Sie Ihre Passport-ID ein.
Password	Geben Sie das Kennwort für Ihre Passport-ID ein.
Proxy-Einstellung	Wenn Sie über einen Proxy auf das Internet zugreifen und Ihre Proxy-Einstellungen auf dem Computer nicht automatisch konfiguriert werden, geben Sie die folgenden Proxy-Daten ein: **Proxy-URL.** Den URL Ihres Proxy-Servers. **Proxy-Benutzername.** Ihr Proxy-Benutzername. **Proxy-Kennwort.** Ihr Proxy-Kennwort.
<Anmeldeinformationen speichern>	Wenn diese Option ausgewählt ist, werden Ihre vorherigen Anmeldeinformationen zum Anmelden bei HPLN verwendet. **Hinweis:** Wenn Sie sich mit einem anderen Benutzernamen anmelden möchten, wählen Sie im VuGen-Hauptmenü **Extras > HPLN-Anmeldeinformationen zurücksetzen**.
Login	Klicken Sie auf diese Schaltfläche, um sich bei HPLN anzumelden.

Dialogfeld "Von HP Live Network herunterladen"

Mit diesem Dialogfeld können Sie Inhaltsdateien von HPLN herunterladen.



Zugriff	Führen Sie einen der folgenden Schritte durch: • Klicken Sie auf der VuGen-Hauptsymbolleiste auf HPLN. • Wählen Sie im VuGen-Hauptmenü **Extras > HPLN** aus.
Wichtige Informationen	• Wenn Sie nicht bei HPLN angemeldet sind, werden Sie zum Dialogfeld **Verbindung zu HP Live Network** weitergeleitet. • Wenn Sie bereits bei HPLN angemeldet sind, wird das Dialogfeld **Von HP Live Network herunterladen** angezeigt. **Hinweis:** Wenn Sie zum ersten Mal in einer Sitzung auf HPLN zugreifen, kann es mehrere Minuten dauern, bis die Funktion das Fenster **Von HP Live Network herunterladen** vollständig anzeigt. • Mit der Schaltfläche **Schließen** wird das Dialogfeld geschlossen und das Herunterladen der aktuell für den Vorgang ausgewählten Inhalte abgeschlossen. • Sie können jeweils nur eine HPLN-Sitzung auf einem Server verwenden. Wenn Sie mehrere Instanzen von VuGen geöffnet haben, können Sie nicht von mehreren Sitzungen gleichzeitig auf HPLN zugreifen.
Relevante Aufgaben	"Herunterladen von Inhalten von HP Live Network (HPLN) in LoadRunner" auf Seite 138 "Hochladen von Inhalten aus LoadRunner zu HP Live Network (HPLN)" auf Seite 140

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
	Inhalte aktualisieren. Füllt die HPLN-Liste mit zum Herunterladen verfügbaren Inhalten erneut auf.
	Inhaltsordner öffnen. Öffnet den Ordner, in den die Inhalte heruntergeladen wurden. **Hinweis:** Da der Dateiname nicht auf dem Bildschirm angezeigt wird, können Sie mit dieser Option den Namen der heruntergeladenen Inhaltsdatei anzeigen.

Element der Oberfläche	Beschreibung
	Inhalt auf HPLN hochladen. Leitet Sie zur Upload-Website von HPLN. Sie können Inhalte mit anderen LoadRunner-Benutzern gemeinsam verwenden, indem Sie Inhalte zu diesem Speicherplatz hochladen. **Hinweis:** Zum Hochladen von Inhalten zu HPLN benötigen Sie die entsprechenden Berechtigungen für Inhaltsbeiträge. Diese können vom Community-Besitzer gewährt werden.
	Hält alle ausstehenden Downloads an.
	Bricht alle ausstehenden Downloads ab.
<HPLN-Inhalte>	Listet die Inhaltstypen auf, die Sie herunterladen können. Folgende Inhaltstypen sind verfügbar: Aktionen/Funktionen, Korrelationsregeln und DFE-Dateien.
<Search>	Geben Sie eine Textzeichenfolge ein, und die angezeigte Inhaltsliste wird automatisch aktualisiert, um den eingegebenen Suchtext wiederzugeben. Der gesamte während der Erstellung des Assets in HPLN eingegebene Text wird in die Textsuche eingeschlossen.
<Aktion/Status des Assets>	Die folgenden Aktionen/Statusangaben können für ein Inhaltselement angezeigt werden: • **Herunterladen.** Klicken Sie hierauf, um eine Inhaltsdatei herunterzuladen. Nachdem der Inhalt erfolgreich heruntergeladen wurde, wird die Schaltfläche **Erneut herunterladen** angezeigt. Die Anzahl der aktuell heruntergeladenen Inhaltsdateien wird im unteren Bereich des Fensters **Von HP Live Network herunterladen** angezeigt. Wenn Sie mehrere Inhaltsdateien gleichzeitig herunterladen, wird die erste heruntergeladen und den anderen Inhaltsdateien wird der Status **Ausstehend** zugewiesen. Wenn der erste Download abgeschlossen ist, beginnt der Download der nächsten Inhaltsdatei. • **Erneut herunterladen.** Klicken Sie hierauf, um eine Inhaltsdatei erneut herunterzuladen. • **Angehalten.** Wird angezeigt, wenn der Download einer Inhaltsdatei durch Klicken auf die Schaltfläche **Pause** angehalten wurde. Klicken Sie auf **Herunterladen**, um den Download-Vorgang der Inhaltsdatei neu zu starten.

Element der Oberfläche	Beschreibung
<Fortschrittsbalken>	Zeigt den Fortschritt des Download-Vorgangs an. Wenn Sie mehrere Inhaltsdateien herunterladen, wird die Anzahl der Downloads in der Form "Element x von y wird von HPLN heruntergeladen" angezeigt.
Close	Schließt das Dialogfeld, beendet den Download-Vorgang der ausgewählten Inhalte jedoch im Hintergrund.

Skript-Workflow

Im Abschnitt **Skript-Workflow** finden Sie ausführliche Informationen zu dem Workflow, der zum Entwickeln eines Vuser-Skripts auszuführen ist.

Erstellen oder Öffnen eines Vuser-Skripts

In diesem Abschnitt wird das Erstellen eines neuen Vuser-Skripts bzw. das Öffnen eines vorhandenen Vuser-Skripts beschrieben.

Erstellen von über Vuser-Skripts – Übersicht

Das Erstellen eines Vuser-Skripts umfasst die nachfolgend aufgeführten Schritte. Dieses Thema enthält eine Übersicht über den ersten Schritt (Erstellen eines Vuser-Skripts).

Der erste Schritt beim Entwickeln eines Vuser-Skripts besteht in der Erstellung eines leeren Skripts. Weitere Informationen zum Erstellen eines leeren Vuser-Skripts finden Sie unter "Erstellen oder Öffnen von Vuser-Skripts" auf der nächsten Seite. Die Inhalte und die Struktur des leeren Vuser-Skripts hängen vom Protokoll des Skripts ab. Daher müssen Sie vor dem Erstellen eines leeren Vuser-Skripts wissen, welches Protokoll für das Skript verwendet werden soll. Sie können den Protocol Advisor verwenden, um ein Protokoll auszuwählen. Er durchsucht Ihre Anwendung nach Elementen von verschiedenen Protokollen und zeigt eine Liste der Protokolle an, die es findet. Weitere Informationen finden Sie unter Protocol Advisor. Nachdem Sie ein leeres Vuser-Skript erstellt haben, können Sie den nächsten Schritt innerhalb der Skripterstellung ausführen, das Aufzeichnen von Benutzeraktionen innerhalb des Skripts. Weitere Informationen finden Sie unter "Aufzeichnen eines Vuser-Skripts" auf Seite 162.

Beim Erstellen eines Vuser-Skripts legt VuGen mehrere Konfigurationsdateien, Datendateien und Quellcodedateien an, aus denen sich das Vuser-Skript zusammensetzt. Diese Dateien enthalten Vuser-Laufzeit- und Setup-Informationen. Weitere Informationen zu den Dateien, aus denen ein Vuser-Skript besteht, finden Sie unter "Skriptverzeichnisdateien" auf Seite 157.

VuGen bietet Ihnen folgende Möglichkeiten:

- Erstellen oder Öffnen eines Skripts von einer Vorlage. Weitere Informationen über die Aufgabe finden Sie unter "Erstellen oder Öffnen von Vuser-Skriptvorlagen" auf Seite 161.
- Öffnen eines .zip-Skripts oder Arbeiten mit einem .zip-Skript. Sie können ein Skript im .zip-Format dekomprimieren oder mit einem Skript im .zip-Format arbeiten. Weitere Informationen über die Aufgabe finden Sie unter "Arbeiten mit .zip-Dateien" auf Seite 159.

- Öffnen eines Skripts aus Application Lifecycle Management. Weitere Informationen finden Sie unter "Arbeiten mit Application Lifecycle Management" auf der nächsten Seite.
- Mit *HP Application Lifecycle Management* (ALM) können Sie Vuser-Skripts, -Szenarios und Analyseergebnisse speichern und abrufen. Sie können Skripts in einem ALM-Projekt speichern und die Skripts in eindeutigen Gruppen organisieren. Weitere Informationen finden Sie unter "Verwalten der Skripts in ALM - Übersicht" auf der nächsten Seite.

Erstellen oder Öffnen von Vuser-Skripts

In dieser Aufgabe wird das Erstellen eines neuen Vuser-Skripts bzw. das Öffnen eines vorhandenen Vuser-Skripts beschrieben.

Erstellen eines neuen Vuser-Skripts

1. Öffnen Sie VuGen, und wählen Sie **Datei > Neues Skript und neue Lösung** aus.
2. Wählen Sie im Dialogfeld **Neues Skript erstellen** in der Liste **Kategorie** den Eintrag **Einzelnes Protokoll** oder **Mehrere Protokolle** aus.
3. Wählen Sie in der Liste **Protokolle** ein Protokoll aus.
4. Geben Sie im Feld **Skriptname** einen Namen für das Skript ein.

   Hinweis: Verwenden Sie als Skriptnamen nicht *init*, *run* oder *end*, da diese Namen von VuGen intern verwendet werden.

5. Klicken Sie auf **Erstellen**, um das Vuser-Skript zu erstellen.

Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Neues Skript erstellen"" auf Seite 157.

Wenn Sie ein neues Vuser-Skript erstellt haben, können Sie Benutzeraktivitäten im Skript aufzeichnen. Weitere Informationen finden Sie unter "Aufzeichnen eines Vuser-Skripts" auf Seite 162.

Erstellen oder Öffnen eines Skripts von einer Vorlage

Weitere Informationen über die Aufgabe finden Sie unter "Erstellen oder Öffnen von Vuser-Skriptvorlagen" auf Seite 161.

Öffnen eines vorhandenen Skripts

Zum Öffnen eines vorhandenen Skripts, das auf Ihrem lokalen Computer oder einem Netzwerklaufwerk gespeichert ist, wählen Sie **Datei > Öffnen > Skript/Lösung** aus.

Öffnen eines .zip-Skripts oder Arbeiten mit einem .zip-Skript

Sie können ein Skript im .zip-Format dekomprimieren oder mit einem Skript im .zip-Format arbeiten. Weitere Informationen über die Aufgabe finden Sie unter "Arbeiten mit .zip-Dateien" auf Seite 159.

Öffnen eines Skripts aus Application Lifecycle Management

Sie können Skripts in HP ALM speichern und sie in VuGen ändern. Weitere Informationen finden Sie unter "Arbeiten mit Application Lifecycle Management" unten.

Arbeiten mit Application Lifecycle Management

Im Abschnitt **Arbeiten mit Application Lifecycle Management** wird beschrieben, wie Sie Ihre Vuser-Skripts durch die Integration mit Application Life Cycle Management verwalten können.

Verwalten der Skripts in ALM - Übersicht

VuGen kann zusammen mit *HP Application Lifecycle Management* (ALM) eingesetzt werden. Mit ALM stehen Ihnen leistungsstarke Funktionen zum Speichern und Abrufen von Vuser-Skripts, -Szenarios und Analyseergebnissen zur Verfügung. Sie können Skripts in einem ALM-Projekt speichern und die Skripts in eindeutigen Gruppen organisieren.

Damit VuGen auf ein ALM-Projekt zugreifen kann, müssen Sie eine Verbindung zwischen VuGen und dem Webserver herstellen, auf dem sich das ALM-Projekt befindet. Sie können eine Verbindung zu einem lokalen oder einem Remotewebserver herstellen.

Weitere Informationen über das Arbeiten mit ALM finden Sie im *HP Application Lifecycle Management-Benutzerhandbuch*.

Versionskontrolle in ALM - Übersicht

VuGen unterstützt Versionskontrollfunktionen in den in ALM-Projekten gespeicherten Vuser-Skripts, die der Versionskontrolle unterliegen.

Die Funktionen der Versionskontrolle ändern den Prozess des Öffnens und Speichern von Skripts. Skripts mit Versionskontrolle sind entweder ein- oder ausgecheckt. Wenn Sie mit einem ausgecheckten Skript arbeiten, werden die von Ihnen vorgenommenen Änderungen erst auf dem ALM-Server gespeichert, wenn Sie das Skript einchecken. Speichern Sie das Skript von VuGen aus, wird eine temporäre Datei auf Ihrem Computer gespeichert, wodurch Ihre Änderungen für den Fall eines Computerabsturzes gesichert sind.

Wenn Sie mit einem eingecheckten Skript arbeiten, ist das Skript schreibgeschützt und Sie können die Änderungen erst speichern, wenn Sie das Skript auschecken.

Wird ein bestimmtes Skript erstmals in ALM gespeichert und verwendet das Projekt die Versionskontrolle, wird das Skript automatisch im ausgecheckten Zustand gestartet.

Verwenden von Skripts in ALM-Projekten

Die folgenden Schritte beschreiben den Workflow für die Verwendung von Vuser-Skripts, die in einem ALM-Projekt gespeichert sind.

Hinweis: Informationen zum Arbeiten mit Skripts in ALM-Projekten mit Versionskontrolle finden Sie unter "Verwenden von Skripts mit Versionskontrolle in ALM-Projekten" auf der nächsten Seite.

1. **Herstellen der Verbindung zu ALM**

   Öffnen Sie eine Verbindung zum ALM-Server und dem Projekt, das das Skript enthält. Weitere Informationen über die Aufgabe finden Sie unter "Herstellen der Verbindung zu ALM" unten.

2. **Öffnen eines Skripts**

   Wählen Sie **Datei > Öffnen > Skript/Lösung**. Wählen Sie im Dialogfeld **VuGen-Skript oder -Lösung öffnen** das gewünschte Skript aus, und klicken Sie auf **Öffnen**.

3. **Speichern eines Skripts**

   Wählen Sie **Datei > Skript speichern** aus. Wenn sich das Skript in einem Projekt mit Versionskontrolle befindet und nicht ausgecheckt wurde, wird das Skript als temporäre Datei auf Ihrem lokalen Computer gespeichert.

Herstellen der Verbindung zu ALM

Um Skripts in ALM zu speichern und aus ALM zu laden, müssen Sie eine Verbindung zu einem ALM-Projekt herstellen. Sie können zu jedem beliebigen Zeitpunkt während des Testablaufs eine Verbindung zu einem ALM-Projekt herstellen und diese trennen.

Sie können sich aus VuGen mit einer Version von HP ALM verbinden und über Ihren Browser mit einer anderen. Weitere Informationen finden Sie im Abschnitt **Wichtige Informationen** unter "Dialogfeld "HP ALM-Verbindung" [VuGen]" auf Seite 154.

Herstellen einer Verbindung zu einem Projekt in ALM

1. Legen Sie den für den ALM-Benutzer erforderlichen Authentifizierungstyp fest: Benutzername/Kennwort oder CAC (Common Access Card). Aktivieren Sie für den CAC-Modus in den Optionen unter **Allgemein** in VuGen die CAC-Authentifizierung. Weitere Informationen finden Sie unter "Allgemeine Optionen" auf Seite 75.

2. Wählen Sie **ALM** > **ALM-Verbindung**. Das Dialogfeld **HP ALM-Verbindung** wird geöffnet.

3. Geben Sie in Abschnitt **Schritt 1: Mit Server verbinden** einen Benutzernamen und ein Kennwort ein (nicht relevant für die CAC-Authentifizierung), und klicken Sie auf **Verbinden**. VuGen stellt die Verbindung zum ALM-Server her.

   Um die Verbindung zu ALM zu trennen, klicken Sie auf **Trennen**.

4. Geben Sie in Abschnitt **Schritt 2: An Projekt anmelden** die Domäne und Projektdetails ein, und klicken Sie dann auf **Anmelden**. VuGen meldet sich beim angegebenen Projekt an.

   Um sich vom Projekt abzumelden, klicken Sie auf **Abmelden**.

5. Klicken Sie auf **Schließen**, um das Dialogfeld **HP ALM-Verbindung** zu schließen.

Verwenden von Skripts mit Versionskontrolle in ALM-Projekten

Die folgenden Schritte beschreiben den Workflow für die Verwendung von Skripts, die in ALM-Projekten mit Versionskontrolle gespeichert sind.

Hinweis: Dieses Verfahren ist nur für Skripts in ALM-Projekten relevant, die Versionskontrolle unterstützen und für die die Performance Center-Erweiterung installiert ist. Wenn diese beiden Bedingungen nicht erfüllt sind, finden Sie weitere Informationen unter "Verwenden von Skripts in ALM-Projekten" auf Seite 150.

1. ### Herstellen der Verbindung zu ALM

   Öffnen Sie eine Verbindung zum ALM-Server und dem Projekt, das das Skript enthält. Weitere Informationen über die Aufgabe finden Sie unter "Herstellen der Verbindung zu ALM" auf der vorherigen Seite.

2. ### Öffnen eines Skripts

   Wählen Sie **Datei > Öffnen > Skript/Lösung**. Wählen Sie im Dialogfeld **VuGen-Skript oder -Lösung öffnen** das gewünschte Skript aus, und klicken Sie auf **Öffnen**.

3. ### Ein- bzw. Auschecken eines Skripts

   Wenn das ALM-Projekt über Versionskontrolle verfügt, wird jedes Skript entweder als eingecheckt oder ausgecheckt definiert. Weitere Informationen finden Sie unter "Versionskontrolle in ALM - Übersicht" auf Seite 150. Um Skripts ein- und auszuchecken, wählen Sie **ALM > Einchecken/Auschecken**.

   Hinweis: Wenn das ALM-Projekt der Versionskontrolle unterliegt, ist die Datei gesperrt, nachdem sie ausgecheckt wurde.

   Wenn das ALM-Projekt nicht der Versionskontrolle unterliegt, ist die Datei nicht gesperrt, nachdem sie vom Projekt ausgecheckt wurde.

4. ### Abbrechen des Auscheckens (optional)

   Wenn Sie ein Skript ausgecheckt haben und die Änderungen nicht speichern möchten, können Sie das Skript ohne zu speichern auf den eingecheckten Status zurücksetzen, indem Sie **ALM > Auschecken rückgängig machen** auswählen.

5. ### Speichern eines Skripts

   Wählen Sie **Datei > Skript speichern** aus. Wenn sich das Skript in einem Projekt mit

Versionskontrolle befindet und nicht ausgecheckt wurde, wird das Skript als temporäre Datei auf Ihrem lokalen Computer gespeichert.

Speichern von VuGen-Vuser-Skripts in ALM-Projekten

Mithilfe der folgenden Schritte können Sie ein Vuser-Skript in einem ALM-Projekt speichern.

1. **Öffnen oder Erstellen des Vuser-Skripts**

   Erstellen oder öffnen Sie das gewünschte Skript in VuGen.

2. **Herstellen der Verbindung zu ALM**

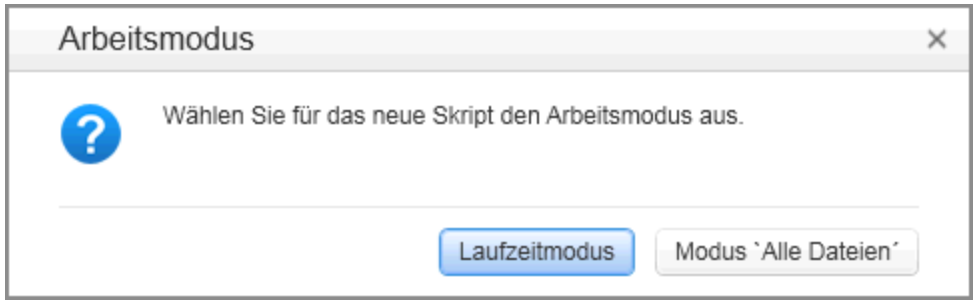
   Öffnen Sie eine Verbindung zum ALM-Server und dem Projekt, in dem Sie das Skript speichern möchten. Weitere Informationen über die Aufgabe finden Sie unter "Herstellen der Verbindung zu ALM" auf Seite 151.

3. **Speichern eines Skripts in ALM**

   a. Wählen Sie **Datei > Skript speichern unter** aus. Das Dialogfeld **Skript speichern unter** wird geöffnet.

   b. Klicken Sie auf **ALM-Testplan**, und geben Sie den Namen und Speicherort für das Skript ein.

   c. Klicken Sie auf **Speichern**. Nach kurzer Zeit wird das Dialogfeld **Arbeitsmodus** geöffnet.



   d. Wählen Sie eine der folgenden Optionen:

   Laufzeitmodus. Kopiert nur die für die Wiedergabe des Skripts erforderlichen Dateien. Mit dieser Option werden keine Dateien mit Aufzeichnungs-Snapshots oder andere nicht benötigte Dateien kopiert. Dies führt zu kürzeren Übertragungszeiten.

   Modus "Alle Dateien". Kopiert alle diesem Skript zugeordneten Dateien. Dies führt zu längeren Übertragungszeiten.

Vergleichen von Vorgängerversionen eines Skripts

Wenn Ihr Vuser-Skript in einem ALM-Projekt mit Versionskontrolle gespeichert wird, können Sie Vorgängerversionen des Skripts vergleichen. In den folgenden Schritten wird die entsprechende Vorgehensweise beschrieben.

1. **Herstellen der Verbindung zu ALM**

   Stellen Sie eine Verbindung zu dem ALM-Server und Projekt her, in dem das Skript enthalten ist, das Sie anzeigen oder ändern möchten. Weitere Informationen über die Aufgabe finden Sie unter "Herstellen der Verbindung zu ALM" auf Seite 151.

2. **Öffnen eines Skripts**

   Wählen Sie **Datei > Öffnen > Skript/Lösung**. Wählen Sie im Dialogfeld **VuGen-Skript oder -Lösung öffnen** das gewünschte Skript aus, und klicken Sie auf **Öffnen**.

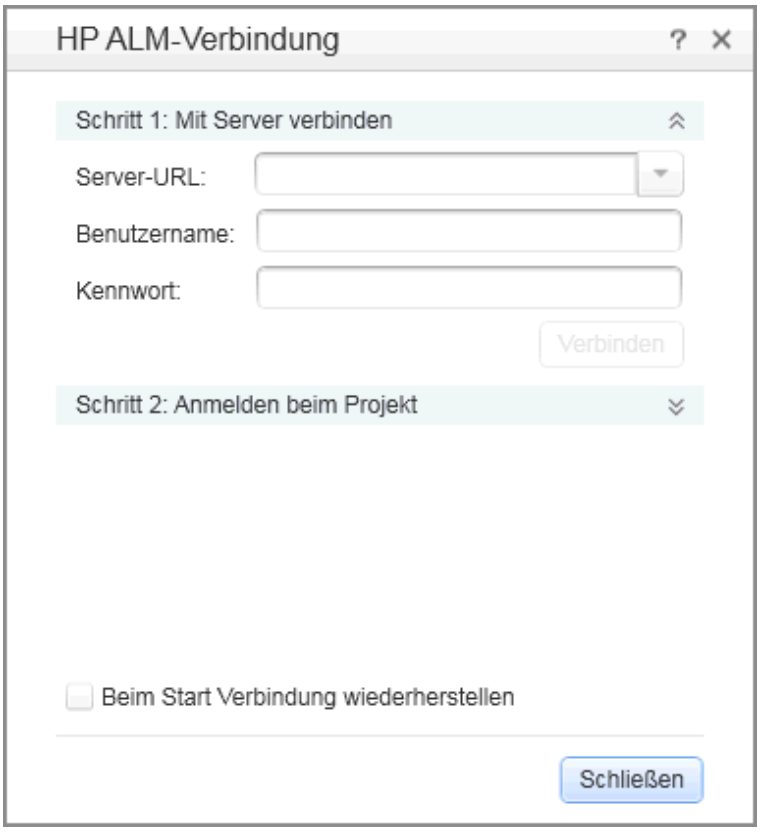
3. **Vergleichen von Vorgängerversionen des Skripts**

   a. Wählen Sie **ALM > Versionshistorie**. Das Dialogfeld **Versionshistorie** wird geöffnet.

   b. Wählen Sie die beiden Vorgängerversionen des Skripts, und klicken Sie dann auf **Versionen vergleichen**. WDiff wird geöffnet und zeigt die zwei Versionen des Skripts an.

Dialogfeld "HP ALM-Verbindung" [VuGen]

Über dieses Dialogfeld können Sie von VuGen aus eine Verbindung zu einem HP ALM-Projekt herstellen.



Zugriff	**ALM > ALM-Verbindung**
Wichtige Informationen	• Sie können sich über VuGen mit einer Version von HP ALM verbinden und über Ihren Browser mit einer anderen Version. • Allerdings können Sie sich nur dann mit unterschiedlichen Versionen von HP ALM verbinden, wenn es sich bei mindestens einer der Versionen um HP ALM 11.00 oder höher handelt. **Hinweis:** Bevor Sie eine Verbindung zu in ALM gespeicherten Ergebnissen über dieses Dialogfeld herstellen, wird empfohlen, dass Sie zuerst eine Verbindung zum HP ALM-Server über Ihren Browser herstellen. Hierdurch werden die ALM-Clientdateien automatisch auf Ihren Computer geladen. • Sie können über das Webgate Customization-Tool (webgatecustomization.exe) erweiterte Einstellungen, beispielsweise eine Proxy-Einstellung, konfigurieren und sich dann über das Dialogfeld **HP ALM-Verbindung** bei ALM anmelden. Sie finden das Webgate Customization-Tool auf Ihrem ALM-Server unter folgender Adresse: `http://<ALM-Server>/qcbin/Apps/`.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Schritt 1: Mit dem Server verbinden	• **Server-URL.** Der URL des Servers, auf dem ALM installiert ist. • **Benutzername.** Ihr Benutzername für das ALM-Projekt (nicht relevant für die CAC-Authentifizierung). • **Kennwort.** Ihr Kennwort für das ALM-Projekt (nicht relevant für die CAC-Authentifizierung). • Verbinden Stellt eine Verbindung zu dem im Feld **Server-URL** angegebenen Server her. • Trennen . Trennt die Verbindung zum aktuellen ALM-Server.
Schritt 2: An Projekt anmelden	• **Domäne.** Die Domäne des ALM-Projekts. Es werden nur Domänen angezeigt, die Projekte enthalten, mit denen Sie basierend auf Ihren Berechtigungen eine Verbindung herstellen dürfen. • **Projekt.** Geben Sie den ALM-Projektnamen ein oder wählen Sie ein Projekt aus der Liste aus. Die Liste enthält nur die Projekte, für die Sie eine Berechtigung zum Herstellen einer Verbindung besitzen. • Anmelden . Führt die Anmeldung am ALM-Projekt durch. • Abmelden . Führt die Abmeldung vom aktuellen ALM-Projekt durch.
Beim Start Verbindung wiederherstellen	Stellt beim nächsten Start von VuGen automatisch mit denselben Anmeldeinformationen wieder eine Verbindung mit dem Server her.

Mehrfachprotokollskripts

Wenn Sie ein Einzelprotokoll aufzeichnen, zeichnet VuGen nur das angegebene Protokoll auf. Wenn Sie im Mehrfachprotokollmodus aufzeichnen, zeichnet VuGen die Aktionen in mehreren Protokollen auf. Mehrfachprotokollskripts werden für die folgenden Protokolle unterstützt: COM, FTP, IMAP, Oracle NCA, POP3, RealPlayer, Windows Sockets (raw), SMTP und Web.

Eine andere Variante zwischen Vuser-Typen ist die Unterstützung von mehreren Aktionen. Die meisten Protokolle unterstützen mehr als einen Aktionsabschnitt. Zur Zeit unterstützen die folgenden Protokolle mehrere Aktionen: Oracle NCA, Web, RTE, General (C Vusers), WAP, und VoiceXML.

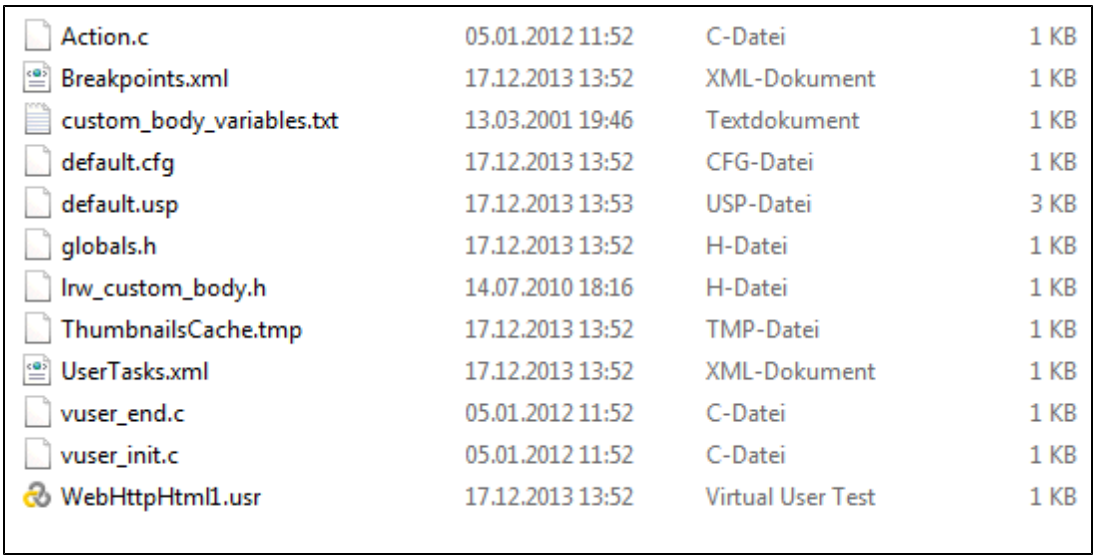
Für die meisten Vuser-Typen erstellen Sie bei jedem Aufzeichnen ein neues Vuser-Skript. Eine Aufzeichnung in ein vorhandenes Skript ist nicht möglich. Beim Aufzeichnen eines Java-, Web-, WAP-, Oracle NCA- oder RTE Vuser-Skripts können Sie jedoch auch innerhalb eines vorhandenen Skripts aufzeichnen.

Da VuGen eine Vielzahl von Protokollen unterstützt, gelten einige der folgenden Aufzeichnungsschritte nur für spezifische Protokolle.

Weitere Informationen für Java-Nutzer finden Sie unter "Java-Protokolle" auf Seite 710

Skriptverzeichnisdateien

Beim Erstellen eines Vuser-Skripts legt VuGen mehrere Konfigurationsdateien, Datendateien und Quellcodedateien an, aus denen sich das Vuser-Skript zusammensetzt. Diese Dateien enthalten Vuser-Laufzeit- und Setup-Informationen. VuGen speichert diese Dateien zusammen mit dem Skript im Skriptordner. Zum Zugreifen auf die Dateien im Skriptordner klicken Sie im **Lösungs-Explorer** mit der rechten Maustaste auf den Skriptnamen, und wählen Sie **Skriptordner öffnen** aus.



Informationen zu den im Skriptordner enthaltenen Dateien finden Sie unter "Bei der Aufzeichnung erstellte Dateien" auf Seite 278.

Dialogfeld "Neues Skript erstellen"

In diesem Dialogfeld können Sie ein neues Vuser-Skript erstellen.

Zugriff	Verwenden Sie eine der folgenden Methoden: • **VuGen > Datei > Neues Skript und neue Lösung** • **VuGen > Datei > Hinzufügen > Neues Skript** • Klicken Sie auf die Schaltfläche in der VuGen-Symbolleiste.

Forts.

Relevante Aufgaben	• "Erstellen oder Öffnen eines Vuser-Skripts" auf Seite 148 • "Aufzeichnen eines Vuser-Skripts" auf Seite 162

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Kategorie	Die Kategorie bezeichnet den Typ der Skripts, die Sie erstellen können: **Mehrere Protokolle.** Erstellt ein Skript unter Verwendung mehrerer Protokolle. **Einzelnes Protokoll.** Erstellt ein Skript unter Verwendung des ausgewählten Protokolls. **Einzelnes Protokoll** ist die Standardkategorie.
Protokoll	In der Protokollliste werden die zu der ausgewählten Kategorie gehörenden Protokolle angezeigt. **Kategorie** / **Inhalt der Protokollliste** **Mehrere Protokolle** — Eine Liste von Protokollen, denen je ein Kontrollkästchen vorangestellt ist. Aktivieren Sie das Kontrollkästchen eines Protokolls, um es in das Vuser-Skript einzubeziehen. **Einzelnes Protokoll** — Eine Liste von Protokollen. Sie können ein Protokoll auswählen. **Mobile** — Protokolle, die mobile Anwendungen aufzeichnen. **Häufig verwendet** — Die beliebtesten Protokolle der LoadRunner-Community. **Letzte** — Die zuletzt verwendeten Protokolle.
Filter	Sie können die Protokollliste filtern, indem Sie den entsprechenden Text eingeben. Geben Sie z. B. "Java" im Filterfeld ein, werden nur die Protokolle in die Liste aufgenommen, die das Wort "Java" enthalten.

Forts.

Skriptname	Geben Sie hier den Namen Ihres Skripts ein. Wenn Sie ein Skript mit einem einzelnen Protokoll erstellen, wird als Standardname <Protokollname>x verwendet, wobei x die numerische Folge des erstellten Skripts darstellt. Der Name des dritten für das Web-HTTP/HTML-Protokoll erstellten Skripts würde also WebHttpHtml3 lauten. Wenn Sie ein Skript mit mehreren Protokollen erstellen, wird als Standardname <Protokollname_multi>x verwendet, wobei Protokollname das erste Protokoll ist, das Sie aus der Liste ausgewählt haben. x steht für die numerische Folge des erstellten Skripts.
Speicherort	Geben Sie hier den Speicherort Ihres Skripts ein. Mit der Schaltfläche **Durchsuchen** können Sie zu dem gewünschten Speicherort im Dateisystem navigieren. Unter **Extras > Optionen > Allgemein > Skripts und Lösungen** können Sie einen Standardspeicherort angeben.
Lösungsname	Diese Option wird nur dann angezeigt, wenn im Lösungs-Explorer keine Lösung geöffnet ist. Sie können einen Namen für die Lösung angeben. Wenn Sie nichts eingeben, wird der Standardname "Ohne Titel" verwendet.
Ordner für Lösung erstellen	Sie können hier einen Ordner für Ihre Lösung erstellen.
Lösungsziel	Zeigt den Dateipfad der Lösung an.
Protocol Advisor	Ermöglicht den Zugriff auf das Dialogfeld **Protocol Advisor**. Weitere Informationen finden Sie unter "Dialogfeld "Protocol Advisor"" auf Seite 135.
	Zeigt die Protokolle in der Listenansicht an.
	Zeigt die Protokolle in der Symbolansicht an.

Arbeiten mit .zip-Dateien

VuGen ermöglicht Ihnen das Arbeiten mit .zip-Dateien unter Verwendung verschiedener Methoden. Vorteile der Arbeit mit .zip-Dateien sind, dass Sie Platz auf der Festplatte sparen und Ihre Skripts in einer einfacher kopiert werden können. Anstatt zahlreiche Dateien von Computer zu Computer zu übertragen, brauchen Sie nur eine ZIP-Datei zu kopieren.

Importieren aus einer ZIP-Datei

Zum Öffnen eines in einer .zip-Datei gespeicherten Skripts wählen Sie **Datei > ZIP-Dateien verwalten > Aus ZIP-Datei Importieren** aus. Nach Auswahl einer ZIP-Datei werden Sie von VuGen aufgefordert, einen Ort zum Speichern der dekomprimierten Dateien anzugeben.

Exportieren in eine ZIP-Datei

Zum Speichern des gesamten Skriptordners als ZIP-Datei wählen Sie **Datei > ZIP-Dateien verwalten > In Zip-Datei exportieren** aus.

Sie können angeben, ob alle Dateien oder nur die Laufzeitdateien gespeichert werden sollen.

ZIP-Datei und E-Mail

Zum Erstellen einer ZIP-Datei und Senden als E-Mail-Anhang wählen Sie **Datei > ZIP-Dateien verwalten > ZIP-Datei und E-Mail**. Wenn Sie im Dialogfeld **ZIP-Datei und E-Mail** auf **OK** klicken, komprimiert VuGen die Datei gemäß Ihren Einstellungen und öffnet ein Formular zum Erstellen einer E-Mail mit der ZIP-Datei als Anhang.

Bearbeiten eines Skripts in einer ZIP-Datei

Um ein in einer ZIP-Datei gespeichertes Skript direkt zu bearbeiten, ohne die Skriptdateien zu extrahieren oder zu speichern, wählen Sie **Datei > ZIP-Dateien verwalten > Skript in ZIP-Datei bearbeiten** aus. Wenn Sie das Skript ändern und speichern, werden die Änderungen direkt in der .zip-Datei gespeichert.

Vuser-Skriptvorlagen

Die benutzerdefinierte Vorlage ermöglicht es Ihnen, ein Skript mit einer spezifischen Konfiguration als Vorlage zu speichern. Sie können diese Vorlage dann als Basis zum Erstellen von zukünftigen Skripts verwenden.

Die Vorlage unterstützt die folgenden Dateien und Daten:

- Laufzeiteinstellungen
- Parameter
- Zusätzliche Dateien
- Aktionen
- Snapshots

Aufzeichnungs- und allgemeine Optionen werden nicht unterstützt, da es sich dabei um allgemeine Einstellungen handelt, die nicht für ein spezifisches Skript relevant sind.

Hinweise und Einschränkungen

- Nachdem Sie ein Skript für ein spezifisches Protokoll konfiguriert und das Skript anschließend als Vorlage gespeichert haben, funktionieren weitere Skripts auf der Basis dieser Vorlage nur mit diesem Protokoll.
- Nachdem Sie Ihre Vorlage erstellt haben, können Sie sie nicht direkt in VuGen bearbeiten. Zum Durchführen von Änderungen öffnen Sie die Vorlage und speichern sie erneut unter einem anderen Namen oder überschreiben Sie die vorhandene Vorlage.

- Wenn Sie ein ursprüngliches Skript aus einer Vorlage neu generieren, verlieren Sie Ihre gesamten manuell vorgenommenen Änderungen.

Erstellen oder Öffnen von Vuser-Skriptvorlagen

In dieser Aufgabe wird beschrieben, wie Sie eine Vuser-Skriptvorlage erstellen, eine Vuser-Skriptvorlage aus einer Vorlage erstellen und diese umbenennen.

Erstellen einer Vuser-Skriptvorlage

1. Öffnen Sie ein Skript in VuGen.
2. Wählen Sie **Datei > Benutzerdefinierte Vorlagen > In Vorlage exportieren** aus.
3. Geben Sie einen Namen und einen Speicherort für die Vorlage ein.
4. Klicken Sie zum Erstellen der Vorlage auf **OK**.

Erstellen eines Vuser-Skripts aus einer Vorlage

Wählen Sie **Datei > Neues Skript und neue Lösung > VuGen > Benutzervorlagen** und die Vorlagendatei aus.

Umbenennen einer Vuser-Skriptvorlage

1. Wählen Sie **Datei > Benutzerdefinierte Vorlagen > Im Explorer verwalten** aus.
2. Im Explorer-Dialogfeld:
   a. Benennen Sie die Inhaltsdatei (.zip) um.
   b. Benennen Sie die Beschreibungsdatei (.xpt) um.
3. Ändern Sie mit einem Text-Editor die folgenden Tags in der .xpt-Datei:
   a. **Namens-Tag:** `<Name>NewName </Name>`
   b. **Datei-Tag für src-Eigenschaft:** `<File name="template_temp.zip" src="NewName.zip" binary="True" />`

Hinweise und Einschränkungen

- Nachdem Sie ein Skript für ein spezifisches Protokoll konfiguriert und das Skript anschließend als Vorlage gespeichert haben, funktionieren weitere Skripts auf der Basis dieser Vorlage nur mit diesem Protokoll.
- Nachdem Sie Ihre Vorlage erstellt haben, können Sie sie nicht direkt in VuGen bearbeiten. Zum Durchführen von Änderungen öffnen Sie die Vorlage und speichern sie erneut unter einem anderen Namen oder überschreiben Sie die vorhandene Vorlage.

Aufzeichnen eines Vuser-Skripts

Im Abschnitt **Aufzeichnen eines Vuser-Skripts** werden Skriptabschnitte, das Aufzeichnen von Skripts, das Arbeiten mit Vorlagen und weitere Aufzeichnungstools beschrieben.

Aufzeichnen – Übersicht

Das Erstellen eines Vuser-Skripts umfasst die nachfolgend aufgeführten Schritte. Dieses Thema enthält eine Übersicht über den zweiten Schritt (Aufzeichnen eines Vuser-Skripts).

Wenn Sie ein leeres Vuser-Skript erstellt haben, können Sie in dem Skript mithilfe von VuGen typische Benutzeraktionen aufzeichnen. Während der Skriptaufzeichnung haben Sie über die unverankerte Aufzeichnungssymbolleiste von VuGen Zugriff auf die wichtigsten Aufzeichnungsfunktionen, z. B. zum Anhalten und Beenden der Aufzeichnung sowie zum Einfügen von Transaktionen und Rendezvous-Punkten. Weitere Informationen zum Aufzeichnen eines Vuser-Skripts finden Sie unter "Aufzeichnen eines Vuser-Skripts" unten.

Jedes Vuser-Skript enthält mindestens drei Abschnitte: "vuser_init", mindestens einen Abschnitt "action" und "vuser_end". Wenn Sie mehrere Iterationen eines Vuser-Skripts ausführen, werden nur die *Aktions*abschnitte des Skripts wiederholt. Die Abschnitte *vuser_init* und *vuser_end* werden nicht wiederholt. Vor der Aufzeichnung können Sie den Abschnitt des Skripts auswählen, in den VuGen die aufgezeichneten Funktionen einfügen soll. Weitere Informationen zu den Skriptabschnitten finden Sie unter "Abschnitte des Vuser-Skripts" auf Seite 260.

Stellen Sie vor dem Starten der Aufzeichnung sicher, dass die Aufzeichnungsoptionen für das Skript richtig festgelegt wurden. Weitere Informationen zu den Aufzeichnungsoptionen finden Sie unter "Aufzeichnungsoptionen" auf Seite 164.

Wenn Sie alle Benutzeraktionen aufgezeichnet haben, generiert VuGen das Vuser-Skript und führt verschiedene Vorgänge aus, die nach der Aufzeichnung ausgeführt werden. Sie können das Skript wiedergeben, um sicherzustellen, dass es wie erwartet funktioniert. Weitere Informationen finden Sie unter "Wiedergeben eines Vuser-Skripts" auf Seite 328.

VuGen ermöglicht die Aufzeichnung von Skripts mithilfe eines LoadRunner-Proxy, um Situationen zu lösen, in denen Sie VuGen nicht auf dem Clientcomputer installieren können. Die Proxy-Aufzeichnung kann bei bestimmten Linux-Computern, Mac OS-Computern und mobilen Geräten erforderlich sein. Weitere Informationen finden Sie unter "Aufzeichnung über einen Proxy - Übersicht" auf Seite 264.

Nach der erfolgreichen Aufzeichnung eines Vuser-Skripts können Sie das Skript wiedergeben. Weitere Informationen finden Sie unter "Übersicht über die Wiedergabe" auf Seite 329.

Aufzeichnen eines Vuser-Skripts

In dieser Aufgabe wird beschrieben, wie ein Vuser-Skript aufgezeichnet wird.

1. Erstellen Sie ein neues Skript oder öffnen Sie ein vorhandenes.

   Weitere Informationen finden Sie unter "Erstellen oder Öffnen eines Vuser-Skripts" auf Seite 148.

2. Ändern der Einstellungen für die Windows-Datenausführungsverhinderung (empfohlen)
   a. Wählen Sie **Start > Systemsteuerung > System**.

   b. Klicken Sie auf der Registerkarte **Erweitert** unter **Systemleistung** auf **Einstellungen**.

   c. Wählen Sie auf der Registerkarte **Datenausführungsverhinderung** die erste Option **Datenausführungsverhinderung nur für erforderliche Windows-Programme und -Dienste aktivieren**.

      Wenn Sie diese Option nicht ändern können, klicken Sie auf **Hinzufügen**. Suchen Sie das Client-Programm, zum Beispiel IEXPLORE.EXE.

   d. Wenn keine dieser beiden Optionen möglich ist, versuchen Sie die Datenausführungsverhinderung vollständig zu deaktivieren.

      i. Öffnen Sie eine Befehlszeile.

      ii. Führen Sie den folgenden Befehl aus: **bcdedit.exe /set {current} nx AlwaysOff**

      iii. Starten Sie den Computer neu.

      iv. Stellen Sie sicher, dass die Einstellungen wirksam wurden, indem Sie folgenden Befehl an der Befehlszeile ausführen: **BCDEdit /enum**

      v. Stellen Sie sicher, dass in der letzten Zeile **nx AlwaysOff** angezeigt wird.

3. Schließen aller Instanzen des Browsers - für die Web-HTTP/HTML-Aufzeichnung von Chrome oder Firefox

   Dies gilt nur für die Aufzeichnung in Chrome oder Firefox. Stellen Sie zudem im Task-Manager sicher, dass keine Chrome- oder Firefox-Prozesse ausgeführt werden.

4. ## Konfigurieren der Aufzeichnungsoptionen - optional

   Zu den Aufnahmeoptionen gehören Optionen, die sich darauf auswirken, wie ein Vuser-Skript nach Aufzeichnung oder Neugenerierung des Skripts erzeugt wird. Informationen zum Konzept und zur Benutzeroberfläche finden Sie unter "Aufzeichnungsoptionen" auf der nächsten Seite.

   Tipp: Wenn der Geschäftsprozess, der aufgezeichnet werden soll, asynchrone "Push"-Kommunikation enthält, wählen Sie **Aufzeichnungsoptionen > HTTP > Erweitert (Knoten)** und aktivieren Sie die Option **Streamingmodus beim Aufzeichnen mit LoadRunner-Proxy verwenden**.

Wenn es Ihnen nicht gelingt, ein Skript fehlerfrei mit VuGen aufzuzeichnen, wählen Sie **Aufzeichnungsoptionen > HTTP > Erweitert (Knoten)** und aktivieren Sie die Option **LR-Proxy zum Aufzeichnen einer lokalen Anwendung verwenden**. Zeichnen Sie nun Ihren Geschäftsprozess auf.

5. **Starten einer Aufzeichnungssitzung**

   Klicken Sie zum Starten der Aufzeichnung auf die Schaltfläche **Aufzeichnen** in der VuGen-Symbolleiste, nehmen Sie die entsprechenden Einstellungen im Dialogfeld "Aufzeichnung starten" vor und klicken Sie auf **Aufzeichnung starten**. Die freischwebende VuGen-Symbolleiste wird eingeblendet und VuGen öffnet Ihre Anwendung und ist bereit, Ihre Aktionen aufzuzeichnen.

   - Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Aufzeichnung starten"" auf Seite 272.

   - Weitere Informationen zu den Skriptabschnitten, in die Sie aufzeichnen können, finden Sie unter "Abschnitte des Vuser-Skripts" auf Seite 260.

6. **Durchführen eines Geschäftsprozesses in Ihrer Anwendung**

   Führen Sie die gewünschten Geschäftsprozesse, die aufgezeichnet werden sollen, durch. Die freischwebende Symbolleiste gibt Ihnen die Möglichkeit, Transaktionen, Rendezvous-Punkte und Kommentare einzufügen. Darüber hinaus können Sie festlegen, in welchen Abschnitten des Skripts aufgezeichnet werden soll. Informationen zur Benutzeroberfläche finden Sie unter "Freischwebende Aufzeichnungssymbolleiste" auf Seite 276.

   Klicken Sie in der unverankerten Symbolleiste auf die Schaltfläche **Anhalten**, wenn Sie mit der Aufzeichnung fertig sind.

   Hinweis: Zum Beenden der Aufzeichnungssitzung klicken Sie in der freischwebenden Symbolleiste auf **Aufzeichnung abbrechen**. Wenn Sie eine Aufzeichnung abbrechen, entfernt VuGen sämtlichen Code, der während der Aufzeichnung in das Skript aufgenommen wurde, und stellt so den Zustand des Skript wieder her, den es vor der Aufzeichnung hatte. Informationen zum Aktivieren bzw. Deaktivieren der Schaltfläche **Aufzeichnung abbrechen** finden Sie unter "Skriptoptionen" auf Seite 83.

Aufzeichnungsoptionen

Im Abschnitt **Aufzeichnungsoptionen** werden die zahlreichen verschiedenen Optionen beschrieben, die sich während der Aufzeichnungs- und Generierungsphasen beim Erstellen eines Skripts auf Ihr Vuser-Skript auswirken.

Knoten "Citrix > Konfiguration"

Ermöglicht das Festlegen der Fenstereigenschaften und Verschlüsselungseinstellungen für den Citrix-Client während der Aufzeichnungssitzung.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Citrix > Konfiguration**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Verschlüsselungsstufe	Die Verschlüsselungsstufe für die ICA-Verbindung: **Standard**, **128-Bit nur für Anmeldung**, **40-Bit**, **56-Bit**, **128-Bit** oder **Serverstandard verwenden**, um den Standardwert des Computers zu verwenden.
Fenstergröße	Die Größe des Client-Fensters. **Standardwert:** 800 x 600.

Citrix > Codegenerierung

Ermöglicht die Konfiguration der Art und Weise, auf die VuGen Informationen während der Aufzeichnung erfasst.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Citrix > Codegenerierung**
Wichtige Informationen	• Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250. • Textsynchronisierungsschritte, die Sie während der Aufzeichnung manuell hinzufügen, sind von den oben genannten Einstellungen nicht betroffen, d. h., sie sind auch dann im Skript enthalten, wenn Sie die genannten Optionen deaktivieren. • Die Synchronisierungsoptionen können auch bei der erneuten Generierung eines Skripts angewendet werden. Wenn Sie beispielsweise bei der ursprünglichen Aufzeichnung eines Skripts die Option **Textsynchronisierungsaufrufe hinzufügen** deaktiviert haben, können Sie das Skript nach der Aufzeichnung erneut generieren und dabei die Textsynchronisierung aktivieren.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Eingaben des Citrix-Agenten bei Codegenerierung verwenden	Die Eingaben des Citrix-Agenten werden verwendet, um ein ausführlicheres Skript mit zusätzlichen Synchronisierungsfunktionen zu generieren. **Standardwert:** Aktiviert. • **Textsynchronisierungsaufrufe automatisch generieren.** Durch diese Option werden **Synchronisieren bei Text**-Schritte vor jedem Mausklick hinzugefügt. **Standardwert:** Deaktiviert.

Knoten "Citrix > Anmelden"

Ermöglicht das Festlegen der Verbindungs- und Anmeldeinformationen für die Aufzeichnungssitzung.

Zugriff	VuGen > Aufzeichnen > Aufzeichnungsoptionen > Citrix > Anmelden
Wichtige Informationen	• Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250. • Wenn Sie keine Anmeldeinformationen angeben, werden Sie zur Eingabe dieser Informationen aufgefordert, wenn der Client auf den angegebenen Server zugreift.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung

Verbindung	• **Netzwerkprotokoll**. Die bevorzugten Protokolle sind TCP/IP und TCP/IP+HTTP. Die meisten Citrix-Server unterstützen TCP/IP. Dies gilt jedoch nicht für Citrix-Clients, deren Versionsnummer mit 11.2 beginnt. Manche Server werden jedoch von den Administratoren so konfiguriert, dass nur TCP/IP mit speziellen HTTP-Kopfzeilen erlaubt ist. Falls ein Kommunikationsproblem auftritt, wählen Sie die Option **TCP/IP + HTTP** aus. • **Server.** Der Name des Citrix-Servers. Um einen neuen Server zur Liste hinzuzufügen, klicken Sie auf **Hinzufügen** und geben den Servernamen ein (und den Port für TCP/IP + HTTP). **Hinweis:** Mehrere Server können Sie nur verwenden, wenn Sie unter **Veröffentlichte Anwendung** eine Anwendung angeben. Wenn Sie ohne bestimmte Anwendung eine Verbindung zum Desktop herstellen, listen Sie nur einen Server auf. • **Veröffentlichte Anwendung.** Der Name der veröffentlichten Anwendung, der auf dem Citrix-Server erkannt wird. Das Dropdownmenü enthält eine Liste der verfügbaren Anwendungen. Wenn Sie keine veröffentlichte Anwendung angeben, verwendet VuGen den Serverdesktop. Wenn Sie eine veröffentlichte Anwendung hinzugefügt oder umbenannt haben, schließen Sie die Aufzeichnungsoptionen und öffnen Sie sie erneut, um die neue Liste anzuzeigen. Sie können den Namen der veröffentlichten Anwendung auch manuell eingeben, wenn Sie ihn kennen. (Dies bietet sich an, wenn die Dropdownliste nicht genau ist.) Um den Namen der veröffentlichten Anwendung auf dem Citrix-Client zu ändern, müssen Sie die Änderung auf dem Citrix-Servercomputer vornehmen. Wählen Sie **Manage Console > Application** aus und erstellen Sie eine neue Anwendung oder benennen Sie eine bestehende Anwendung um. **Hinweis**: Wenn Sie keine veröffentlichte Anwendung festlegen, funktioniert der Citrix-Lastausgleich nicht. Wenn Sie Lastausgleich beim Zugriff auf den Serverdesktop verwenden möchten, registrieren Sie den Desktop als veröffentlichte Anwendung auf dem Servercomputer und wählen Sie diesen Namen in der Dropdownliste **Veröffentlichte Anwendung** aus.
Verbindungsparameter definieren	Ermöglicht die manuelle Definition der Anmelde- und Verbindungsinformationen.

Anmeldeinformationen	Geben Sie unter **Benutzername**, **Kennwort** und **Domäne** die entsprechenden Werte des Citrix-Benutzers ein. Optional können Sie auch den Clientnamen angeben, durch den der MetaFrame-Server den Client identifiziert.
ICA-Datei für Verbindungsparameter verwenden	Geben Sie eine ICA-Datei mit den Protokollkonfigurationsinformationen an.

Knoten "Citrix > Recorder"

Ermöglicht es Ihnen festzulegen, wie Fensternamen generiert werden, wenn die Fenstertitel sich während der Aufzeichnung ändern. Außerdem können Sie angeben, ob Snapshots der Bildschirme zusammen mit den Skriptdateien gespeichert werden sollen und ob Textsynchronisierungsfunktionen generiert werden sollen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Citrix > Recorder**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Snapshots speichern	Speichert einen Snapshot des Citrix-Clientfensters für jeden relevanten Skriptschritt. Das Aktivieren dieser Option wird empfohlen, damit Sie die aufgezeichneten Aktionen besser nachvollziehen können. Durch das Speichern von Snapshots verbrauchen Sie jedoch mehr Speicherplatz und die Aufzeichnungssitzung wird langsamer.

Fenstername	In manchen Anwendungen ändert sich der Name des aktiven Fensters während der Aufzeichnung. Wenn Sie das Skript unverändert wiedergeben, verwendet der Vuser den ursprünglichen Fensternamen und die Wiedergabe schlägt möglicherweise fehl. Sie können eine Namenskonvention für Fenster angeben, durch die VuGen ein gemeinsames Präfix oder Suffix zum Identifizieren eines Fensters verwendet: • **Fenstername unverändert übernehmen.** Der Fenstername wird so übernommen, wie er im Fenstertitel angezeigt wird (Standardeinstellung). • **Gemeinsames Präfix bei neuen Fensternamen verwenden.** Die gemeinsame Zeichenfolge am Anfang der Fenstertitel wird als Fenstername verwendet. • **Gemeinsames Suffix bei neuen Fensternamen verwenden.** Die gemeinsame Zeichenfolge am Ende der Fenstertitel wird als Name verwendet. Alternativ können Sie die Fensternamen nach der Aufzeichnung im Skript ändern. Suchen Sie in der Skriptansicht den Fensternamen und ersetzen Sie den Anfang oder das Ende des Fensternamens durch den Platzhalter *. **Beispiel:** ctrx_sync_on_window ("My Application*", ACTIVATE, ...CTRX_ LAST);

Knoten "COM/DCOM > Filter"

Damit können Sie die aufzuzeichnenden COM/DCOM-Objekte definieren.

Zugriff	Führen Sie einen der folgenden Schritte durch: • **VuGen > Aufzeichnen > Aufzeichnungsoptionen > COM/DCOM > Filter** • **VuGen > Aufzeichnen > Aufzeichnungsoptionen > COM/DCOM > Filter**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung

DCOM Profil	**Wählen Sie einen der folgenden Filtertypen:** • **Standardfilter.** Der Filter, der standardmäßig zur Aufzeichnung eines COM-Vuser-Skripts verwendet wird. • **Neuer Filter.** Ein leerer Filter, der auf den Einstellungen der Standardumgebung basiert. Beachten Sie, dass Sie einen Namen für diesen Filter angegeben müssen, bevor Sie ihn zur Aufzeichnung verwenden können. Mit den Schaltflächen **Speichern unter** und **Löschen** können Sie auch die aktuellen Einstellungen speichern und einen Filter löschen.

Liste "DCOM-Listenereinstellungen"	Zeigt eine Struktur aus Typbibliotheken an. Durch Erweitern der einzelnen Knoten können Sie alle in einer Typbibliothek verfügbaren Klassen anzeigen. Wenn Sie die Klassenstruktur erweitern, können Sie alle Schnittstellen anzeigen, die von einer Klasse unterstützt werden. Deaktivieren Sie das Kontrollkästchen neben einem Bibliotheksnamen, um die betreffende Bibliothek auszuschließen. Hierdurch werden auch alle dort enthaltenen Klassen ausgeschlossen. Wenn Sie die Struktur erweitern, können Sie auch einzelne Klassen oder Schnittstellen durch Deaktivieren des zugehörigen Kontrollkästchens ausschließen. Eine Schnittstelle kann von verschiedenen Klassen unterschiedlich implementiert werden. Wenn Sie eine Schnittstelle ausschließen, die von anderen, nicht ausgeschlossenen Klassen implementiert wird, wird ein Dialogfeld geöffnet. Sie werden gefragt, ob die Schnittstelle auch in allen anderen Klassen, die sie implementieren, ausgeschlossen werden soll. Beachten Sie, dass das Deaktivieren des Kontrollkästchens neben einer Schnittstelle funktionell äquivalent zum Auswählen dieser Schnittstelle im Dialogfeld **Ausgeschlossene Schnittstellen** ist. • **Umgebungen.** Die Umgebungen, die aufgezeichnet werden sollen: **ADO Objects**, **RDS Objects** und **Remote Objects**. Deaktivieren Sie die Objekte, die nicht aufgezeichnet werden sollen. • **Typbibliotheken.** Eine Typbibliotheksdatei (.**tlb** oder .**dll**), die das aufzuzeichnende COM-Objekt enthält. Alle COM-Objekte werden durch eine Typbibliothek dargestellt. Sie können die Typbibliothek aus der Registry, auf einem Microsoft Transaction Server oder aus dem Dateisystem auswählen. **Typbibliotheken.** Im unteren Teil des Dialogfelds werden die folgenden Informationen zu den einzelnen Typbibliotheken angezeigt. • **Beschreibung.** Der Name der Typbibliothek (tlb-Datei). • **Pfad.** Der Pfad der Typbibliothek. • **Guid.** Die GUID (Global Unique Identifier) der Typbibliothek.

Hinzufügen...	Fügt eine COM-Typbibliothek hinzu. • **Registrierung durchsuchen.** Zeigt eine Liste der Typbibliotheken an, die sich in der Registrierung des lokalen Computers befinden. Wählen Sie die gewünschte Bibliothek durch Aktivieren des zugehörigen Kontrollkästchens aus und klicken Sie auf **OK**. • **Dateisystem durchsuchen**. Ermöglicht die Auswahl der Typbibliotheken aus dem lokalen Dateisystem. • **MTS durchsuchen**. Ermöglich das Hinzufügen einer Komponente aus einem Microsoft Transaction Server. Im Dialogfeld **MTS-Komponenten** werden Sie aufgefordert, den Namen des MTS-Servers einzugeben. Geben Sie den Namen des MTS-Servers ein und klicken Sie auf **Verbindung**. Denken Sie daran, dass zum Aufzeichnen von MTS-Komponenten ein MTS-Client auf dem Computer installiert sein muss. Wählen Sie mindestens ein MTS-Komponentenpaket in der Liste aus und klicken Sie auf **OK**. Sobald das Paket unter **Typbibliotheken** angezeigt wird, können Sie einzelne Komponenten auswählen.
Entfernen	Entfernt eine COM-Typbibliothek.

Ausschließen...	Schließt Schnittstellen im Filter über das Dialogfeld **Ausgeschlossene Schnittstellen** aus. In diesem Dialogfeld werden die markierten Schnittstelleneinträge ausgeschlossen. Sie haben auch die Möglichkeit, Schnittstellen hinzuzufügen, die nicht aufgelistet werden. Klicken Sie im Dialogfeld **Ausgeschlosssene Schnittstellen** auf **Schnittstelle hinzufügen...** und geben Sie die GUID-Nummer (Schnittstellen-ID) und den Namen der Schnittstelle ein. Sie können die GUID aus der von VuGen erstellten Datei **interfaces.h** kopieren. Diese wird im VuGen-Fenster in der Auswahlstruktur in der Spalte auf der linken Seite aufgelistet. Sie können **Schnittstelle hinzufügen...** auch verwenden, um Schnittstellen auszuschließen, die unnötigerweise vom Skript aufgerufen werden, aber nicht im Filter aufgeführt sind. Eine Schnittstelle kann von verschiedenen Klassen unterschiedlich implementiert werden. Wenn Sie eine Schnittstelle ausschließen, die von anderen, nicht ausgeschlossenen Klassen implementiert wird, zeigt VuGen eine Warnung an. Wenn Sie die Option **Nicht mehr fragen** aktivieren und das Dialogfeld schließen, dann wird der Status aller Instanzen dieser Schnittstelle in allen anderen Klassen automatisch für diesen Filter geändert, sobald Sie den Status der Schnittstelle in einem Objekt ändern. Klicken Sie auf **Ja zu Allen**, um den Status aller Instanzen dieser Schnittstelle in allen anderen Klassen zu ändern. Klicken Sie auf **Nein zu Allen**, um den Status für alle anderen Instanzen unverändert zu lassen. Klicken Sie auf **Nächste Instanz**, um die nächste Klasse anzuzeigen, die diese Schnittstelle verwendet.

COM/DCOM > Optionen (Knoten)

Ermöglicht das Festlegen zusätzlicher Optionen für COM-Aufzeichnungssitzungen. Diese betreffen die Verarbeitung von Objekten, das Generieren von Protokolldateien und VARIANT-Definitionen.

Die DCOM-Skriptoptionen gelten für alle Programmiersprachen. Anhand der folgenden Einstellungen können Sie die Skriptoptionen für DCOM-Methoden und die Schnittstellenverarbeitung konfigurieren.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > COM/DCOM > Optionen**

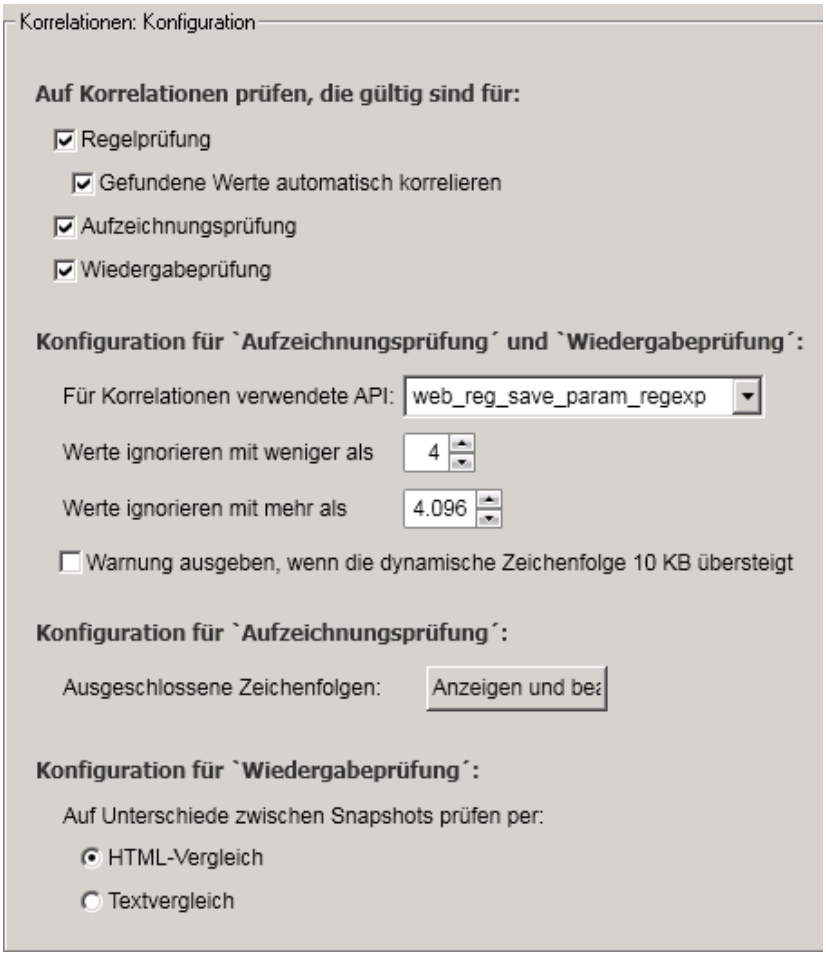
Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
ADO-Datensatzfilterung	Mehrere Datensatzoperationen werden zu einer einzeiligen FETCH-Anweisung zusammengefasst (standardmäßig aktiviert).

Temporäre VARIANT-Typen als global deklarieren	Temporäre VARIANT-Typen werden als globale und nicht als lokale Variablen definiert (standardmäßig aktiviert).
Array in separaten Bereichen füllen	Jedes Array wird in einen separaten Bereich eingefügt (standardmäßig aktiviert).
Struktur in separaten Bereichen füllen	Jede Struktur wird in einen separaten Bereich eingefügt (standardmäßig aktiviert).
COM-Ausnahmen generieren	COM-Funktionen und -Methoden, die während der Aufzeichnung Ausnahmen hervorgerufen haben, werden generiert (standardmäßig deaktiviert).
COM-Statistik generieren	Es werden Leistungsstatistiken und zusammenfassende Informationen erzeugt (standardmäßig deaktiviert).
Größe von SafeArray-Protokoll begrenzen	Begrenzt die Anzahl der pro COM-Aufruf im SafeArray-Protokoll erzeugten Elemente auf 16 (standardmäßig deaktiviert).
COM-Objekte freigeben	Die Freigabe von nicht mehr verwendeten COM-Objekten wird aufgezeichnet (standardmäßig aktiviert).
Datensatzinhalt speichern	Speichert den Inhalt des Datensatzes in Tabellen, um diese in VuGen anzeigen zu können (standardmäßig aktiviert).
Gebundene Monikerobjekte abgefangen	Alle gebundenen Monikerobjekte werden abgefangen (standardmäßig deaktiviert).

Korrelationen > Konfiguration

In diesem Bereich können Sie die Einstellungen für die Registerkarte **Korrelation** konfigurieren.

Beispiel für die Benutzer-oberfläche	
Zugriff	**VuGen > Aufzeichnungsoptionen > Korrelationen > Konfiguration**
Wichtige Informationen	• "Registerkarte "Korrelation" [Design Studio] - Übersicht" auf Seite 282 • "Korrelationen > Regeln" auf Seite 177 • Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.
Relevante Aufgaben	"Korrelieren von Skripts mit Design Studio" auf Seite 288

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Auf Korrelationen prüfen, die gültig sind für:	
Regelprüfung	Wenden Sie die Korrelationsregeln bei der Korrelationsprüfung an. Weitere Informationen finden Sie unter "Korrelationen > Regeln" auf der nächsten Seite.
Gefundene Werte automatisch korrelieren	Design Studio korreliert die gefundenen dynamischen Werte anhand der Regeln automatisch.
Aufzeichnungsprüfung	Prüft auf Korrelationen mit der aufzeichnungsbasierten Engine.
Wiedergabeprüfung	Prüft auf Korrelationen mit der wiedergabebasierten Engine.
Konfiguration für "Aufzeichnungsprüfung" und "Wiedergabeprüfung"	
Für Korrelationen verwendete API	Wählen Sie die API-Funktion aus, die für die Korrelation verwendet werden soll: **Begrenzungsbasiert** web_reg_save_param_ex **Regulärer Ausdruck**: web_reg_save_param_regexp **Hinweis:** Wenn Sie die API-Funktion ändern, werden die Änderungen erst nach einer neuen Prüfung angewendet.
Zeichenfolgen ausschließen	Hier können Sie Zeichenfolgen eingeben, die bei der Aufzeichnungs- und Wiedergabeprüfung ignoriert werden sollen. Weitere Informationen finden Sie unter "Ausschließen von Zeichenfolgen aus der Korrelationsprüfung" auf Seite 320.
Werte ignorieren mit weniger als [ ]	Dynamische Werte, die kürzer sind als der hier definierte Wert, werden bei der Aufzeichnungs- oder Wiedergabeprüfung ignoriert. Die Standardeinstellung ist 4 Zeichen.
Werte ignorieren mit mehr als [ ]	Dynamische Werte, die länger sind als der hier definierte Wert, werden bei der Aufzeichnungs- oder Wiedergabeprüfung ignoriert. Die Standardeinstellung ist 400 Zeichen.
Warnung ausgeben, wenn die dynamische Zeichenfolge 10 KB übersteigt	Hiermit wird eine Warnung ausgegeben, wenn Sie versuchen, eine Zeichenfolge mit 10 KB oder mehr zu korrelieren.
Groß-/Kleinschreibung bei der Suche nach korrelierbaren Werten ignorieren	Groß-/Kleinschreibung wird während der Korrelationsprüfung nicht berücksichtigt.
Konfiguration für 'Aufzeichnungsprüfung'	

Element der Oberfläche	Beschreibung
Heuristische Ebene	Hiermit können Sie die Filterebene festlegen, mit der der Umfang der zurückgegebenen Korrelationsergebnisse gesteuert wird. Je höher die Filterebene, desto kürzer dauert die Ausführung der Prüfung. **High.** Design Studio führt eine detaillierte Prüfung durch und gibt einen extrem feinen Ergebnissatz zurück. **Medium.** Design Studio führt eine weniger detaillierte Prüfung durch und gibt mehr Ergebnisse zurück. Dies ist die Standardeinstellung. **Low.** Design Studio führt eine ausführlichere Prüfung durch und gibt dabei die meisten Ergebnisse zurück. Diese Einstellung kann viele unerwünschte Ergebnisse erzeugen.
Konfiguration für 'Wiedergabeprüfung'	
Auf Unterschiede zwischen Snapshots prüfen per	Wählen Sie eine Vergleichsmethode aus: • **HTML-Vergleich.** Zeigt nur die Unterschiede im HTML-Code an. • **Textvergleich.** Zeigt alle Text-, HTML- und Binär-Unterschiede an.

Korrelationen > Regeln

In diesem Dialogfeld können Sie Korrelationsregeln verwalten, mit denen dynamische Werte während der Codegenerierung automatisch korreliert werden. Sie können:

- Neue Anwendung hinzufügen
- Neue Regel hinzufügen
- Regel löschen
- Regeln exportieren
- Regeln importieren
- Regel testen

Zugriff	Verwenden Sie eine der folgenden Methoden: • **VuGen > Aufzeichnen > Aufzeichnungsoptionen > Korrelation > Regeln** • **Studio** > **Korrelationen > Regeln**
Wichtige Informationen	"Korrelieren – Übersicht" auf Seite 281 "Registerkarte "Korrelation" [Design Studio] - Übersicht" auf Seite 282
Relevante Aufgaben	"Korrelieren von Skripts mit Design Studio" auf Seite 288

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Anwendungs-liste >	Die Liste der Anwendungen und zugehörigen Regeln. • Aktivieren Sie das Kontrollkästchen neben der Anwendung, um sie während der Aufzeichnung zu aktivieren. • Erweitern Sie die Anwendungsstruktur, um das Kontrollkästchen neben den Regeln zu aktivieren und die Regeln während der Aufzeichnung zu aktivieren.
Neue Anwendung	Fügt eine neue Anwendung zur **<Anwendungsliste>** hinzu.
Neue Regel	Eingabe einer neuen Regel für die ausgewählte Anwendung in den Korrelationsregeln. Weitere Informationen finden Sie unter "Bereich "Neue Regel"" unten.
Löschen	Löscht die ausgewählte Regel oder Anwendung aus der Liste.
Importieren...	Importiert eine Datei mit Definitionen von Korrelationsregeln.
Exportieren...	Exportiert eine Datei mit einer Definition einer Korrelationsregel.
Test...	Testet eine Korrelationsregel. Weitere Informationen finden Sie unter "Dialogfeld "Testumgebung für Zeichenfolgenersetzung"" auf Seite 181.

Bereich "Neue Regel"

Ermöglicht das Definieren einer neuen benutzerdefinierten Regel.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Korrelation > Regeln > Neue Regel**
Wichtige Informationen	Dieser Bereich steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie in der "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Öffnet das "Dialogfeld "Erweiterte Korrelationseigenschaften"" auf der nächsten Seite.
Aktion	Wählen Sie den Aktionstyp für die Regel aus den folgenden Optionen aus: • **Im gesamten Textkörper nach Parametern suchen.** Durchsucht den gesamten Text, nicht nur Links, Formularaktionen oder Cookies. VuGen durchsucht anhand der angegebenen Begrenzungen den Text nach einer Übereinstimmung. • **In Links und Formularaktionen nach Parametern suchen.** Durchsucht Links und Formularaktionen nach Text für die Parametrierung. Diese Methode eignet sich für Anwendungsserver mit bekannten Kontextregeln. Sie definieren eine linke, eine rechte und eine alternative rechte Begrenzung sowie eine Instanz der linken Begrenzung im aktuellen Link. • **In den Kopfzeilen von Cookies nach Parametern suchen.** Mit der vorherigen Regel vergleichbar, nur dass der Wert aus dem Cookie-Text (genauso wie er im Aufzeichnungsprotokoll dargestellt wird) extrahiert wird, anstatt aus einem Link oder einer Formularaktion. • **Formularfeldwerte parametrieren.** Speichert den benannten Wert des Formularfeldes in einem Parameter. Dabei wird ein Parameter erstellt und vor dem Aktionsschritt des Formulars im Skript eingefügt. Für diese Option müssen Sie den Feldnamen angeben. • Bei der Methode **Text für das Hinzufügen einer web_reg_add_cookie-Funktion** wird eine Funktion **web_reg_add_cookie** eingefügt, wenn eine bestimmte Zeichenfolge im Puffer gefunden wird. Dabei wird die Funktion nur für die Cookies mit dem angegebenen Präfix hinzugefügt. Für diese Option müssen Sie den Suchtext und das Cookie-Präfix angeben.
Prüfungstyp	Legen Sie als Prüfungstyp entweder "Regulärer Ausdruck" oder "Begrenzungsbasiert" fest.

RegExp-Zeichenfolge	Legen Sie den regulären Ausdruck fest. Dieses Element wird nur für den Prüfungstyp "Regulärer Ausdruck" verwendet.
Linke Begrenzung	Die äußere linke Begrenzung für die Anwendung der Regel. Dieses Element wird nur für den Prüfungstyp "Begrenzungsbasiert" verwendet.
Groß-/Kleinschreibung	Bei der Suche nach Begrenzungen wird die Groß- und Kleinschreibung beachtet.
Parameterpräfix	Verwendet ein Präfix in allen anhand dieser Regel automatisch erzeugten Parametern. Mit Präfixen wird das Überschreiben bestehender Benutzerparameter verhindert. Außerdem können Sie mit Präfixen den Parameter im Skript erkennen. Beispiel: In Siebel-Web sucht eine der integrierten Regeln nach dem Präfix **Siebel_row_id**.
Rechte Begrenzung	Die äußere rechte Begrenzung für die Anwendung der Regel. Unter Verwendung des Dropdownmenüs können Sie diese Begrenzung als Ende der Zeichenfolge, einen Zeilenumbruch oder benutzerdefinierten Text definieren. Das Element wird nur für den Prüfungstyp "Begrenzungsbasiert" verwendet.
'#' für beliebige Ziffern verwenden	Alle Ziffern werden durch ein Nummernzeichen ersetzt. Die Nummernzeichen dienen als Platzhalter, damit Sie Textzeichenfolgen mit beliebigen Ziffern finden können. **Beispiel:** Wenn Sie diese Option aktivieren und **HP###** als linke Begrenzung angeben, sind **HP193** und **HP284** gültige Übereinstimmungen. Dieses Element wird nur für den Prüfungstyp "Begrenzungsbasiert" verwendet.

Dialogfeld "Erweiterte Korrelationseigenschaften"

Ermöglicht das Festlegen der erweiterten Optionen für Korrelationsregeln.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Korrelation > Regeln > Neue Regel > Erweitert**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie in der "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung

Forts.

Alternative rechte Begrenzung	Alternatives Kriterium für die rechte Begrenzung, wenn die zuvor angegebene Begrenzung nicht gefunden wird. Wählen Sie eine der folgenden Optionen: **Benutzerdefinierter Text, Zeilenumbruch, Seitenende.**
Immer neuen Parameter erstellen	Erstellt einen neuen Parameter für diese Regel, selbst wenn der durch den Parameter ersetzte Wert seit der vorherigen Instanz nicht geändert wurde.
Vorkommen linke Begrenzung	Die Anzahl der Vorkommen der linken Begrenzung, damit diese als Übereinstimmung gilt.
Länge	Die Länge der Zeichenfolge ab dem angegebenen Offset zum Speichern des Parameters. Wird dieser Wert nicht angegeben, wird der Parameter bis zum Ende des gefunden Werts fortgesetzt.
Offset	Der Offset der Zeichenfolge innerhalb des gefunden Werts.
Nur bei genauen Übereinstimmungen durch Parameter ersetzen	Ersetzt einen Wert nur dann durch einen Parameter, wenn der Text exakt dem gefundenen Wert entspricht.
Rückwärts suchen	Führt eine Rückwärtssuche durch.

Dialogfeld "Testumgebung für Zeichenfolgenersetzung"

Ermöglicht Ihnen, Korrelationsregeln vor ihrer Anwendung zu testen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Korrelation > Regeln > Test**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Führt den Test aus.
Angewendete Regeln	Eine Liste der Regeln, die während des Tests angewendet wurden.
Zu ersetzende Zeichenfolge	Geben Sie die zu ersetzende Zeichenfolge ein.
Ergebnis der Ersetzung	Die Ergebnisse des Tests.

Datenbank > Datenbank (Knoten)

Ermöglicht das Festlegen der Aufzeichnungsoptionen für Datenbankprotokolle.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Datenbank > Datenbank**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Öffnet das "Datenbank > Dialogfeld "Erweiterte Aufzeichnungsoptionen"" unten.
Automatische Transaktionen	Markiert jede Funktion **lrd_exec** und **lrd_fetch** als Transaktion. Sind diese Optionen aktiviert, fügt VuGen die Funktionen **lr_start_transaction** und **lr_end_transaction** vor bzw. hinter jeder Funktion **lrd_exec** oder **lrd_fetch** ein. **Standardwert:** Deaktiviert.
Skriptoptionen	Generiert Kommentare in aufgezeichneten Skripts, um die **lrd_stmt**-Optionswerte zu beschreiben. Darüber hinaus können Sie die maximale Länge einer Skriptzeile angeben. **Standardwert:** 80 Zeichen**.**
Latenzzeit	VuGen zeichnet automatisch die Latenzzeit des Bedieners auf. Sie können einen Schwellenwert festlegen, bis zu dem die aufgezeichnete Latenzzeit ignoriert wird. Wenn die aufgezeichnete Latenzzeit über dem Schwellenwert liegt, fügt VuGen eine Anweisung **lr_think_time** vor LRD-Funktionen ein. Liegt die aufgezeichnete Latenzzeit unter dem Schwellenwert, wird keine Anweisung **lr_think_time** generiert. **Standardwert:** fünf Sekunden.

Datenbank > Dialogfeld "Erweiterte Aufzeichnungsoptionen"

Ermöglicht das Festlegen der erweiterten Aufzeichnungsoptionen für Datenbankprotokolle.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Datenbank > Datenbank > Erweitert**

Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Puffergröße für Codegenerierung	Mit dieser Option geben Sie die maximale Größe des Codegenierungspuffers in Kilobyte an. **Standardwert:** 128 Kilobyte.
Optionen für CtLib-Funktionen	Sie können festlegen, dass VuGen einen Zeitstempel für das Senden von Daten oder eine Anweisung für einen erweiterten Ergebnissatz generiert. • **Zeitstempel für das Senden von Daten generieren.** Generiert die Anweisung **lrd_send_data** mit den Schlüsselwörtern **TotalLen** und **Log** für den Parameter **mpszReqSpec**. Im Dialogfeld **Erweiterte Aufzeichnungsoptionen** können Sie festlegen, dass VuGen auch das Schlüsselwort **TimeStamp** generieren soll. Wenn Sie diese Einstellung bei einem vorhandenen Skript ändern, müssen Sie das Vuser-Skript erneut generieren, indem Sie **Aufzeichnen > Skript neu generieren** wählen. Es wird nicht empfohlen, das Schlüsselwort **Timestamp** standardmäßig zu generieren. Der während der Aufzeichnung generierte Zeitstempel stimmt nicht mit dem bei der Wiedergabe generierten Zeitstempel überein, sodass die Skriptausführung fehlschlägt. Sie sollten diese Option nur nach der erfolglosen Ausführung eines Skripts verwenden, in dem **lrd_result_set** nach **lrd_send_data** fehlschlägt. Der generierte Zeitstempel kann mit einem Zeitstempel korreliert werden, der von einer früheren Funktion **lrd_send_data** generiert wurde. • **Anweisung für erweiterten Ergebnissatz generieren.** Generiert beim Vorbereiten des Ergebnissatzes die Funktion **lrd_result_set**. Durch diese Einstellung generiert VuGen die erweiterte Form der Funktion **lrd_result_set**, die **lrd_result_set_ext** lautet. Diese Funktion bereitet nicht nur einen Ergebnissatz vor, sondern gibt auch einen Rückgabecode und -typ von **ct_results** aus.
Aufzeichnungsengine	Sie können festlegen, dass VuGen alle Skripts mit der älteren LRD-Aufzeichnungsengine erfasst, um Kompatibilität mit früheren VuGen-Versionen zu gewährleisten. **Hinweis:** Diese Option ist nur für Einzelprotokollskripts verfügbar.

Optionen für Aufzeichnungsprotokoll	Sie können die Detailebene für Trace- und ASCII-Protokolldateien festlegen. Die verfügbaren Ebenen für die Tracedatei lauten **Aus**, **Fehlerverfolgung**, **Kurze Verfolgung** und **Vollständige Verfolgung**. Mit der Option **Fehlerverfolgung** werden nur Fehlermeldungen protokolliert. Die Option **Kurze Verfolgung** protokolliert Fehler und listet die während der Aufzeichnung generierten Funktionen auf. Mit der Option **Vollständige Verfolgung** werden alle Meldungen, Benachrichtigungen und Warnungen protokolliert. Sie können auch festlegen, dass VuGen die Aufzeichnungssitzung im ASCII-Format protokollieren soll. Die verfügbaren Ebenen lauten **Aus**, **Kurzdetail** und **Volldetail**. Mit **Kurzdetail** werden alle Funktionen und mit **Volldetail** alle generierten Funktionen und Meldungen im ASCII-Code protokolliert.

Datenformaterweiterung > Kettenkonfiguration (Knoten)

Ermöglicht das Hinzufügen, Löschen und Ändern von Ketten sowie das Verwalten der in den Ketten enthaltenen DFEs.

Zugriff	Verwenden Sie eine der folgenden Methoden: • **VuGen > Aufzeichnen > Aufzeichnungsoptionen > Datenformaterweiterungen > Kettenkonfiguration** • **VuGen > Wiedergeben > Laufzeiteinstellungen > Datenformaterweiterungen > Codegenerierung**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.
Siehe auch	• "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912 • "Liste der Datenformaterweiterungen" auf Seite 922

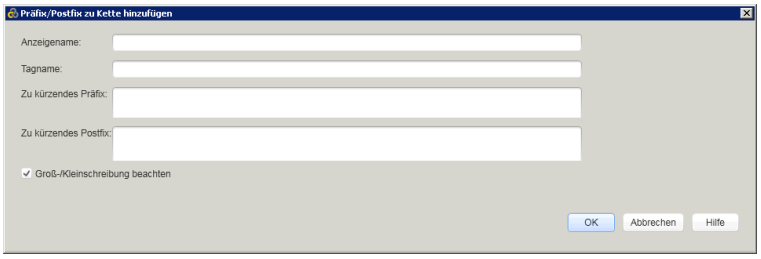
Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Bereich "Ketten"	Zeigt eine Liste der DFE-Ketten an, die für das Skript definiert sind.
	Kette hinzufügen. Ermöglicht Ihnen, eine neue Kette hinzuzufügen.

	Kettenname bearbeiten. Ermöglicht Ihnen, den Namen der ausgewählten Kette zu ändern.
	Kette löschen. Löscht die ausgewählte Kette.
Kette: Bereich <Kettenname>	
	DFE hinzufügen. Ermöglicht Ihnen das Hinzufügen einer DFE zu der im Bereich **Ketten** ausgewählten Kette. Weitere Informationen zu Datenformaterweiterungen finden Sie unter "Liste der Datenformaterweiterungen" auf Seite 922.
	DFE bearbeiten. Wenn Sie eine Präfix/Postfix-Erweiterung ausgewählt haben, können Sie die Details im Dialogfeld **Präfix/Postfix zu Kette hinzufügen** bearbeiten. Weitere Informationen zu diesem Dialogfeld finden Sie unter "Dialogfeld "Präfix/Postfix zu Kette hinzufügen"" auf der nächsten Seite.
	DFE löschen. Löscht die ausgewählte DFE aus der Kette.
	Nach oben/unten verschieben. Verschiebt die ausgewählte Datenformaterweiterung in der Kette nach oben oder unten. Erweiterungen werden in der Reihenfolge ausgeführt, in der sie in der Erweiterungsliste angezeigt werden.
Name	Der Anzeigename der Datenformaterweiterung.
Markierung	Die eindeutige ID der Erweiterung.
Anbieter	Der Ersteller der Datenformaterweiterung.
Verarbeitung fortsetzen	Bestimmt das Verhalten der Kette nach dem Anwenden der DFE: • **True:** Die Daten werden an die nächste DFE in der Kette weitergegeben, unabhängig davon, ob die Daten konvertiert wurden. • **False:** Wenn die DFE die empfangenen Daten konvertiert hat, wird die Kette beendet - auf die Daten werden keine weiteren DFEs angewendet. Wenn die DFE die empfangenen Daten nicht konvertiert hat, werden die Daten an die nächste DFE in der Kette übergeben. **Hinweis:** Wenn die Kette nur eine einzige DFE enthält, ist die Einstellung **Verarbeitung fortsetzen** nicht von Bedeutung.

Dialogfeld "Präfix/Postfix zu Kette hinzufügen"

Über dieses Dialogfeld können Sie der ausgewählten Kette eine Präfix/Postfix-Erweiterung hinzufügen oder diese bearbeiten.



Zugriff	1. Wechseln Sie zu **Aufzeichnen > Aufzeichnungsoptionen > Datenformaterweiterungen > Kettenkonfiguration**. 2. Klicken Sie im Bereich **Kette: <Name verketten>** auf die Schaltfläche . 3. Wählen Sie **Präfix/Postfix-Erweiterung**, und klicken Sie auf **OK**.
Siehe auch	"Datenformaterweiterung > Kettenkonfiguration (Knoten)" auf Seite 184 "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912 "Liste der Datenformaterweiterungen" auf Seite 922

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Groß-/Kleinschreibung beachten	Legt die Erweiterung zum Kürzen des definierten Präfix und Postfix der Zeichenkette fest, wenn die Schreibweise übereinstimmt.
Anzeigename	Der Name der Präfix/Postfix-Erweiterung.
Zu kürzendes Postfix	Der zu kürzende Abschnitt ab dem Ende der Zeichenkette.
Zu kürzendes Präfix	Der zu kürzende Abschnitt ab dem Anfang der Zeichenkette.
Tagname	Die eindeutige ID der Präfix/Postfix-Erweiterung.

Datenformaterweiterung hinzufügen

In diesem Dialogfeld können Sie den Typ der Datenformaterweiterung auswählen.

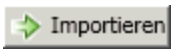
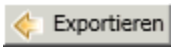
Beispiel für die Benutzeroberfläche	
Zugriff	**VuGen > Aufzeichnungsoptionen > Datenformaterweiterung > Kettenkonfiguration >**
Wichtige Informationen	• "Datenformaterweiterung > Kettenkonfiguration (Knoten)" auf Seite 184 • "Dialogfeld "Präfix/Postfix zu Kette hinzufügen"" auf der vorherigen Seite
Relevante Aufgaben	"Anwenden von DFE-Ketten auf Abschnitte der HTTP-Nachricht" auf Seite 920
Datenformaterweiterung	**Beschreibung**
Base64-Erweiterung	Decodiert Zeichenfolgen, die mit einem BASE64-Encoder codiert wurden.
GWT-Erweiterung	Wandelt GWT-Daten in das XML-Format um.
Erweiterung für URL-Codierung	Zur Dekodierung von Zeichenfolgen, die mit einem URL-Format codiert wurden.
JSON-zu-XML-Erweiterung	Wandelt JSON-Daten in das XML-Format um.
XML-Erweiterung	Überprüft Daten auf ihre Übereinstimmung mit dem XML-Syntax. Diese Prüfung ermöglicht es VuGen, Korrelationen auf Grundlage von XPath auszuführen und Snapshot-Daten in einer XML-Anzeige darzustellen.
Präfix/Postfix-Erweiterung	Damit können Sie Daten am Anfang und/oder Ende einer Zeichenfolge ausschneiden, die nicht decodiert werden sollen. Sie können so viele Präfix/Postfix-Erweiterungen hinzufügen und anpassen, wie erforderlich sind. Jede erstellte Präfix/Postfix-Erweiterung sollte einen eindeutigen Anzeige- und Tagnamen erhalten.
Binär-zu-XML-Erweiterung	Wandelt binäre Microsoft WCF-XML-Daten in das XML-Format um.
Remedy-zu-XML-Erweiterung	Wandelt Remedy-Anforderungsdaten in XML-Format um. Beachten Sie, dass diese Erweiterung keine Remedy-Antwortdaten umwandelt, da diese im JavaScript-Code sind.
XSS-Erweiterung	Ermöglicht es Ihnen, Sites zu testen, die XSS-Abwehrcode (Cross Site Scripting) verwenden.

Datenformaterweiterung > Codegenerierung (Knoten)

Aktiviert Datenformaterweiterungen während der Codegenerierung und ermöglicht Ihnen das Definieren von Ketten für jeden Abschnitt der HTTP-Nachricht.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Datenformaterweiterungen > Codegenerierung**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.
Siehe auch	• "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912 • "Liste der Datenformaterweiterungen" auf Seite 922

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
Datenformaterweiterung aktivieren	Ermöglicht Ihnen das Auswählen von Ketten für jeden Nachrichtenabschnitt der HTTP-Nachricht. Diese Option ist standardmäßig deaktiviert.
Konfiguration	
Format	• **Code und Snapshots**. (Standard) Aktiviert Datenformaterweiterungen in Code- und Snapshotdaten. • **Snapshots**. Aktiviert Datenformaterweiterungen in Snapshotdaten, formatiert jedoch nicht die Daten im Skript selbst.
Formatierte Daten überprüfen	Überprüft die Ergebnisse formatierter Daten, indem sie zurück in den ursprünglichen Zustand konvertiert und mit den ursprünglichen Daten abgeglichen werden. **Hinweis**: Nur für Base64-Erweiterung verfügbar.
Kettenzuweisung	
	Importiert die Datenformaterweiterungen aus einer Datei.
	Exportiert die Datenformaterweiterungen in eine Datei.

<Liste der Nachrichtenabschnitte>	Zeigt eine Liste der folgenden Abschnitte der im Skript enthaltenen HTTP-Nachricht an: • Text • Kopfzeilen • Cookies • Abfragezeichenfolge Wenn Sie in der Liste einen Nachrichtenabschnitt auswählen, gibt der Titel des Abschnittskettenbereichs (im Folgenden beschrieben) Ihre Auswahl wieder und im Bereich wird die Liste der Ketten für diesen Abschnitt angezeigt.
<Abschnittsketten>	
	Kette hinzufügen. Fügt eine Kette zum ausgewählten Nachrichtenabschnitt hinzu. **Hinweis**: • Nur für Kopfzeilen- und Cookie-Abschnitte aktiviert. Ermöglicht Ihnen das Hinzufügen zusätzlicher Ketten zum ausgewählten Nachrichtenabschnitt. • Damit VuGen die Kette dem Kopfzeilen- oder Cookie-Abschnitt korrekt zuordnen kann, muss der Name in der Spalte **Name** mit dem Namen des Kopfzeilen- oder Cookie-Abschnitts übereinstimmen.
	Kette löschen. Entfernt die Kette aus dem entsprechenden Nachrichtenabschnitt. **Hinweis:** Sie können die Standardoptionen aus keinem der Nachrichtenabschnitte löschen.
	Zurücksetzen. Löscht die in der Spalte **Kette** ausgewählte Kette.

Flex > RTMP (Knoten)

Dieser Knoten ermöglicht das Einbeziehen des Schritts flex_rtmp_recive_stream in Flex-RTMP-Skripts.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Flex > RTMP**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Flex_ rtmp_ receive_ stream-Schritt generieren	Generiert beim Aufzeichnen eines Datenstroms einen einzelnen Schritt. Dieser Schritt führt keine bestimmten Aktionen aus, wie Pause und Suchen. Wenn diese Aktionen für Ihr Skript erforderlich sind, deaktivieren Sie das Kontrollkästchen, um alle Schritte zu empfangen und zu senden. In diesem Fall müssen Sie jedoch Ihr Skript wie in der *Readme*-Datei beschrieben manuell ändern.

Flex > Konfiguration (Knoten) (Aufzeichnen)

Ermöglicht das Festlegen eines externen JVM-Pfads (Java Virtual Machine).

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Flex > Konfiguration**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Externe JVM verwenden	Ermöglicht die Verwendung einer externen JVM. Wenn Sie diese Option auswählen, müssen Sie den Pfad angeben: **Externer JVM-Pfad** Der vollständige Pfad der externen JVM. VuGen muss neu gestartet werden, damit die Änderungen wirksam werden.
GraniteDS-Konfiguration verwenden	Definiert die serverseitige Datendienstkonfiguration. Wenn Sie diese Option auswählen, wählen Sie nicht **Flex LCDS/BlazeDS-JAR-Dateien verwenden** aus, um die Nachrichten zu serialisieren. Stellen Sie sicher, dass die Datei "granit-config.xml" mit der auf dem Server bereitgestellten Datei übereinstimmt.

Maximal zulässige Größe der formatierten Anforderung/Antwort zur Ausgabe	Geben Sie hier die maximale Zeichenlänge für Anforderungen oder Antworttexte zur Ausgabe in die Protokolldateien ein. Diese Option wirkt sich nur auf die Schritte **flex_amf_call** und **flex_remoting_call** aus. Wenn Sie beispielsweise den Wert 1048576 Zeichen (1 MB) angeben, werden nur Antworten oder Anfragen mit einer Länge von weniger als einem Megabyte im Codegenerierungs- oder Wiedergabeprotokoll erfasst.

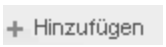
Flex > Externalisierbare Objekte (Knoten) (Aufzeichnen)

Mit diesem Dialogfeld können Sie konfigurieren, wie LoadRunner externalisierbare Objekte in Flex-Skripts behandelt.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Flex > Externalisierbare Objekte**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.
Relevante Aufgaben	• Serialisieren mit der externen Java-Serialisierungskomponente • "Serialisieren von Flex-Skripts " auf Seite 704
Siehe auch	• "Flex – Übersicht" auf Seite 682 • "Externalisierbare Objekte in Flex-Skripts" auf Seite 700

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Externalisierbare Objekte nicht serialisieren	Generiert Skripts unter Verwendung der Standardeinstellungen.

Serialisieren von Objekten mit	Wählen Sie eine geeignete Option aus: • Wählen Sie **LoadRunner AMF Serializer** aus, wenn Sie nicht den Adobe LiveCycle Data Services- oder Adobe BlazeDS-Server verwenden. • Wählen Sie **Benutzerdefinierte Java-Klassen** und dann eine oder beide der folgenden verfügbaren Optionen aus: ▪ Wählen Sie **Flex LCDS/BlazeDS-JAR-Dateien verwenden** aus, wenn Sie Flex LCDS- oder BlazeDS-JAR-Dateien für die Serialisierung der Nachrichten verwenden. Falls Sie **GraniteDS-Konfiguration verwenden** im Knoten **Konfiguration** ausgewählt haben, wählen Sie **Flex LCDS/BlazeDS-JAR-Dateien verwenden** nicht aus. ▪ Wählen Sie **Weitere JAR-Dateien verwenden** aus, um zusätzliche JAR-Dateien für die Serialisierung der Nachrichten hinzuzufügen. Sie müssen die JAR-Dateien vom Server kopieren und ihren Speicherort in der Liste **Klassenpfadeinträge** (siehe unten) angeben. Kopieren Sie nur die JAR-Dateien, die die externalisierbare Klasse enthalten. Außerdem müssen Sie sicherstellen, dass sich diese Dateien auf allen Lastgenerator-Computern am gleichen Speicherort befinden. Wenn Sie JAR-Dateien mit denselben Namen wie die Flex LCDS- oder Blaze DS-JAR-Dateien hinzufügen, indem Sie das erste Kontrollkästchen auswählen, werden diese Dateien überschrieben.
Liste der Klassenpfadeinträge	
	Nach unten. Verschiebt einen Klassenpfadeintrag in der Liste nach unten.
	Nach oben. Verschiebt einen Klassenpfadeintrag in der Liste nach oben.
	**Klassenpfad hinzufügen.** Fügt eine neue Zeile zur Klassenpfadliste hinzu.
	Klassenpfadordner hinzufügen. Fügt alle Dateien des Ordners zur Klassenpfadliste hinzu.
	Löschen. Entfernt einen Klassenpfad unwiderruflich.

Allgemein > Codegenerierung

In diesem Bereich des Dialogfelds **Aufzeichnungsoptionen** können Sie festlegen, welche Aufgaben VuGen nach der Generierung eines Vuser-Skripts automatisch durchführt.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Allgemein > Codegenerierung**

Relevante Aufgaben	"Erstellen eines asynchronen Vuser-Skripts" auf Seite 480 "Korrelieren von Skripts mit Design Studio" auf Seite 288

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Korrelationsprüfung	Weist VuGen an, das Vuser-Skript zu analysieren und nach dynamischen Werten zu suchen, die möglicherweise korreliert werden müssen. Diese Prüfung wird durchgeführt, nachdem ein neues Skript aufgezeichnet wurde und nachdem ein vorhandenes Skript neu generiert wurde.
Asynchrone Prüfung	Weist VuGen an, das Vuser-Skript zu analysieren und nach asynchroner Kommunikation zu suchen. Diese Prüfung wird durchgeführt, nachdem ein neues Skript generiert wurde und nachdem ein vorhandenes Skript neu generiert wurde.
Asynchron-Optionen...	Öffnet das "Dialogfeld "Asynchron-Optionen"" auf Seite 506.

Allgemein > Protokolle (Knoten)

Ermöglicht das Festlegen der benutzerdefinierten Einstellungen für die Skriptgenierung durch Festlegen der Skriptsprache und -optionen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Allgemein > Protokolle**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Liste "Aktive Protokolle"	Eine Liste der Protokolle, die das Mehrfachprotokollskript einbeziehen. Sie können die Protokollliste in VuGen ändern, für die der Code während der Aufzeichnungssitzung generiert wird. Wählen Sie die Kontrollkästchen neben den Protokollen aus, die in der nächsten Aufzeichnungssitzung protokolliert werden sollen. Nehmen Sie die Auswahl der Kontrollkästchen neben den Protokollen zurück, die in der nächsten Aufzeichnungssitzung nicht protokolliert werden sollen.

Allgemein > Aufzeichnen (Knoten)

Ermöglicht es Ihnen, mit der Auswahl der Aufzeichnungsebene festzulegen, welche Informationen aufgezeichnet werden sollen und welche Funktionen zum Generieren eines Vuser-Skripts verwendet werden sollen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Allgemein > Aufzeichnen**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
HTML erweitert...	Öffnet das "Dialogfeld "Erweitertes HTML"" auf der nächsten Seite.
URL erweitert...	Öffnet das "Dialogfeld "Erweiterter URL"" unten.
HTML-basiertes Skript	Dies ist die Standardaufzeichnungsebene für Web-Vuser (HTTP/HTML). Sie weist VuGen an, HTML-Aktionen im Kontext der jeweils aktuellen Webseite aufzuzeichnen. Während der Aufzeichnungssitzung zeichnet sie nicht alle Ressourcen auf, sondern lädt sie bei der Wiedergabe herunter. Diese Option wird für Browseranwendungen mit Applets und VB Script empfohlen.
URL-basiertes Skript	Zeichnet alle Anfragen und Ressourcen des Servers auf. Jede HTTP-Ressource wird automatisch als URL-Schritt (**web_url**-Anweisung) bzw. bei Formularen als **web_submit_data** aufgezeichnet. Die Funktionen **web_link**, **web_image** und **web_submit_form** werden nicht generiert und Frames werden nicht aufgezeichnet. Diese Option wird für Nicht-Browser-Anwendungen empfohlen.

Dialogfeld "Erweiterter URL"

Ermöglicht das Festlegen der erweiterten Optionen für Skripts, die den URL-Aufzeichnungsmodus verwenden.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Allgemein > Aufzeichnen > URL erweitert**
Wichtige Informationen	Dieses Dialogfeld steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Standardwerte	Stellt die Standardeinstellungen dieses Dialogfelds wieder her.
Gleichzeitige Gruppen für Ressourcen nach zugehöriger Quell-HTML-Seite erstellen	Zeichnet die Ressourcen in einer gleichzeitigen Gruppe (umschlossen von den Anweisungen **web_concurrent_start** und **web_concurrent_end**) im Anschluss an den URL auf. Zu den Ressourcen zählen Dateien wie z. B. Bild- und **js**-Dateien. Bei Deaktivierung dieser Option werden die Ressourcen als separate **web_url**-Schritte aufgelistet, aber nicht als gleichzeitige Gruppe markiert.
Enable EUC-Encoded Web Pages	(Nur japanisches Windows) Weist VuGen an, EUC-Codierung zu verwenden. Weitere Informationen finden Sie unter "EUC-Codierung (nur japanisches Windows)" auf Seite 255.
Nur web_custom_request verwenden	Zeichnet alle HTTP-Anforderungen als benutzerdefinierte Anforderungen auf. VuGen generiert für alle Anforderungen, gleich welchen Inhalts, eine **web_custom_request**-Funktion. Wird für Nicht-Browser-Anwendungen empfohlen.

Dialogfeld "Erweitertes HTML"

Ermöglicht das Festlegen erweiterter Optionen für HTTP-basierte Skripts.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Allgemein > Aufzeichnen > HTML erweitert**
Wichtige Informationen	Dieses Dialogfeld steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Standardwerte	Stellt die Standardeinstellungen dieses Dialogfelds wieder her.

Nicht-HTML-generierte Elemente	Viele Webseiten enthalten Elemente, die nicht das HTML-Format aufweisen (Applets, XML, ActiveX-Elemente oder JavaScript usw.) Nicht-HTML-Elemente enthalten in der Regel eigene Ressourcen oder rufen diese ab. Die folgenden Optionen bieten Ihnen die Möglichkeit, die Aufzeichnung von Nicht-HTML-generierten Elementen durch VuGen zu steuern. • **Innerhalb des aktuellen Skriptschritts ausführen** Generiert keine neue Funktion für die einzelnen Nicht-HTML-generierten Ressourcen. Die Option listet alle Ressourcen als Argumente der entsprechenden Funktionen auf, z. B. **web_url**, **web_link** und **web_submit_data**. Die Ressourcen sind Argumente der Webfunktionen und werden durch das Kennzeichen **EXTRARES** angezeigt. • **In separaten Schritten unter Verwendung gleichzeitiger Gruppen aufzeichnen**. Erstellt für jede Nicht-HTML-generierte Ressourcen eine neue Funktion und schließt diese nicht als Elemente in die Seitenfunktionen ein (z. B. **web_url** und **web_link**). Alle für eine Ressource generierten **web_url**-Funktionen werden in eine gleichzeitige Gruppe eingefügt (umschlossen von **web_concurrent_start** und **web_concurrent_end**). • **Nicht aufzeichnen.** Zeichnet keine Nicht-HTML-generierten Ressourcen auf.
Skripttyp	• **Ein Skript zur Beschreibung von Benutzeraktionen.** Generiert Funktionen, die der ausgeführten Aktion direkt entsprechen. Sie erstellt Funktionen für URL (**web_url**), Links (**web_link**), Bilder (**web_image**) und die Formularübergabe (**web_submit_form**). Das resultierende Skript ist äußerst intuitiv und ähnelt einer kontextsensitiven Aufzeichnung. • **Ein Skript, das nur explizite URLs enthält.** Zeichnet alle Links, Bilder und URLs als **web_url**-Anweisungen auf. Formulare werden als **web_submit_data** aufgezeichnet. Die Funktionen **web_link**, **web_image** und **web_submit_form** werden nicht generiert. Das resultierende Skript ist weniger intuitiv. Dieser Modus ist für Instanzen nützlich, bei denen viele Links einer Website den gleichen Linktext aufweisen. Wenn die Website mit der ersten Option aufgezeichnet wird, steht für den Link eine Ordnungszahl (Instanz), bei Auswahl der zweiten Option wird jeder Link mit seinem URL aufgelistet. Auf diese Weise werden die Parametrierung und die Korrelation für diesen Schritt vereinfacht.

Allgemein > Skript (Knoten)

Ermöglicht das Festlegen der benutzerdefinierten Einstellungen für die Skriptgenierung durch Festlegen der Skriptsprache und -optionen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Allgemein > Skript**

Forts.

Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Für jede Aktion einen Kommentar hinzufügen	Vor jedem Meldungsaufruf werden informative Protokollmeldungen eingefügt (nur andere als C). **Standardwert:** Aktiviert.
Bei Aufzeichnungsende alle AUT-Prozesse schließen	Alle AUT-Prozesse (Application Under Test) werden automatisch geschlossen, wenn VuGen die Aufzeichnung beendet. **Standardwert:** Deaktiviert.
Arrays korrelieren	Arrays aller Datentypen (Zeichenfolgen, Strukturen, Zahlen usw.) werden verfolgt und korreliert. **Standardwert:** Aktiviert.
Lange Datentypen korrelieren	Lange Datentypen (Ganzzahl, lange Ganzzahl, 64-Bit-Zeichen, Gleitkomma, doppelte Genauigkeit) werden korreliert. **Standardwert:** Deaktiviert.
Einfache Zeichenfolgen korrelieren	Einfache Nicht-Array-Zeichenfolgen und Ausdrücke werden korreliert. **Standardwert:** Deaktiviert.
Kurze Datentypen korrelieren	Kurze Datentypen (Byte, Zeichen, kurze Ganzzahl) werden korreliert. **Standardwert:** Deaktiviert.
Strukturen korrelieren	Komplexe Strukturen werden verfolgt und korreliert. **Standardwert:** Aktiviert.
Elementare Werte als lokal deklarieren	Variablen mit elementaren Werten werden als lokale Variablen, und nicht als Klassenvariablen, deklariert (nur C, C# und .NET). **Standardwert:** Aktiviert.
Explizite Deklaration von Varianten	Variantentypen werden explizit deklariert, um ByRef-Varianten zu handhaben (nur Visual Basic for Applications). **Standardwert:** Aktiviert.
Feste Latenzzeit nach Transaktionsende generieren	Nach dem Ende jeder Transaktion wird eine feste Latenzzeit (in Sekunden) eingefügt. Bei Aktivierung dieser Option können Sie einen Wert für die Latenzzeit angeben. **Standardwert:** Deaktiviert, wenn aktiviert: 3 Sekunden.
Protokoll mit aufgezeichneten Ereignissen generieren	Es wird ein Protokoll mit allen Ereignissen generiert, die während der Aufzeichnung stattgefunden haben. **Standardwert:** Deaktiviert.

Element der Oberfläche	Beschreibung
Latenzzeit generieren, wenn Schwellenwert überschritten wird	Für die Latenzzeit wird ein Schwellenwert verwendet. Unterschreitet die aufgezeichnete Latenzzeit den Schwellenwert, generiert VuGen keine Latenzzeitanweisung. Sie geben auch den Schwellenwert an. Der Standardwert beträgt 3. Wenn die Latenzzeit weniger als 3 Sekunden beträgt, generiert VuGen keine Latenzzeitanweisungen. Wenn Sie diese Option deaktivieren, generiert VuGen keine Latenzzeiten. **Standardwert:** Aktiviert, 3 Sekunden.
Ausgabeparameterwerte einfügen	Nach dem Aufruf werden Ausgabeparameterwerte eingefügt (nur C, C# und .NET). **Standardwert:** Deaktiviert.
Informationen nach Aufruf einfügen	Nach jedem Meldungsaufruf werden informative Protokollmeldungen eingefügt (nur andere als C). **Standardwert:** Aktiviert.
Maximale Anzahl von Zeilen in Aktionsdatei	Wenn die Anzahl der Aktionen den festgelegten Schwellenwert überschreitet, wird eine neue Datei erstellt. Der Standardschwellenwert beträgt 60000 Zeilen (nur C, C# und .NET). **Standardwert:** Deaktiviert.
Lange Zeichenfolgen durch Parameter ersetzen	Zeichenfolgen, die die maximale Länge überschreiten, werden in einem Parameter gespeichert. Die ursprüngliche maximale Länge beträgt 100 Zeichen. Die Parameter sowie die vollständigen Zeichenfolgen werden im folgenden Format in der Datei **lr_strings.h** im Ordner des jeweiligen Skripts gespeichert: **const char <paramName_uniqueID> ="string".** Diese Option ermöglicht eine bessere Skriptlesbarkeit. Sie hat keine Auswirkungen auf die Leistungsfähigkeit des Skripts. **Standardwert:** Aktiviert.
Variablen für elementare Rückgabewerte wiederverwenden	Die gleichen Variablen werden für elementare Werte wiederverwendet, die von Methodenaufrufen empfangen wurden. Hierdurch wird die Einstellung **Elementare Werte als lokal deklarieren** außer Kraft gesetzt. **Standardwert:** Aktiviert.
Prozesse protokollieren, die als lokale COM-Server erstellt wurden	Wenn einer der Unterprozesse als lokaler COM-Server erstellt wurde, wird die Aktivität der aufgezeichneten Anwendung verfolgt (nur C und COM). **Standardwert:** Aktiviert.
Vollständige Namen verwenden	Beim Deklarieren neuer Variablen wird der vollständige Name verwendet (nur C# und .NET). **Standardwert:** Deaktiviert.

Element der Oberfläche	Beschreibung
Hilfsfunktionen für Arrays verwenden	Mithilfe von Hilfsfunktionen werden Komponenten aus Varianten-Arrays extrahiert (nur Java und VB Scripting). **Standardwert:** Deaktiviert.
Hilfsfunktionen für Objekte verwenden	Mithilfe von Hilfsfunktionen werden Objektreferenzen aus Varianten extrahiert, wenn sie als Funktionsargumente übergeben werden (nur Java und VB Scripting). **Standardwert:** Deaktiviert.
Aufzeichnung für geschützte Anwendung verwenden	Verwenden Sie diese Option, wenn VuGen Ihre Anwendung nicht aufzeichnen kann. Möglicherweise blockiert Ihre Anwendung den Zugriff auf VuGen. Wenn Sie diese Option aktivieren, kann der Zugriff und die Aufzeichnung ermöglicht werden. **Standardwert:** Deaktiviert.
Warnung ausgeben, wenn bei der Anwendung, die aufgezeichnet wird, ein Fehler auftritt	Bei Auswahl dieser Option werden Sie von VuGen aufgefordert, die Aufzeichnung abzubrechen, wenn die aufgezeichnete Anwendung abstürzt oder über einen Zeitraum von 3 Minuten keine Ereignisse aufgezeichnet werden. Wenn Sie die Aufzeichnung abbrechen, wird kein Skript generiert. **Standardwert:** Aktiviert. **Hinweis:** Diese Option ist nur für Web (HTTP/HTML)-Vuser-Skripts verfügbar.

GUI-Eigenschaften > Webereigniskonfiguration (Knoten)

Ermöglicht das Festlegen der in einem Skript aufgezeichneten Detailebene (Webereignisaufzeichnung).

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > GUI-Eigenschaften > Webereigniskonfiguration**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung

Standard	• Zeichnet Klickereignisse bei Standardwebobjekten, wie Bildern, Schaltflächen und Optionsfeldern immer auf. • Zeichnet das Übergabeereignis innerhalb von Formularen immer auf. • Zeichnet Klickereignisse bei anderen Objekten mit einem verbundenen Handler oder Verhalten auf. • Zeichnet das Mouseover-Ereignis bei Bildern und Image-Maps nur auf, wenn das Ereignis nach dem Mouseover-Ereignis am selben Objekt ausgeführt wird.
Benutzerdefinierte Einstellungen	Öffnet das "Dialogfeld "Benutzerdefinierte Konfiguration der Aufzeichnung von Webereignissen"" unten, in dem Sie die Ereignisaufzeichnungskonfiguration anpassen können.
Hoch	Zusätzlich zu den auf der Ebene **Mittel** aufgezeichneten Objekten zeichnet diese Ebene Mouseover-, Mousedown- und Doppelklickereignisse bei Objekten mit verbundenen Handlern oder Verhalten auf.
Mittel	Zusätzlich zu den auf der Ebene **Standard** aufgezeichneten Objekten zeichnet diese Ebene Klickereignisse an den Tagobjekten <DIV>, <SPAN> und <TD> HTML auf.

Dialogfeld "Benutzerdefinierte Konfiguration der Aufzeichnung von Webereignissen"

Ermöglicht es Ihnen, die Ebene der Webereignisaufzeichnung anzupassen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > GUI-Eigenschaften > Webereigniskonfiguration > Benutzerdefinierte Einstellungen**
Wichtige Informationen	Dieses Dialogfeld steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Objektliste>	Eine Liste der Webobjekte. Alle Webobjekte können gemäß den anderen Einstellungen in diesem Dialogfeld angepasst werden.

Element der Oberfläche	Beschreibung
<Menü "Objekt">	• **Hinzufügen.** Fügt der Objektliste ein neues HTML-Tagobjekt hinzu. Geben Sie den Namen des Tags ein. • **Löschen.** Löscht ein Objekt aus der Objektliste.
Menü "Ereignis"	• **Hinzufügen.** Fügt in der Spalte **Ereignisname** dieses Objekts ein Ereignis hinzu. • **Löschen.** Löscht ein Ereignis aus der Spalte **Ereignisname** dieses Objekts.
Ereignisname	Eine Liste der dem Objekt zugeordneten Ereignisse.
Menü "Datei"	• **Konfiguration laden.** Lädt eine zuvor erstellte benutzerdefinierte Konfiguration. • **Konfiguration speichern unter.** Speichert die aktuelle Konfiguration.
Abhören	Die Kriterien, die festlegen, wann VuGen ein Ereignis abhört. • **Immer**. Hört das Ereignis immer ab. • **Bei Handler**. Hört das Ereignis ab, wenn ein Handler mit ihm verbunden ist. Ein Handler ist ein Code auf einer Webseite, normalerweise eine in einer Skriptsprache geschriebene Funktion oder Routine, die die Kontrolle erhält, wenn das entsprechende Ereignis auftritt. • **Bei Verhalten**. Hört das Ereignis ab, wenn ein DHTML-Verhalten mit ihm verbunden ist. Ein DHTML-Verhalten verkapselt eine spezifische Funktionalität oder ein spezifisches Verhalten auf einer Seite. Bei Anwendung auf ein Standard-HTML-Element auf einer Seite, verbessert ein Verhalten das Standardverhalten dieses Elements. • **Bei Handler oder Verhalten**. Hört das Ereignis ab, wenn ein Handler oder Verhalten mit ihm verbunden ist. • **Nie**. Hört das Ereignis nie ab. Weitere Informationen finden Sie unter "Tipps zum Arbeiten mit der Ereignisabhörung und -aufzeichnung" auf Seite 258.

Element der Oberfläche	Beschreibung
Aufzeichnen	Die Kriterien, die festlegen, wann VuGen ein Ereignis aufzeichnet. • **Aktiviert.** Zeichnet das Ereignis jedes Mal auf, wenn es bei einem Objekt auftritt, solange VuGen das Ereignis beim ausgewählten Objekt oder bei einem anderen Objekt, an das das Ereignis über das Event-Bubbling weitergeleitet wird, abhört. Event-Bubbling ist der Prozess, bei dem ein Ereignis, das bei einem untergeordneten Objekt auftritt, innerhalb des HTML-Codes in der Hierarchiereihe nach oben aufsteigen kann, bis es auf einen Ereignishandler trifft, der das Ereignis verarbeiten kann. • **Deaktiviert.** Zeichnet das angegebene Ereignis nicht auf und ignoriert gegebenenfalls Event-Bubbling. • **Aktiviert bei nächstem Ereignis.** Mit **Aktiviert** identisch, außer dass das Ereignis nur aufgezeichnet wird, wenn das nachfolgende Ereignis in demselben Objekt auftritt. Angenommen ein Mouseover-Verhalten ändert einen Bild-Link. Möglicherweise möchten Sie das Mouseover-Ereignis nicht jedes Mal aufzeichnen, wenn Sie die Maus über dieses Bild bewegen. Da nur das Bild, das nach dem Mouseover-Ereignis angezeigt wird, das Link-Ereignis aktiviert, ist es jedoch wichtig, dass das Mouseover-Ereignis vor einem Klickereignis beim selben Objekt aufgezeichnet wird. Weitere Informationen finden Sie unter "Tipps zum Arbeiten mit der Ereignisabhörung und -aufzeichnung" auf Seite 258.
Einstellungen zurücksetzen	Setzt die benutzerdefinierten Einstellungen auf die von Ihnen gewählten Einstellungen zurück: **Standard**, **Mittel** oder **Hoch**.

HTTP > Erweitert (Knoten)

Ermöglicht es Ihnen, die Codegenerierungsoptionen anzupassen, die sich auf Latenzzeit, Zurücksetzen von Kontexten, Speichern von Snapshots und das Generieren der Funktion **web_reg_find** beziehen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > HTTP-Eigenschaften > Erweitert**
Wichtige Informationen	• Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250. • Einige Optionen innerhalb dieses Knotens sind bei Verwendung eines Mehrfachprotokoll-Skripts nicht verfügbar.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Öffnet das "Dialogfeld "Kopfzeilen"" auf Seite 206.
	Öffnet das "Dialogfeld "Inhaltstypenfilter"" auf Seite 207.
	Öffnet das "Dialogfeld "Nicht-Ressourcen"" auf Seite 208.
Kontext für jede Aktion zurücksetzen	Setzt alle HTTP-Kontexte zwischen Aktionen zurück. Durch das Zurücksetzen von Kontexten kann ein neuer Benutzer, der eine Browsersitzung beginnt, mit dem Vuser genauer emuliert werden. Durch diese Option wird der HTML-Kontext so zurückgesetzt, dass zu Beginn der Aktion immer eine kontextlose Funktion aufgezeichnet wird. Außerdem wird der Zwischenspeicher geleert und Benutzernamen und Passwörter werden zurückgesetzt. **Hinweis:** Diese Option ist nur für Web- und Oracle NCA-Protokolle verfügbar.
Snapshot-Ressourcen lokal speichern	Speichert während der Aufzeichnung und der Wiedergabe eine lokale Kopie der Snapshot-Ressourcen, wodurch die Snapshots genauer erstellt und schneller angezeigt werden.
web_reg_find-Funktionen für Seitentitel generieren	Generiert **web_reg_find**-Funktionen für alle HTML-Seitentitel. VuGen fügt die Zeichenfolge aus dem Titeltag der Seite hinzu und verwendet sie als Argument für **web_reg_find**. • **web_reg_find-Funktionen für Subframes generieren.** Generiert **web_reg_find**-Funktionen für Seitentitel in allen Sub-Frames der aufgezeichneten Seite. **Hinweis:** Diese Option ist nur für Web- und Oracle NCA-Protokolle verfügbar.
Kommentare zu HTTP-Fehlern beim Aufzeichnen zu Skript hinzufügen	Fügt für jeden HTTP-Anforderungsfehler ein Kommentar im Skript hinzu. Anfragen gelten als fehlgeschlagen, wenn während der Aufzeichnung ein Serverantwortwert von 400 oder höher generiert wird.

Zeichensatz unterstützen	• **UTF-8.** Ermöglicht die Unterstützung von UTF-8-Kodierung. Dadurch konvertiert VuGen nicht-ASCII UTF-8-Zeichen in die Codierung des Computers, für den die jeweilige Landessprache konfiguriert wurde, damit die Zeichen in VuGen korrekt angezeigt werden können. Sie sollten diese Option nur für nicht englische, UTF-8-codierte Seiten aktivieren. Die Sprache der aufgezeichneten Site muss mit der Betriebssystemsprache übereinstimmen. Sie können keine nicht englischen Webseiten mit anderen Codierungen (z. B. UTF-8 zusammen mit ISO-8859-1 oder shift_jis) im selben Skript aufzeichnen. • **EUC-JP.** Wählen Sie bei Verwendung der japanischen Version von Windows diese Option aus, um Unterstützung für Websites mit der Zeichencodierung EUC-JP zu ermöglichen. VuGen konvertiert dann alle EUC-JP-Zeichenfolgen in die Codierung des Computers, für den die jeweilige Landessprache konfiguriert wurde, damit die Zeichen in VuGen korrekt angezeigt werden können. VuGen konvertiert alle EUC-JP-Zeichenfolgen (japanisches UNIX) in die SJIS-Codierung (japanisches Windows) des Computers, für den die jeweilige Landessprache konfiguriert wurde, und fügt eine Funktion **web_sjis_to_euc_param** zum Skript hinzu (nur Kanji).

<table>
<tr>
<td>Servernamen parametrieren</td>
<td>VuGen identifiziert Servernamen und IP-Adressen, wenn Sie ein Vuser-Skript erneut generieren. Diese Servernamen und IP-Adressen sind in bestimmten Argumenten enthalten, die speziellen Funktionen im Vuser-Skript zugeordnet sind. [Weitere Details finden Sie in der nachstehenden Tabelle.] VuGen ersetzt die identifizierten Servernamen und IP-Adressen mit Parametern. Durch die Parametrierung der Servernamen und IP-Adressen können Sie das Vuser-Skript in verschiedenen Umgebungen ausführen, indem Sie einfach die Werte für den Server und die IP-Adresse in der Parameterdatei ändern. Eine Einführung zu den Parametern finden Sie unter "Parametrierung – Übersicht" auf Seite 436.

Hinweis: Um Daten für die Parametrierung zu identifizieren, sucht VuGen die Argumente, die für die folgenden Funktionen aufgelistet sind:

<table>
<tr><th>API-Funktion</th><th>Argumente</th></tr>
<tr><td>web_url</td><td>• URL
• Referrer</td></tr>
<tr><td>web_custom_request</td><td>• URL
• Referrer</td></tr>
<tr><td>web_image</td><td>• URL
• Referrer</td></tr>
<tr><td>web_submit_data</td><td>• Aktion
• URL
• Referrer</td></tr>
<tr><td>web_submit_form</td><td>• Aktion
• URL
• Referrer</td></tr>
</table>

Diese Option ist standardmäßig deaktiviert.</td>
</tr>
</table>

LR-Proxy zum Aufzeichnen einer lokalen Anwendung verwenden	Bietet eine alternative Möglichkeit zum Aufzeichnen, wenn der standardmäßige VuGen-Aufzeichnungsmechanismus mit Ihrer Anwendung nicht kompatibel ist. Dieser wird angewendet, wenn Sie eine Webbrowser- oder Windows-Anwendung aufzeichnen möchten. **Hinweis:** Deaktivieren Sie diese Option nach dem Aufzeichnen, um den Standardmodus wiederherzustellen.
Streamingmodus beim Aufzeichnen mit LoadRunner-Proxy verwenden	Mithilfe des Streamingmodus können vom Server empfangene HTTP-Datenteile mittels Pufferung an die Anwendung weitergeleitet werden. Dies ermöglicht die Aufzeichnung von asynchroner Kommunikation des Typs "Push". **Hinweis:** Wenn diese Option aktiviert ist, ist die Symbolleiste für die Remoteaufzeichnung deaktiviert.

Dialogfeld "Kopfzeilen"

Ermöglicht es Ihnen, mit jeder HTTP-Anforderung an den Server automatisch zusätzliche Standard-HTTP-Kopfzeilen zu senden.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > HTTP > Erweitert > Kopfzeilen**
Wichtige Informationen	• Dieses Dialogfeld steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250. • Die folgenden Standardkopfzeilen gelten als riskant: Authorization, Connection, Content-Length, Cookie, Host, If-Modified-Since, Proxy-Authenticate, Proxy-Authorization, Proxy-Connection, Referer und WWW-Authenticate. Sie werden nur aufgezeichnet, wenn sie in der Kopfzeilenliste ausgewählt wurden.

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
	Plus. Fügt einen neuen Eintrag hinzu.
	Minus. Löscht einen Eintrag.
Liste wiederherst.	Stellt in der aktuellen Liste wieder die Standardwerte und -einträge her.

Forts.

Standardwerte	Stellt in allen Listen wieder die Standardwerte und -einträge her.
<Dropdownmenü>	Steuert die Optionen für dieses Dialogfeld: • **Kopfzeilen nicht aufzeichnen** • **Kopfzeilen aus Liste aufzeichnen** • **Kopfzeilen aufzeichnen, die nicht in Liste sind**
<Inhaltstypliste>	Eine Liste der Kopfzeilen, die aufgezeichnet werden bzw. nicht aufgezeichnet werden. Die Listen variieren je nach ausgewähltem Dropdownelement. Jedes Element kann mit dem zugehörigen Kontrollkästchen ausgewählt oder ausgeschlossen werden.

Dialogfeld "Inhaltstypenfilter"

Ermöglicht das Filtern von Inhaltstypen für ein aufgezeichnetes Skript. Sie können den Inhaltstyp festlegen, den Sie im Skript aufzeichnen oder ausschließen möchten.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > HTTP > Erweitert > Inhaltstypen**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
	Plus. Fügt einen neuen Eintrag hinzu.
	Minus. Löscht einen Eintrag.
Liste wiederherst.	Stellt in der aktuellen Liste wieder die Standardwerte und -einträge her.
Standardwerte	Stellt in allen Listen wieder die Standardwerte und -einträge her.

Forts.

<Dropdownmenü>	Steuert die Optionen für dieses Dialogfeld: • **Inhaltstypen nicht filtern.** • **Inhaltstypen in Liste ausschließen.** • **Inhaltstypen ausschließen, die nicht in Liste.**
<Inhaltstypliste>	Eine Liste der Inhaltstypen, die gefiltert werden bzw. nicht gefiltert werden. Die Listen variieren je nach ausgewähltem Dropdownelement. Jedes Element kann mit dem zugehörigen Kontrollkästchen ausgewählt oder ausgeschlossen werden.

Dialogfeld "Nicht-Ressourcen"

Beim Aufzeichnen eines Skripts gibt VuGen an, ob die Ressource während der Wiedergabe mit dem Attribut **Resource** der Funktion **web_url** abgerufen wird. Ist das Attribut **Resource** auf 0 gesetzt, wird die Ressource während der Skriptausführung abgerufen. Ist das Attribut **Resource** auf 1 gesetzt, überspringt der Vuser den Ressourcentyp.

Sie können ausschließen, dass bestimmte Inhaltstypen als Ressourcen behandelt werden. Sie können z. B. in VuGen festlegen, dass der Ressourcentyp **gif** nicht als Ressource behandelt und daher ohne Bedingungen heruntergeladen werden soll.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > HTTP > Erweitert > Nicht-Ressourcen**
Wichtige Informationen	Dieses Dialogfeld steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
	Hinzufügen. Fügt einen neuen Eintrag zur Liste hinzu.
	Entfernen. Löscht einen Eintrag aus der Liste.
	Stellt die Standardliste wieder her.
<Liste der Nicht-Ressourcen-Inhaltstypen>	Liste der Elemente, die nicht als Ressourcen aufgezeichnet werden sollten. Jedes Element kann mit dem zugehörigen Kontrollkästchen ausgewählt oder ausgeschlossen werden.

Flex > Konfiguration (Knoten) (Aufzeichnen)

Ermöglicht das Festlegen eines externen JVM-Pfads (Java Virtual Machine).

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Flex > Konfiguration**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Externe JVM verwenden	Ermöglicht die Verwendung einer externen JVM. Wenn Sie diese Option auswählen, müssen Sie den Pfad angeben: **Externer JVM-Pfad** Der vollständige Pfad der externen JVM. VuGen muss neu gestartet werden, damit die Änderungen wirksam werden.
GraniteDS-Konfiguration verwenden	Definiert die serverseitige Datendienstkonfiguration. Wenn Sie diese Option auswählen, wählen Sie nicht **Flex LCDS/BlazeDS-JAR-Dateien verwenden** aus, um die Nachrichten zu serialisieren. Stellen Sie sicher, dass die Datei "granit-config.xml" mit der auf dem Server bereitgestellten Datei übereinstimmt.
Maximal zulässige Größe der formatierten Anforderung/Antwort zur Ausgabe	Geben Sie hier die maximale Zeichenlänge für Anforderungen oder Antworttexte zur Ausgabe in die Protokolldateien ein. Diese Option wirkt sich nur auf die Schritte **flex_amf_call** und **flex_remoting_call** aus. Wenn Sie beispielsweise den Wert 1048576 Zeichen (1 MB) angeben, werden nur Antworten oder Anfragen mit einer Länge von weniger als einem Megabyte im Codegenerierungs- oder Wiedergabeprotokoll erfasst.

Flex > Externalisierbare Objekte (Knoten) (Aufzeichnen)

Mit diesem Dialogfeld können Sie konfigurieren, wie LoadRunner externalisierbare Objekte in Flex-Skripts behandelt.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Flex > Externalisierbare Objekte**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.
Relevante Aufgaben	• Serialisieren mit der externen Java-Serialisierungskomponente • "Serialisieren von Flex-Skripts " auf Seite 704
Siehe auch	• "Flex – Übersicht" auf Seite 682 • "Externalisierbare Objekte in Flex-Skripts" auf Seite 700

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Externalisierbare Objekte nicht serialisieren	Generiert Skripts unter Verwendung der Standardeinstellungen.
Serialisieren von Objekten mit	Wählen Sie eine geeignete Option aus: • Wählen Sie **LoadRunner AMF Serializer** aus, wenn Sie nicht den Adobe LiveCycle Data Services- oder Adobe BlazeDS-Server verwenden. • Wählen Sie **Benutzerdefinierte Java-Klassen** und dann eine oder beide der folgenden verfügbaren Optionen aus: ▪ Wählen Sie **Flex LCDS/BlazeDS-JAR-Dateien verwenden** aus, wenn Sie Flex LCDS- oder BlazeDS-JAR-Dateien für die Serialisierung der Nachrichten verwenden. Falls Sie **GraniteDS-Konfiguration verwenden** im Knoten **Konfiguration** ausgewählt haben, wählen Sie **Flex LCDS/BlazeDS-JAR-Dateien verwenden** nicht aus. ▪ Wählen Sie **Weitere JAR-Dateien verwenden** aus, um zusätzliche JAR-Dateien für die Serialisierung der Nachrichten hinzuzufügen. Sie müssen die JAR-Dateien vom Server kopieren und ihren Speicherort in der Liste **Klassenpfadeinträge** (siehe unten) angeben. Kopieren Sie nur die JAR-Dateien, die die externalisierbare Klasse enthalten. Außerdem müssen Sie sicherstellen, dass sich diese Dateien auf allen Lastgenerator-Computern am gleichen Speicherort befinden. Wenn Sie JAR-Dateien mit denselben Namen wie die Flex LCDS- oder Blaze DS-JAR-Dateien hinzufügen, indem Sie das erste Kontrollkästchen auswählen, werden diese Dateien überschrieben.

Liste der Klassenpfadeinträge	
	Nach unten. Verschiebt einen Klassenpfadeintrag in der Liste nach unten.
	Nach oben. Verschiebt einen Klassenpfadeintrag in der Liste nach oben.
	Klassenpfad hinzufügen. Fügt eine neue Zeile zur Klassenpfadliste hinzu.
	Klassenpfadordner hinzufügen. Fügt alle Dateien des Ordners zur Klassenpfadliste hinzu.
	Löschen. Entfernt einen Klassenpfad unwiderruflich.

.NET > Filter (Knoten)

Ermöglicht das Festlegen der Aufzeichnungsoptionen für .NET-Vuser-Skripts.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Microsoft .NET > Filter**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.
Siehe auch	• "Übersicht über .NET-Filter" auf Seite 796 • ".NET-Filter - Weiterführende Themen" auf Seite 797 • "Richtlinien für das Festlegen von .NET-Filtern" auf Seite 798

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Öffnet das Dialogfeld **Neuen Filter erstellen**, in dem Sie einen neuen Filter erstellen können. Weitere Informationen finden Sie unter "Dialogfeld "Neuen Filter erstellen" [.NET Protocol]" auf der nächsten Seite.
	Öffnet den Filter-Manager, in dem Sie alle Microsoft .NET-Protokollfilter anzeigen und ändern können. Weitere Informationen finden Sie unter "Filter-Manager [.NET Protocol]" auf der nächsten Seite.
Benutzerdefinierter Filter	Zeigt die Filter an, die Sie zuvor auf dem gleichen Computer erstellt haben.

Umgebungsfilter	Listet die vorhandenen Umgebungsfilter auf: .NET Remoting, ADO.NET, Enterprise Services und WCF (Windows Communication Foundation of Framework 3.0).
Neuer Filter	Gibt an, dass Sie einen neuen Filter erstellen möchten.

Dialogfeld "Neuen Filter erstellen" [.NET Protocol]

In diesem Dialogfeld können Sie einen neuen Filter für .NET-Vuser-Skripts erstellen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Microsoft .NET > Filter > Neuer Filter > Erstellen**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Auf Grundlage eines benutzerdefinierten Filters	Erstellt einen neuen Filter auf Grundlage eines benutzerdefinierten Filters. Den benutzerdefinierten Filter können Sie im Dropdownmenü auswählen.
Auf Grundlage eines Umgebungsfilters	Erstellt einen neuen Filter auf Grundlage eines Umgebungsfilters. Mit den Kontrollkästchen neben den Umgebungsfiltern können Sie angeben, auf welchen Umgebungsfiltern der Filter basieren soll.
Mit leerem Filter starten	Erstellt einen neuen Filter, der nicht auf einem bereits bestehenden Filter basiert.

Filter-Manager [.NET Protocol]

In diesem Dialogfeld können Sie .NET-Filter erstellen und bearbeiten.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Microsoft .NET > Filter > Umgebungsfilter / Benutzerdefinierter Filter > Filter-Manager**
Siehe auch	"Übersicht über .NET-Filter" auf Seite 796 ".NET-Filter - Weiterführende Themen" auf Seite 797 "Richtlinien für das Festlegen von .NET-Filtern" auf Seite 798

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
	Schließt das ausgewählte Element ein. Wenn Sie einen übergeordneten Knoten manuell einschließen, schließt der Filtermanager die jeweils untergeordneten Elemente ein, sofern es keine anderslautenden Regeln gibt. Wenn Sie beispielsweise eine Klasse einschließen, schließt der Filtermanager alle untergeordneten Methoden ein, sofern Sie keine Methode ausdrücklich ausgeschlossen haben.
	Schließt das ausgewählte Element aus. Die untergeordneten Elemente werden ebenfalls ausgeschlossen, sofern sie nicht aufgrund einer anderen Regel eingeschlossen werden. Wenn Sie eine **Klasse** ausschließen, wendet der Filter-Manager das Attribut **Ausschließen** auf die Klasse an, erlaubt jedoch der Aufzeichnungsengine das Aufzeichnen der Aktivität innerhalb der Methode der ausgeschlossenen Klasse. Wenn Sie allerdings eine **Methode** ausschließen, wendet der Filter-Manager das Attribut **Alle ausschließen** an, das die Aufzeichnungsengine am Aufzeichnen einer Aktivität innerhalb der Methoden der ausgeschlossenen Klasse hindert. Erfahrene Benutzer können diese Einstellung in der Filterdatei ändern.

	Entfernt die manuell eingegebene Ein- bzw. Ausschlussregel. In diesem Fall kann das Element von übergeordneten Elementen beeinflusst werden. Die Ein- und Ausschlussregeln haben folgende Eigenschaften: • Die Regeln sind hierarchisch strukturiert. Wenn Sie eine Regel in eine Klasse einschließen oder aus einer Klasse ausschließen, folgen die abgeleiteten Klassen der gleichen Regel, sofern keine anders lautenden Regeln vorgegeben wurden. • Die für eine Klasse geltende Regel wirkt sich nur auf die zu dieser Klasse gehörenden öffentlichen Methoden, abgeleiteten Klassen und inneren Klassen aus. • Die für einen Namespace geltende Regel wirkt sich auf alle Klassen sowie deren öffentlichen Methoden aus. • Beachten Sie, dass sich das Hinzufügen bzw. Entfernen von Assemblys nicht unbedingt auf die darin enthaltenen Klassen auswirkt. Sie können ein Assembly entfernen und die entsprechenden Methoden werden aufgrund der hierarchischen Struktur des Filters trotzdem aufgezeichnet. • Bei der Filtererstellung folgen einige Methoden, wie z. B. **.cctor()** und **Dispose(bool)**, nicht den hierarchischen Standardregeln. **Hinweis:** Durch das Rücksetzen eines übergeordneten Knotens wird das manuelle, auf den untergeordneten Knoten angewendete Ein- bzw. Ausschließen nicht überschrieben. Wenn Sie eine Methode beispielsweise manuell **ausschließen** und dann die entsprechende Klasse rücksetzen, die standardmäßig alle untergeordneten Knoten **einschließt**, bleibt Ihre Methode ausgeschlossen. Eigenschaften und Ereignisse können nur angezeigt werden und lassen sich über den Filtermanager weder ein- noch ausschließen. Außerdem sind mehrere systemspezifische Elemente geschützt und können dementsprechend nicht geändert werden. Tipps zum Ein- und Ausschließen von Elementen in einem Filter finden Sie unter "Richtlinien für das Festlegen von .NET-Filtern" auf Seite 798.
	Navigiert zum vorherigen oder nächsten, vom Benutzer besuchten Strukturknoten.

	Nicht öffentliche Elemente anzeigen. Standardmäßig zeigt der Filter-Manager nur öffentliche Klassen und Klassenmitglieder an. Durch Klicken auf die Schaltfläche werden auch nicht öffentliche Elemente in der Strukturansicht angezeigt. Wenn Sie eine Klasse einschließen, die nicht öffentliche Elemente enthält, werden diese nicht automatisch zum Filter hinzugefügt. Sie müssen jedes nicht öffentliche Element explizit in den Filter aufnehmen.
Auswirkungsprotokoll	Im **Auswirkungsprotokoll** werden die letzten Änderungen und ihre Auswirkung auf die Filter angezeigt. Die Benutzeraktionen erscheinen angefangen bei der letzten Änderung in absteigender Reihenfolge. Für jedes Element, das von einem manuellen Ein- oder Ausschließen betroffen ist, enthält das Protokoll Angaben zu den Auswirkungen des Elements. Ferner enthält das Protokoll ein Link zu dem entsprechenden Element im Filtermanager. Zum Anzeigen des Auswirkungsprotokolls klicken Sie in der Symbolleiste des Filter-Managers auf die Schaltfläche **Auswirkungsprotokoll** oder Sie wählen **Aktionen > Auswirkungsprotokoll anzeigen** im Fenster **Filter-Manager** aus.
<Filter-Manager-Struktur>	In der Filter-Manager-Struktur dienen Symbole zur Darstellung von Elementen und ihrem Status: Informationen zu den einzelnen Symbolen finden Sie in der Tabelle unten. • Elementsymbole verweisen auf den Elementtyp: Assembly, Namespace, Klasse, Methode, Struktur, Eigenschaft, Ereignisse oder Schnittstellen. • Ein Häkchen oder X neben dem Elementsymbol gibt an, ob ein Element ein- oder ausgeschlossen ist. • Ein fett gedruckt angezeigtes Element gibt an, ob das Element ausdrücklich ein- oder ausgeschlossen wurde. Das ist u. U. das Ergebnis eines manuellen Ein- bzw. Ausschließens durch den Benutzer oder einer vordefinierten Regel im Umgebungsfilter. Wenn Sie einen fett gedruckt angezeigten Knoten zurücksetzen, wird er in seinem ursprünglichen, nicht fett gedruckten Zustand angezeigt.
Verweis hinzufügen	Öffnet das Dialogfeld **Verweis hinzufügen** mit einer Liste von .NET Framework-Komponenten oder Assemblys. Weitere Informationen finden Sie unter "Dialogfeld "Verweis hinzufügen" [.NET Protocol]" auf der nächsten Seite.
Löschen	Löscht den ausgewählten angepassten Filter. Der Filtermanager fordert Sie zur Bestätigung auf.

Neu	Öffnet das Dialogfeld **Neuen Filter erstellen**, in dem Sie einen leeren Filter oder einen Filter anhand eines existierenden Filters erstellen können. Weitere Informationen finden Sie unter "Dialogfeld "Neuen Filter erstellen" [.NET Protocol]" auf Seite 212.
Verweis entfernen	Entfernt die Assembly, die im Filter-Manager ausgewählt ist, zusammen mit allen zugehörigen Elementen. Der Filtermanager fordert Sie zur Bestätigung auf.
Speichern	Speichert die an den Filtern vorgenommenen Änderungen.
Auswirkungsprotokoll anzeigen	Öffnet das Auswirkungsprotokoll für den ausgewählten Filter. Das **Auswirkungsprotokoll** zeigt die Knoten in der Struktur, die von den letzten Aktionen betroffen waren.

Die folgende Tabelle zeigt die Symbole der Filter-Manager-Struktur, die die verschiedenen Elemente repräsentieren:

Symbol	Beschreibung	Symbol	Beschreibung
	Assembly		Schnittstelle
	Assembly, das nicht geladen werden konnte		Methode
	Assembly, das teilweise geladen werden konnte		Statische Methode
	Klasse	{}	Namespace
	Konstruktor		Eigenschaft
	Statischer Konstruktor		Statische Eigenschaft
	Ereignis		Struktur
	Statisches Ereignis		

Dialogfeld "Verweis hinzufügen" [.NET Protocol]

In diesem Dialogfeld können Sie Verweise zu .NET-Filtern hinzufügen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Microsoft .NET > Filter > Umgebungsfilter / Benutzerdefinierter Filter > Filter-Manager > Verweis hinzufügen**

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Komponentenliste>	Eine Liste der .NET Framework-Komponenten oder Assemblys im Ordner **Public Assemblies**. • Um eines der aufgelisteten Elemente hinzuzufügen, markieren Sie das Element und klicken auf **Auswählen**. Durch Klicken bei gedrückter Strg-Taste können Sie mehrere Komponenten gleichzeitig auswählen. Im unteren Fensterbereich erscheinen die ausgewählten Referenzen. • Um ein Assembly hinzuzufügen, das nicht in der Liste erscheint, klicken Sie auf **Durchsuchen** und suchen in Ihrem Dateisystem oder in Ihrem Netzwerk nach der Referenz.
<Liste der ausgewählten Komponenten>	Die Liste der ausgewählten Komponenten. Die Spalte **Typ** enthält den Eintrag **.NET** für Komponenten aus dem Ordner **Public Assemblies** und den Eintrag **Datei** für Komponenten, die durch Auswahl von **Durchsuchen** hinzugefügt wurden. • Um das Element aus der Liste zu löschen, wählen Sie es im unteren Fensterbereich aus und klicken auf **Entfernen**.

GUI-Eigenschaften > Erweitert (Knoten)

Ermöglicht das Festlegen erweiterter Aufzeichnungsoptionen für Click & Script-Vuser.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > GUI-Eigenschaften > Erweitert**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Eigenschaften der Aufzeichnungseinstellungen

Element der Oberfläche	Beschreibung
Renderingbezogene Eigenschaftswerte aufzeichnen	Zeichnet die Werte der renderingbezogenen Eigenschaften von DOM-Objekten auf (z. B. **offsetTop**), sodass sie bei der Wiedergabe verwendet werden können. Beachten Sie, dass dies die Wiedergabegeschwindigkeit erheblich reduzieren kann. **Standardwert:** Deaktiviert.

'Klicken' nach Mausereignissen aufzeichnen	Zeichnet Mausklicks durch Erfassen der Mausereignisse anstatt Erfassen der click()-Methode auf. Aktivieren Sie diese Option, wenn die aufgezeichnete Anwendung die DOM click()-Methode verwendet, um die Generierung von mehreren Funktionen für dieselbe Benutzeraktion zu vermeiden. **Standardwert:** Aktiviert.
Daten der Socket-Ebene aufzeichnen	Aktiviert die Aufzeichnung von Daten auf Socketebene. Wenn Sie diese Option deaktivieren, müssen Sie den Anfangs-URL vor dem Aufzeichnen manuell hinzufügen. Außerdem können Sie das Skript in diesem Fall nicht auf einer HTML-Ebene neu generieren **Standardwert:** Aktiviert.
Snapshots für Ajax-Schritte generieren	Aktiviert die Generierung von Snapshots für Ajax-Schritte. Die Auswahl dieser Option kann zu Fehlern während der Aufzeichnung führen. **Standardwert:** Deaktiviert.

Eigenschaften der Codegenerierungseinstellungen

Element der Oberfläche	Beschreibung
Generierung von nicht kontextbezogenen Schitten aktivieren	Erstellt ein URL-basiertes Skript für ActiveX-Steuerelemente und Java-Applets, sodass sie wiedergegeben werden. Da diese Funktionen nicht Teil der nativen Aufzeichnung sind, werden sie als Aufzeichnung außerhalb des Kontexts bezeichnet. **Standardwert:** Deaktiviert.
Automatische Browsertitelprüfung aktivieren	Aktiviert die automatische Browsertitelprüfung. **Standardwert:** Deaktiviert.
Ausführen einer Titelprüfung für	• **jede Navigation.** Führt nur nach einer Navigation eine Titelverifizierung durch. Wenn ein Benutzer mehrere Operationen auf derselben Seite ausführt, wie z. B. Ausfüllen eines Formulars mit mehreren Feldern, bleibt der Titel unverändert und es ist keine Verifizierung erforderlich. • **jeden Schritt.** Führt für jeden Schritt eine Titelverifizierung durch, um sicherzustellen, dass kein Schritt den Browsertitel geändert hat. Ein geänderter Browsertitel kann zu einem Fehlschlagen des Skripts führen. • **Eine Titelüberprüfung mit dem URL ausführen, wenn der Titel fehlt.** Führt bei Browserfenstern ohne Titel für jeden Schritt eine Titelverifizierung mithilfe ihres URL durch.

.NET > Aufzeichnen - Knoten "Aufzeichnen"

Dieser Bildschirm ermöglicht das Festlegen der Aufzeichnungsoptionen für .NET-Vuser-Skripts.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Microsoft .NET > Aufzeichnen**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Codegenerierung	Ermöglicht es Ihnen anzugeben, ob Warnungen, eine Stapelverfolgung oder alle Ereignisabonnements während der Codegenerierung angezeigt werden sollen. • **Warnungen anzeigen.** Anzeigen von Warnmeldungen, die während des Codegenerierungsprozesses ausgegeben werden. • **Stapelverfolgung anzeigen.** Anzeigen der aufgezeichneten Stapelverfolgung, falls diese verfügbar ist. • **Alle Ereignisabonnements anzeigen.** Codegenerierung für alle aufgezeichneten Ereignisabonnements. Durch Deaktivieren dieser Option generiert VuGen nur Code für Ereignisse, bei denen sowohl der Veröffentlicher (das Objekt, das das Ereignis aufruft) als auch der Abonnent (das Objekt, das über das Ereignis informiert wird) im Filter enthalten sind. **Standardwert:** Deaktiviert.

Debugoptionen	Ermöglicht die Verfolgung des Staples und die Angabe seiner Größe. • **Stapelverfolgung.** Verfolgt die Inhalte des Stacks für jeden Aufruf im Skript. So können Sie bestimmen, welche Klassen und Methoden die Anwendung genutzt hat. Dies kann hilfreich sein, um die Referenzen, Namespaces, Klassen oder Methoden zu ermitteln, die der Filter abdecken muss. Das Aktivieren der Verfolgung kann die Anwendungsleistung während der Aufzeichnung beeinträchtigen. **Standardwert:** Deaktiviert. • **Grenzwert für die Stapelverfolgung.** Die maximale Anzahl von Aufrufen, die im Stack gespeichert werden. Wenn die Anzahl der Aufrufe diesen Wert überschreitet, nimmt VuGen eine Kürzung vor. **Standardwert:** 20 Aufrufe.
Filter	• **Alle Assemblys standardmäßig ignorieren (erweitert).** Ignorieren aller Assemblys, die vom ausgewählten Filter nicht explizit einbezogen werden. Wenn Sie diese Option deaktivieren, sucht VuGen nach einer passenden Filterregel für alle Assemblys, die während der Aufzeichnung geladen wurden.
Protokollierung	Mit den Protokollierungsoptionen können Sie die Detailebene für die Protokolldatei der Aufzeichnung festlegen. • **Protokollschweregrad.** Festlegen der Protokollierungsebene auf **Nur Fehler** (Standardeinstellung) oder **Debuggen**. Die Schweregradeinstellung gilt für alle Protokolle, die Sie weiter unten aktivieren. Sie sollten immer die Option **Nur Fehler** für Protokolle verwenden, außer wenn Sie vom HP Support explizit andere Anweisungen erhalten, da die Aufzeichnungszeit durch detaillierte Protokollierung erheblich ansteigen kann. • **Instrumentationsprotokoll.** Protokollieren von Meldungen in Bezug auf den Instrumentationsprozess. **Standardwert:** Aktiviert. • **Aufzeichnungprotokoll.** Protokollieren von Meldungen, die während der Aufzeichnung ausgegeben werden. **Standardwert:** Aktiviert. • **Codegenerierungsprotokoll.** Protokollieren von Meldungen, die während der Codegenerierungsphase ausgegeben werden. **Standardwert:** Aktiviert.
Remoteobjekte	Weitere Informationen zu dieser Eigenschaft finden Sie unter "Eigenschaft "Remoteobjekte"" auf der nächsten Seite.

Serialisierung	• **Serialisierungsformat.** Das Format der Serialisierungsdatei, die VuGen bei der Aufzeichnung einer Klasse mit Serialisierungsunterstützung erstellt: **Binär**, **XML** oder **Beide**. Der Vorteil des Binärformats besteht darin, dass es komprimierter und daher schneller ist. Nachteilig ist allerdings, dass Sie beim Binärformat die Daten nicht so bearbeiten können wie in XML. • **Lange Arrays serialisieren.** Für lange Arrays mit serialisierbaren Objekten (z. B. einen Array mit elementaren Objekten) können Sie den Serialisierungsmechanismus von VuGen verwenden. Durch das Aktivieren dieser Option wird der Aufruf **LrReplayUtils.GetSerializedObject** generiert, wenn die Arraygröße den Schwellenwert erreicht oder überschreitet. • **Schwellenwert für die Größe langer Arrays.** Der Schwellenwert für die Arraygröße, ab dem ein Array als **lang** angesehen wird. Wenn die Arraygröße diesen Wert erreicht oder überschreitet, wird der Array von VuGen serialisiert, wenn serialisierbare Objekte erkannt werden. **Tipp:** Für XML-Serialisierung können Sie den Inhalt der XML-Datei anzeigen. Zum Anzeigen der Datei wählen Sie im Kontextmenü die Option **XML-Datei anzeigen**.

Eigenschaft "Remoteobjekte"

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
In-Process-Objekte aufzeichnen	Aktivitäten zwischen Client und Server werden aufgezeichnet, wenn der Server im selben Prozess wie der Client gehostet wird. Da die Aktionen keinen echten Client/Server-Datenverkehr darstellen, sind sie in der Regel nicht von Interesse. Wenn prozessinterne Methoden relevant sind, wie z. B. bei bestimmten Enterprise Service-Anwendungen, können Sie diese Optionen aktivieren, um die Methoden zu erfassen. **Standardwert:** Deaktiviert.

Asynchrone Aufrufe	Gibt an, wie VuGen asynchrone Aufrufe von Remoteobjekten und deren Callback-Methoden verarbeiten soll. • **Ursprüngliche Callbacks standardmäßig aufrufen.** Beim Generieren und Wiedergeben des Skripts wird das ursprüngliche Callback der aufgezeichneten Anwendung verwendet. Wenn die Callback-Methode durch einen Filter explizit ausgeschlossen wird, gilt dieser Ausschluss selbst bei Aktivieren dieser Option. • **Asynchrone Callbacks generieren.** Diese Option bestimmt, wie Callbacks von VuGen verarbeitet werden, wenn die ursprünglichen Callbacks nicht aufgezeichnet wurden. Weitere Informationen finden Sie unter "Asynchrone Aufrufe" auf Seite 792.
WCF-Duplexbindung	• **Dummy-Callback-Handler generieren.** Ersetzt das ursprüngliche Callback bei der Duplexkommunikation durch ein Dummy-Callback, das die folgenden Aktionen durchführt: ▪ **Speichern von Argumenten.** Wenn der Server den Handler bei der Wiedergabe aufruft, speichert er die Methodenargumente in einem Schlüsselwert im Adressraum. ▪ **Synchronisieren bei der Wiedergabe.** Die Skriptausführung wird angehalten, bis die nächste Antwort eintrifft. VuGen platziert die Synchronisierung an der Stelle, an der das Callback während der Aufzeichnung aufgetreten ist. Dies wird im Skript durch eine Warnung angedeutet: • **Eindeutige Client-Basisadresse generieren.** Wenn die Anwendung duale HTTP-Bindung nutzt, gilt Folgendes: Da HTTP an sich kein Duplexprotokoll ist, verwendet das Framework einen Standardport für den Empfang von Antwortdaten, die an das Callback übergeben werden. Das Ausführen mehrerer Instanzen der Anwendung über dieselbe Portnummer schlägt möglicherweise fehl. Diese Option ersetzt die ursprüngliche Portnummer der Client-Basisadresse durch einen eindeutigen Port. Hintergrundinformationen zur WCF-Duplexbindung finden Sie unter "Aufzeichnen von WCF-Duplexkommunikation" auf Seite 788.

.NET > Gemeinsam genutzte DLLs

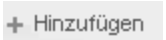
In diesem Dialogfeld können Sie die Liste der gemeinsam genutzten DLLs angeben, bevor Sie ein Vuser-Skript aufzeichnen. Wenn eine DLL in der Liste der gemeinsam genutzten DLLs vorhanden ist, greift der Vuser auf die DLL am freigegebenen Speicherort zu, sobald das Vuser-Skript ausgeführt und die DLL benötigt wird. Die DLL wird nicht auf den Lastgenerator kopiert. Sie können somit bei einer Vuser-Ausführung Festplattenspeicherplatz auf dem Lastgenerator sparen, wenn eine DLL zur Liste der gemeinsam genutzten DLLs hinzugefügt wird.

Hinweis: Der angegebene Speicherort einer gemeinsam genutzten DLL muss von allen Lastgeneratoren, auf denen der Vuser ausgeführt wird, aufgerufen werden können.

Nach der Aufzeichnung eines Vuser-Skripts wird die Liste der gemeinsam genutzten DLLs aus den Aufzeichnungsoptionen in die Laufzeiteinstellungen kopiert. Weitere Informationen zum Anzeigen und Ändern der Laufzeiteinstellungen finden Sie unter ".NET > Gemeinsam genutzte DLLs" auf Seite 393.

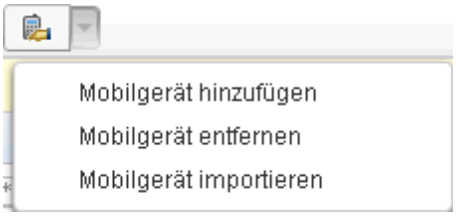
Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Microsoft .NET > Gemeinsam genutzte DLLs**
Siehe auch	".NET > Gemeinsam genutzte DLLs" auf Seite 393

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
DLL-Einträge	Die Liste der gemeinsam genutzten DLLs, auf die VuGen zugreift, während das Vuser-Skript aufgezeichnet wird. Die Reihenfolge, in der die DLLs angezeigt werden, ist von Bedeutung. Wenn eine bestimmte DLL erforderlich ist, greift VuGen auf die erste Instanz der DLL in der Liste zu. Wenn eine bestimmte DLL aktuell nicht verfügbar ist, deaktivieren Sie das Kontrollkästchen links neben dem DLL-Eintrag. Die DLL verbleibt in der Liste der freigegebenen DLLs. Um die DLL zu aktivieren, aktivieren Sie das Kontrollkästchen links neben dem DLL-Eintrag.
	Nach unten. Verschiebt den ausgewählten DLL-Eintrag in der Liste nach unten.
	Nach oben. Verschiebt den ausgewählten DLL-Eintrag in der Liste nach oben.
	**DLL hinzufügen.** Ermöglicht das Hinzufügen einer DLL zur Liste der gemeinsam genutzten DLLs.
	DLL-Ordner hinzufügen. Diese Option ist immer deaktiviert.
	Löschen. Entfernt die ausgewählte DLL aus der Liste der gemeinsam genutzten DLLs.

Eigenschaften von Mobile TruClient > Knoten "Mobilgerät"

In diesem Bereich können Sie die Eigenschaften eines Mobilgeräts auswählen, wenn Sie ein TruClient Ajax - Mobile-Skript aufzeichnen.

Zugriff	• **VuGen > Aufzeichnungsoptionen > TruClient Ajax - MobileEigenschaften > Mobilgerät** • Wählen Sie in der VuGen-Hauptsymbolleiste die Schaltfläche **Mobilgerät**. 
Relevante Aufgaben	"Hinzufügen, Entfernen und Importieren von Mobilgeräteeinstellungen für Mobile TruClient" auf Seite 783 "Aufzeichnen eines Skripts mit Mobile TruClient" auf Seite 783

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Mobilgerät	Wählen Sie den mobilen Gerätetyp aus, den Sie testen möchten.
Benutzer-Agent:	Geben Sie die Kopfzeilenzeichenfolge an, die an den Server gesendet wird, um das Mobilgerät zu identifizieren. Nach der Auswahl eines Geräts wird der Standard-Kopfzeilenwert angezeigt. Diese Kopfzeile kann jedoch geändert werden.
Anzeigen	Geben Sie die Höhe und Breite des Mobilgerätebildschirms an. Mobile TruClient öffnet ein Browserfenster entsprechend den Displayeinstellungen.

Netzwerk > Portzuordnung (Knoten)

Ermöglicht das Festlegen der Portzuordnung für die Aufzeichnungsoptionen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Netzwerk > Portzuordnung**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
Neu	Öffnet das Dialogfeld **Servereintrag**, in dem eine neue Zuordnung hinzugefügt werden kann. Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Servereintrag"" auf der nächsten Seite.
Bearbeiten	Öffnet das Dialogfeld **Servereintrag**, in dem Sie den ausgewählten Eintrag bearbeiten können. Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Servereintrag"" auf der nächsten Seite.
Optionen...	Öffnet das Dialogfeld **Erweiterte Einstellungen für die Portzuordnung**, in dem Sie die automatische Ermittlung des Kommunikationsprotokolls und der SSL-Ebene aktivieren können. Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Erweiterte Einstellungen für die Portzuordnung"" auf Seite 228.
<Liste der Portzuordnungen>	Eine Liste der Portzuordnungen. Durch Klicken auf das Kontrollkästchen neben einem Eintrag können Sie diesen vorübergehend deaktivieren. Wenn Sie einen Eintrag deaktivieren, ignoriert VuGen den gesamten Verkehr an dieser Server:Port-Kombination. Ein Port-Eintrag sollte deaktiviert werden, wenn die entsprechenden Daten nicht relevant sind oder das Protokoll nicht unterstützt wird.
Aufzeichnungsebene	Die Ebene, auf der die Daten aufgezeichnet werden (nur für HTTP-basierte Protokolle relevant): • **Daten auf Socket-Ebene.** Aufzeichnen von Daten durch Abfangen auf der Socket-Ebene. In diesem Fall gelten die Port-Zuordnungen (Standardeinstellung). • **Daten auf WinINet-Ebene.** Aufzeichnen von Daten unter Verwendung von Hooks auf die in bestimmten HTTP-Anwendungen eingesetzte **WinINet.dll**-API. Die gängigste Anwendung, die solche Hooks verwendet, ist Internet Explorer. Port-Zuordnungen sind für diese Ebene nicht relevant. • **Daten auf Socket-Ebene und WinINet-Ebene.** Zeichnet Daten unter Anwendung beider Methoden auf. Die WinINet-Ebene sendet Informationen für Anwendungen, in denen die **WinINet.dll** eingesetzt wird. Die Socket-Ebene sendet nur dann Daten, wenn sie ermittelt hat, dass diese nicht aus der **WinINet.dll** stammen. Die Port-Zuordnung gilt für Daten, die nicht aus der **WinINet.dll** stammen.

Serveradresszuordnungen auf Netzwerkebene für	Gibt die Zuordnungen pro Protokoll an. Wenn Sie beispielsweise nur die FTP-Zuordnungen anzeigen möchten, wählen Sie **FTP** aus.

Dialogfeld "Servereintrag"

Ermöglicht die Definition eines Servers aus der Serverliste im Knoten **Netzwerk - Portzuordnung**.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Netzwerk > Portzuordnung > Neuer Eintrag / Eintrag bearbeiten**
Wichtige Informationen	Dieses Dialogfeld steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Abschnitt	Element der Oberfläche	Beschreibung
Socket-Dienst	**Zielserver**	Die IP-Adresse oder der Hostname des Zielservers, für den der Eintrag gilt. **Standardwert:** Alle Server.
	Dienst-ID	Ein Protokoll oder Dienstname, anhand dessen der Recorder den Verbindungstyp identifiziert (z. B. HTTP, FTP usw.). Sie können auch einen neuen Namen angeben. Der Name darf nicht mehr als 8 Zeichen umfassen.
	Aufzeichnungstyp	Der Aufzeichnungstyp: direkt oder über einen Proxy-Server.

	Port	Der Port des Zielservers, für den der Eintrag gilt. Der Wert **0** steht für alle Ports. Wenn Sie nicht alle Port- und Servernamen angeben, verwendet VuGen die folgenden Prioritäten, um die Daten einem Dienst zuzuordnen: • Priorität 1: Port und Server angegeben • Priorität 2: Port nicht angegeben, Server angegeben • Priorität 3: Port angegeben, Server nicht angegeben • Priorität 4: Port und Server nicht angegeben Ein Zuordnungseintrag mit hoher Priorität kann nicht von einem Eintrag mit niedrigerer Priorität außer Kraft gesetzt werden. Wenn Sie beispielsweise festlegen, dass der Verkehr auf Port 25 des Servers **twilight** als SMTP behandelt wird und anschließend angeben, dass alle Server auf Port 25 als HTTP verarbeitet werden sollen, werden die Daten als SMTP behandelt. • **Erzwungene Zuordnung**. Wenn Sie eine Zuordnung für eine Portnummer, einen Servernamen oder eine Server:Port-Kombination festlegen, zwingt VuGen den Netzwerkservice, diesen Dienst zu verwenden. Wenn Sie beispielsweise angeben, dass jeder Server auf Port 80 FTP verwenden soll, setzt VuGen das FTP-Protokoll bei der Aufzeichnung dieser Kommunikation ein, auch wenn die tatsächliche Kommunikation womöglich in HTTP ist. In diesem Fall ist das Vuser-Skript möglicherweise leer.
	Diensttyp	Der Diensttyp, derzeit auf TCP eingestellt.
	Verbindungstyp	Die Sicherheitsebene der Verbindung: **Einfach** (nicht gesichert), **SSL** oder **Auto**. Bei Auswahl von **Auto**, prüft der Recorder die ersten 4 Byte auf eine SSL-Signatur. Wenn er eine SSL-Signatur erkennt, nimmt er an, dass SSL verwendet wird.

SSL-Konfiguration	**SSL-Version**	Die bevorzugte SSL-Version für die Kommunikation mit der Clientanwendung und dem Server. **Standardwert:** SSL 2/3. Einige Dienste benötigen jedoch ausschließlich SSL 3.0 bzw. ausschließlich SSL 2.0. Für bestimmte drahtlose Anwendungen ist der Transport Layer Security-Algorithmus TLS 1.x erforderlich.
	SSL-Verschlüsselungen	Die bevorzugte SSL-Verschlüsselung für die Verbindung mit einem sicheren Remoteserver.
	Das angegebene clientseitige Zertifikat verwenden	Das clientseitige Standardzertifikat für die Verbindung mit einem Remoteserver. Geben Sie ein Zertifikat an oder suchen Sie in den Formaten **txt**, **crt** oder **pem** nach einem Zertifikat und geben Sie ein Passwort an.
	Das angegebene Proxyserverzertifikat verwenden	Das Standardserverzertifikat, das Clientanwendungen vorgelegt wird, die ein Serverzertifikat anfordern. Geben Sie ein Zertifikat an oder suchen Sie in den Formaten **txt**, **crt** oder **pem** nach einem Zertifikat und geben Sie ein Passwort an. Klicken Sie auf **SSL testen**, um die Authentifizierungsdaten mit den Serverdaten zu vergleichen.

Dialogfeld "Erweiterte Einstellungen für die Portzuordnung"

Ermöglicht das Festlegen der erweiterten Einstellungen für die Portzuordnung. Weitere Informationen finden Sie unter "Automatische Erkennung der Portzuordnung" auf Seite 254.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Netzwerk > Portzuordnung > Optionen**
Wichtige Informationen	Dieses Dialogfeld steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung

Automatische SSL-Erkennung zulassen	Dient zur automatischen Erkennung von SSL-Kommunikationen. Geben Sie die Version und die Standardverschlüsselung an, die erkannt werden sollen. Beachten Sie, dass dies nur für Port-Zuordnungen gilt, für die im Feld **Verbindungstyp** der Wert **Auto** festgelegt wurde, oder für Zuordnungen, die nicht definiert wurden. Wurde für einen Server, Port oder eine Server:Port-Kombination der Wert **Einfach** oder **SSL** definiert, kann die automatische SSL-Erkennung nicht angewendet werden.
Automatische Erkennung SOCKET-basierter Kommunikation aktivieren	Dient zur automatischen Erkennung des Kommunikationstyps. Erhöhen Sie ggf. schrittweise die maximale Anzahl der Transitionen, bis VuGen das Protokoll erkennen kann. Sie können auch die maximale Puffergröße nach und nach um je 1024 Byte (1 KB) erhöhen, bis VuGen das Protokoll erkennen kann. Dies ermöglicht VuGen, eine größere Datenmenge auf eine Signatur zu prüfen.
Protokollebene	Legt die Protokollebene für die automatische Socket-Erkennung fest.

RDP > Codegenerierung > Erweitert (Knoten)

Ermöglicht es Ihnen zu steuern, wie VuGen ein RDP-Skript erstellt. Das Ändern dieser Einstellungen wird nur fortgeschrittenen Benutzern empfohlen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > RDP > Codegenerierung - Erweitert**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Zwischenablageparameter korrelieren	Durch diese Option wird der vom Benutzer gesendete aufgezeichnete Zwischenablagetext durch den korrelierten Parameter ersetzt, der denselben Text enthält, wie vom Server empfangen.
Zeitüberschreitung für Doppelklick (ms)	Die maximale Zeit (in Millisekunden), die zwischen zwei aufeinanderfolgenden Mausklicks vergehen darf, damit diese als Doppelklick angesehen werden. **Standardwert:** 500 Millisekunden.
Präfix für Zwischenablageparameter	Das Präfix für Zwischenablageparameter, die im aktuellen Skript generiert werden. Diese Option eignet sich beim Zusammenführen von Skripts, da Sie für jedes Skript ein eigenes Präfix festlegen können. **Standardwert: ClipboardDataParam_**.

Präfix für Snapshotnamen	Das Präfix für Dateinamen von Snapshots, die im aktuellen Skript generiert werden. Diese Option eignet sich beim Zusammenführen von Skripts, da Sie für jedes Skript ein eigenes Präfix festlegen können. **Standardwert: snapshot_**.

RDP > Codegenerierung > Agent (Knoten)

Ermöglicht Ihnen zu steuern, wie sich der Microsoft Terminal Server-Agent während der Aufzeichnung mit VuGen verhält.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > RDP > Codegenerierung - Agent**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
RDP-Agent verwenden	Das Skript wird mithilfe von Informationen generiert, die der RDP-Agent erfasst. Der LoadRunner RDP-Agent muss auf dem Server installiert sein. Weitere Informationen finden Sie unter "Installieren/Deinstallieren des RDP-Agenten" auf Seite 828. **Hinweis:** Zum Verwenden dieser Funktion müssen Sie die Laufzeiteinstellungen für **Mit RDP-Agent wiedergeben** aktivieren. Weitere Informationen finden Sie unter "RDP > RDP-Agent (Knoten)" auf Seite 398.
RDP-Agentenprotokoll aktivieren	Aktivieren des RDP-Agentenprotokolls. • **Detailstufe für RDP-Agentenprotokoll.** Konfigurieren der Detailebene des RDP-Agentenprotokolls; dabei steht **Standard** für die niedrigste Detailebene und **Extended Debug** für die höchste Detailebene. • **Ziel für RDP-Agentenprotokoll.** Konfigurieren des Ziels für die RDP-Agentenprotokolldaten. Mit der Option **File** werden die Protokollmeldungen nur auf dem Remoteserver gespeichert. Mit der Option **Stream** werden die Protokollmeldungen an den VuGen-Computer gesendet. Mit der Option **Datei und Datenstrom** werden die Protokollmeldungen an beide Ziele gesendet. • **Ordner für RDP-Agentenprotokoll.** Der Verzeichnispfad auf dem Remoteserver, in dem die RDP-Agentenprotokolldatei generiert wird.

RDP > Codegenerierung > Grundlegend (Knoten)

Ermöglicht das Steuern der Skripterstellung in VuGen, z. B. Detailebene, Auslöser und Zeitüberschreitungen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > RDP > Codegenerierung - Grundlegend**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Verbindungsnamen immer generieren	Ist diese Option aktiviert, enthält der Funktionsaufruf den Parameter **ConnectionName**. Ist diese Option nicht aktiviert, enthalten die Funktionen diesen Parameter nur, wenn **rdp_ connect_server** mehrmals im Skript vorkommt. **Standardwert:** Deaktiviert.
Automatische Generierung von Synchronisierungspunkten	Durch Synchronisierungspunkte kann die Wiedergabe des Skripts unterbrochen werden, damit durch das Öffnen eines Fensters oder Dialogs oder eines anderen Steuerelements eine bestimmte Bedingung erfüllt wird. Durch diese Option wird automatisch eine Funktion **sync_on_image** vor Mausklicks und Mausziehbewegungen generiert (standardmäßig aktiviert). **Sync radius** steht für den Abstand der Mausbewegung von den Seiten des Rechtecks, das den Synchronisierungsbereich definiert. Die Standardeinstellung beträgt 20 Pixel. Wählen Sie eine der folgenden Optionen: • **Keine.** Synchronisierungspunkte werden nicht automatisch hinzugefügt. • **Rechteck.** Synchronisierungspunkte werden als Rechtecke erzeugt, deren Mittelpunkt die Klick- oder Ziehposition bildet. • **Erweitert.** Synchronisierungspunkte werden so erzeugt, dass nur die gewünschte Position (z. B. eine Schaltfläche) ausgewählt wird und dass auf Änderungen der Benutzeroberfläche reagiert wird (z. B. Verschieben der Schaltfläche). Wird ein Synchronisierungsbereich nicht erkannt, werden die Einstellungen für rechteckige Synchronisierung verwendet.

Element der Oberfläche	Beschreibung
Aufrufe bei Mausbewegungen generieren	Generieren von Aufrufen vom Typ **rdp_mouse_move** im Skript. Durch das Aktivieren dieser Option nimmt die Skriptgröße deutlich zu. **Standardwert:** Deaktiviert.
Unformatierte Tastaturaufrufe generieren	Generieren von **rdp_raw_key_up/down**-Aufrufen, als ob die Skriptebene **Rohdaten** festgelegt wäre. Mausaufrufe werden weiterhin gemäß der Skriptebene generiert. Ist diese Option deaktiviert, generiert VuGen die Tastaturaufrufe gemäß der Skriptebene. Lautet die Skriptebene **Rohdaten**, wird diese Option ignoriert. **Standardwert:** Deaktiviert.
Unformatierte Mausaufrufe generieren	Generieren von **rdp_mouse_button_up/down**-Aufrufen, als ob die Skriptebene **Rohdaten** festgelegt wäre. Tastaturaufrufe werden weiterhin gemäß der Skriptebene generiert. Ist diese Option deaktiviert, generiert VuGen die Mausaufrufe gemäß der Skriptebene. Lautet die Skriptebene **Rohdaten**, wird diese Option ignoriert. **Standardwert:** Deaktiviert.
Skriptgenerierungsebene	Die Skriptebene und der Typ der API-Funktionen zum Generieren des Skripts. • **High.** Generieren von Skripts auf hoher Ebene. Tastaturereignisse werden in einen Aufruf **rdp_type** übersetzt. Zwei aufeinanderfolgende Mausklicks mit denselben Koordinaten werden in einen Doppelklick übersetzt. • **Low.** Generieren von Skripts auf niedriger Ebene. Das Drücken/Loslassen von Tasten wird in ein Ereignis **rdp_key** übersetzt. Die Alt-, Strg- und Umschalttasten werden als Parameter **KeyModifier** für andere Funktionen verwendet. Das Drücken/Loslassen/Bewegen der Maus wird in Mausklick- bzw. Mausziehereignisse übersetzt. • **Rohdaten.** Generieren eines Skripts auf Rohebene, indem Eingabeereignisse aus Netzwerkpuffern extrahiert und einfachste Aufrufe generiert werden: Taste drücken/loslassen, Maus drücken/loslassen/bewegen. Der Parameter **KeyModifier** wird nicht verwendet.

RDP > Clientstart (Knoten)

Ermöglicht das Festlegen der Optionen für den RDP-Clientstart.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > RDP > Clientstart**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
RDP-Clientanwendung ausführen	Stellt durch Ausführen des Terminaldienste-Clients eine Verbindung zum Terminalserver her.
Benutzerdefinierte Verbindungsdatei verwenden	Stellt mithilfe einer vorhandenen Verbindungsdatei eine Verbindung zum Terminalserver her. Die Datei muss die Erweiterung ***.rdp** aufweisen. Sie können das Dateisystem oder das Netzwerk nach der Datei durchsuchen.
Standardverbindungsdatei verwenden	Stellt mithilfe der Datei **Default.rdp** in Ihrem Dokumentverzeichnis eine Verbindung zum Terminalserver her.

Aufzeichnungseigenschaften > Corba-Optionen (Knoten)

Über diese Optionen können Sie die CORBA-spezifischen Aufzeichnungseigenschaften und verschiedene Callback-Optionen festlegen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Aufzeichnungseigenschaften > Corba-Optionen**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Rückrufverbindung aufzeichnen	Legt fest, dass VuGen für jedes Callback-Objekt eine Verbindungsanweisung für die Verbindung mit dem ORB generieren soll. **Standardwert:** Deaktiviert.
Nur DLL aufzeichnen	Legt fest, dass VuGen nur auf einer DLL-Ebene aufzeichnen soll. **Standardwert:** Deaktiviert.
Eigenschaften aufzeichnen	Legt fest, dass VuGen System- und angepasste Eigenschaften in Verbindung mit dem Protokoll aufzeichnen soll. **Standardwert:** Aktiviert.

CORBA-Objekte auflösen	Wenn es der Korrelation nicht gelingt, ein CORBA-Objekt aufzulösen, erstellen Sie es erneut unter Verwendung seiner binären Daten. **Standardwert:** Deaktiviert.
IDL-Konstrukte anzeigen	Zeigt das IDL-Konstrukt an, das bei der Übergabe als ein Parameter an einen CORBA-Aufruf verwendet wird. **Standardwert:** Aktiviert.
Lokale Anbieterklassen verwenden	Verwendet die lokalen Anbieterklassen und fügt den Ordner **srv** zum **BOOT**-Klassenpfad hinzu. Wenn Sie diese Option deaktivieren, verwendet VuGen Netzwerkklassen und fügt die Klassen des Skripts zum Klassenpfad hinzu. **Standardwert:** Aktiviert.
Anbieter	Die CORBA-Anbieter: **Inprise Visibroker**, **Iona OrbixWeb** oder **Bea Weblogic**.

Aufzeichnungseigenschaften > Korrelationsoptionen (Knoten)

Über diese Optionen können Sie die automatische Korrelation aktivieren und ihre Tiefe steuern.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Aufzeichnungseigenschaften > Korrelationsoptionen**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Erweiterte Korrelation	Aktiviert die Korrelation für komplexe Objekte, z. B. Arrays und CORBA-Containerkonstrukte und -Arrays. Diese Art der Korrelation wird auch als tiefe Korrelation bezeichnet. **Standardwert:** Aktiviert.
Auflistungstyp korrelieren	Korreliert Objekte aus der Collection-Klasse für JDK 1.2 oder höher. **Standardwert:** Deaktiviert.
Zeichenfolgenarrays korrelieren	Korreliert Zeichenfolgen innerhalb von Zeichenfolgenarrays während der Aufzeichnung. Bei deaktivierter Option werden Zeichenfolgen in Arrays nicht korreliert und das Skript enthält die tatsächlichen Werte. **Standardwert:** Aktiviert.

Zeichenfolgen korrelieren	Korreliert Zeichenfolgen während der Aufzeichnung im Skript. Bei deaktivierter Option werden die aufgezeichneten Werte im Skript in Anführungszeichen gefasst und alle anderen Korrelationsoptionen werden ignoriert. **Standardwert:** Deaktiviert.
Korrelationsgrad	Gibt den Grad der tiefen Korrelation und die Anzahl der zu prüfenden inneren Container an. **Standardwert:** 15.

Aufzeichnungseigenschaften > Protokolloptionen (Knoten)

Über diese Optionen können Sie die Ebene der während der Aufzeichnung generierten Debuginformationen festlegen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Aufzeichnungseigenschaften > Protokolloptionen**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Klassensicherung	Erstellt ein Abbild aller geladenen Klassen im Skriptordner. **Standardwert:** Deaktiviert.
Digest-Berechnung	Generiert einen Digest aller aufgezeichneten Objekte. **Standardwert:** Deaktiviert. • **Aus Digest ausschließen**. Eine Liste von nicht in die Digest-Berechnung aufzunehmenden Objekte. **Syntax:** java.lang.Object class format, delimiter = ","

Protokollebene	Die Ebene des zu generierenden Aufzeichnungsprotokolls: • **None**. Es wird keine Protokolldatei erstellt. • **Brief**. Generiert ein Standardaufzeichnungsprotokoll und eine Umleitung der Ausgabe. • **Detailed**. Generiert ein ausführliches Protokoll für Methoden, Argumente und Rückgabewerte. • **Debug.** Zeichnet Hooking- und Aufzeichnungsdebuginformationen zusammen mit allen oben genannten Informationen auf.
Threads synchronisieren	Legt fest, dass VuGen für Multithreadanwendungen eine Synchronisierung zwischen den verschiedenen Threads durchführen soll. **Standardwert:** Deaktiviert.

Aufzeichnungseigenschaften > Recorder-Optionen (Knoten)

Über diese Optionen können Sie das aufzuzeichnende Java-Protokoll sowie weitere protokollspezifische Aufzeichnungsoptionen festlegen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Aufzeichnungseigenschaften > Recorder-Optionen**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Byte-Arrayformat	Das Format von Bytearrays in einem Skript: Regular, Unfolded Serialized Objects oder Folded Serialized Objects. Verwendet beim Aufzeichnen von sehr langen Bytearrays eine der serialisierten Objektoptionen. **Standardwert:** Standard.
Bytes als Zeichen	Zeigt lesbare Zeichen als Zeichen mit der erforderlichen Umwandlung an - nicht in Byte- oder hexadezimaler Form. **Standardwert:** Aktiviert.

Element der Oberfläche	Beschreibung
Kommentieren von Zeilen mit	Kommentiert alle Zeilen im Skript aus, die eine der angegebenen Zeichenfolgen enthalten. Um mehrere Zeichenfolgen anzugeben, trennen Sie die Einträge mit Kommata. **Standardwert:** Jede Zeile mit einer Zeichenfolge, die <undefined> enthält, wird auskommentiert.
Erweiterungsliste	Eine mit Kommas getrennte Liste aller unterstützten Erweiterungen. Jede Erweiterung hat ihre eigene Hook-Datei. **Standardwert:** JNDI.
Funktionsprüfung einfügen	Fügt einen Verifizierungscode ein, der den während der Wiedergabe erhaltenen Rückgabewert mit dem während der Aufzeichnung generierten erwarteten Rückgabewert vergleicht. Diese Option gilt nur für elementare Rückgabewerte. **Standardwert:** Deaktiviert.
Übergeordnete Klasse vor der Klasse laden	Ändert die Reihenfolge des Ladens, sodass übergeordnete Klassen vor untergeordneten Klassen geladen werden. Dies unterstützt bei der Identifizierung von Hooking für Strukturen mit tiefer Vererbung **Standardwert:** Aktiviert.
LoadRunner-Rückruf aufzeichnen	Zeichnet das LoadRunner-Stubobjekt als ein Callback auf. Wenn diese Option deaktiviert ist, zeichnet VuGen die ursprüngliche Klasse als das Callback auf. **Standardwert:** Aktiviert.
Aufgezeichnetes Protokoll	Gibt an, welches Protokoll aufgezeichnet werden soll: RMI, CORBA, JMS oder Jacada **Standardwert:** RMI.
Entfernen von Zeilen mit	Entfernt alle Zeilen, die eine der angegebenen Zeichenfolgen aus dem Skript enthalten. Um mehrere Zeichenfolgen anzugeben, trennen Sie die Einträge mit Kommata. Diese Funktion ist nützlich, um das Skript für ein spezifisches Testziel anzupassen.
Unlesbare Zeichenfolgen als Bytes	Stellt Zeichenfolgen mit unlesbaren Zeichen als Bytearrays dar. Diese Option gilt für Zeichenfolgen, die als Parameter an Aufrufe übergeben werden. **Standardwert:** Aktiviert.
_JAVA_OPTIONS-Kennzeichen verwenden	Zwingt JVM-Versionen 1.2 und höher, die Umgebungsvariable _JAVA_ OPTION zu verwenden, die die gewünschten JVM-Parameter enthält. **Standardwert:** Deaktiviert.
DLL-Hooking verwenden, um LoadRunner-Unterstützung anzufügen	Verwendet DLL-Hooking, um LoadRunner-Unterstützung automatisch an jede JVM anzuhängen.

Aufzeichnungseigenschaften > Knoten "Serialisierungsoptionen"

Über diese Optionen können Sie steuern, wie Objekte serialisiert werden. Serialisierung ist oft für die Anzeige von Objekten in einer ACII-Darstellung relevant, um ihre Werte zu parametrisieren.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Aufzeichnungseigenschaften > Serialisierungsoptionen**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der zugeordneten Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.
Siehe auch	"Korrelieren von Skripts - Java-Skripts - Serialisierung" auf Seite 305

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung

Unfold Serialized Objects	Erweitert serialisierte Objekte in einer ASCII-Darstellung und ermöglicht die Anzeige der ASCII-Werte der Objekte, um eine Parametrisierung durchzuführen. • **Limit Object Size (bytes).** Begrenzt serialisierbare Objekte auf den angegebenen Wert. Objekte, die größer als dieser Wert sind, werden im Skript nicht in ASCII-Darstellung angezeigt. **Standardwert:** 3072 Byte. • **Ignore Serialized Objects.** Listet die serialisierten Objekte auf, die nicht erweitert werden sollen, wenn sie im aufgezeichneten Skript angetroffen werden. Trennen Sie Objekte mit Kommata. **Syntax:** java.lang.Object class format, delimiter = "," • **Serialization Delimiter.** Gibt das Trennzeichen an, das die Elemente in der ASCII-Darstellung von Objekten trennt. VuGen parametrisiert nur Zeichenfolgen, die zwischen diesen Trennzeichen stehen. Das Standardtrennzeichen lautet '#'. • **Unfold Arrays.** Erweitert Arrayelemente von serialisierten Objekten in einer ASCII-Darstellung. Wenn Sie diese Option deaktivieren und ein Objekt ein Array enthält, wird das Objekt nicht erweitert. **Standardwert:** Aktiviert - alle deserialisierten Objekte werden vollständig erweitert. ▪ **Limit Array Entries.** Legt fest, dass der Recorder Arrays mit mehr als der angegebenen Anzahl von Elementen nicht öffnen soll. **Standardwert:** 200.

RTE > Konfiguration (Knoten)

Sie können die Aufzeichnungsoptionen so einstellen, dass sie dem Zeichensatz entsprechen, der während der Terminalemulation verwendet wird.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > RTE > Konfiguration**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung

Zeichensatz	Muss dem Zeichensatz entsprechen, der während der Terminalemulation verwendet wird. Der Standard-Zeichensatz ist ANSI. Für Kanji und andere Multi-Byte-Plattformen können Sie DBCS (Double-byte Character Set) festlegen.

RTE > RTE (Knoten)

Ermöglicht das Festlegen der allgemeinen RTE-Aufzeichnungsoptionen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > RTE > RTE**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Automatisch Synchronisierungsbefehle generieren	Generiert automatisch eine Reihe von **TE-Synchronisierungsfunktionen** und fügt diese während der Aufzeichnung im Skript ein. • **Cursor.** Generiert eine **TE_wait_cursor**-Funktion vor jeder **TE_type**-Funktion. • **Prompt.** Generiert eine **TE_wait_text**-Funktion vor jeder **TE_type**-Funktion (falls zutreffend). • **X-System.** Generiert jedes Mal, wenn während der Aufzeichnung ein neuer Bildschirm angezeigt wird, eine Funktion **TE_wait_sync**. **Hinweis:** VuGen generiert nur bei der Aufzeichnung von VT-Terminals aussagekräftige Funktionen des Typs **TE_wait_text**. Verwenden Sie das automatische Generieren der Funktion **TE_wait_text** nicht bei der Aufzeichnung von Block-Mode-Terminals (IBM).

Generiert automatische X-System-Transaktionen	Zeichnet die Zeit auf, die das System während einer Szenarioausführung im X System-Modus verbringt. Hierzu wird nach jeder Funktion **TE_wait_sync** eine Funktion **TE_wait_sync_transaction** eingefügt. Jede Funktion **TE_wait_sync_transaction** erstellt eine Transaktion mit dem Namen **default**. Durch jede Funktion **TE_wait_sync_transaction** wird die Zeit aufgezeichnet, die das System im vorangegangenen X-System-Zustand verbracht hat.
Bildschirmkopfkommentare generieren	Generiert während der Aufzeichnung eines Vuser-Skripts Bildschirmkopfkommentare und fügt diese in das Skript ein. Ein generierter Kommentar enthält den Text, der in der ersten Zeile des Terminalemulatorfensters angezeigt wird. **Hinweis:** Sie können automatische Kommentare nur generieren, wenn Sie Block-Mode-Terminalemulatoren wie z. B. den IBM 5250 verwenden.
Zeitüberschreitung für die Tastaturaufzeichnung	Wenn Text während der Aufzeichnung in einen Terminalemulator eingegeben wird, überwacht VuGen die Texteingabe. Nach jeder Tastenbedienung wartet VuGen bis zu einer bestimmten Zeit auf die nächste Tastenbedienung. Erfolgt innerhalb der festgelegten Zeit keine weitere Tastenbedienung, geht VuGen davon aus, dass der Befehl abgeschlossen ist.

SAPGUI > Automatisch anmelden (Knoten)

Ermöglicht die automatische Anmeldung zu Beginn der Aufzeichnung. Die Anmeldefunktionen befinden sich im Abschnitt **vuser_init** des Skripts.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > SAPGUI > Automatisch anmelden**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Automatische Anmeldung aktivieren	Ermöglicht die automatische Anmeldung zu Beginn der Aufzeichnung. Geben Sie den Servernamen, den Benutzer, das Kennwort, den Clientnamen und die Sprachen der Benutzeroberfläche für den SAP-Server ein.

SAPGUI > Codegenerierung (Knoten)

Ermöglicht das Festlegen der Codegenerierungseinstellungen für das SAPGUI-Protokoll.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > SAPGUI > Codegenerierung**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Immer Objekt-ID in Kopfzeilendatei generieren	Trägt die Objekt-IDs in einer separaten Kopfzeile ein und nicht im Skript. Wenn Sie diese Option deaktivieren, generiert VuGen die IDs gemäß der Zeichenfolgelänge, die in der allgemeinen Skripteinstellung angegeben wurde. Dies führt zu einem kompakteren und übersichtlicheren Skript.
Schritte zum Füllen mit Daten generieren	Generiert Schritte, durch die Daten in Tabellen und Rastern eingetragen werden, ohne Einzelschritte für jede Zelle auszuführen.
Anmeldung als einen Schritt generieren	Generiert eine einzelne Methode **sapgui_logon** für alle Anmeldevorgänge. Dies vereinfacht den Code. Wenn Anmeldeprobleme auftreten, deaktivieren Sie diese Option.

SAPGUI > Knoten "Allgemein"

Ermöglicht das Festlegen der allgemeinen Aufzeichnungsoptionen für das SAPGUI-Protokoll.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > SAPGUI > Allgemein**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung

Bildschirmsnapshots erfassen	Gibt an, wie die Snapshots der SAPGUI-Bildschirme während der Aufzeichnung gespeichert werden: **ActiveScreen-Snapshots**, **Reguläre Snapshots** oder **Keine**. Die Option **ActiveScreen snapshots** sorgt für höhere Interaktivität und umfassendere Bildschirminformationen nach der Aufzeichnung, erfordert aber mehr Ressourcen.
Ändern von Ereignissen während der Aufzeichnung	**Process context menus by text.** Verarbeitet die Kontextmenüs anhand ihres Textes und generiert Funktionen des Typs **sapgui_ toolbar_select_context_menu_item_by_text**. Bei deaktivierter Option verarbeitet VuGen die Kontextmenüs anhand ihrer IDs und generiert eine Funktion **sapgui_toolbar_select_context_menu_item** für Kontextmenüs. Dies ist bei der Arbeit mit japanischen Schriftzeichen vorteilhaft.

Knoten "Silverlight > Dienste"

Ermöglicht die Verwaltung von WSDL-Dateien in Silverlight-Protokollskripts.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Silverlight > Dienste**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.
Relevante Aufgaben	"Importieren von WSDL-Dateien" auf Seite 879

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Dienstliste>	Die Liste der importierten WSDL-Dateien und der zugehörigen Speicherorte. Über die Symbolleiste können Sie WSDL-Dateien hinzufügen, löschen und bearbeiten. Klicken Sie zusätzlich auf die Schaltfläche **Protokoll und Sicherheit**, um Protokoll- und Sicherheitsdaten zu bearbeiten.
WSDL-Dateien automatisch erkennen und Dienste während der Codegenerierung importieren	Versucht automatisch, die im Skript verwendeten WSDL-Dateien zu finden und zu importieren.

Keine WSDL-Dateien verwenden	Deaktiviert WSDL-Dateien im Skript und generiert stattdessen SOAP-Anforderungen. Dies führt zu einem Skript mit einer tieferen Ebene, steigert jedoch im Allgemeinen die Skriptleistung.
Dienstendpunkt	Der Speicherort des Endpunkts, an dem eine bestimmte WSDL verfügbar ist.
In Skript enthaltene WSDL-Dateien verwenden	Lässt WSDL-Dateien zu, die automatisch oder manuell importiert werden.
WSDL-Speicherort	Der Speicherort der ausgewählten WSDL-Datei.

Dialogfeld "Dienst hinzufügen/bearbeiten"

Ermöglicht das Suchen und Importieren von WSDL-Dateien in ein Silverlight-Protokollskript.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Silverlight > Dienste > Hinzufügen**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.
Relevante Aufgaben	"Importieren von WSDL-Dateien" auf Seite 879

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Verbindungseinstellungen	Öffnet das Dialogfeld **Verbindungseinstellungen**, in dem Sie die Proxy- und Authentifizierungsinformationen für die angegebene WSDL-Datei konfigurieren können. Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Verbindungseinstellungen"" auf der nächsten Seite.
WSDL auswählen von	• **URL.** Die WSDL-Datei wird durch Angabe des URL ausgewählt. • **Datei.** Die WSDL-Datei wird durch Angabe des lokalen Pfads ausgewählt. • **Zuvor importiert.** Die WSDL-Datei kann aus der WSDL-Historie (Liste der zuvor importierten WSDL-Dateien) ausgewählt werden.

Dienstendpunkt	Der Speicherort des Endpunkts, an dem eine bestimmte WSDL verfügbar ist.
WSDL-Speicherort	Der URL oder der lokale Pfad der WSDL.

Dialogfeld "Verbindungseinstellungen"

Konfiguriert die Proxy- und Authentifizierungsinformationen für WSDL-Dateien in Silverlight-Protokollskripts.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Silverlight > Dienste > Hinzufügen > Verbindungseinstellungen**
Wichtige Informationen	• Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250. • Diese Einstellungen sind nur für das Importieren von WSDL-Dateien relevant. Um Authentifizierungs- und Proxyinformationen für WSDL-Dateien zu konfigurieren, die während der Wiedergabe verwendet werden sollen, fügen Sie eine Funktion **web_set_user_step** mit den gewünschten Werten hinzu.
Relevante Aufgaben	"Importieren von WSDL-Dateien" auf Seite 879

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Authentifizierung	Die Authenfizierungseinstellungen werden aktiviert, indem Sie **Authentifizierungseinstellungen verwenden** auswählen und Ihren Benutzernamen und das entsprechende Kennwort eingeben.
Proxy	Die Proxyeinstellungen werden aktiviert, indem Sie **Proxyeinstellungen verwenden** auswählen und Ihren Benutzernamen, das entsprechende Kennwort sowie Server und Portnummer eingeben.

Dialogfeld "Daten für das Szenario zu Protokoll und Sicherheit"

Ermöglicht die Konfiguration der Einstellungen von Daten für das Szenario zu Protokoll und Sicherheit.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Silverlight > Dienste > Schaltfläche "Protokoll- und Sicherheitsdaten"**
Wichtige Informationen	• Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250. • Diese Einstellungen sind nur für das Importieren von WSDL-Dateien relevant. Um Authentifizierungs- und Proxyinformationen für WSDL-Dateien zu konfigurieren, die während der Wiedergabe verwendet werden sollen, fügen Sie eine Funktion **web_set_user_step** mit den gewünschten Werten hinzu. • Die Einstellungen in diesem Dialogfeld werden während der Codegenerierung zurückgesetzt.
Relevante Aufgaben	"Importieren von WSDL-Dateien" auf Seite 879

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Port	Ein einzelner Endpunkt für eine WSDL-Bindung. **Hinweis:** [redacted]
Transport	Das Transportschichtprotokoll, das VuGen zum Senden von Dienstanforderungen an den Server verwendet. Zur Auswahl stehen die Protokolle HTTP, HTTPS und LrHTTP. **Hinweis:** HTTP ist nicht mit der Sicherheitseinstellung "UserNameOverTransport" kompatibel. Bei HTTPS muss "UserNameOverTransport" ausgewählt werden.
Codierung	Die Codierungsmethode, die für Serviceanforderungen verwendet werden muss, die an den Server gesendet werden.
WS-Adressierungsversion	Die WS-Adressierungsversion für die ausgewählte WSDL-Datei.
Sicherheit	

Authentifizierungsmodus	Aktivieren Sie die Authentifizierung, indem Sie den Modus **UserNameOverTransport** auswählen. **Standardmodus:** Keine.
Benutzername	Wird die Authentifizierung aktiviert, ist ein gültiger Benutzername erforderlich.
Kennwort	Wird die Authentifizierung aktiviert, ist ein gültiges Kennwort erforderlich.

Datenverkehrsanalyse > Filter für Datenverkehr

In diesem Dialogfeld können Sie den eingehenden und ausgehenden Verkehr filtern.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Datenverkehrsanalyse > Filter für Datenverkehr**
Wichtige Informationen	• Weitere Informationen finden Sie unter "Analysieren von Datenverkehr" auf Seite 765. • Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie in der "Tabelle für Protokollkompatibilität" auf Seite 250.
Relevante Aufgaben	"Erstellen eines Skripts durch Analysieren des Datenverkehrs" auf Seite 771

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	**Beschreibung**
Beim Generieren des Skripts	Kontrollkästchen **Alle IP-Adressen in der Liste einschließen**, um Verkehr von den angegebenen IP-Adressen einzuschließen. Kontrollkästchen **Alle IP-Adressen in der Liste ausschließen**, um Verkehr von den angegebenen IP-Adressen auszuschließen.
<Registerkarte "Eingehender Datenverkehr"> Ermöglicht Ihnen das Identifizieren von IP-Adressen für eingehenden Verkehr.	
	IP-Adresse hinzufügen
Quell-IP	Festlegen der IP-Adresse des Servers.
	IP-Adresse löschen

Element der Oberfläche	Beschreibung
<Registerkarte "Ausgehender Datenverkehr"> Ermöglicht Ihnen das Identifizieren von IP-Adressen für ausgehenden Verkehr.	
	IP-Adresse hinzufügen
Ziel-IP	Festlegen der IP-Adresse des Servers.
Zielport	Festlegen des Zielports des Servers.
	IP-Adresse löschen

Knoten "WinSock"

Ermöglicht das Festlegen der WinSock-Aufzeichnungsoptionen.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > WinSock**

Wichtige Informationen	Bei Verwendung von Konvertierungstabellen auf Solaris-Computern müssen Sie die folgenden Umgebungsvariablen auf allen Computern festlegen, die Skripts ausführen:
	`setenv LRSDRV_SERVER_FORMAT 0025`
	`setenv LRSDRV_CLIENT_FORMAT 04e4`

Relevante Aufgaben	"Aufzeichnen eines Windows-Sockets-Skripts" auf Seite 1057

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Fügt einen neuen Eintrag zur Liste der ausgeschlossenen Sockets hinzu.
	Entfernt den ausgewählten Eintrag aus der Liste der ausgeschlossenen Sockets.
Ausgeschlossenes Socket nicht in Protokoll aufnehmen	Schließt die Sockets in der Liste aus dem Protokoll aus. Durch Deaktivieren dieser Option wird die Protokollierung für die ausgeschlossenen Sockets aktiviert. Ihren Aktionen wird in der Protokolldatei der Hinweis "Exclude" vorangestellt.

Ausschlusseinstellungen / Socketliste	Der Host und der Port der Sockets, die von der Aufzeichnung oder der erneuten Generierung des Skripts ausgeschlossen werden. Verwenden Sie die folgende Syntax: • Bei dem Format **Host:Port** wird ein bestimmter Port ausgeschlossen. • Bei dem Format **Host** werden alle Ports des angegebenen Hosts ausgeschlossen. • Bei dem Format **:Port** wird ein bestimmter Port auf dem lokalen Host ausgeschlossen. • Bei dem Format ***:Port** wird ein bestimmter Port auf allen Hosts ausgeschlossen.
Codierungsmethode	**OEM-Codierung verwenden.** Aktiviert eine Datencodierung, die nicht englische Zeichen unterstützt. **ASCII-Codierung verwenden.** Aktiviert eine Datencodierung, die auf englische Zeichen beschränkt ist. Verwenden Sie diese Option, um die Methode LR 9.5x für die Datencodierung zu replizieren.
Latenzzeit-Schwellenwert	Während der Aufzeichnung fügt VuGen automatisch die Latenzzeitschritte hinzu, wenn Sie zwischen Aktionen Pausen einlegen. Sie können einen Schwellenwert festlegen, bis zu dem die aufgezeichnete Latenzzeit ignoriert wird. **Standardwert:** fünf Sekunden.
Konvertierungstabellen	Eine Konvertierungstabelle ermöglicht es Ihnen, das Format für die Aufzeichnungssitzungen anzugeben, wenn Sie WinSock als Einzelprotokoll verwenden. Sie können sie für die Codegenerierung heranziehen, wenn Sie ein WinSock-Mehrfachprotokoll verwenden. Dieses gilt für Benutzer, die auf Mainframecomputern oder AS/400-Servern ausgeführt werden. Sowohl Server als auch Client beziehen das Datenformat aus den auf Ihrem System installierten Übersetzungstabellen. Wählen Sie aus dem Listenfeld eine Übersetzungsoption aus. Die ersten vier Zeichen des Listenfeldelements stehen für das Serverformat. Die letzten vier Zeichen stehen für das Clientformat. Die Übersetzungstabellen befinden sich im Ordner **ebcdic** des VuGen-Installationsordners. Wenn Ihr System andere Übersetzungstabellen verwendet, kopieren Sie sie in den Ordner **ebcdic**. **Hinweis:** Daten im ASCII-Format erfordern keine Übersetzung. Wählen Sie in diesem Fall die Standardoption **Keine** aus.

Aufzeichnungsoptionen – Sonstige Themen

Dieser Abschnitt enthält eine Vielzahl unterschiedlicher Themen in Zusammenhang mit den Aufzeichnungsoptionen.

Tabelle für Protokollkompatibilität

In der folgenden Tabelle werden die Vuser-Protokolle sowie die Knoten mit Aufzeichnungsoptionen aufgeführt, die für jedes Protokoll verfügbar sind.

Protokoll	Knoten mit Aufzeichnungsoptionen
AJAX - Click & Script-Protokoll	• Allgemein - Skript, Aufzeichnen • GUI-Eigenschaften - Erweitert, Webereigniskonfiguration • Netzwerk - Portzuordnung • HTTP-Eigenschaften - Erweitert, Korrelation
Ajax TruClient	• Keine
C VUser	• Keine
Citrix ICA	• Allgemein - Skript • Citrix - Konfiguration, Recorder, Codegenerierung, Anmelden
COM/DCOM	• Allgemein - Skript • COM/DCOM - Filter, Optionen
DNS	• Keine
FTP	• Allgemein - Skript • Netzwerk - Portzuordnung
Flex (AMF/RTMP)	• Allgemein - Skript, Protokolle, Aufzeichnen • Flex - RTMP, Konfiguration, Externalisierbare Objekte, AMF • Netzwerk - Portzuordnung • HTTP-Eigenschaften - Erweitert, Korrelation
IMAP	• Allgemein - Skript • Netzwerk - Portzuordnung

Protokoll	Knoten mit Aufzeichnungsoptionen
Java over HTTP	• Allgemein - Aufzeichnen • Java-Umgebungseinstellungen - Java VM, Classpath • Netzwerk - Portzuordnung • HTTP-Eigenschaften - Browser, Proxy für Aufzeichnung, Erweitert, Korrelation
Java Record Replay	• Java-Umgebungseinstellungen - Java VM, Classpath • Aufzeichnungseigenschaften - Recorder-Optionen, Serialisierungsoptionen, Korrelationsoptionen, Protokolloptionen, Corba-Optionen
Java-Vuser	• Keine
LDAP	• Allgemein - Skript
MMS (Media Player)	• Keine
Microsoft .NET	• Allgemein - Skript • .NET - Aufzeichnen, Filter
Microsoft Remote Desktop Protocol (RDP)	• Allgemein - Skript • RDP - Clientstart, Codegenerierung (Standard, Erweitert, Agent) • Netzwerk - Portzuordnung
MAPI (Microsoft Exchange)	• Keine
MMS (Multimedia Messaging Service)	• Allgemein - Skript
ODBC	• Allgemein - Skript • Datenbank - Datenbank
Oracle (2-Tier)	• Allgemein - Skript • Datenbank - Datenbank

Protokoll	Knoten mit Aufzeichnungsoptionen
Oracle NCA	• Allgemein - Skript, Protokolle, Aufzeichnen • Netzwerk - Portzuordnung • HTTP-Eigenschaften - Erweitert, Korrelation
Oracle Web Applications 11i	• Allgemein - Skript, Aufzeichnen • GUI-Eigenschaften - Erweitert, Webereigniskonfiguration • Netzwerk - Portzuordnung • HTTP-Eigenschaften - Erweitert, Korrelation
POP3	• Allgemein - Skript • Netzwerk - Portzuordnung
SAP Click and Script	• Allgemein - Skript, Aufzeichnen • GUI-Eigenschaften - Erweitert, Webereigniskonfiguration • Netzwerk - Portzuordnung • HTTP-Eigenschaften - Erweitert, Korrelation
SAP - Web	• Allgemein - Skript, Aufzeichnen • Netzwerk - Portzuordnung • HTTP-Eigenschaften - Erweitert, Korrelation
SAP GUI	• Allgemein - Skript • SAP GUI - Allgemein, Codegenerierung, Automatisch anmelden
Siebel - Web	• Allgemein - Skript, Protokolle, Aufzeichnen • Netzwerk - Portzuordnung • HTTP-Eigenschaften - Erweitert, Korrelation

Protokoll	Knoten mit Aufzeichnungsoptionen
Silverlight	• Allgemein - Skript, Protokolle, Aufzeichnen • Silverlight – Dienste • Netzwerk - Portzuordnung • HTTP-Eigenschaften - Erweitert, Korrelation • Datenformaterweiterung - Kettenkonfiguration, Codegenerierung
SMTP	• Allgemein - Skript • Netzwerk - Portzuordnung
Web (HTTP/HTML)	• Allgemein - Skript, Protokolle, Aufzeichnen • Netzwerk - Portzuordnung • HTTP-Eigenschaften - Erweitert, Korrelation • Datenformaterweiterung - Kettenkonfiguration, Codegenerierung
Webdienste	• Allgemein - Skript, Protokolle, Aufzeichnen • Datenverkehrsanalyse - Filter für Datenverkehr • Netzwerk - Portzuordnung • HTTP-Eigenschaften - Erweitert, Korrelation
Windows-Sockets	• Sockets - Winsock

Übersicht über die Portzuordnung

Bei der Aufzeichnung von Vuser-Skripts, die den Netzwerkverkehr auf einer Socket-Ebene erfassen (HTTP, SMTP, POP3, FTP, IMAP, Oracle NCA und WinSocket) können Sie Optionen für die Port-Zuordnung festlegen. Diese Optionen bieten die Möglichkeit, den Verkehr einer bestimmten Kombination von Server und Port dem gewünschten Kommunikationsprotokoll zuzuordnen.

Für die Zuordnung stehen die Kommunikationsprotokolle FTP, HTTP, IMAP, NCA, POP3, SMTP und SOCKET zur Verfügung. Um eine Zuordnung zu erstellen, geben Sie einen Servernamen, eine Portnummer oder eine vollständige Server:Port-Kombination an. Sie können beispielsweise festlegen, dass der gesamte Verkehr des Servers *twilight* auf Port 25 als SMTP gehandhabt werden soll. Sie könnten auch festlegen, dass der gesamte Verkehr des Servers *viper*, unabhängig vom Port, dem FTP-Protokoll zugeordnet werden soll. Es ist außerdem möglich, den gesamten Verkehr auf Port 23, unabhängig vom Servernamen, SMTP zuzuordnen.

Wenn bei der Aufzeichnung im Mehrfachprotokollmodus mindestens ein Protokoll auf der Socket-Ebene aufzeichnet, ist der Knoten **Portzuordnung** verfügbar. Die einzige Ausnahme bildet die Aufzeichnung von HTTP oder WinSock als Einzelprotokollskript.

Automatische Erkennung der Portzuordnung

Über die erweiterten Optionen von VuGen für die Port-Zuordnung können Sie die Optionen für die **automatische Erkennung** konfigurieren. Die automatische Erkennung von VuGen analysiert die an den Server gesendeten Daten. Sie überprüft die Daten auf eine Signatur, ein Muster im Dateninhalt, das zur Identifizierung des Protokolls dient. Zum Zweck der Signaturerkennung, werden alle Sendepuffer bis zum ersten Empfangspuffer zusammengenommen Alle Sendepuffer, die gesendet wurden, bis ein Empfangspuffer zurückgegeben wurde, werden als eine einzige Datentransition angesehen. Standardmäßig werden keine Zuordnungen definiert und VuGen wendet die automatische Erkennung an. In einigen Protokollen bestimmt VuGen den Typ in einer einzigen Transition (z. B. HTTP). Bei anderen Netzwerkprotokollen sind mehrere Transitionen erforderlich, bis der Typ bestimmt ist. Zu diesem Zweck erstellt VuGen einen temporären Puffer für jede Server:Port-Kombination. Wenn VuGen den Protokolltyp nicht durch Lesen der ersten Transitionspuffer bestimmen kann, werden die Daten in einem temporären Puffer gespeichert. Die eingehenden Puffer werden solange gelesen, bis eine Signatur eines bestimmten Protokolls erkannt wird.

Standardmäßig erlaubt VuGen 4 Transitionen und setzt einen temporären Puffer von 2048 Byte ein, um eine Protokollsignatur zu erkennen. Wenn VuGen den Typ auch nach Erreichen der maximalen Anzahl von Transitionen oder der maximalen Puffergröße nicht erkannt hat, ordnet es die Daten dem WinSock-Protokoll zu. Wenn Sie VuGen nicht angewiesen haben, das WinSock-Protokoll aufzuzeichnen (bei der Auswahl des Mehrfachprotokolls), werden die Daten verworfen.

Sie haben die Möglichkeit, die maximale Anzahl der Puffer zu ändern, die von VuGen zum Erkennen des Protokolltyps gelesen werden sollen. Sie können auch die Größe des temporären Puffers festlegen. Wenn die Datenmenge in den ersten Sendepuffern die Größe des temporären Puffers übersteigt, kann VuGen den Protokolltyp nicht automatisch erkennen. Vergrößern Sie in diesem Fall den temporären Puffer.

Bei Verwendung der obigen Netzwerkebenenprotokolle wird eine automatische Erkennung des Protokolltyps durch VuGen empfohlen. In den meisten Fällen ist der VuGen-Recorder in der Lage, die Signaturen dieser Protokolle zu erkennen. Anschließend verarbeitet er diese automatisch gemäß den Protokollspezifikationen. In bestimmten Fällen kann VuGen das Protokoll jedoch nicht erkennen. Beispiel:

- Ähnelt die Protokollsignatur einem bestehenden Protokoll, führt dies zu einer fehlerhaften Verarbeitung.
- Das Protokoll weist keine eindeutige Signatur auf.
- Das Protokoll verwendet SSL-Verschlüsselung und kann daher auf der WinSock-Ebene nicht erkannt werden.

In allen oben genannten Fällen können Sie Informationen eingeben, um den Server und den Port, die das Protokoll hosten, eindeutig zu identifizieren.

EUC-Codierung (nur japanisches Windows)

Bei der Arbeit mit Nicht-Windows-Standardzeichensätzen müssen Sie eventuell eine Codekonvertierung durchführen. Ein Zeichensatz wird durch die Zuordnung eines Satzes von Zeichen zu einem Satz von Ganzzahlen gebildet. Bei dieser Zuordnung entsteht eine eindeutige Kombination aus Zeichen und Ganzzahl, die für ein bestimmtes Alphabet gültig ist. Extended UNIX Code (EUC) und Shift Japan Industry Standard (SJIS) sind Nicht-Windows-Standardzeichensätze, mit denen japanische Schriftzeichen auf Websites dargestellt werden.

Unter Windows wird die SJIS-Codierung, unter UNIX die EUC-Codierung verwendet. Wird auf dem Webserver UNIX ausgeführt und auf dem Client Windows, werden die Zeichen der Website auf dem Client aufgrund der Unterschiede in den Codierungsmethoden nicht korrekt dargestellt. Davon ist auch die Darstellung von EUC-codierten japanischen Zeichen in Vuser-Skripts betroffen.

Bei der Aufzeichnung ermittelt VuGen die Codierung der Webseite anhand von deren HTTP-Header. Enthält der HTTP-Header keine Informationen zum Zeichensatz, wird das HTML-Metatag geprüft.

Wenn Ihnen im Vorfeld bekannt ist, dass eine bestimmte Webseite EUC-codiert ist, können Sie VuGen unter Verwendung der Aufzeichnungsoptionen anweisen, die richtige Codierung zu verwenden. Um eine Seite in EUC-Codierung aufzuzeichnen, aktivieren Sie die Option **EUC** in den Aufzeichnungsoptionen (Knoten **Aufzeichnen**) (nur in der japanischen Windows-Version angezeigt).

Durch die Aktivierung der Option **EUC** wird VuGen gezwungen, eine Webseite in EUC-Codierung aufzuzeichnen, auch wenn diese nicht EUC-codiert ist. Daher sollte die Option nur aktiviert werden, wenn VuGen die Codierung nicht aus dem HTTP-Header oder dem HTML-Metatag ermitteln kann und Ihnen im Vorfeld bekannt ist, dass die Seite EUC-codiert ist.

Während der Aufzeichnung empfängt VuGen eine EUC-codierte Zeichenfolge vom Webserver und konvertiert diese in SJIS. Die SJIS-Zeichenfolge wird in der Skriptfunktion **Action** gespeichert. Eine erfolgreiche Wiedergabe ist jedoch nur möglich, wenn die Zeichenfolge wieder in EUC konvertiert wird, bevor sie an den Webserver zurückgesendet wird. Aus diesem Grund fügt VuGen vor der Funktion **Action** die Funktion **web_SyS_to_euc_param** ein, welche die SJIS-Zeichenfolge wieder in EUC konvertiert.

Im folgenden Beispiel ruft der Benutzer eine EUC-codierte Webseite auf und klickt auf einen Link. VuGen zeichnet die Funktion **Action** auf und fügt die Funktion **web_sjis_to_euc_param** vor der Funktion **Action** in das Skript ein.

```
web_sjis_to_euc_param("param_link","Search");
web_link("LinkStep","Text={param_link}");
```

Weitere Informationen finden Sie unter "Dialogfeld "Erweiterter URL"" auf Seite 194.

Übersicht über die benutzerdefinierten Einstellungen für die Skriptgenerierung

Bevor Sie eine Sitzung aufzeichnen, ermöglicht VuGen das Festlegen der Sprache für die Skriptgenerierung. Die für die Skriptgenerierung verfügbaren Sprachen sind vom Protokoll abhängig. Die gebräuchlichsten verfügbaren Sprachen sind C und Java. Standardmäßig generiert VuGen ein Skript in der für das jeweilige Protokoll gebräuchlichsten Sprache; diese kann jedoch in den **Aufzeichnungsoptionen** des Skripts geändert werden.

Informationen zur Benutzeroberfläche finden Sie unter "Allgemein > Skript (Knoten)" auf Seite 196.

Tipp: Ein in einer Sprache aufgezeichnetes Skript kann nach der Aufzeichnung in einer anderen Sprache erneut generiert werden. Weitere Informationen über die Aufgabe finden Sie unter "Erneutes Generieren eines Vuser-Skripts" auf Seite 271.

Nachdem Sie eine Generierungssprache ausgewählt haben, können Sie die sprachspezifischen Aufzeichnungsoptionen aktivieren, die dem Recorder mitteilen, welche Elemente das Skript umfassen soll und wie das Skript generiert werden soll.

Wenn mindestens eines der aufgezeichneten Protokolle mehrfachprotokollfähig ist, ist der Knoten **Skript** verfügbar, es sei denn, Sie zeichnen HTTP oder WinSock als Einzelprotokollskript auf.

Skriptsprachenoptionen

Beim Aufzeichnen einer Sitzung erstellt VuGen ein Skript, das die Aktionen emuliert. Die Standardsprache für die Skriptgenerierung ist C. In der folgenden Liste wird angegeben, welche Protokolle für die einzelnen Sprachen geeignet sind:

- **C.** Für Aufzeichnungsanwendungen, die komplexe COM-Konstrukte und C++-Objekte verwenden.
- **C #.** Für Aufzeichnungsanwendungen, die komplexe Anwendungen und Umgebungen (nur MS .NET-Protokoll) verwenden.
- **Visual Basic .NET.** Für VB .NET-Anwendungen, die alle Funktionen von VB einsetzen.

Nach der Aufzeichnungssitzung können Sie das Skript mithilfe von regulärem C-, C#-, Visual Basic-, oder .NET-Code und Ablaufanweisungen ändern.

Übersicht über Aufzeichnungsebenen

In VuGen können Sie angeben, welche Informationen aufgezeichnet werden sollen und welche Funktionen bei der Generierung eines Vuser-Skripts verwendet werden sollen, indem Sie im Knoten **Allgemein > Aufzeichnen** des Dialogfelds **Aufzeichnungsoptionen** eine Aufzeichnungsebene auswählen. Die ausgewählte Aufzeichnungsebene wird von Ihren Anforderungen und Ihrer

Umgebung bestimmt. Die verfügbaren Ebenen sind **HTML-basiertes Skript** und **URL-basiertes Skript**. Informationen zur Benutzeroberfläche finden Sie unter "Allgemein > Aufzeichnen (Knoten)" auf Seite 194.

Die folgenden Beispiele enthalten Skripts, die drei Aufzeichnungsebenen verwenden:

HTML-basiertes Skript

Erzeugt für jede HTML-Benutzeraktion einen eigenen Schritt. Die Schritte sind intuitiv, geben jedoch die wirkliche Emulation des JavaScript-Codes nicht wieder.

```
/* HTML-based mode - a script describing user actions*/
...
web_url("WebTours",
        "URL=http://localhost/WebTours/",
         "Resource=0",
         "RecContentType=text/html",
         "Referer=",
         "Snapshot=t1.inf",
         "Mode=HTML",
        LAST);
web_link("Click Here For Additional Restrictions",
        "Text=Click Here For Additional Restrictions",
        "Snapshot=t4.inf",
        LAST);
web_image("buttonhelp.gif",
        "Src=/images/buttonhelp.gif",
        "Snapshot=t5.inf",
        LAST);
...
```

URL-basiertes Skript

Zeichnet alle Browseranfragen und Ressourcen des Servers auf, die aufgrund der Benutzeraktionen gesendet wurden. Dabei werden alle HTTP-Ressourcen automatisch als URL-Schritte aufgezeichnet (**web_url**-Anweisungen). Für normale Browseraufzeichnungen ist der Einsatz des URL-basierten Modus nicht empfehlenswert, da er anfälliger für Probleme hinsichtlich der Korrelation ist. Zur Aufzeichnung von Seiten wie Applets und Anwendungen ohne Browser ist dieser Modus dagegen optimal geeignet.

Da alle Aktionen als **web_url**-Schritte (und nicht als **web_link**- oder **web_image**-Schritte usw.) aufgezeichnet werden, erweisen sich URL-basierte Skripts im Vergleich zu HTML-basierten Skripts als weniger intuitiv.

```
/* URL-based mode - only web_url functions */
...
web_url("spacer.gif",
        "URL=http://graphics.hplab.com/images/spacer.gif",
        "Resource=1",
        "RecContentType=image/gif",
         "Referer=",
        "Mode=HTTP",
        LAST);
```

```
web_url("calendar_functions.js",
        "URL=http://www.im.hplab.com/travelp/calendar_functions.js",
        "Resource=1",
        "RecContentType=application/x-javascript",
         "Referer=",
        "Mode=HTTP",
        LAST);
...
```

Die Aufzeichnungsebene und die erweiterten Aufzeichnungsoptionen können während der Aufzeichnung gewechselt werden, wenn kein Mehrfachprotokoll-Skript aufgezeichnet wird. Fortgeschrittenen Benutzern steht die Option zum Kombinieren von Aufzeichnungsebenen beim Leistungstest zur Verfügung.

Außerdem können Sie ein Skript nach der Aufzeichnung erneut generieren und dabei eine andere Methode als bei der ursprünglichen Aufzeichnung einsetzen. Ein auf HTML-Basis aufgezeichnetes Skript kann beispielsweise auf einer URL-basierten Ebene erneut generiert werden. Wenn Sie ein Skript erneut generieren möchten, wählen Sie **Aufzeichnen > Skript neu generieren** aus und klicken auf **Optionen**, um die Aufzeichnungsoptionen für die erneute Generierung festzulegen.

Übersicht über die Serialisierung

VuGen verwendet die Serialisierung, wenn während der Aufzeichnung ein unbekanntes Objekt gefunden wird, sofern das Objekt die Serialisierung unterstützt. Bei einem unbekannten Objekt kann es sich um ein Eingabeargument handeln, das nicht im Filter enthalten war und dessen Erstellung daher nicht aufgezeichnet wurde. Serialisierung verhindert Kompilierungsfehler, die entstehen, wenn ein unbekanntes Argument an eine Methode übergeben wird. Bei serialisierten Objekten empfiehlt es sich häufig, einen benutzerdefinierten Filter für die Aufzeichnung dieser Objekte einzurichten. Weitere Informationen finden Sie unter "Serialisieren von Flex-Skripts " auf Seite 704.

Tipps zum Arbeiten mit der Ereignisabhörung und -aufzeichnung

Es kann manchmal schwierig sein, die idealen Abhör- und Aufzeichnungseinstellungen zu finden. Beachten Sie beim Definieren dieser Einstellungen die folgenden Richtlinien:

- Um ein Ereignis bei einem Objekts aufzuzeichnen, müssen Sie festlegen, dass VuGen das Ereignis abhört und aufzeichnet, wenn es auftritt. Sie können ein Ereignis bei einem untergeordneten Objekt abhören, auch wenn ein übergeordnetes Objekt den Handler oder das Verhalten enthält, oder Sie können ein Ereignis bei einem übergeordneten Objekt abhören, auch wenn das untergeordnete Objekt den Handler oder das Verhalten enthält.

Sie müssen jedoch das Aufzeichnen für das Ereignis beim Quellenobjekt festlegen (das Objekt, bei dem das Ereignis tatsächlich auftritt, unabhängig davon, welches übergeordnete Objekt den Handler oder das Verhalten enthält).

Angenommen eine Tabellenzelle mit einem Ereignis-Handler **onmouseover** enthält zwei Bilder. Wenn ein Benutzer eines der beiden Bilder mit dem Mauszeiger berührt, wird das Ereignis durch Event-Bubbling an die Zelle weitergeleitet. Es enthält Informationen zu dem tatsächlich berührten Bild. Sie können dieses Mouseover-Ereignis wie folgt aufzeichnen:

- Setzen Sie **Listen** für das WebTable-Mouseover-Ereignis auf **If Handler** (so dass VuGen das Ereignis "hört", wenn es auftritt) und deaktivieren Sie zugleich die Aufzeichnung des Ereignisses. Setzen Sie anschließend **Listen** für das Bild-Mouseover-Ereignis auf **Never** und setzen Sie seinen Aufzeichnungsstatus zugleich auf **Enable** (um das Mouseover-Ereignis an dem Bild aufzuzeichnen, nachdem es auf WebTable-Ebene abgehört wurde.).
- Setzen Sie **Listen** für das Bild-Mouseover-Ereignis auf **Always** (um das Mouseover-Ereignis abzuhören, auch wenn der Bild-Tag kein Verhalten oder keinen Handler enthält) und setzen Sie den Aufzeichnungsstatus für das Bildobjekt auf **Enabled** (um das Mouseover-Ereignis für das Bild aufzuzeichnen).

- Wenn VuGen viele Ereignisse abhören muss, kann es zu Leistungseinbußen kommen. Versuchen Sie daher die Abhörungseinstellungen auf die erforderlichen Objekte zu begrenzen.
- In seltenen Situationen kann das Abhören des Objekts, für das das Ereignis auftritt (das Quellenobjekt) zu einem Konflikt mit dem Ereignis führen.

Beispiel für Click and Script bei einer Aufzeichnung außerhalb des Kontexts

Im folgenden Beispiel wurde das Skript mit aktivierter Option für die Aufzeichnung außerhalb des Kontexts neu generiert.

```
web_image_link("Search Flights Button",
        "Snapshot=t5.inf",
        DESCRIPTION,
        "Alt=Search Flights Button",
        "FrameName=navbar",
        ACTION,
        "ClickCoordinates=58,9",
        LAST);
web_add_cookie("MSO=SID=;1141052844; DOMAIN=localhost");
web_add_cookie("MTUserInfo=hash=;47=;firstName=;Joseph=;expDate=;
                %0A=;creditCard=;=;address1=;234%20Willow%20Drive=;
                lastName=;Marshall%0A=;address2=;San%20Jose%2FCA%2F94085=;
                username=;jojo; DOMAIN=localhost");
web_url("FormDateUpdate.class",
        "URL=http://localhost:1080/WebTours/FormDateUpdate.class",
         "Resource=0",
         "RecContentType=text/html",
         "Referer=",
        "UserAgent=Mozilla/4.0 (Windows 2000 5.0) Java/1.4.2_08",
        "Mode=HTTP",
```

```
        LAST);
...
```

Wenn Sie diese Option deaktivieren, generiert VuGen keinen Code für die ActiveX-Steuerelemente und Java-Applets. Im folgenden Beispiel wurde nur die Funktion **web_image_link** generiert, nicht jedoch die Funktionen **web_url**, die die Klassendateien enthalten.

```
web_image_link("Search Flights Button",
        "Snapshot=t5.inf",
        DESCRIPTION,
        "Alt=Search Flights Button",
        "FrameName=navbar",
        ACTION,
        "ClickCoordinates=58,9",
        LAST);
```

Weitere Informationen finden Sie unter "GUI-Eigenschaften > Erweitert (Knoten)" auf Seite 217.

Abschnitte des Vuser-Skripts

Jedes Vuser-Skript enthält mindestens drei Abschnitte: vuser_init, eine oder mehrere Aktionen und vuser_end. Vor und während der Aufzeichnung können Sie den Abschnitt des Skripts auswählen, in den VuGen die aufgezeichneten Funktionen einfügen soll. Die folgende Tabelle zeigt, was in den einzelnen Abschnitten aufgezeichnet werden kann und wann diese ausgeführt werden:

Skriptabschnitt	Für die Aufzeichnung...	Wird ausgeführt, wenn...
vuser_init	einer Anmeldung an einen Server.	der Vuser initialisiert (geladen) wird.
Aktionen	Clientaktivität.	sich der Vuser im Status **Wird ausgeführt** befindet.
vuser_end	einer Abmeldeprozedur.	der Vuser endet oder angehalten wird.

Wenn Sie mehrere Iterationen eines Vuser-Skripts ausführen, werden nur die *Aktions*abschnitte des Skripts wiederholt. Die Abschnitte *vuser_init* und *vuser_end* werden nicht wiederholt. Weitere Informationen zu den Iterationseinstellungen finden Sie unter "Allgemein > Ausführungslogik (Knoten)" auf Seite 368.

Sie verwenden den VuGen-Skripteditor zum Anzeigen und Bearbeiten des Inhalts von jedem der Skriptabschnitte. Sie können nur den Inhalt eines einzigen Abschnitts auf einmal anzeigen. Zum Anzeigen eines Abschnitts im Skripteditor doppelklicken Sie im Lösungs-Explorer auf den Namen des Abschnitts.

Beim Arbeiten mit Vuser Skripts, die Java-Klassen verwenden, fügen Sie Ihren gesamten Code in die Aktionsklasse ein. Die Aktionsklasse enthält drei Methoden: **init**, **action** und **end**. Diese Methoden entsprechen den Abschnitten von Skripts, die mit anderen Protokollen entwickelt wurden. Sie fügen Initialisierungsroutinen in die Methode **init**, Clientaktionen in die Methode **action** und Abmeldeprozeduren in die Methode **end** ein.

Weitere Informationen finden Sie unter "Java-Vuser-Protokoll - Manuelles Programmieren von Skripts" auf Seite 730.

```
public class Actions{
    public int init() {
        return 0;}
    public int action() {
        return 0;}
    public int end() {
        return 0;}
}
```

Hinweis: Die Transaktionsaufschlüsselung für Oracle DB ist für Aktionen, die im Abschnitt "vuser_init" aufgezeichnet wurden, nicht verfügbar.

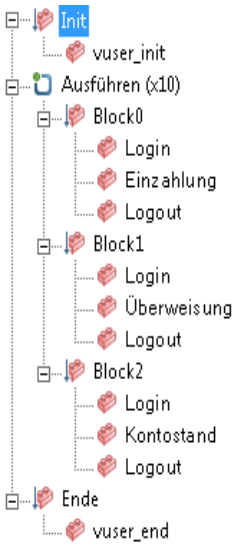
Beispiel für die Struktur eines Skriptabschnitts

Jedes Vuser-Skript ist in drei Bereiche unterteilt: *vuser_init*, *Run (Actions)* und *vuser_end*. Der Vuser kann angewiesen werden, den Abschnitt *Run* bei der Skriptausführung zu wiederholen. Eine Wiederholung wird auch als *Iteration* bezeichnet.

Die Abschnitte *vuser_init* und *vuser_end* eines Vuser-Skripts werden beim Ausführen mehrerer Iterationen nicht wiederholt.

Beim Ausführen von Skripts mit mehreren Aktionen können Sie angeben, wie die Aktionen vom Vuser ausgeführt werden sollen:

Im folgenden Beispiel führt *Block0* eine Einzahlung aus, *Block1* eine Überweisung und *Block2* schickt eine Kontostandsanfrage ab. Die Aktionen *Login* und *Logout* sind den drei Blöcken gemein.



Abfolge. Sie haben die Möglichkeit, die Reihenfolge der Aktionen innerhalb des Skripts festlegen. Außerdem können Sie angeben, ob die Aktionen sequentiell oder zufällig ausgeführt werden sollen.

Iterationen. Es besteht die Möglichkeit, die Anzahl der Iterationen für den gesamten *Run*-Abschnitt

oder für einzelne Aktionen bzw. Aktionsblöcke festzulegen. Dies ist beispielsweise bei der Emulation gewerblicher Websites nützlich, wenn für die Suche nach einem Produkt zahlreiche Abfragen ausgeführt werden, jedoch nur ein Kauf erfolgt.

Gewichtung. Für Aktionsblöcke, deren Aktionen zufällig ausgeführt werden, können Sie die *Gewichtung* bzw. den Prozentsatz, den jede Aktion innerhalb eines Blocks einnimmt, festlegen.

In den meisten Fällen entspricht der Name der Kopfzeilendatei dem Protokollpräfix. Datenbankfunktionen, die mit dem Präfix **lrd** beginnen, sind beispielsweise in der Datei **lrd.h** aufgelistet.

Kopfzeilendateien

Kopfzeilendateien enthalten im Allgemeinen Vorwärtsdeklarationen der Klassen, Unterroutinen, Variablen und anderen Bezeichner. In den meisten Fällen entspricht der Name der Kopfzeilendatei dem Protokollpräfix. Datenbankfunktionen, die mit dem Präfix **lrd** beginnen, sind beispielsweise in der Datei **lrd.h** aufgelistet.

In der folgenden Tabelle werden die Kopfzeilendateien der gebräuchlichsten Protokolle aufgeführt:

Protokoll	File
Ajax (Click & Script)	**web_ajax.h**
Citrix	**ctrxfuncs.h**
COM/DCOM	**lrc.h**
Datenbank	**lrd.h**
FTP	**mic_ftp.h**
Allgemeine C-Funktion	**lrun.h**
IMAP	**mic_imap.h**
LDAP	**mic_mldap.h**
MAPI	**mic_mapi.h**
Oracle NCA	**orafuncs.h**
POP3	**mic_pop3.h**
RDP	**lrrdp.h**
SAP GUI	**as_sapgui.h**
SAP (Click & Script)	**sap_api.h**
Siebel	**lrdsiebel.h**
SMTP	**mic_smtp.h**

Protokoll	File
Terminalemulator	**lrrte.h**
WAP	**as_wap.h**
Web (HTML\HTTP)	**as_web.h**
Web (Click & Script)	**web_api.h**
Webdienste	**wssoap.h**
Windows-Sockets	**lrs.h**

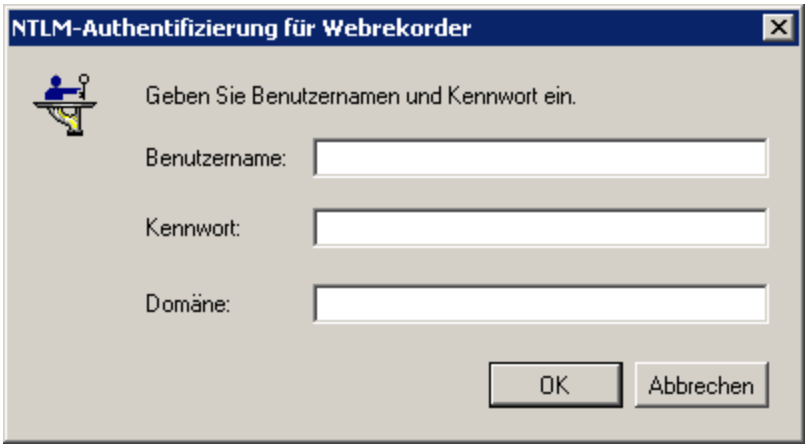
Bereitstellen von Authentifizierungsinformationen

Der folgende Abschnitt gilt nur für Mehrfachprotokollskripts.

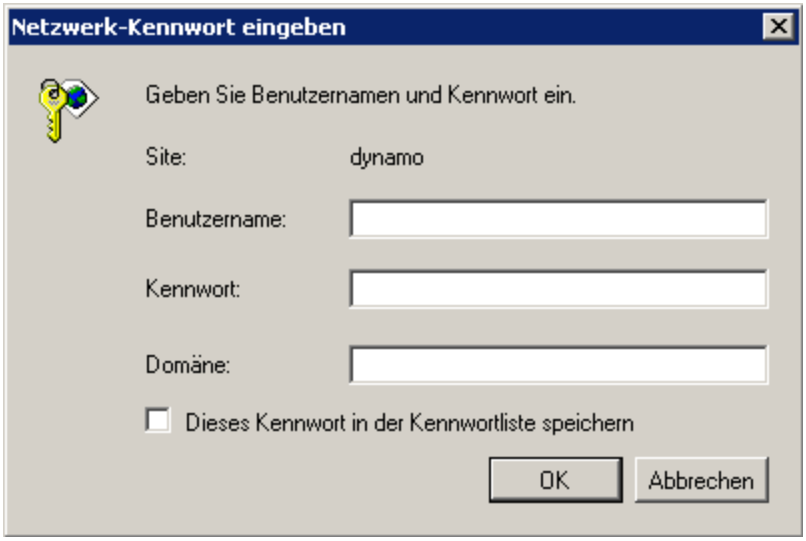
Beim Aufzeichnen einer Web-Sitzung, die NTLM-Authentifizierung verwendet, kann Ihr Server die Eingabe von Einzelheiten, wie Benutzername und Kennwort verlangen.

Zunächst versucht IE (Internet Explorer), die NT-Authentifizierungsinformationen des aktuellen Benutzers zu verwenden.

- Wenn IE die Anmeldung mit diesen Informationen erfolgreich durchführen kann und Sie ein Skript aufzeichnen, werden Sie von VuGen am Ende der Aufzeichnung zur Eingabe eines Kennworts aufgefordert. VuGen ruft den Benutzernamen und die Domäneninformationen automatisch ab. Falls erforderlich können Sie den Benutzernamen auch im Dialogfeld **NTLM-Authentifizierung für Webrekorder** bearbeiten.



- Wenn IE die Anmeldung mit diesen Informationen nicht erfolgreich durchführen kann, fordert er Sie zur Eingabe eines Benutzernamens und Kennworts über das Standard-Authentifizierungsdialogfeld des Browsers auf.



Generieren einer "web_set_user"-Funktion

Beim Durchführen von NTLM-Authentifizierung fügt VuGen eine Funktion **web_set_user** zum Skript hinzu.

- Bei erfolgreicher Authentifizierung generiert VuGen eine Funktion **web_set_user** mit Ihrem Benutzernamen, dem verschlüsselten Kennwort und Host.

```
web_set_user("domain1\\dashwood",
         lr_decrypt("4042e3e7c8bbbcfde0f737f91f"),
        "sussex:8080");
```

- Wenn Sie das Dialogfeld **NTLM-Authentifizierung für Webrekorder** ohne Eingabe von Informationen abbrechen, generiert VuGen eine Funktion **web_set_user**, die Sie manuell bearbeiten können.

```
web_set_user("domain1\\dashwood,
        "Enter NTLM Password Here",
        "sussex:8080");
```

Hinweis: Wenn Sie ein Kennwort manuell eingeben, erscheint es unverändert im Skript, und dies stellt ein Sicherheitsproblem dar. Zum Verschlüsseln des Kennworts klicken Sie mit der rechten Taste auf das Kennwort und wählen Sie **Zeichenfolge verschlüsseln** aus. VuGen verschlüsselt die Zeichenkette und generiert eine Funktion **lr_decrypt**, die zum Decodieren des Kennworts bei der Wiedergabe verwendet wird. Weitere Informationen zum Verschlüsseln von Zeichenketten finden Sie unter "Verschlüsseln von Text" auf Seite 432.

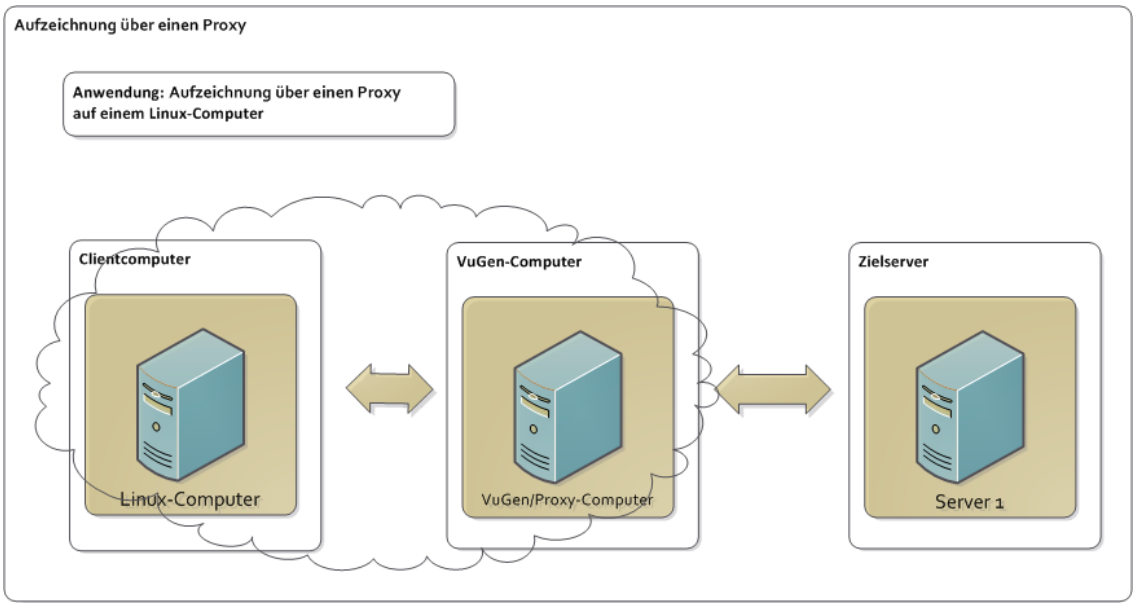
Aufzeichnung über einen Proxy - Übersicht

VuGen ermöglicht die Aufzeichnung von Skripts mithilfe eines LoadRunner-Proxy, um Situationen zu lösen, in denen Sie VuGen nicht auf dem Clientcomputer installieren können. Dies kann bei

bestimmten Linux-Computern, Mac OS-Computern und mobilen Geräten der Fall sein.

Wenn Sie diese Option nutzen, fungiert der VuGen-Computer als Proxyserver, der den gesamten Datenverkehr vom Clientcomputer zum Zielserver erfasst. Nach der Aufzeichnung des Geschäftsprozesses erstellt VuGen ein Skript.

Das folgende Diagramm veranschaulicht den grundlegenden Workflow:



Hinweise zum Aufzeichnen über einen Proxy:

- Sie können nur ein Web-HTTP/HTML-Skript erstellen, wenn Sie über einen Proxy aufzeichnen.
- Der Client muss Proxykonfigurationen zulassen, d. h. Sie müssen in der Lage sein, den Port und die Adresse des VuGen-Computers auf dem Clientgerät oder -computer anzugeben.
- Das Clientgerät bzw. der Clientcomputer und der VuGen-Computer müssen sich im selben Netzwerk befinden.
- Da VuGen den Cache des Browsers und die Verlaufseinstellungen auf dem Clientcomputer nicht umgehen kann, muss der Browserverlauf des Clientcomputers vor dem Aufzeichnen des Geschäftsprozesses gelöscht werden. Dann ist VuGen in der Lage, Ihren Geschäftsprozess über einen Proxy exakt aufzuzeichnen.

HTTP-Weiterleitung an mehrere Ziele

Wenn mehrere Zielcomputer vorhanden sind, kann der VuGen-Proxy die Daten entsprechend der Host-HTTP-Kopfzeile korrekt an den richtigen Zielserver weiterleiten.

Weiterleiten an einen Zielserver über einen Internet-Proxyserver

Sie können den VuGen-Computer so konfigurieren, dass er eine Verbindung mit dem Internet-Proxy Ihrer Organisation herstellt. Wählen Sie hierzu im Dialogfeld **Aufzeichnung starten** den Modus **Remoteanwendung über LoadRunner-Proxy** aus.

Weitere Informationen finden Sie unter "Dialogfeld "Aufzeichnung starten"" auf Seite 272.

Aufzeichnen eines Skripts über einen Proxy

Dieses Thema beschreibt verschiedene Methoden für die Aufzeichnung eines Skripts über einen Proxyserver.

Hinweis: In allen Fällen befinden sich der Clientcomputer und der VuGen-Computer im selben Netzwerk.

Anwendungsszenario 1: Sie möchten einen Geschäftsprozess aufzeichnen, können aber VuGen auf dem Clientcomputer oder Gerät nicht installieren.

1. ### Erstellen eines Skripts

   Wählen Sie **Neues Skript > Web - HTTP/HTML** aus.

2. ### Aufzeichnung starten

   a. Wählen Sie im Dialogfeld **Aufzeichnung starten** nacheinander **Aufzeichnungsmodus > Aufzeichnen > Remoteanwendung über LoadRunner-Proxy** aus. Weitere Informationen finden Sie unter "Dialogfeld "Aufzeichnung starten"" auf Seite 272.

   b. Geben Sie den Port an, den der LoadRunner-Proxy überwachen soll. Standardmäßig ist das Port 8888.

   c. Aktivieren Sie **Aufzeichnungssymbolleiste auf Clientcomputer anzeigen**. Auf diese Weise können Sie die Aufzeichnungssymbolleiste auf dem Clientcomputer sehen und diese verwenden.

   d. Klicken Sie auf **Aufzeichnung starten**.

3. Löschen Sie auf dem Clientcomputer die Daten im Browsercache, u. a. temporäre Internetdateien und Cookies.

4. ### Konfigurieren der Proxyeinstellungen auf dem Clientcomputer

   Legen Sie in den Proxyeinstellungen den VuGen-Computer als Proxyserver fest. Zu diesem Zweck geben Sie die Computeradresse und den Port an, den der LoadRunner-Proxy überwachen soll.

   Es folgen einige Beispielkonfigurationen:

Browser/Betriebssystem	Pfad	Konfiguration
Internet Explorer	▪ **Internetoptionen > Verbindungen > LAN-Einstellungen > Proxyserver** ▪ **Systemsteuerung > Internetoptionen**	a. Wählen Sie **Proxyserver für LAN verwenden** aus. b. Geben Sie den **Port** an. c. Geben Sie die **Adresse** an.
Firefox	**Tools > Options > Network > Advanced > Connection > Settings...**	a. Wählen Sie **Manual proxy configuration** aus. b. Geben Sie den **HTTP Proxy** an. c. Geben Sie den **Port** an. d. Aktivieren Sie **Use this proxy server for all protocols**.

5. **Aufzeichnen des Geschäftsprozesses**
   a. Navigieren Sie zu Ihrer Anwendung.
   b. Führen Sie die Schritte des Geschäftsprozesses aus, den Sie aufzeichnen möchten.
6. **Generieren des Skripts**
   a. Wählen Sie in der **Aufzeichnungssymbolleiste** auf dem Clientcomputer oder auf der **unverankerten Aufzeichnungssymbolleiste** auf dem VuGen-Computer **Aufzeichnung beenden** aus.
   b. VuGen generiert das Skript.

Anwendungsszenario 2: Sie möchten einen Geschäftsprozess aufzeichnen, können aber VuGen nicht auf dem Computer bzw. auf dem Gerät installieren, auf dem die Anwendung ausgeführt wird. Für den Internetzugriff vom Clientcomputer aus ist ein Proxy erforderlich.

1. **Erstellen eines Skripts**
   Wählen Sie **Neues Skript > Web - HTTP/HTML** aus.

2. **Aufzeichnung starten**

   a. Wählen Sie im Dialogfeld **Aufzeichnung starten** nacheinander **Aufzeichnungsmodus > Aufzeichnen > Remoteanwendung über LoadRunner-Proxy** aus.

   b. Ändern Sie ggf. den Port, den der LoadRunner-Proxy überwachen soll. Der Standardport lautet 8888.

   c. Aktivieren Sie **Aufzeichnungssymbolleiste auf Clientcomputer anzeigen**. Auf diese Weise können Sie die Aufzeichnungssymbolleiste auf dem Clientcomputer sehen und diese verwenden.

   d. Während einer Proxy-Aufzeichnung wird auf dem lokalen Computer keine Browsersitzung gestartet. Aus diesem Grund wird im Dialogfeld **Aufzeichnung starten** ein Link bereitgestellt, über den Sie eine Verbindung mit dem Internet-Proxy konfigurieren können.

      Wählen Sie in Internet Explorer **Extras > Internetoptionen > Verbindungen**. Klicken Sie auf **LAN-Einstellungen** und geben Sie Port und Adresse für den Internet-Proxy des Clientcomputers ein.

      Hinweis: Die Option **Automatisches Konfigurationsskript verwenden** wird nicht unterstützt.

   e. Wählen Sie **Aufzeichnung starten** aus.

3. Löschen Sie auf dem Clientcomputer die Daten im Browsercache, u. a. temporäre Internetdateien und Cookies.

4. **Konfigurieren der Proxyeinstellungen auf dem Clientcomputer**

   Legen Sie auf dem Clientcomputer in den Browsereinstellungen IP und Port des VuGen-Computers fest. In der folgenden Tabelle wird erläutert, wie Sie die Proxyeinstellungen konfigurieren.

Browser/Betriebssystem	Pfad	Konfiguration
Internet Explorer	▪ **Internetoptionen > Verbindungen > LAN-Einstellungen > Proxyserver** ▪ **Systemsteuerung > Internetoptionen**	a. Wählen Sie **Proxyserver für LAN verwenden** aus. b. Geben Sie den **Port** an. c. Gebern Sie unter **Adresse** die IP-Adresse des VuGen-Computers ein.

Browser/Betriebssystem	Pfad	Konfiguration
Firefox	**Tools > Options > Network > Advanced > Connection > Settings...**	a. Wählen Sie **Manual proxy configuration** aus. b. Geben Sie den **HTTP Proxy** an. c. Geben Sie den **Port** an. d. Aktivieren Sie **Use this proxy server for all protocols**.

5. **Aufzeichnen des Geschäftsprozesses**
   a. Navigieren Sie zu Ihrer Anwendung.
   b. Führen Sie die Schritte des Geschäftsprozesses aus, den Sie aufzeichnen möchten.
6. **Generieren des Skripts**
   a. Wählen Sie auf der **Aufzeichnungssymbolleiste** auf dem Clientcomputer oder auf der **unverankerten Aufzeichnungssymbolleiste** auf dem VuGen-Computer **Aufzeichnung beenden** aus.
   b. VuGen generiert das Skript.

Anwendungsszenario 3: Die Anwendung kommuniziert über ein SSL-Protokoll.

1. Sie müssen das LoadRunner-SSL-Zertifikat auf den Clientcomputer importieren.

   Hinweis: Als Anwendungsentwickler können Sie bestimmte Richtlinien für das Serverzertifikat festlegen, wenn Sie SSL verwenden. Aber nur wenn das LoadRunner-Zertifikat mit der Richtlinie übereinstimmt, erachtet der Client den Server als vertrauenswürdig; dann kann die SSL-Verbindung festgelegt werden.

2. Zum Herunterladen des Zertifikats navigieren Sie zu `http://<Name des VuGen-Computers>:port/proxyroot.cer` oder `http://<IP-Adresse des VuGen-Computers>:port/proxyroot.cer`.

   Hinweis: Bei Sicherheitseinschränkungen navigieren Sie zu `http://<Name des VuGen-Computers>:port/proxyroot.dat` oder `http://<IP-Adresse des VuGen-Computers>:port/proxyroot.dat`. Wenn das Zertifikat heruntergeladen wurde, ändern

Sie die Erweiterung `.dat` wieder in `.cer`, um das Zertifikat zu importieren.

3. Importieren Sie das SSL-Zertifikat. Die folgende Tabelle enthält Beispiele für den Pfad für verschiedene Browser.

Browser	Pfad
Internet Explorer	**Internetoptionen > Inhalte > Vertrauenswürdige Stammzertifizierungsstellen > Importieren**
Firefox	**Tool > Options > Advanced > View Certificates > Authorities > Import**

Anwendungsszenario 4: Die möchten eine Proxy-Aufzeichnung einer lokalen Anwendung ausführen, für die der System-Proxy verwendet wird, wobei VuGen und die Clientanwendung auf demselben Computer ausgeführt werden.

1. Erstellen eines Skripts

   Wählen Sie **Neues Skript > Web - HTTP/HTML** aus.

2. Legen Sie die Aufzeichnungsoptionen fest.

   Öffnen Sie die Aufzeichnungsoptionen (wählen Sie **Aufzeichnen > Aufzeichnungsoptionen** und dann den Knoten **HTTP-Eigenschaften > Erweitert** aus). Aktivieren Sie **LR-Proxy zum Aufzeichnen einer lokalen Anwendung verwenden**.

3. Aufzeichnung starten

   a. Öffnen Sie das Dialogfeld "Aufzeichnung starten" (Strg + R).

   b. Wählen Sie im **Aufzeichnungsmodus Aufzeichnen: Webbrowser** aus.

4. Führen Sie den Geschäftsprozess aus.

VuGen setzt den Proxy nach der Aufzeichnung automatisch auf die ursprüngliche Einstellung zurück. Wenn die Aufzeichnung nicht ordnungsgemäß beendet wurde, z. B. wenn Ihre Anwendung abgestürzt ist, müssen Sie den Proxy möglicherweise manuell zurücksetzen. **Internetoptionen > Verbindungen > LAN-Einstellungen > Proxyserver**.

Wenn Sie eine Java-Anwendung bzw. einen anderen Browser als Internet Explorer aufzeichnen möchten und die Anwendung nicht auf den Systemproxy eingestellt wurde, müssen Sie den Proxyserver der Anwendung manuell einstellen.

Importieren von Aktionen in ein Skript

Für Vuser-Typen, die mehrere Aktionen unterstützen, können Sie Aktionen von einem anderen Vuser-Skript in Ihr Skript importieren. Sie können nur Aktionen von Vusern desselben Typs

importieren. Beachten Sie, dass mit der importierten Aktion verbundene Parameter mit dem Skript zusammengeführt werden. In den folgenden Schritten wird beschrieben, wie Aktionen in das aktuelle Skript importiert werden.

1. Wählen Sie **Entwurf > Aktion > Aktion importieren** aus, oder klicken Sie mit der rechten Maustaste auf den Lösungs-Explorer, und wählen Sie im Kontextmenü die Option **Aktion importieren** aus. Das Dialogfeld **Aktion in VuGen-Skript importieren** wird geöffnet.

2. Klicken Sie auf **Durchsuchen**, um ein Vuser-Skript auszuwählen. Eine Liste der Aktionen des Skripts enthält der Abschnitt **Zu importierende Aktion**.

3. Wählen Sie die einzubeziehenden Aktionen aus, und klicken Sie auf **Importieren**. Die importierten Aktionen werden im **Lösungs-Explorer** angezeigt.

Erneutes Generieren eines Vuser-Skripts

Wenn Sie das ursprünglich aufgezeichnete Skript wiederherstellen möchten, können Sie es erneut generieren. Dieses Funktion eignet sich ideal zum Debuggen oder zur Fehlerbehebung in einem beschädigten Skript. Bei der erneuten Generierung eines Skripts werden alle manuell zu den aufgezeichneten Aktionen im Skript hinzugefügten Erweiterungen entfernt. Wenn Sie Parameter zu Ihrem Skript hinzugefügt haben, stellt VuGen die ursprünglichen Werte wieder hier. Die Parameterliste wird jedoch nicht gelöscht; Sie können früher erstellte Parameter erneut einfügen. Beachten Sie, dass durch die erneute Generierung nur die aufgezeichneten Aktionen bereinigt werden, nicht jedoch die manuell hinzugefügten.

In dieser Aufgabe wird beschrieben, wie ein Vuser-Skript erneut generiert wird.

1. ### Starten der erneuten Generierung

   Wählen Sie **Aufzeichnen > Skript neu generieren** aus. VuGen gibt eine Warnung aus, dass alle manuellen Änderungen überschrieben werden.

2. ### Ändern der Optionen zur erneuten Generierung - optional

   Klicken Sie auf **Optionen**, um das Dialogfeld **Optionen für erneute Generierung** zu öffnen.

   In einem Mehrfachprotokollskript können Sie über den Knoten **Allgemein > Protokolle** ändern, welche Protokolle aufgezeichnet werden sollen, wenn das Skript erneut generiert wird. Informationen zur Benutzeroberfläche finden Sie unter "Allgemein > Protokolle (Knoten)" auf Seite 193.

   Zum Ändern der Skript-Optionen wählen Sie den Knoten **Allgemein > Skript** aus und aktivieren bzw. deaktivieren das entsprechende Kontrollkästchen. Informationen zur Benutzeroberfläche finden Sie unter "Allgemein > Skript (Knoten)" auf Seite 196.

Hinweis: Wenn für ein Flex-, Silverlight- oder Java over HTTP-Vuser-Skript Fehler während der Codegenerierung festgestellt wurden, werden die Fehler im Bereich **Fehler** von VuGen angezeigt. In diesem Fehlerbereich werden Details zu den einzelnen Fehlern zusammen mit empfohlenen Maßnahmen angezeigt. Befolgen Sie diese Anweisungen und generieren Sie das

Skript neu.

Dialogfeld "Aufzeichnung starten"

Mithilfe dieses Dialogfelds können Sie die Geschäftsprozesse aufzeichnen.

Zugriff	• **VuGen > Aufzeichnen > Aufzeichnen** • [VuGen]-Schaltfläche **Aufzeichnung starten**
Wichtige Informationen	Dieses Dialogfeld wird dynamisch angepasst und ändert sich je nach den ausgewählten Optionen und verwendeten Protokollen.
Relevante Aufgaben	• "Aufzeichnen eines Vuser-Skripts" auf Seite 162 • "Skriptoptionen" auf Seite 83 • "Aufzeichnen eines Skripts über einen Proxy" auf Seite 266

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Alle Protokolle (außer Java)

Element der Oberfläche	Beschreibung
In Aktion aufzeichnen	Der Abschnitt für die Aufzeichnung. Sie können eine eigene Aktion hinzufügen. Geben Sie den Aktionsnamen im Feld **In Aktion aufzeichnen** ein und klicken Sie auf **Hinzufügen**. Die neue Aktion wird zum Skript hinzugefügt.

Forts.

Element der Oberfläche	Beschreibung
Aufzeichnungsmodus	Der zum Aufzeichnen des Geschäftsprozesses verwendete Modus. Sie können folgende Elemente aufzeichnen: • **Webbrowser.** Web und Oracle NCA zeichnen beispielsweise Webanwendungen auf. • **Windows-Anwendung.** Windows Socket-Vuser zeichnen beispielsweise eine Windows-Anwendung auf. • **Remoteanwendung über LoadRunner-Proxy.** Zum Aufzeichnen des Datenverkehrs, wenn VuGen nicht auf dem Clientcomputer ausgeführt werden kann. Beispiel: Linux-Computer, Mac OS-Computer und mobile Geräte. Wenn Sie diesen Modus auswählen, können Sie folgende Optionen angeben: a. **LoadRunner-Proxy überwacht Port:** Der Port, den der LoadRunner-Proxy überwachen soll. b. **Aufzeichnungssymbolleiste auf Clientcomputer anzeigen**. Auf diese Weise können Sie die Aufzeichnungssymbolleiste auf dem Clientcomputer verwenden.
Applikation	Wählen Sie einen Browser oder einen Pfad der Windows-Anwendung aus, die Sie aufzeichnen möchten. Die folgenden Browser werden unterstützt: • Microsoft Internet Explorer • Mozilla Firefox • Google Chrome **Hinweis:** Um in Chrome oder Firefox aufzeichnen zu können, schließen Sie alle Fenster, und überprüfen Sie im Windows-Task-Manager, dass keine Chrome- oder Firefox-Instanzen mehr ausgeführt werden, bevor Sie mit der Aufzeichnung beginnen.
URL-Adresse	Die Anfangs-URL-Adresse. Diese Option wird nur für Internetanwendungen angezeigt.

Forts.

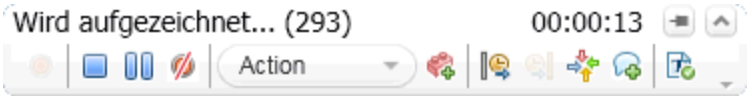
Element der Oberfläche	Beschreibung
Programmargumente	(Nur für Windows-Anwendungen) Das Befehlszeilenargument für die unter **Aufzuzeichnende Anwendung** angegebene ausführbare Datei. Wenn Sie beispielsweise `plus32.exe` (aufzuzeichnende Anwendung) mit den Befehlszeilenoptionen `peter@neptune` angeben, wird der Benutzer `Peter` beim Starten von `plus32.exe` mit dem Server `Neptune` verbunden.
Aufzeichnung starten	Sie können den Geschäftsprozess mit folgenden Optionen aufzeichnen: • **Sofort** - Die Aufzeichnung beginnt, sobald Sie auf die Schaltfläche **Aufzeichnung starten** klicken. • **In verzögertem Modus** - In den folgenden Fällen ist es möglicherweise ratsam, die Aufzeichnung nicht sofort vorzunehmen: ▪ Wenn Sie mehrere Aktionen aufzeichnen; denn in diesem Fall muss der Start nur in einer Aktion ausgeführt werden. ▪ In Fällen, in denen Sie zu einem spezifischen Punkt in der Anwendung navigieren möchten, bevor das Aufzeichnen gestartet wird. ▪ Wenn Sie in ein vorhandenes Skript aufzeichnen.
Arbeitsverzeichnis	Bei Anwendungen, bei denen die Angabe eines Arbeitsverzeichnisses erforderlich ist.
Aufzeichnungsoptionen	Öffnet das Dialogfeld **Aufzeichnungsoptionen**. Informationen zur Benutzeroberfläche finden Sie unter . "Aufzeichnungsoptionen" auf Seite 164.
Schaltfläche "Aufzeichnung starten"	Startet die Aufzeichnung des Geschäftsprozesses auf Grundlage der darüber ausgewählten Optionen: **Sofort** oder **In verzögertem Modus**.

Java-Protokolle

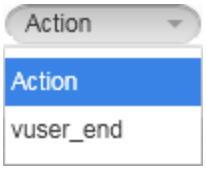
Element der Oberfläche	Beschreibung
In Aktion aufzeichnen	**Der Abschnitt für die Aufzeichnung.**
Aufzeichnen:	• **Java-Applet** zum Aufzeichnen eines Java-Applets über den Appletviewer von Sun. • **Java-Anwendung** zum Aufzeichnen einer Java-Anwendung. • **Internet Explorer** zum Aufzeichnen eines Applets innerhalb eines Browsers. • **Programmdatei\Batch** zum Aufzeichnen von Applets oder Anwendungen, die direkt über eine Batch-Datei oder den Namen einer ausführbaren Datei gestartet werden. • **Listener**, damit VuGen auf eine Batch-Datei wartet, die vor dem Aufzeichnen die Konfiguration initialisiert und eine Anwendung ausführt. Für diesen Modus müssen Sie die Systemvariable _JAVA_OPTIONS as --**Xrunjdkhook** mit jdk1.2.x oder aktueller definieren. (Für JDK 1.x müssen Sie die Umgebungsvariable _**classload_hook=JDKhook** definieren. Für JDK 1.6 definieren Sie _JAVA_OPTIONS als **-agentlib:jdhook**.)
URL-Adresse	Der URL für den Aufzeichnungsbeginn (für Internet Explorer-Aufzeichnungen)
Parameter	Alle zusätzlichen Parameter, die von der Anwendung benötigt werden.
Hauptklasse	Der *vollständige* Pfad der Java-Klasse mit der Hauptmethode. **Hinweis:** Diese Option steht nur für Anwendungen des Typs **Java-Anwendung** zur Verfügung.
Applet-Pfad	Diese Option steht nur für Java-Applets zur Verfügung.
Internet Explorer-Pfad	Diese Option steht nur für Anwendungen des Typs Internet Explorer zur Verfügung.
Programmdatei\Batch	Diese Option steht nur für Anwendungen des Typs **Programmdatei\Batch** zur Verfügung.
Arbeitsverzeichnis	Ein Arbeitsverzeichnis ist nur notwendig, wenn Ihre Anwendung den Ordner kennen muss, in dem das Arbeitsverzeichnis gespeichert ist (beispielsweise zum Lesen von Eigenschaftendateien oder zum Schreiben von Protokolldateien).
Aufzeichnungsoptionen	Öffnet das Dialogfeld **Aufzeichnungsoptionen**. Informationen zur Benutzeroberfläche finden Sie unter "Aufzeichnungsoptionen" auf Seite 164.

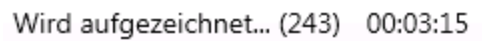
Freischwebende Aufzeichnungssymbolleiste

Die freischwebende Aufzeichnungssymbolleiste ermöglicht Ihnen die Aufzeichnung Ihrer Vuser-Skripts und bietet einfachen Zugriff auf allgemeine Skriptbefehle.

Beispiel für die Benutzeroberfläche	
Zugriff	Die freischwebende Aufzeichnungssymbolleiste wird angezeigt, wenn die Skriptaufzeichnung beginnt.
Wichtige Informationen	• Die freischwebende Aufzeichnungssymbolleiste kann angedockt werden. Weitere Informationen finden Sie unter "VuGen-Layouts - Übersicht" auf Seite 56. • Sie können die Symbolleiste mit der Schaltfläche verankern. • Sie können die Symbolleiste mit der Schaltfläche und erweitern oder ausblenden.
Relevante Aufgaben	"Erstellen oder Öffnen eines Vuser-Skripts" auf Seite 148

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Fortsetzen der Skriptaufzeichnung, nachdem die Aufzeichnung angehalten wurde.
	Beenden der Skriptaufzeichnung.
	Aufzeichnung unterbrechen.
	Abbrechen der Aufzeichnung.
	Auswählen einer Aktion für die Aufzeichnung.
	Erstellen einer neuen Aktion für die Aufzeichnung.

Element der Oberfläche	Beschreibung
	Einfügen eines Schrittes **Transaktion starten** in Ihr Skript. Weitere Informationen finden Sie unter "Übersicht über Transaktionen" auf Seite 425.
	Fügt einen Schritt **Transaktion beenden** in Ihr Skript ein. Weitere Informationen finden Sie unter "Übersicht über Transaktionen" auf Seite 425.
	Einfügen eines Schrittes **Rendezvous-Punkt** in Ihr Skript. Weitere Informationen finden Sie unter "Rendezvous-Punkte " auf Seite 427.
	Fügen Sie einen Kommentar in Ihr Skript ein.
	Einfügen eines Schrittes **Textprüfung** in Ihr Skript. Weitere Informationen finden Sie unter "Überprüfen von Text und Bildern (Web-Vuser-Skripts) – Übersicht" auf Seite 901.
	Zeigt Folgendes an: • Wie viele Ereignisse in Ihrem Skript aufgezeichnet wurden. • Die verstrichene Zeit seit dem Beginn der Aufzeichnung exklusive der Zeit, während der das Skript angehalten war.
	Verankern der Aufzeichnungssymbolleiste oder Lösen der Verankerung.
	Ein- bzw. Ausblenden der Symbolleistenschaltflächen.

Element der Oberfläche	Beschreibung
	Ausblenden der Aufzeichnungssymbolleiste. Die Symbolleiste wird wieder angezeigt, wenn Sie die Anzeige aktualisieren oder zur nächsten Seite navigieren. Das Ausblenden der Symbolleiste kann hilfreich sein, wenn die Symbolleiste Steuerelemente in der Anwendung, die bedient werden sollen, verdeckt und so den Zugriff auf die Steuerelemente verhindert. **Hinweis:** Die Schaltfläche **Symbolleiste ausblenden** wird nur für die Proxy-Aufzeichnung angezeigt.

Bei der Aufzeichnung erstellte Dateien

Angenommen, dem aufgezeichneten Skript wurde der Name **vuser** zugeordnet und es wurde unter **c:\tmp** gespeichert. Die nachfolgende Liste enthält die wichtigsten Dateien, die nach dem Aufzeichnen generiert werden:

Dateiname	Details

vuser.usr	Enthält Informationen zu Vuser-Typ, AUT, Aktionsdateien usw.
	
	Beispiel einer Datei "vuser.usr"
	
	`[General]`
	`Type=Oracle_NCA`
	`DefaultCfg=default.cfg`
	`BuildTarget=`
	`ParamRightBrace=>`
	`ParamLeftBrace=<`
	`NewFunctionHeader=0`
	`MajorVersion=5`
	`MinorVersion=0`
	`ParameterFile=nca_test3.prm`
	`GlobalParameterFile=`
	`[Transactions]`
	`Connect=`
	`[Actions]`
	`vuser_init=init.c`
	`Actions=run.c`
	`vuser_end=end.c`

vuser.bak	Eine Kopie von "Vuser.usr" vor dem letzten Speichern.

default.cfg	Enthält eine Liste aller in VuGen festgelegten Laufzeiteinstellungen (Latenzzeit, Iterationen, Protokoll, Web).
	Beispiel einer Datei "default.cfg"
	`[General]`
	`XlBridgeTimeout=120`
	`[ThinkTime]`
	`Options=NOTHINK`
	`Factor=1`
	`LimitFlag=0`
	`Limit=1`
	`[Iterations]`
	`NumOfIterations=1`
	`IterationPace=IterationASAP`
	`StartEvery=60`
	`RandomMin=60`
	`RandomMax=90`
	`[Log]`
	`LogOptions=LogBrief`
	`MsgClassData=0`
	`MsgClassParameters=0`
	`MsgClassFull=0`

vuser.asc	Die ursprünglich aufgezeichneten API-Aufrufe.
vuser.grd	Enthält die Spaltenkopfzeilen für Tabellen in Datenbankskripts.
default.usp	Enthält die Ausführungslogik des Skripts einschließlich der Festlegungen zur Ausführung von actions-Abschnitten.
init.c	Exakte Kopie der Funktion "Vuser_init", wie im VuGen-Hauptfenster gezeigt.
run.c	Exakte Kopie der Funktion "Action", wie im VuGen-Hauptfenster gezeigt.
end.c	Exakte Kopie der Funktion "Vuser_end", wie im VuGen-Hauptfenster gezeigt.
vdf.h	Eine Kopfzeilendatei von C-Variablendefinitionen, die im Skript verwendet werden.
\Data	Im Ordner "Data" sind alle aufgezeichneten Daten gespeichert, die primär als Sicherung verwendet werden. Sobald sich die Daten in diesem Ordner befinden, werden sie nicht mehr bearbeitet oder verwendet. Beispielsweise ist Vuser.c eine Kopie von run.c.

Fehlerbehebung und Einschränkungen - Aufzeichnung

In diesem Abschnitt werden die Fehlerbehebung und die Einschränkungen für die Aufzeichnung von Skripts mit VuGen beschrieben.

Fehlerbehebung bei fehlenden Schritten

Problem: In Ihrem Skript fehlen Schritte, die Sie aufgezeichnet haben.

Unter **Ausgabebereich > Codegenerierung** wird folgende Warnung angezeigt:

```
Warning: One or more responses are missing or have missing packets. Therefore, a
step may appear to be missing in the script.
This issue can be caused if the recording was stopped before all the responses w
ere received.
If the script is generated from a .pcap file, check if the file has missing pack
ets.
```

Dies kann geschehen, wenn Sie auf **Aufzeichnung beenden** klicken, bevor der Datenverkehr vollständig empfangen wurde.

Schritte für die Behebung: Zeichnen Sie das Skript erneut auf. Stellen Sie sicher, dass alle Seiten und Ressourcen heruntergeladen wurden, bevor Sie auf die Schaltfläche **Aufzeichnung beenden** klicken.

Aufzeichnen mit Internet Explorer 10

Problem: Bei Aufzeichnungen im Internet Explorer (IE) 10 verwendet der Browser Seiten aus dem Cache und zeichnet möglicherweise nicht alle Schritte auf.

Schritte für die Behebung: Konfigurieren Sie den IE 10 vor jeder Aufzeichnung erneut und wählen Sie zur Aktualisierung von Webseiten die Option **Immer vom Server aktualisieren** aus. Drücken Sie dazu nach dem Starten der Aufzeichnungssitzung im IE die Taste F12, um den Bereich "Entwicklertools" zu öffnen. Wählen Sie in diesem Bereich die Option **Cache > Immer vom Server aktualisieren** aus. Diese Option befindet sich in der Regel unten im Browserfenster.

Warnmeldung zu Zertifikat

Wenn Sie VuGen als Benutzer ohne Administratorrechte öffnen, wird während der Aufzeichnung möglicherweise in einem Popup eine Zertifikatwarnung angezeigt. Die Meldung wird automatisch geschlossen und wirkt sich nicht auf die Aufzeichnung aus.

McAfee-Schutzsoftware

McAfee-Schutzsoftware (z. B. Antivirus, insbesondere die Komponente **Host Intrusion Prevention**) hat möglicherweise Auswirkungen auf einige LoadRunner-Komponenten und blockiert die Skriptaufzeichnung.
Lösung: Entfernen Sie die McAfee-Software. Details finden Sie im McAfee-Knowledgebase-Artikel über das manuelle Entfernen von McAfee Agent 4.x unter https://kc.mcafee.com/corporate/index?page=content&id=KB65863.

Firefox als Standardbrowser

Wenn Firefox als Standardbrowser festgelegt ist, funktioniert die Option **Proxyeinstellungen vom Standardbrowser abrufen** nicht (**Laufzeiteinstellungen > Internetprotokoll > Proxy**), und es wird eine direkte Verbindung verwendet.

64-Bit-Aufzeichnung

Im Allgemeinen sollten 64-Bit-Anwendungen, die von einer 32-Bit-Clientversion portiert wurden, ähnlich wie der 32-Bit-Client funktionieren. Es besteht das geringe Risiko, dass neue Clients die Leistung von systemeigenen 64-Bit-Anwendungen in Anspruch nehmen. Wenn beispielsweise 64-

Bit-long-Typen für IDs in Datenbanktabellen verwendet werden, wird der ID-Wert abgeschnitten, und die Abfrage schlägt fehl.

Folgende Richtlinien gelten:

- Die Umgebung für 64-Bit-Aufzeichnungen muss Windows 7 x64 oder Windows 8 x64 (Windows 8 x64 wurde in Service Pack 11.52 hinzugefügt) und eine zu testende 64-Bit-Anwendung (Application Under Test, AUT) aufweisen.
- Die Aufzeichnung unter einem 64-Bit-Betriebssystem für 32-Bit- und 64-Bit-Anwendungen (ausgeführt als 64-Bit-Anwendung) wird unterstützt.
- Es ist nicht möglich, eine Seite aufzuzeichnen, für die Clientzertifikat mit der 64-Bit-Version von Internet Explorer erforderlich ist.
- Die Wiedergabe wird nur bei 32-Bit unterstützt.
- Java über HTTP-Protokoll: Für die Wiedergabe ist JVM 32-Bit erforderlich.
- .NET-Protokoll: Es sind zwei 64-Bit-Typen für .NET-Anwendungen verfügbar (**BeliebigeCPU** und **pure 64-Bit**). LoadRunner unterstützt nur **BeliebigeCPU**. Derzeit gibt es keine Lösung für pure 64-Bit-Anwendungen.
  Für die Wiedergabe verwendet LoadRunner dieselben AnyCPU-DLLs, die für die Aufzeichnung verwendet wurden.

  - **Hinweis**: LoadRunner 11.50 und höher erfordert die Installation von .NET Framework 4. Dieses Paket unterstützt beide Versionen der Bibliotheken für 32- und 64-Bit-Systeme.
- Oracle 2-Tier: Sowohl der 32-Bit- als auch der 64-Bit-Client muss installiert werden (der 32-Bit-Client ist für die Wiedergabe erforderlich).

Correlation Studio

Korrelieren – Übersicht

Das Erstellen eines Vuser-Skripts umfasst die nachfolgend aufgeführten Schritte. Dieses Thema enthält eine Übersicht über den dritten Schritt (Korrelieren eines Vuser-Skripts).

Korrelation wird verwendet, wenn ein aufgezeichnetes Skript einen dynamischen Wert enthält (beispielsweise eine Sitzungs-ID) und daher nicht erfolgreich wiedergegeben werden kann. Um dieses Problem zu beheben, wandeln Sie den dynamischen Wert in eine Variable um und ermöglichen so die erfolgreiche Wiedergabe des Skripts.

Viele Anwendungen und Websites identifizieren beispielsweise eine Sitzung anhand des aktuellen Datums und der aktuellen Uhrzeit. Wenn Sie versuchen, ein auf einer solchen Website aufgezeichnetes Skript wiederzugeben, tritt möglicherweise ein Fehler auf, da sich die aktuelle

Uhrzeit von der aufgezeichneten Uhrzeit unterscheidet. Durch das Korrelieren der Daten können Sie die dynamischen Daten speichern und während der gesamten Ausführung des Szenarios verwenden.

Wenn eine Korrelation erstellt wird, fügt VuGen eine Funktion hinzu, die den dynamischen Wert in einen Parameter extrahiert. Entsprechende Vorkommen des ursprünglichen Werts werden durch einen Parameter ersetzt.

Weitere Informationen finden Sie unter "Registerkarte "Korrelation" [Design Studio] - Übersicht" unten.

Registerkarte "Korrelation" [Design Studio] - Übersicht

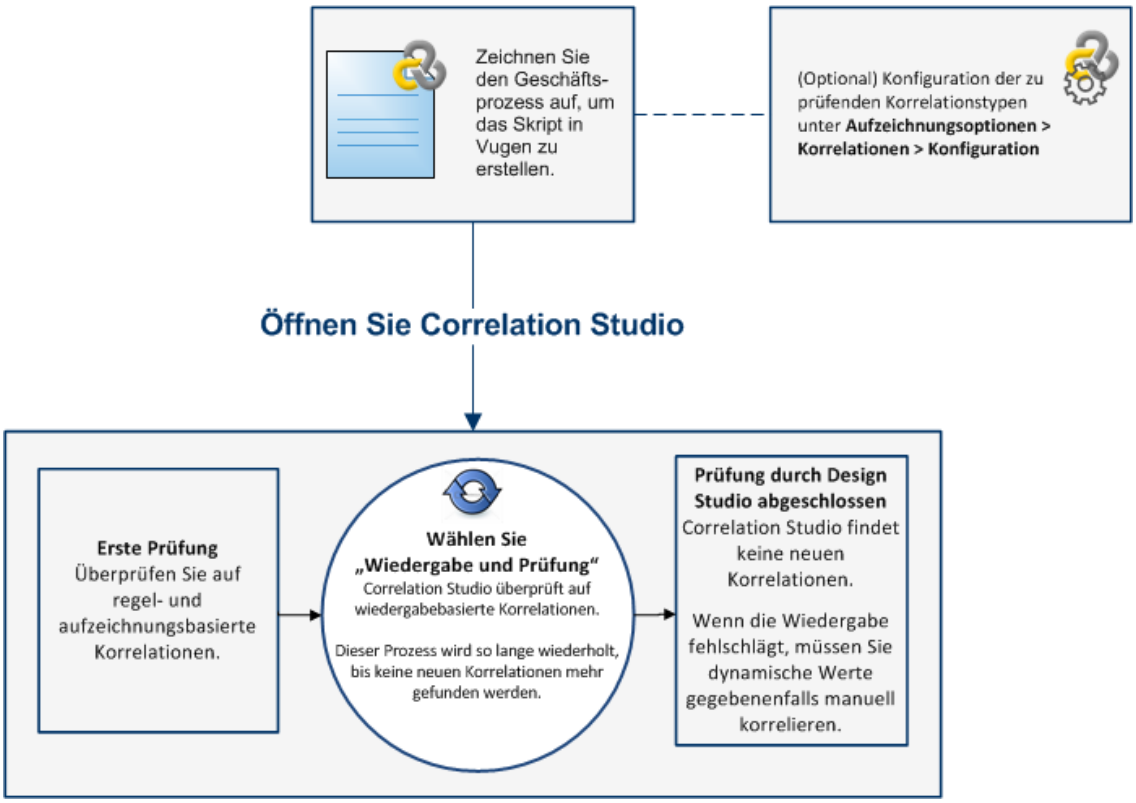
Mit der Registerkarte **Korrelation** können Sie dynamische Werte in Ihren webbasierten Vuser-Skripts korrelieren und verwalten. Weitere Informationen über die Konzepte zur Korrelation finden Sie unter "Korrelieren – Übersicht" auf der vorherigen Seite.

Mit der Registerkarte **Korrelation** können Sie folgende Aktionen durchführen:

- Prüfung auf Korrelationen anhand von Regeln sowie aufzeichnungs- und wiedergabebasierten Engines
- Korrelierung von Roh- und formatierten Daten
- Hinzufügen und Bearbeiten von Regeln
- Rückgängigmachen von Korrelationen
- Überprüfung von Details, die zu einem bestimmten dynamischen Wert in einem Snapshot gehören

Wenn Sie ein Skript anhand eines webbasierten Protokolls aufzeichnen, ändern sich viele Werte bei jedem Senden einer Anforderung an den Server dynamisch. Ein Beispiel für einen dynamischen Wert ist eine `Sitzungs-ID`, die einen Datums- und Zeitstempel für den Zeitpunkt enthalten kann, an dem die Websitzung geöffnet wurde. Weitere Informationen finden Sie unter "Korrelieren von Skripts mit Design Studio" auf Seite 288.

Im folgenden Flussdiagramm ist der Prozess für die Korrelierung von Werten in Ihrem Skript mit der Registerkarte **Korrelation** dargestellt:



Sie werden feststellen, dass die Registerkarte **Korrelation** anhand verschiedener Prozesse prüft, ob dynamische Werte vorliegen.

Korrelationstypen

Design Studio verwendet drei verschiedene Prozesse, um dynamische Werte, die möglicherweise korreliert werden müssen, automatisch zu finden.

- Regeln

  Design Studio prüft zunächst, ob durch Regeln definierte dynamische Werte vorliegen, wenn die Regelprüfung aktiviert wurde. Weitere Informationen finden Sie unter "Korrelationsregeln" auf der nächsten Seite.

- Aufzeichnen

  Design Studio prüft, ob dynamische Werte nach der Codegenerierung auftreten. Mit dieser Methode wird ein signifikanter Prozentsatz der dynamischen Werte in Ihrem Skript gefunden.

- Wiedergabe

  Design Studio prüft, ob dynamische Werte nach der Wiedergabe auftreten. Diese Methode muss möglicherweise mehrmals wiederholt werden.

Sie können auswählen, welche Prüfungstypen die Registerkarte **Korrelation** verwenden soll, indem Sie die Einstellungen unter **Aufzeichnungsoptionen > Korrelationen > Konfiguration** konfigurieren. Generell empfiehlt es sich, alle Prüfungstypen zu aktivieren.

In der folgenden Tabelle wird das erwartete Verhalten bei unterschiedlichen Skriptstatusangaben erläutert:

Skriptstatus beim Öffnen der Registerkarte "Korrelation"	**Verhalten in der Registerkarte "Korrelation"** **(Alle Prüfungstypen aktiviert)**
Skript enthält aufgezeichnete Daten	Wenn Design Studio geöffnet wird, wird die Prüfung auf regel- und aufzeichnungsbasierte Korrelationen durchgeführt. Anschließend können Sie die Wiedergabe und die Prüfung auf wiedergabebasierte Korrelationen durchführen. Wiederholen Sie diesen Prozess, bis Design Studio keine neuen Korrelationen mehr findet.
Skript enthält aufgezeichnete Daten und wurde wiedergegeben	Wenn die Registerkarte **Korrelation** geöffnet wird, wird die Prüfung auf alle Korrelationstypen durchgeführt. Anschließend können Sie die Wiedergabe und die Prüfung auf zusätzliche wiedergabebasierte Korrelationen durchführen. Wiederholen Sie diesen Prozess, bis Design Studio keine neuen Korrelationen mehr findet.

Korrelationsregeln

Wenn Sie vor der Aufzeichnung die dynamischen Werte kennen, die korreliert werden müssen, können Sie Korrelationsregeln erstellen, die diese Werte automatisch während der Aufzeichnung ermitteln. Wenn **Gefundene Werte automatisch korrelieren** im Knoten **Aufzeichnungsoptionen > Korrelationen > Konfiguration** ausgewählt wurde, werden die anhand der Regeln gefundenen Werte automatisch korreliert. Darüber hinaus stellt VuGen einige vordefinierte Korrelationsregeln für unterstützte Anwendungsserver bereit. Sie können die Regeln unter "Korrelationen > Regeln" auf Seite 177 aktivieren bzw. deaktivieren.

Details und Auftreten von Snapshots

Design Studio stellt Details zu jedem Snapshot-Schritt bereit, der dynamische Werte enthält. Mithilfe dieser Details können Sie feststellen, welche Werte in Ihrem Skript korreliert werden müssen. Zusätzlich zu den Snapshot-Details zeigt die Registerkarte **Korrelation** alle Vorkommnisse des dynamischen Werts in Ihrem Skript an. Sie können bestimmte Vorkommnisse für die Korrelation auswählen oder alle korrelieren. Weitere Informationen finden Sie unter "Dialogfeld "Design Studio" [Registerkarte "Korrelation"]" auf Seite 325.

Bestimmen der zu korrelierenden Werte

Sobald Sie eine Liste der Unterschiede erzeugt haben, müssen Sie bestimmen, welche davon korreliert werden sollen. Wenn Sie fälschlicherweise einen Unterschied korrelieren, der nicht korreliert werden muss, kann die Wiedergabe beeinträchtigt werden.

Folgende Zeichenfolgen machen eine Korrelation höchstwahrscheinlich erforderlich:

- **Anmeldezeichenfolge.** Eine Anmeldezeichenfolge mit dynamischen Daten, beispielsweise einer Sitzungs-ID oder einem Zeitstempel.
- **Datum/Zeitstempel.** Eine beliebige Zeichenfolge, die ein Datum, einen Zeitstempel oder andere Benutzeranmeldedaten verwendet.
- **Allgemeines Präfix.** Ein allgemeines Präfix, z. B. **SessionID** oder **CustomerID**, gefolgt von einer Zeichenfolge.

Wenn Sie sich nicht sicher sind, ob ein Unterschied korreliert werden muss, korrelieren Sie nur den entsprechenden Unterschied und führen dann das Skript aus. Überprüfen Sie das Wiedergabeprotokoll, um festzustellen, ob das Problem behoben wurde.

Sie müssen außerdem Unterschiede korrelieren, bei denen einige der aufgezeichneten und abgespielten Zeichenfolgen identisch sind, andere jedoch unterschiedlich. Beispiel: SessionID-Zeichenfolgen mit identischen Prä- und Suffixen, bei denen die Zeichen dazwischen jedoch unterschiedlich sind, müssen korreliert werden.

Ändern von gespeicherten Parametern

Nach dem Speichern eines Wertes in einem Parameter sind möglicherweise Änderungen am Parameter erforderlich, bevor er in Ihrem Skript verwendet werden kann. Zum Durchführen von arithmetischen Operationen an einem Parameter muss der Parameter zunächst über die **atoi**- oder **atol**-C-Funktionen von einer Zeichenfolge in eine Ganzzahl geändert werden. Nach Ändern des Wertes in eine Ganzzahl müssen Sie ihn wieder in eine Zeichenfolge konvertieren, um die neue Variable in Ihrem Skript zu verwenden.

Im folgenden WinSock-Beispiel wurden die Daten an Offset 67 als Parameter **param1** gespeichert. Mithilfe von **atol** hat VuGen die Zeichenfolge in eine lange Ganzzahl konvertiert. Nach Erhöhen des Wertes von **param1** um 1, hat VuGen die Ganzzahl mithilfe von **sprintf** wieder in eine Zeichenfolge konvertiert und als eine neue Zeichenfolge unter **new_param1** gespeichert. Der Wert des Parameters wird über **lr_output_message** angezeigt. Dieser neue Wert kann an einer späteren Stelle im Skript verwendet werden.

```
lrs_receive("socket2", "buf47", LrsLastArg);lrs_save_param("socket2",
                  NULL, "param1", 67, 5);
lr_output_message ("param1: %s", lr_eval_string("<param1>"));
sprintf(new_param1, "value=%ld", atol(lr_eval_string("<param1>")) + 1);
lr_output_message("ID Number:"%s" lr_eval_string("new_param1"));
```

Korrelation und Parametrierung im Vergleich

Parametrierung wird verwendet, wenn Sie einen Wert in eine Variable ändern möchten, um ein realistischeres Skript zu erhalten. Wenn Sie beispielsweise ein Formular auf einer Website ausfüllen, möchten Sie möglicherweise, dass der für ein bestimmtes Feld eingegebene Wert variiert.

Korrelation wird verwendet, wenn ein aufgezeichnetes Skript einen dynamischen Wert enthält (beispielsweise eine Sitzungs-ID) und nicht wiedergegeben werden kann. Um dieses Problem zu beheben, ändern Sie den dynamischen Wert in eine Variable und ermöglichen so die erfolgreiche Wiedergabe des Skripts.

Dienstprogramm für die Wdiff-Korrelation

Mit dem Wdiff-Dienstprogramm können Sie aufgezeichnete Vuser-Skripts und Wiedergabeergebnisse vergleichen, um zu bestimmen, welche Werte korreliert werden müssen.

Für eine effiziente Verwendung von *WDiff* zeichnen Sie dieselbe Operation zweimal auf und vergleichen die Skripts (oder Datendateien für WinSock). WDiff zeigt Unterschiede gelb hinterlegt an. Beachten Sie, dass nicht alle Unterschiede auf einen zu korrelierenden Wert hinweisen. Beispielsweise ist für bestimmte Empfangspuffer, die die Zeit der Ausführung angeben, keine Korrelation erforderlich.

Weitere Informationen über die Aufgabe finden Sie unter "Suchen nach Werten, für die eine Korrelation erforderlich ist" auf Seite 312.

Korrelieren von Java-Skripts

Der Java-Recorder von VuGen versucht, die Anweisungen im generierten Skript automatisch zu korrelieren. Korrelationen werden nur für Java-Objekte ausgeführt. Stößt der Recorder während der Aufzeichnung auf einen elementaren Java-Datentyp (Byte, Zeichen, Logisch, Ganzzahl, Gleitkomma, doppelte Genauigkeit, Kurz und Lang), werden die Argumentwerte im Skript nicht mit Variablen verknüpft. VuGen korreliert automatisch alle Objekte, Objektarrays und Arrays aus elementaren Datentypen. Beachten Sie, dass Java-Arrays und -Zeichenfolgen auch als Objekte gelten.

In VuGen gibt es mehrere Ebenen der Korrelation: Standard, Erweitert, Zeichenfolgen. Die Korrelation wird über die Aufzeichnungsoptionen aktiviert bzw. deaktiviert. Wenn keine der oben genannten Methoden angewendet werden kann, steht zusätzliche die Methode der Serialisierung zur Verfügung, um Skripts zu verarbeiten.

Standardkorrelation

Die Standardkorrelation ist die automatische Korrelation, die während der Aufzeichnung einfacher Objekte (d. h. keine Objektarrays, Vektoren und Containerkonstrukte) ausgeführt wird.

Wenn die aufgezeichnete Anwendung eine Methode aufruft, die ein Objekt zurückgibt, zeichnet der Korrelationsmechanismus von VuGen diese Objekte auf. Bei Ausführung des Skripts vergleicht VuGen die generierten Objekte mit den aufgezeichneten Objekten. Bei Übereinstimmung wird das entsprechende Objekt verwendet. Im folgenden Beispiel werden die beiden CORBA-Objekte **my_bank** und **my_account** verwendet. Das erste Objekt, **my_bank**, wird aufgerufen, das zweite Objekt, **my_account**, wird korreliert und als Parameter in der letzten Zeile des Segments übergeben:

```
public class Actions {
  // Public function: init
      public int init() throws Throwable {
       Bank my_bank = bankHelper.bind("bank", "shunra");
```

```
        Account my_account = accountHelper.bind("account","shunra");
        my_bank.remove_account(my_account);
       }
:
}
```

Erweiterte Korrelation

Die erweiterte oder **tiefe** Korrelation ist die automatische Korrelation, die während der Aufzeichnung komplexer Objekte, d. h. von Objektarrays und CORBA-Containerkonstrukten, ausgeführt wird.

Bei der tiefen Korrelation werden CORBA-Konstrukte (Strukturen, Vereinigungen, Sequenzen, Arrays, Halter, Any-Datentypen) als Container behandelt. Hierdurch wird es möglich, innere Mitglieder von Containern, zusätzliche Objekte oder andere Container zu referenzieren. Immer, wenn ein Objekt aufgerufen oder als Parameter weitergegeben wird, wird es auch mit den inneren Mitgliedern der Container verglichen.

Im folgenden Beispiel führt VuGen eine tiefe Korrelation durch Referenzierung eines Arrayelements aus. Das Objekt **remove_account** empfängt ein Objekt vom Typ **account** als Parameter. Während der Aufzeichnung durchsucht der Korrelationsmechanismus das zurückgegebene Array **my_ accounts** und stellt fest, dass sein sechstes Element als Parameter weitergegeben werden soll.

```
public class Actions {
   // Public function: init
   public int init() throws Throwable {
      my_banks[] = bankHelper.bind("banks", "shunra");
      my_accounts[] = accountHelper.bind("accounts","shunra");
      my_banks[2].remove_account(my_accounts[6]);
    }
:
}
```

Im folgenden Segment wird die erweiterte Korrelation weiter veranschaulicht. Das Skript ruft das Objekt **send_letter** auf, das ein Argument vom Typ **address** erhalten hat. Der Korrelationsmechanismus ruft das innere Mitglied, **address**, im sechsten Element des Arrays **my_ accounts** ab.

```
public class Actions {
   // Public function: init
   public int init() throws Throwable {
      my_banks = bankHelper.bind("bank", "shunra");
      my_accounts = accountHelper.bind("account", "shunra");
      my_banks[2].send_letter(my_accounts[6].address);
    }
:
}
```

Zeichenfolgenkorrelation

Der Begriff Zeichenfolgenkorrelation bezieht sich auf die Darstellung eines aufgezeichneten Werts entweder als eine Zeichenfolge oder als eine Variable. Wenn Sie die Zeichenfolgenkorrelation deaktivieren (die Standardeinstellung), wird der tatsächlich aufgezeichnete Wert der Zeichenfolge explizit innerhalb des Skripts ausgewiesen. Wenn Sie die Zeichenfolgenkorrelation aktivieren, wird

für jede Zeichenfolge eine Variable generiert, die später im Skript wiederverwendet werden kann.

Im folgenden Segment ist die Zeichenfolgenkorrelation aktiviert, d. h. der von der Methode **get_id** zurückgegebene Wert wird in einer Variable vom Typ **string** zur späteren Verwendung im Skript gespeichert.

```
public class Actions {
   // Public function: init
   public int init() throws Throwable {
      my_bank = bankHelper.bind("bank", "shunra");
      my_account1 = accountHelper.bind("account1", "shunra");
      my_account2 = accountHelper.bind("account2", "shunra");
      string = my_account1.get_id();
      string2 = my_account2.get_id();
      my_bank.transfer_money(string, string2);
    }
:
}
```

Korrelieren von Skripts mit Design Studio

In diesem Abschnitt wird die Verwendung der Registerkarte **Korrelation** zur Korrelation von Vuser-Skripts beschrieben.

Voraussetzungen

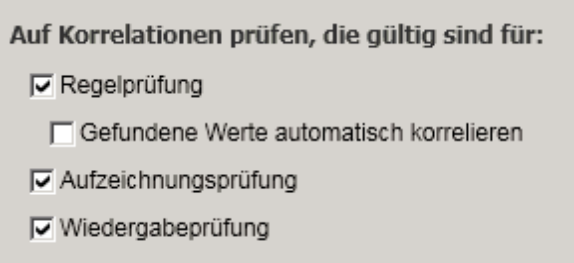
1. Zeichnen Sie ein Skript mit einem der folgenden Protokolle auf:
   - Web HTTP/HTML
   - Flex
   - RTMP/RTMPT
   - Citrix
   - SAP Web
   - Oracle NCA

   Hinweis: Sie können nur die Web-HTTP/HTML-Schritte innerhalb Ihres Oracle NCA-Skripts korrelieren, und das Web-HTTP/HTML-Protokoll muss aktiv sein. Wählen Sie zum Aktivieren des Protokolls **Aufzeichnungsoptionen > Protokoll > Aktive Protokolle > Web HTTP/HTML**.

   Weitere Informationen über die manuelle Korrelation finden Sie unter "Korrelieren von Skripts - Oracle NCA" auf Seite 301.

2. Überprüfen Sie, ob unter **Aufzeichnen > Aufzeichnungsoptionen > Korrelationen >**

Konfiguration alle Prüfungstypen aktiviert sind.



Verwenden der Registerkarte "Korrelation"

1. **Erste Prüfung:** Klicken Sie auf die Schaltfläche **Design Studio**. Das Dialogfeld **Design Studio** wird geöffnet, das eine Prüfung auf antwortbasierte (oder aufzeichnungsbasierte) Korrelationen durchführt und Korrelationsregeln anwendet. Die Statusanzeige im Dialogfeld gibt an, ob die erste Prüfung erfolgreich war.

   Details zur Registerkarte **Korrelation** finden Sie unter "Registerkarte "Korrelation" [Design Studio] - Übersicht" auf Seite 282.

2. Sie können die zu korrelierenden Werte auswählen, indem Sie den Wert im Raster markieren und auf die Schaltfläche **Korrelieren** klicken.

   Erfolgt die Korrelation eines Werts, fügt VuGen eine **web_reg_save_param_***-Funktion hinzu und speichert den Originalwert in einem Kommentar im Skript.

   Sie können die Details der Korrelation überprüfen, indem Sie das **Details**-Chevron im Dialogfeld erweitern. Weitere Informationen finden Sie unter "Dialogfeld "Design Studio" [Registerkarte "Korrelation"]" auf Seite 325.

   Details zur Registerkarte **Korrelation** finden Sie unter "Registerkarte "Korrelation" [Design Studio] - Übersicht" auf Seite 282.

3. Sie können auf die Schaltfläche **Als Regel hinzufügen** klicken, um einen Regeltyp hinzuzufügen. Außerdem können Sie auf die Schaltfläche **Regel bearbeiten** klicken und die entsprechende Regel bearbeiten, sofern der dynamische Wert durch eine Regel korreliert wurde. Weitere Informationen finden Sie unter Aufzeichnungsoptionen > Korrelation > Regeln.

4. **Wiedergabe und Prüfung:** Nachdem Sie die Werte korreliert haben, klicken Sie auf die Schaltfläche **Wiedergabe und Prüfung**. Die Statusanzeige auf der Registerkarte **Korrelation** gibt an, ob weitere Korrelationen gefunden wurden. Auch hier können Sie die zu korrelierenden Werte auswählen, indem Sie den Wert im Raster markieren und auf die Schaltfläche **Korrelieren** klicken. Möglicherweise müssen Sie diesen Schritt mehrmals wiederholen.

5. **Prüfung abgeschlossen:** Findet Design Studio keine weiteren neuen Korrelationen, wird in der Statusanzeige die Meldung "Prüfung durch Design Studio abgeschlossen" angezeigt.

Tipp: Wenn Design Studio nicht alle korrelationsbasierten Fehler beheben kann, versuchen Sie, die Fehler mithilfe der manuellen Korrelation zu lösen. Weitere Informationen finden Sie unter "Manuelles Korrelieren von Skripts" unten.

Manuelles Korrelieren von Skripts

Wenn mithilfe der Überprüfung auf Korrelationen nicht alle korrelationsbasierten Fehler in Ihrem Skript behoben werden konnten, können Sie versuchen, Ihr Skript folgendermaßen manuell zu korrelieren:

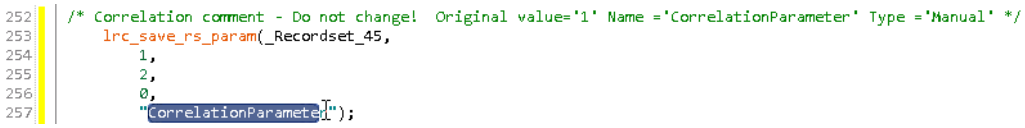
1. **Suchen Sie nach Werten, für die eine manuelle Korrelation erforderlich ist.** Es gibt eine Reihe unterschiedlicher Möglichkeiten, um manuell nach Werten zu suchen, für die eine Korrelation erforderlich ist. Weitere Informationen finden Sie unter "Suchen nach Werten, für die eine Korrelation erforderlich ist" auf Seite 312.

2. **Korrelieren Sie den Wert.**

   Wählen Sie eine der folgenden Methoden aus:

   - **Korrelieren aus Snapshots.** Markieren Sie den zu korrelierenden Wert und wählen Sie im Kontextmenü **Korrelation erstellen** aus.

     Erfolgt die Korrelation eines Werts, fügt VuGen den Korrelationsparameter hinzu und speichert den Originalwert in einem Kommentar im Skript.



   - **Manuelles Hinzufügen von Korrelationsfunktionen.** Fügen Sie in Ihrem Skript die relevanten Korrelationsfunktionen manuell hinzu. Weitere Informationen finden Sie unter "Korrelieren von Skripts - Web (Manuell)" auf Seite 294.

Korrelieren von Skripts aus einem Snapshot

Die folgenden Schritte beschreiben, wie Sie Skripts aus einem Snapshot korrelieren.

Diese Aufgabe gilt für die folgenden Protokolle:

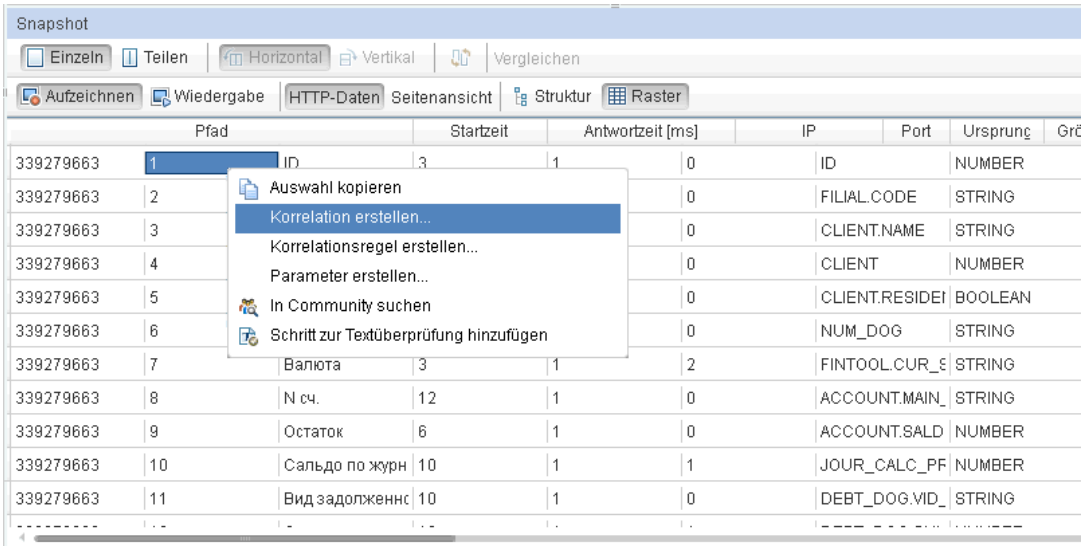
- Datenbankprotokolle
- RTMP

- COM-Protokolle

1. Öffnen Sie den **Ausgabebereich**.

   Wählen Sie **Ansicht** > **Ausgabefenster** aus, um die Registerkarten am unteren Fensterrand anzuzeigen. Suchen Sie in der Registerkarte **Wiedergabe** nach Fehlern. In den meisten Fällen können diese Fehler durch eine Korrelation korrigiert werden.

2. Wählen Sie den entsprechenden Schritt im **Schrittenavigator** aus, und zeigen Sie den Schritt im **Snapshot-Bereich** an. Klicken Sie mit der rechten Maustaste auf den Wert im Snapshot und wählen Sie **Korrelation erstellen**. Dadurch wird das Fenster **Design Studio** geöffnet.



3. Sie können den zu korrelierenden Wert auswählen, indem Sie den Wert im Raster markieren und auf die Schaltfläche **Korrelieren** klicken.

4. Erfolgt die Korrelation eines Werts, fügt VuGen den Korrelationsparameter hinzu und speichert den Originalwert in einem Kommentar im Skript.

```
/* Correlation comment - Do not change!  Original value='1' Name ='CorrelationParameter' Type ='Manual' */
    lrc_save_rs_param(_Recordset_45,
        1,
        2,
        0,
        "CorrelationParameter");
```

Korrelieren von Winsock-Skripts

VuGen Design Studio stellt die Benutzeroberfläche für die Korrelierung von Vuser-Skripts bereit. Diese ist für Arbeit mit dynamischen Daten erforderlich. In WinSock-Vuser-Skripts werden häufig Ports verwendet, deren Nummern dynamisch zugewiesen werden. Einige Anwendungen nutzen immer denselben Port, andere verwenden jeweils den nächsten verfügbaren. Wenn beim Wiedergeben eines Skripts der aufgezeichnete Port nicht mehr verfügbar ist, tritt ein Fehler auf. Um

dieses Problem zu umgehen, ist eine Korrelierung erforderlich, d. h. die tatsächlichen Laufzeitwerte werden gespeichert und im Skript verwendet.

VuGen verwendet die Funktionen **lrs_save_param** und **lrs_save_searched_string**, um Winsock-Skripts zu korrelieren. Das bedeutet, dass die empfangenen Daten für eine spätere Verwendung im Test gespeichert werden. Bei der Korrelation werden empfangene Daten gespeichert, daher gilt sie lediglich für Empfangspuffer und nicht für Sendepuffer. Es wird empfohlen, im Empfangspuffer eine Zeichenfolge mit dynamischen Daten für die Korrelation auszuwählen. Verwenden Sie in einem nachfolgenden Sendepuffer denselben Parameter.

Korrelation von Winsock-Skripts

Beginnen Sie die Korrelation von Winsock-Vuser-Skripts im Snapshot-Bereich. Die Korrelationsfunktionen sind sowohl auf der Registerkarte **Text** als auch auf der Registerkarte **Hexadezimal** des Snapshot-Bereichs enthalten.

1. Wählen Sie im Snapshot-Bereich die Daten aus, die Sie korrelieren möchten.

2. Klicken Sie mit der rechten Maustaste auf die Auswahl und wählen Sie **Korrelation erstellen** oder **Grenzkorrelation erstellen** aus. Design Studio wird mit der Registerkarte **Korrelation** geöffnet.

   Sie können auf die Anzahl der Vorkommen in der Spalte **Ersetzen/Gefunden** klicken, um die genaue Anzahl der Vorkommen auszuwählen, die Sie korrelieren möchten.

3. Klicken Sie auf **Details**, um die Details zur Korrelation anzuzeigen.

4. Vergewissern Sie sich, dass die Registerkarte **Ursprünglicher Schritt für Snapshot** sichtbar ist. Der Typ ist entweder **Datenbereich** oder **Begrenzungsbasiert**.

5. Klicken Sie auf **Korrelieren**, um die Korrelation des Vuser-Skripts durchzuführen.

6. Klicken Sie auf **Schließen**, um Design Studio zu schließen. Beachten Sie, dass VuGen die geeigneten Korrelationsfunktionen und Kommentare in das Skript eingefügt hat.

Weitere Informationen zur Verwendung von Design Studio finden Sie unter "Registerkarte "Korrelation" [Design Studio] - Übersicht" auf Seite 282.

Parametrierung vs. Korrelation

Diese Art der Korrelierung darf nicht mit der einfachen Parametrierung verwechselt werden. Die einfache Parametrierung (**Entwurf > Parameter > Neuen Parameter erstellen**) gilt nur für Daten im Sendepuffer. Hierbei richten Sie einen Parameter ein und weisen ihm mehrere Werte zu. VuGen verwendet die verschiedenen Werte in den einzelnen Skriptausführungen oder Iterationen. Details finden Sie unter "Korrelation und Parametrierung im Vergleich" auf Seite 285.

Informationen zur manuellen Korrelation eines Winsock-Vuser-Skripts finden Sie unter "Korrelieren von Skripts - Winsock (Manuell)" auf der nächsten Seite.

Korrelieren von Skripts - Winsock (Manuell)

In diesem Abschnitt wird die Verwendung des Editors zur manuellen Korrelation eines Winsock-Vuser-Skripts beschrieben.

1. Fügen Sie die Anweisung **lrs_save_param_ex** an der Stelle in Ihr Skript ein, an der Sie die Pufferinhalte speichern möchten. Sie können Benutzer-, statische und Empfangspuffer speichern.

```
lrs_save_param_ex (socket, type, buffer, offset, length, encoding, paramete
r);
```

2. Zeigen Sie den Pufferinhalt an, indem Sie **data.ws** im Bereich **Aktion** des VuGen-Hauptfensters auswählen (dieses wird standardmäßig im Editor angezeigt). Suchen Sie die Daten, die Sie durch den Inhalt des gespeicherten Puffers ersetzen möchten. Ersetzen Sie alle Vorkommen des aufgezeichneten Werts durch den Parameternamen, den Sie in Parameterklammern setzen. Die Standardparameterklammern sind **< >** oder **( )**. Sie können die Parameterklammern unter **Extras** > **Optionen** > Registerkarte **Parameter** ändern.

   Im folgenden Beispiel hat ein Benutzer eine Telnet-Sitzung ausgeführt. Der Benutzer hat die Prozess-ID (PID) mit dem Befehl **ps** bestimmt und eine Anwendung beendet, die auf dieser PID basiert.

```
frodo:/u/jay>ps
   PID TTY      TIME CMD
14602 pts/18    0:00 clock
14569 pts/18    0:03 tcsh
frodo:/u/jay>kill 14602
[3]    Exit 1                clock
frodo:/u/jay>
```

   Während der Ausführung der Sitzung ändert sich die PID der Prozedur (Linux weist bei jeder Ausführung eindeutige PIDs zu); daher ist die Beendigung der aufgezeichneten PID wirkungslos. Verwenden Sie zur Umgehung dieses Problems die Funktion **lrs_save_param_ex**, die die aktuelle PID als Parameter speichert. Ersetzen Sie die Konstante durch den Parameter.

3. Bestimmen Sie in der Datei **data.ws** den Empfangspuffer für die Daten, **buf47**.

```
recv buf47 98
    "\r"
    "\x00"
    "\r\n"
    "   PID TTY      TIME CMD\r\n"
    " 14602 pts/18   0:00 clock\r\n"
    " 14569 pts/18   0:02 tcsh\r\n"
    "frodo:/u/jay>"
.
.
```

```
.
send buf58
    "kill 14602"
```

4. Legen Sie im Bereich **Aktionen** den von buf47 verwendeten Socket fest. Im vorliegenden Beispiel ist dies **socket1**.

```
lrs_receive("socket1", "buf47", LrsLastArg);
```

5. Bestimmen Sie den Offset und die Länge der zu speichernden Datenzeichenfolge. Der Offset der **PID** ist 11, die Länge beträgt 5 Byte. Weitere Informationen zum Anzeigen der Daten finden Sie unter "Datenpuffer" auf Seite 1061.

6. Fügen Sie im Bereich **Aktionen** nach der **lrs_receive**-Funktion des entsprechenden Puffers eine **lrs_save_param_ex**-Funktion ein. In diesem Beispiel lautet der Puffer **buf47**. Die PID wird als Parameter mit dem Namen **param1** gespeichert. Mit der Funktion **lr_output_ message** geben Sie den Parameter aus.

```
lrs_receive("socket1", "buf79", LrsLastArg);
lrs_save_param("socket1", "user", buf47, 11, 5, ascii, param1);

lr_output_message ("param1: %s", lr_eval_string("<param1>"));
lr_think_time(10);
lrs_send("socket1", "buf80", LrsLastArg);
```

7. Legen Sie in der Datendatei **data.ws** die Daten fest, die durch einen Parameter, die PID, ersetzt werden sollen.

```
send buf58
    "kill 14602"
```

8. Ersetzen Sie den Wert durch den Parameter, den Sie in eckige Klammern setzen.

```
send buf58
    "kill <param1>"
```

Korrelieren von Skripts - Web (Manuell)

Im Rahmen dieser Aufgabe wird das manuelle Korrelieren von Webskripts durch Ändern des Codes beschrieben.

1. ## Suchen nach der Zeichenfolge und den zugehörigen Details

   Identifizieren Sie die Anweisung mit den dynamischen Daten und Mustern, die die Speicherorte der Daten charakterisieren. Bei diesen Mustern kann es sich um Begrenzungen oder XPaths handeln.

   a. **Identifizieren von Mustern mithilfe von Begrenzungen**

Verwenden Sie die nachstehenden Richtlinien zum Bestimmen und Festlegen der Begrenzungen der dynamischen Daten:

- Analysieren Sie den Speicherort der dynamischen Daten innerhalb der HTTP-Antwort.
- Ermitteln Sie die Zeichenfolge, die sich unmittelbar links von den dynamischen Daten befindet. Diese Zeichenfolge definiert die linke Begrenzung der dynamischen Daten.
- Ermitteln Sie die Zeichenfolge, die sich unmittelbar rechts von den dynamischen Daten befindet. Diese Zeichenfolge definiert die rechte Begrenzung der dynamischen Daten.
- Die rechten und linken Begrenzungen sollten so eindeutig wie möglich sein, um die Zeichenketten besser zu finden.
- **web_reg_save_param_ex** sucht nach den Zeichen zwischen den (jedoch nicht einschließlich der) angegebenen Begrenzungen und speichert die Informationen ab dem ersten Byte nach der linken Begrenzung bis zum ersten Byte vor der rechten Begrenzung. **web_reg_save_param_ex** unterstützt keine eingebetteten Begrenzungszeichen.
  Beispiel: Wenn der Eingabepuffer `{a{b{c}` lautet und **{** als linke Begrenzung und **}** als rechte Begrenzung angegeben ist, ist die erste Instanz **c** und es gibt keine weiteren Instanzen. Die rechten und linken Begrenzungen wurden gefunden, eingebettete Begrenzungen sind jedoch nicht zulässig, daher ist **c** die einzige gültige Entsprechung.

Standardmäßig beträgt die maximale Länge einer beliebigen Begrenzungszeichenfolge 256 Zeichen. Schließen Sie eine Funktion **web_set_max_html_param_len** in das Skript ein, um die maximal zulässige Länge zu erhöhen. Folgende Funktion erhöht beispielsweise die maximale Länge auf 1024 Zeichen:

Diese Längenbeschränkungen gelten nicht, wenn für die linken oder rechten Begrenzungen nichts angegeben wurde.

b. **Identifizieren von Mustern mithilfe von XPaths**

Verwenden Sie den Ausschnitt für Snapshots, um manuell nach dem XPath für die gewünschte Zeichenfolge zu suchen.

Die maximale Länge einer Begrenzungszeichenfolge ist auf 256 Zeichen beschränkt. Fügen Sie die Funktion **web_set_max_html_param_len**in Ihr Skript ein, um die maximal zulässige Länge zu erhöhen. Folgende Funktion erhöht beispielsweise die maximale Länge auf 1024 Zeichen:

Diese Längenbeschränkungen gelten nicht, wenn für die linken oder rechten Begrenzungen nichts angegeben wurde.

2. Hinzufügen der web_reg_save_param_*-Funktion

Fügen Sie die Funktion **web_reg_save_param_ex** oder **web_reg_save_param_xpath** im Skript vor der Anweisung mit den dynamischen Daten hinzu.

a. **web_reg_save_param_ex**

Diese Funktion durchsucht Serverantworten in Webschritten nach der linken und rechten Begrenzung der Zeichenfolge und speichert die Zeichenfolge in einem Parameter, der im Argument der Funktion benannt wurde. Wenn die angegebene Anzahl an Vorkommen gefunden wurde, durchsucht **web_reg_save_param_ex** keine weiteren Antworten mehr. Weitere Informationen finden Sie in der Function Reference **(Hilfe > Funktionsreferenz**).

b. **web_reg_save_param_xpath**

Mithilfe dieser Funktion werden die Serverantworten in Webschritten für einen bestimmten XPath durchsucht. Die Zeichenfolge, die sich im angegebenen XPath befindet, wird in einem Parameter gespeichert, der im Argument der Funktion benannt wird. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

3. ### Ersetzen von Daten durch einen Parameter

Wählen Sie im VuGen-Hauptfenster **Bearbeiten** > **Ersetzen** aus, um das Dialogfeld **Suchen und ersetzen** zu öffnen. Durchsuchen Sie das gesamte Skript nach dynamischen Daten und ersetzen diese durch einen Parameter. Geben Sie dem Parameter einen beliebigen Namen und setzen Sie ihn zwischen geschweifte Klammern: `{param_name}`. Sie können maximal 64 Parameter pro Skript aufnehmen.

Korrelieren von Skripts - Siebel-Protokoll

Mit den folgenden Schritten können Sie Siebel-Web-Vuser-Skripts korrelieren.

Korrelationsbibliothek

Um Sie bei der Korrelation zu unterstützten, hat Siebel als Teil von Siebel Application Server 7.7 eine Korrelationsbibliothek veröffentlicht. Diese Bibliothek wird ausschließlich von Siebel angeboten. Die Bibliotheksdatei, **ssdtcorr.dll**, befindet sich bei Windows-Installationen im Verzeichnis **siebsrvr\bin** und bei Linux-Installationen im Verzeichnis **siebsrvr/lib**.

ssdtcorr.dll muss auf allen Systemen, auf denen Load Generator oder Controller installiert ist, zur Verfügung stehen. Zur Unterstützung dieser Bibliothek ist mindestens VuGen 8.0 erforderlich. In den folgenden Schritten wird beschrieben, wie Sie die Korrelation mit dieser Bibliothek aktivieren.

1. Kopieren Sie die DLL-Datei in den Ordner **bin** der Produktinstallation.
2. Öffnen Sie ein Mehrfachprotokoll-Skript mit dem **Siebel-Web**-Vuser-Typ.
3. Aktivieren Sie die UTF-8-Unterstützung im Knoten **Aufzeichnungsoptionen > HTTP-Eigenschaften > Erweitert**.
4. Klicken Sie in den Aufzeichnungsoptionen auf den Knoten **Korrelationen** und dann auf **Importieren**. Importieren Sie die Regeldatei **WebSiebel77Correlation.cor** aus dem Ordner **\dat\webrulesdefaultsetting**. Wenn Sie Warnmeldungen erhalten, klicken Sie auf **Überschreiben**.

Um die Standardkorrelation wiederherzustellen, löschen Sie alle Siebel-Regeln und klicken auf **Standardwerte verwenden**.

Wenn Sie die Siebel-Korrelationsbibliothek verwenden, vergewissern Sie sich, dass die Regeln für die SWE-Zählung (bei denen unter **Linke Begrenzung** die Zeichenfolge **SWEC** angegeben ist) nicht deaktiviert sind.

Korrelationsregeln

VuGens integrierte Regeln für den Siebel-Server erkennen dessen Variablen und Zeichenketten und speichern diese für die spätere Verwendung automatisch innerhalb des Skripts. Die Regeln enthalten die Kriterien für die Begrenzungen, die nur für Siebel-Serverzeichenfolgen gelten.

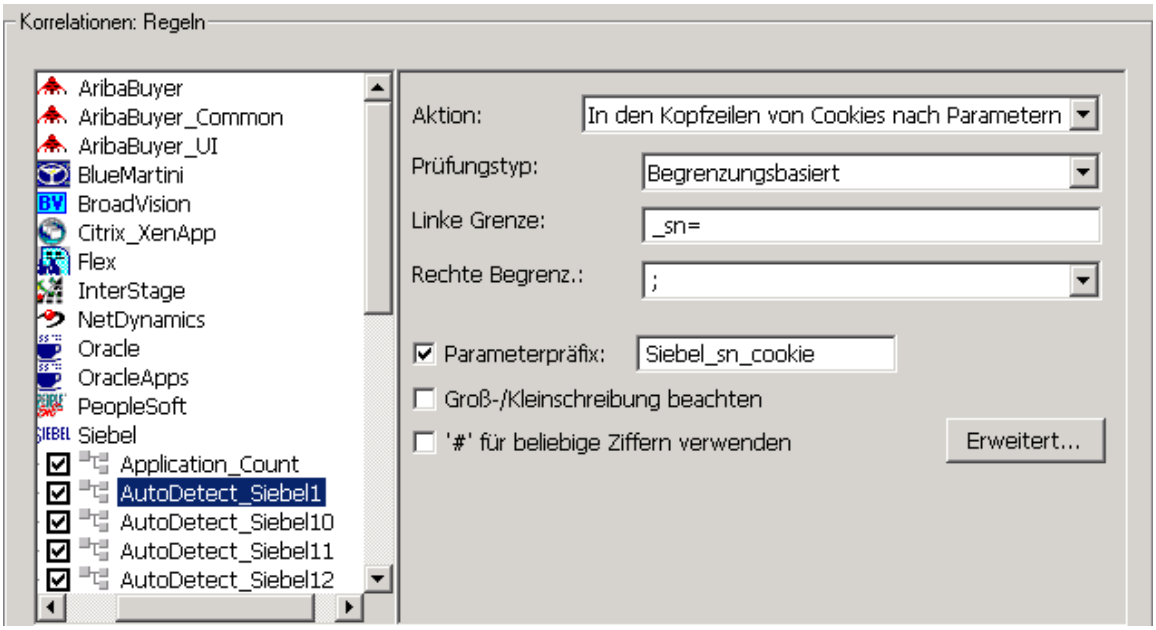
Sobald VuGen ein übereinstimmendes Begrenzungskriterium erkennt, wird der Wert innerhalb der Begrenzung in einem Parameter gespeichert. Der Wert kann eine einfache Variable oder eine öffentliche Funktion sein.

Normalerweise ist es nicht erforderlich, Regeln zu deaktivieren. In einigen Fällen ist es jedoch von Vorteil, nicht zutreffende Regeln zu deaktivieren. Prüfregeln für japanische Inhalte können beispielsweise beim Testen rein deutschsprachiger Anwendungen deaktiviert werden.

Ein weiterer denkbarer Grund, eine Regel zu deaktivieren, ergibt sich, wenn der Controller explizit vorgibt, dass eine Fehlerbedingung generiert werden muss. Überprüfen Sie die Regeleigenschaften in den Aufzeichnungsoptionen und ermitteln Sie die für Ihre Anwendung erforderlichen Bedingungen.

Korrelation einfacher Variablen

Im folgenden Beispiel lautet das Kriterium für die linke Begrenzung `_sn=`. Bei jedem Vorkommen von `_sn=` als linke und von `;` als rechte Begrenzung erstellt VuGen einen Parameter mit dem Präfix **Siebel_sn_cookie**.



Im folgenden Beispiel hat VuGen die Begrenzung **_sn** erkannt. Der Parameter wurde im Parameter **Siebel_sn_cookie6** und in der Funktion **web_url** verwendet.

```
/* Registering parameter(s) from source
web_reg_save_param("Siebel_sn_cookie6",
"LB/IC=_sn=",
"RB/IC=;",
"Ord=1",
"Search=headers",
"RelFrameId=1",
LAST);
...
web_url("start.swe_3",
"URL=http://cannon.hplab.com/callcenter_enu/start.swe?SWECmd=GotoPostedAction=;S
WEDIC=true=;_sn={Siebel_sn_cookie6}=;SWEC={Siebel_SWECount}=;SWEFrame=top._swecl
ient=;SWECS=true",
"TargetFrame=",
"Resource=0",
"RecContentType=text/html",
"Referer=http://cannon.hplab.com/callcenter_enu/start.swe?SWECmd=GetCachedFrame
=;_sn={Siebel_sn_cookie6}=;SWEC={Siebel_SWECount}=;SWEFrame=top._swe",
"Snapshot=t4.inf",
"Mode=HTML",
LAST);
```

Korrelation von Funktionen

Manchmal handelt es sich bei der vorgefundenen Begrenzung um eine Funktion. Bei Funktionen wird in der Regel zur Speicherung der Laufzeitwerte ein Array verwendet. Um diese Werte zu korrelieren, analysiert VuGen das Array und speichert die Argumente im folgenden Format in einem separaten Parameter:

```
<Parametername> = <aufgezeichneter_Wert> (Anzeigename)
```

Der Anzeigename ist der Wert, der in der Siebel-Anwendung neben dem Wert angezeigt wird.

VuGen fügt einen Kommentarabschnitt mit den gesamten Parameterdefinitionen ein.

```
/* Registering parameter(s) from source task id 159
    // {Siebel_Star_Array_Op33_7} = ""
    // {Siebel_Star_Array_Op33_6} = "1-231"
    // {Siebel_Star_Array_Op33_2} = ""
    // {Siebel_Star_Array_Op33_8} = "Opportunity"
    // {Siebel_Star_Array_Op33_5} = "06/26/2003 19:55:23"
    // {Siebel_Star_Array_Op33_4} = "06/26/2003 19:55:23"
    // {Siebel_Star_Array_Op33_3} = ""
    // {Siebel_Star_Array_Op33_1} = "test camp"
    // {Siebel_Star_Array_Op33_9} = ""
    // {Siebel_Star_Array_Op33_rowid} = "1-6F"
    // */
```

Darüber hinaus erstellt VuGen, wenn es eine Funktion entdeckt, für **web_reg_save_param** einen neuen Parameter namens **AutoCorrelationFunction**. VuGen ermittelt ebenfalls das Präfix der Parameter und verwendet dieses als Parametername. Im folgenden Beispiel lautet das Präfix **Siebel_Star_Array_Op33**.

```
web_reg_save_param("Siebel_Star_Array_Op33",
    "LB/IC=`v`",
    "RB/IC=`",
    "Ord=1",
    "Search=Body",
    "RelFrameId=1",
    "AutoCorrelationFunction=flCorrelationCallbackParseStarArray",
    LAST);
```

VuGen verwendet die Parameter zu einem späteren Zeitpunkt innerhalb des Skripts. Im folgenden Beispiel lautet der Parametername **web_submit_data**.

web_submit_data("start.swe_14", "Action=http://cannon.hplab.com/callcenter_enu/start.swe", "Method=POST", "RecContentType=text/html", "Referer=", "Snapshot=t15.inf", "Mode=HTML", ITEMDATA, "Name=SWECLK", "Value=1", ENDITEM, "Name=SWEField", "Value=s_2_1_13_0", ENDITEM, "Name=SWER", "Value=0", ENDITEM, "Name=SWESP", "Value=false", ENDITEM, "Name=s_2_2_29_0", "Value={**Siebel_Star_Array_Op33_1**}", ENDITEM, "Name=s_2_2_30_0", "Value={**Siebel_Star_Array_Op33_2**}", ENDITEM, "Name=s_2_2_36_0", "Value={**Siebel_Star_Array_Op33_3**}", ENDITEM, ...

Bei der Wiedergabe führen die Vuser ein Callback der öffentlichen Funktion unter Verwendung der als Parameter gespeicherten Arrayelemente aus.

Hinweis: Die Korrelation für den **SWEC**-Parameter wird nicht mithilfe der Korrelationsregeln ausgeführt. VuGen verarbeitet diese automatisch mithilfe eines integrierten Erkennungsmechanismus. Weitere Informationen finden Sie unter SWEC Correlation.

SWEC-Korrelation

SWEC ist ein von Siebel-Servern verwendeter Parameter, der dazu dient, die Anzahl der Benutzerklicks zu speichern. Der SWEC-Parameter ist in der Regel Teil eines Arguments innerhalb einer URL- oder POST-Anweisung. Beispiel:

```
GET /callcenter_enu/start.swe?SWECmd=GetCachedFrame=;_sn=2-mOTFXHWBAAGb5Xzv9Ls2Z
45QvxGQnOnPVtX6vnfUU_=;SWEC=1=;SWEFrame=top._swe._sweapp HTTP/1.1
```

oder

```
POST /callcenter_enu/start.swe HTTP/1.1
...
\r\n\r\n
SWERPC=1=;SWEC=0=;_sn=2-mOTFXHWBAAGb5Xzv9Ls2Z45QvxGQnOnPVtX6vnfUU_=;SWECmd=Invok
eMethod...
```

VuGen reagiert auf die Änderungen des SWEC-Parameters durch Erhöhung eines Zählers vor jedem relevanten Schritt. VuGen speichert den aktuellen Wert des SWEC-Parameters in einer

separaten Variable (Siebel_SWECount_var). Vor jedem Schritt speichert VuGen den Zählerwert in einem VuGen-Parameter (Siebel_SWECount).

Im folgenden Beispiel verwendet **web_submit_data** den dynamischen Wert des SWEC-Parameters, **Siebel_SWECount**.

```
Siebel_SWECount_var += 1;
lr_save_int(Siebel_SWECount_var, "Siebel_SWECount");
web_submit_data("start.swe_8",
    "Action=http://cannon.hplab.com/callcenter_enu/start.swe",
    "Method=POST",
    "TargetFrame=",
     "RecContentType=text/html",
     "Referer=",
    "Snapshot=t9.inf",
     "Mode=HTML",
    "EncodeAtSign=YES",
    ITEMDATA,
    "Name=SWERPC", "Value=1", ENDITEM,
    "Name=SWEC", "Value={Siebel_SWECount}", ENDITEM,
    "Name=SWECmd", "Value=InvokeMethod", ENDITEM,
    "Name=SWEService", "Value=SWE Command Manager", ENDITEM,
    "Name=SWEMethod", "Value=BatchCanInvoke", ENDITEM,
    "Name=SWEIPS",...
    LAST);
```

Der SWEC-Parameter kann darüber hinaus auch in dem verweisenden URL vorkommen. Der Wert weicht allerdings in der Regel von dem Wert im angeforderten URL ab. VuGen verarbeitet diese Abweichung automatisch.

Korrelieren von SWECount-Parametern

Im SWECount-Parameter wird in der Regel ein niedriger ein- oder zweistelliger Wert gespeichert. Es ist oft schwer zu bestimmen, an welcher Stelle der aufgezeichnete Wert durch einen Parameter ersetzt werden soll.

In der Funktion **web_submit_data** ersetzt VuGen diesen nur im SWEC-Feld.

In URLs ersetzt VuGen den Wert nur, wenn dieser nach den Zeichenfolgen **SWEC=** oder **SWEC`** auftritt.

Der Parametername ist für alle SWECount-Korrelationen derselbe.

Korrelieren von ROWID-Parametern

In einigen Fällen wird der Zeilen-ID (**rowid**) deren Länge im hexadezimalen Format vorangestellt. Da die Länge variiert, muss dieser Wert korreliert werden.

Die folgende Zeichenfolge besteht aus einem Wert für die Länge und der Zeilen-ID: `xxx6_1-4ABCyyy`. Hierbei ist 6 die Länge und `1-4ABC` die Zeilen-ID.

Wenn Sie Parameter definieren, um die Zeichenfolge folgendermaßen zu korrelieren,

```
xxx{rowid_Length}_{rowid}yyy
```

generiert VuGen anhand dieser erweiterten Korrelation die folgende Funktion vor der Zeichenfolge:

```
web_save_param_length("rowid", LAST);
```

Diese Funktion ruft den Wert von **rowid** ab und speichert die Länge im hexadezimalen Format im Parameter **rowid_length**.

Korrelieren von SWET-Parametern (Zeitstempel)

Der SWETS-Wert im Skript gibt die Anzahl der Millisekunden an, die seit dem 1. Januar 1970 um Mitternacht vergangen sind.

VuGen ersetzt alle nicht-leeren Zeitstempel im Skript durch den Parameter {SiebelTimeStamp}. Bevor ein Wert in diesem Parameter gespeichert wird, generiert VuGen die folgende Funktion:

```
web_save_timestamp_param("SiebelTimeStamp", LAST);
```

Diese Funktion speichert den aktuellen Zeitstempel im Parameter **SiebelTimeStamp**.

Korrelieren von Skripts - Oracle NCA

In den folgenden Schritten werden die unterschiedlichen Objekte in Oracle NCA-Skripts beschrieben, für die eine Korrelation erforderlich ist.

Korrelieren von Anweisungen für den Lastausgleich

VuGen unterstützt den Lastausgleich für mehrere Anwendungsserver. Sie korrelieren die HTTP-Rückgabewerte mit den Parametern **nca_connect_server**. Der Vuser stellt anschließend bei der Ausführung des Tests die Verbindung zum relevanten Server her und wendet den Lastausgleich an. Mit den folgenden Schritten können Sie Anweisungen für den Lastenausgleich korrelieren.

1. **Zeichnen Sie ein Skript mit mehreren Protokollen auf.**

   Zeichnen Sie ein Skript mit mehreren Protokollen für Oracle NCA- und Webprotokolle auf. Führen Sie die gewünschten Aktionen durch und speichern Sie das Skript.

2. **Definieren Sie Parameter für den Host und die Hostargumente.**

   Definieren Sie zwei Variablen, **serverHost** und **serverArgs**, für die Parametrierung:

   ```
   web_set_max_html_param_len("512");
   web_reg_save_param("serverHost", "NOTFOUND=ERROR",
       "LB=<PARAM name=\"serverHost\" value=\"","RB=\">", LAST);
   web_reg_save_param("serverArgs", "NOTFOUND=ERROR",
       "LB=<PARAM name=\"serverArgs\" value=\"","RB=\">", LAST);
   ```

3. **Weisen Sie serverHost und serverArgs Werte zu:**
   `web_url("step_name", "URL=http://server1.acme.com/test.htm", LAST);`

4. **Ändern Sie die Anweisung nca_connect_server von:**
   `nca_connect_server("199.203.78.170", 9000"/*version=107*/, "module=e:\\appsnca...fndnam=apps ");`

in:

```
nca_connect_server("{ serverHost }", "9000"/*version=107*/, "{serverArgs}");
```

Das Skript sollte jetzt folgendermaßen aussehen:

```
web_set_max_html_param_len("512");
web_reg_save_param("serverHost", "NOTFOUND=ERROR",
    "LB=<PARAM name=\"serverHost\" value=\"","RB=\">", LAST);
web_reg_save_param("serverArgs", "NOTFOUND=ERROR",
    "LB=<PARAM name=\"serverArgs\" value=\"","RB=\">", LAST);
    web_url("step_name", "URL=http://server1.acme/test.htm", LAST);
nca_connect_server("{serverHost}","9000"/*version=107*/,"{serverArgs}");
```

Korrelieren der "icx_ticket"-Variable

Die Variable **icx_ticket** ist Bestandteil der Informationen, die in den Funktionen **web_url** und **nca_connect_server** gesendet werden:

```
web_url("fnd_icx_launch.runforms",
"URL=http://ABC-123:8002/pls/VIS/fnd_icx_launch.runforms\?ICX_TICKET=5843A55058947ED3=;RESP_APP=AR=;RESP_KEY=RECEIVABLES_MANAGER=;SECGRP_KEY=STANDARD", LAST);
```

Dieser Wert für **icx_ticket** ist bei jeder Aufzeichnung anders. Er enthält Cookie-Informationen, die vom Client gesendet wurden. Um die Aufzeichnung zu korrelieren, fügen Sie vor dem ersten Auftreten des aufgezeichneten Wertes von **icx_ticket** den Eintrag **web_reg_save_param** hinzu:

```
web_reg_save_param("icx_ticket", "LB=TICKET=", "RB==;RES", LAST);
...
web_url("fnd_icx_launch.runforms",
"URL=http://ABC-123:8002/pls/VIS/fnd_icx_launch.runforms\?ICX_TICKET={icx_ticket}=;RESP_APP=AR=;RESP_KEY=RECEIVABLES_MANAGER=;SECGRP_KEY=STANDARD", LAST);
```

Hinweis: Die linke und rechte Begrenzung von **web_reg_save_param** ist abhängig vom Setup der Anwendung jeweils unterschiedlich.

Korrelieren der "JServSessionIdroot"-Werte

Beim Wert **JServSessionIdroot** handelt es sich um ein Cookie, das von der Anwendung zum Speichern der Sitzungs-ID festgelegt wird. In den meisten Fällen korreliert VuGen diesen Wert automatisch und fügt eine Funktion **web_reg_save_param** ein. Wenn VuGen diese Funktion nicht automatisch hinzugefügt hat, fügen Sie sie manuell hinzu und ersetzen dadurch alle Vorkommnisse durch den Parameternamen.

Um zu ermitteln, welcher Wert korreliert werden muss, öffnen Sie das Ausführungsprotokoll (**Ansicht > Ausgabe**) und suchen nach dem Antworttext.

```
vuser_init.c(8):     Set-Cookie: JServSessionIdroot=my1sanw2n1.JS4; path=/\r\n
vuser_init.c(8):     Content-Length: 79\r\n
vuser_init.c(8):     Content-Type: text/plain\r\n
vuser_init.c(8):     \r\n
vuser_init.c(8):     81-byte response body for "http://ABC-
```

```
123/servlet/oracle.forms.servlet.ListenerServlet?ifcmd=getinfo=;
                          ifhost=mercury=;ifip=123.45.789.12" (RelFrameId=1)
vuser_init.c(8):     /servlet/oracle.forms.servlet.ListenerServlet?JServSessionI
droot=my1sanw2n1.JS4\r\n
```

Um diesen dynamischen Wert zu korrelieren, fügen Sie vor dem ersten Auftreten eine Funktion **web_reg_save_param** ein und ersetzen dadurch den Variablenwert im gesamten Skript durch den Parameternamen. In diesem Beispiel heißen die rechte und linke Begrenzung **\r** und **\n**, Sie müssen jedoch Ihre jeweilige Umgebung überprüfen, um die genauen Begrenzungen für Ihre Umgebung zu bestimmen.

```
web_reg_save_param("NCAJServSessionId","LB=\r\n\r\n","RB=\r","ORD=1",LAST);
web_url("f60servlet",
    "URL= http://ABC-"123/servlet/oracle.forms.servlet.ListenerServlet?ifcmd=get
info=;" "ifhost=mercury=;ifip=123.45.789.12", LAST);
web_url("oracle.forms.servlet.ListenerSer",
    "URL=http://ABC-123{NCAJServSessionId}?ifcmd=getinfo=;" "ifhost=mercury=;ifi
p=123.45.789.12", LAST);
```

Korrelieren von Skripts - Microsoft .NET

Im Rahmen dieser Aufgabe wird das Korrelieren von Microsoft .NET-Vuser-Skripts beschrieben.

Korrelieren von Skripts mithilfe von ADO.net-Umgebungen

1. **Suchen Sie den Datensatz in Ihrem Skript.**

   Zeigen Sie das Vuser-Skript im Editor an und erweitern Sie die entsprechende **DATASET_ XML**-Anweisung. Klicken Sie auf **Ansicht > Snapshot**.

	CustomerID	CompanyName	ContactName	ContactTitle	Address
1	ABC	ABC Company	John Smith	Owner	One My Way
2	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57
3	ANATR	Ana Trujillo Emparec	Ana Trujillo	Owner	Avda. de la C
4	ANTON	Antonio Moreno Tac	Antonio Moreno	Owner	Mataderos 2
5	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover

2. **Suchen Sie den Wert.**

   Suchen Sie den Wert, den Sie korrelieren möchten. Zeigen Sie dazu die Datentabelle im Snapshot-Bereich an und klicken Sie auf **Suchen > Schnellsuche**, um das Dialogfeld **Suchen** zu öffnen.

   Klicken Sie im Dialogfeld auf **In Suche einschließen** und aktivieren Sie anschließend das Kontrollkästchen **Snapshots**.

3. **Erstellen Sie eine Korrelation.**

Klicken Sie in der Tabelle mit der rechten Maustaste auf den Wert, den Sie korrelieren möchten, und wählen Sie im Kontextmenü die Option **Korrelation erstellen** aus. Das Dialogfeld **Korrelation erstellen** wird geöffnet.

4. **Geben Sie einen Parameternamen an.**

   Geben Sie einen Parameternamen an, der mit der zuvor definierten Variable identisch ist. Klicken Sie auf **OK**. VuGen fragt, ob Sie nach allen Vorkommen suchen möchten. Klicken Sie auf **OK**.

   VuGen fügt vor den Daten jeweils eine Funktion **lr.save_string** hinzu. Beispiel:

   ```
   lr.save_string("MyCustomerID", CustomerAndOrdersDataSet_3.Tables["Customers"]
   .Rows[0]["CompanyName"].ToString());
   ```

5. **Referenzieren Sie den Parameter an einer späteren Stelle im Skript.**

   Wählen Sie den Wert aus, den Sie durch einen Parameter ersetzen möchten, und wählen Sie dann im Kontextmenü die Option **Durch einen Parameter ersetzen** aus. Fügen Sie den Namen der gespeicherten Variable im Feld **Parametername** ein. Klicken Sie auf OK. Sie müssen bestätigen, dass VuGen alle Werte durch einen Parameter ersetzen und die Funktion **lr.eval_string** zum Auswerten des Zeichenfolgewerts verwenden soll.

   ```
   lr.message("The customer ID is""+ lr.eval_string("{MyCustomerID}") + ");
   ```

Im Gegensatz zu anderen Protokollen enthält das Skript direkte Aufrufe an die Anwendungs- oder Frameworkmethode. Deshalb können Sie den Zeichenfolgewert nicht durch {paramName} ersetzen, sondern müssen die Funktion **lr.eval_string** zum Auswerten des Parameterwerts verwenden.

Korrelieren mit Ausgabeparametern

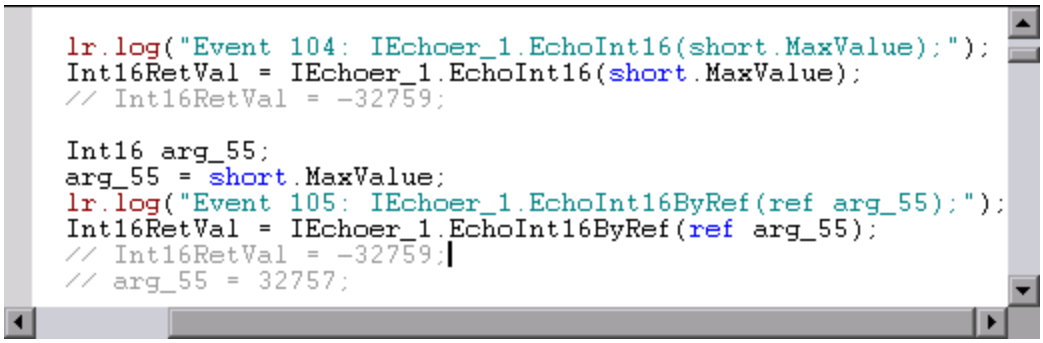
Bei elementaren Werten sollten Sie das Skript mit Ausgabeparameterwerten generieren und die Ausgabeparameter auf Korrelationen überprüfen.

1. Wählen Sie **Aufzeichnen > Aufzeichnungsoptionen** und dann den Knoten **Allgemein > Skript** aus.

2. Aktivieren Sie das Kontrollkästchen **Ausgabeparameterwerte einfügen**. Klicken Sie auf **OK**, um das Dialogfeld **Aufzeichnungsoptionen** zu schließen.

3. Wählen Sie **Aufzeichnen > Skript neu generieren** aus, um das Skript erneut zu generieren.

4. Suchen Sie in den kommentierten elementaren Ausgabewerten nach Korrelationen.



Weitere Informationen zu Korrelationsfunktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Korrelieren von Skripts - Java-Skripts - Serialisierung

In RMI sowie in einigen Fällen bei CORBA erstellt die Client-AUT mithilfe der Schnittstelle **java.io.serializable** eine neue Instanz eines Java-Objekts. Die Instanz wird dann als Parameter für einen Serveraufruf übergeben. Im folgenden Segment wird die Instanz **p** erstellt und als Parameter übergeben.

```
// AUT code:
java.awt.Point p =  new java.awt.Point(3,7);
map.set_point(p);
:
```

Der automatische Korrelationsmechanismus ist hier wirkungslos, da das Objekt nicht von einem vorherigen Aufruf zurückgegeben wurde. In diesem Fall aktiviert VuGen den Serialisierungsmechanismus und speichert das weitergegebene Objekt als Parameter. Die Informationen werden in einer binären Datendatei im Benutzerordner abgelegt. Zusätzliche Parameter werden dann in weiteren, fortlaufend nummerierten binären Datendateien gespeichert. VuGen generiert den folgenden Code:

```
public class Actions {
  // Public function: init
      public int init() throws Throwable {
          java.awt.Point p = (java.awt.Point)lr.deserialize(0, false);
          map.set_point(p);
      }
:
}
```

Die an **lr.deserialize** übergebene Ganzzahl stellt die Anzahl von binären Datendateien im Vuser-Ordner dar.

Verwenden Sie zur Parametrierung der aufgezeichneten Werte die öffentliche Methode **setLocation** (Informationen hierzu finden Sie in der JDK-Funktionsreferenz). Im folgenden Beispiel wird die Methode **setLocation** verwendet, um den Wert des Objekts **p** festzulegen.

```
public class Actions {
  // Public function: init
      public int init() throws Throwable {
          java.awt.Point p = (java.awt.Point)lr.deserialize(0, false);
          p.setLocation(2,9);
          map.set_point(p);
      }
:
:
}
```

In bestimmten Fällen kann die öffentliche Methode **setLocation** nicht angewendet werden. Alternativ können Sie das API Ihrer Klasse verwenden, wenn diese Accessormethoden vom Typ **get** oder **set** beinhaltet. Wenn Sie mit AUT-Klassen arbeiten, die keine get/set-Methoden enthalten oder private Methoden verwenden, oder wenn Sie mit dem API der Klassen nicht vertraut sind, können Sie den integrierten Serialisierungsmechanismus von VuGen verwenden. Dieser Mechanismus erlaubt Ihnen, Objekte in ihrer ASCII-Darstellung zu erweitern und das Skript manuell zu parametrieren. Sie aktivieren diesen Mechanismus über das Dialogfeld **Aufzeichnungsoptionen**. Details finden Sie unter "Aufzeichnungseigenschaften > Knoten "Serialisierungsoptionen"" auf Seite 238.

VuGen generiert eine Methode vom Typ **lr.deserialize**, die die Daten deserialisiert bzw. komplexe Datenstrukturen als serielle Zeichenfolgen anzeigt. Nachdem eine Struktur in ihre Komponenten aufgespalten wurde, lässt sie sich einfacher parametrieren. Die Methode **lr.deserialize** empfängt zwei Argumente, eine Zeichenfolge und eine Ganzzahl. Die Zeichenfolge ist der Wert des Parameters, der bei der Wiedergabe des Skripts ersetzt werden soll. Die Ganzzahl ist die Indexnummer der zu ladenden Binärdatei.

Wenn Sie sich dafür entscheiden, Objekte in Ihrem Skript nicht zu erweitern (durch Deaktivieren des Kontrollkästchens **Serialisierte Objekte erweitern**), können Sie den Serialisierungsmechanismus steuern, indem Sie Argumente an die Methode **lr.deserialize** übergeben. Das erste Argument ist eine Ganzzahl, die die Anzahl der zu ladenden Binärdateien darstellt. Das zweite Argument ist ein logischer Wert:

true	Der Serialisierungsmechanismus von VuGen wird verwendet.
false	Der standardmäßige Java-Serialisierungsmechanismus wird verwendet.

Das folgende Segment zeigt ein generiertes Skript, bei dem der Serialisierungsmechanismus aktiviert war.

```
public class Actions {
  // Public function: init
      public int init() throws Throwable {
          _string = "java.awt.Point __CURRENT_OBJECT = {" +
                    "int x = "#5#" +
                    "int y = "#8#" +
          "}";
          java.awt.Point p = (java.awt.Point)lr.deserialize(_string,0);
          map.set_point(p);
      }
```

```
:
}
```

Die Zeichenfolgenwerte werden zwischen Trennzeichen gesetzt. Das Standardtrennzeichen ist **#**. Das Standardtrennzeichen kann in den Aufzeichnungsoptionen auf der Registerkarte **Serialisierung** geändert werden. Die Trennzeichen beschleunigen den Analysevorgang der Zeichenfolgen bei der Wiedergabe des Skripts.

Beim Bearbeiten der Zeichenfolgen müssen die folgenden Regeln beachtet werden:

- Die Reihenfolge der Zeilen darf nicht geändert werden. Der Parser liest die Werte einzeln nacheinander ein, nicht die Namen der Mitglieder.
- Es dürfen nur Werte zwischen Trennzeichen modifiziert werden.
- Objektreferenzen müssen unverändert bleiben. Auf Objektreferenzen wird nur hingewiesen, um die interne Konsistenz zu erhalten.
- "_NULL_" kann als Wert auftreten und repräsentiert dann die Java-Nullkonstante. Sie kann nur durch Zeichenfolgenwerte ersetzt werden.
- Objekte können an beliebigen Stellen im Skript deserialisiert werden. Sie können beispielsweise alle Objekte in der Methode **init** deserialisieren und die Werte dann in der Methode **action** verwenden.
- Behalten Sie die interne Konsistenz der Objekte bei. Wenn ein Vektor beispielsweise das Mitglied **element count** aufweist und Sie ein Element hinzufügen, müssen Sie die Anzahl der Elemente entsprechend anpassen.

Der Vektor im folgenden Segment enthält zwei Elemente:

```
public class Actions {
  // Public function: init
      public int init() throws Throwable {
          _string = "java.util.Vector CURRENTOBJECT = {" +
                    "int capacityIncrement = "#0#" +
                    "int elementCount = #2#" +
                    "java/lang/Object elementData[] = {" +
                           "elementData[0] = #Erstes Element#" +
                           "elementData[1] = #Zweites Element#" +
                           "elementData[2] = _NULL_" +
                            ....
                           "elementData[9] = _NULL_" +
                    "}" +
          "}";
          _vector = (java.util.Vector)lr.deserialize(_string,0);
          map.set_vector(_vector);
      }
:
}
```

Im folgenden Beispiel wurde eines der Vektorelemente geändert – der Wert "_NULL_" wurde in "Third element" geändert. Aufgrund des hinzugefügten Elements wurde das Mitglied **elementCount** auf **3** gesetzt.

```
public class Actions {
  // Public function: init
      public int init() throws Throwable {
          _string = "java.util.Vector CURRENTOBJECT = {" +
                    "int capacityIncrement = "#0#" +
                    "int elementCount = #1,36kg" +
                    "java/lang/Object elementData[] = {" +
                           "elementData[0] = #Erstes Element#" +
                           "elementData[1] = #Zweites Element#" +
                           "elementData[2] =  #Third Element#" +
                             ....
                           "elementData[9] = _NULL_" +
                    "}" +
          "}";
          _vector = (java.util.Vector)lr.deserialize(_string,0);
          map.set_vector(_vector);
      }
:
}
```

Wegen der Komplexität des Serialisierungsmechanismus, der die Objekte für die ASCII-Darstellung öffnet, führt das Öffnen umfangreicher Objekte während der Aufzeichnung dazu, dass die Skriptgenerierung mehr Zeit in Anspruch nimmt. Zur Verkürzung der Generierungsdauer können Sie Flags setzen, die den Serialisierungsmechanismus beschleunigen.

Wenn Sie die Methode **lr.deserialize** in ein Skript aufnehmen, wird empfohlen, sie zur Methode **init** hinzuzufügen und nicht zu **action**. Hierdurch wird die Serialisierung beschleunigt, da VuGen die Zeichenfolgen nur einmal deserialisiert. Wenn **lr.deserialize** Teil der Methode **action** ist, deserialisiert VuGen die Zeichenfolgen bei jeder Iteration.

Korrelieren von Skripts - Java

Der Java-Recorder von VuGen versucht, die Anweisungen im generierten Skript automatisch zu korrelieren. Korrelationen werden nur für Java-Objekte ausgeführt. Stößt der Recorder während der Aufzeichnung auf einen elementaren Java-Datentyp (Byte, Zeichen, Logisch, Ganzzahl, Gleitkomma, doppelte Genauigkeit, Kurz und Lang), werden die Argumentwerte im Skript nicht mit Variablen verknüpft. VuGen korreliert automatisch alle Objekte, Objektarrays und Arrays aus elementaren Datentypen. Beachten Sie, dass Java-Arrays und -Zeichenfolgen auch als Objekte gelten.

In VuGen gibt es mehrere Ebenen der Korrelation: Standard, Erweitert, Zeichenfolgen. Die Korrelation wird über die Aufzeichnungsoptionen aktiviert bzw. deaktiviert. Wenn keine der oben genannten Methoden angewendet werden kann, steht zusätzliche die Methode der Serialisierung zur Verfügung, um Skripts zu verarbeiten.

Standardkorrelation

Die Standardkorrelation ist die automatische Korrelation, die während der Aufzeichnung einfacher Objekte (d. h. keine Objektarrays, Vektoren und Containerkonstrukte) ausgeführt wird.

Wenn die aufgezeichnete Anwendung eine Methode aufruft, die ein Objekt zurückgibt, zeichnet der Korrelationsmechanismus von VuGen diese Objekte auf. Bei Ausführung des Skripts vergleicht VuGen die generierten Objekte mit den aufgezeichneten Objekten. Bei Übereinstimmung wird das entsprechende Objekt verwendet. Im folgenden Beispiel werden die beiden CORBA-Objekte **my_bank** und **my_account** verwendet. Das erste Objekt, **my_bank**, wird aufgerufen, das zweite Objekt, **my_account**, wird korreliert und als Parameter in der letzten Zeile des Segments übergeben:

```
public class Actions {
  // Public function: init
      public int init() throws Throwable {
       Bank my_bank = bankHelper.bind("bank", "shunra");
       Account my_account = accountHelper.bind("account","shunra");
       my_bank.remove_account(my_account);
      }
:
}
```

Erweiterte Korrelation

Die erweiterte oder **tiefe** Korrelation ist die automatische Korrelation, die während der Aufzeichnung komplexer Objekte, d. h. von Objektarrays und CORBA-Containerkonstrukten, ausgeführt wird.

Bei der tiefen Korrelation werden CORBA-Konstrukte (Strukturen, Vereinigungen, Sequenzen, Arrays, Halter, Any-Datentypen) als Container behandelt. Hierdurch wird es möglich, innere Mitglieder von Containern, zusätzliche Objekte oder andere Container zu referenzieren. Immer, wenn ein Objekt aufgerufen oder als Parameter weitergegeben wird, wird es auch mit den inneren Mitgliedern der Container verglichen.

Im folgenden Beispiel führt VuGen eine tiefe Korrelation durch Referenzierung eines Arrayelements aus. Das Objekt **remove_account** empfängt ein Objekt vom Typ **account** als Parameter. Während der Aufzeichnung durchsucht der Korrelationsmechanismus das zurückgegebene Array **my_accounts** und stellt fest, dass sein sechstes Element als Parameter weitergegeben werden soll.

```
public class Actions {
   // Public function: init
   public int init() throws Throwable {
      my_banks[] = bankHelper.bind("banks", "shunra");
      my_accounts[] = accountHelper.bind("accounts","shunra");
      my_banks[2].remove_account(my_accounts[6]);
    }
:
}
```

Im folgenden Segment wird die erweiterte Korrelation weiter veranschaulicht. Das Skript ruft das Objekt **send_letter** auf, das ein Argument vom Typ **address** erhalten hat. Der Korrelationsmechanismus ruft das innere Mitglied, **address**, im sechsten Element des Arrays **my_accounts** ab.

```
public class Actions {
   // Public function: init
   public int init() throws Throwable {
      my_banks = bankHelper.bind("bank", "shunra");
      my_accounts = accountHelper.bind("account", "shunra");
      my_banks[2].send_letter(my_accounts[6].address);
    }
:
}
```

Zeichenfolgenkorrelation

Der Begriff Zeichenfolgenkorrelation bezieht sich auf die Darstellung eines aufgezeichneten Werts entweder als eine Zeichenfolge oder als eine Variable. Wenn Sie die Zeichenfolgenkorrelation deaktivieren (die Standardeinstellung), wird der tatsächlich aufgezeichnete Wert der Zeichenfolge explizit innerhalb des Skripts ausgewiesen. Wenn Sie die Zeichenfolgenkorrelation aktivieren, wird für jede Zeichenfolge eine Variable generiert, die später im Skript wiederverwendet werden kann.

Im folgenden Segment ist die Zeichenfolgenkorrelation aktiviert, d. h. der von der Methode **get_id** zurückgegebene Wert wird in einer Variable vom Typ **string** zur späteren Verwendung im Skript gespeichert.

```
public class Actions {
   // Public function: init
   public int init() throws Throwable {
      my_bank = bankHelper.bind("bank", "shunra");
      my_account1 = accountHelper.bind("account1", "shunra");
      my_account2 = accountHelper.bind("account2", "shunra");
      string = my_account1.get_id();
      string2 = my_account2.get_id();
      my_bank.transfer_money(string, string2);
    }
:
}
```

Korrelieren von Skripts - XPath-Korrelation in Flex Vuser-Skripts

In diesem Abschnitt wird die Verwendung der XPath-Korrelation in Flex Vuser-Skripts beschrieben. Zum Korrelieren verwenden Sie die XML-Ansicht im Snapshot-Bereich. Bevor Sie die XPath-Korrelation implementieren können, müssen Sie die Aufzeichnungsoptionen konfigurieren.

Informationen zur Verwendung der herkömmlichen Korrelation in Flex-Vuser-Skripts finden Sie unter "Korrelieren von Skripts mit Design Studio" auf Seite 288.

1. **Konfigurieren der Aufzeichnungsoptionen**
   a. Wählen Sie **Aufzeichnen > Aufzeichnungsoptionen** aus.

b. Klicken Sie unter **Flex** auf **Externalisierbare Objekte**.

c. Klicken Sie auf **Serialisieren von Objekten mit** und wählen Sie **Benutzerdefinierte Java-Klassen** aus.

d. Klicken Sie auf die Schaltfläche **Fügt eine JAR- oder ZIP-Datei hinzu** .

e. Suchen Sie auf den LiveCycle-Installationsdatenträgern die folgenden drei Dateien und fügen Sie sie der Liste **Klassenpfadeinträge** hinzu:

i. **flex.jar**

ii. **flex-messaging-common.jar**

iii. **flex-messaging-core.jar**

Außerdem müssen Sie sicherstellen, dass sich diese Dateien auf allen Lastgenerator-Computern am gleichen Speicherort befinden.

2. Zeichnen Sie das Vuser-Skript auf:

3. Klicken Sie im Editor in den Schritt **flex_amf_call**, der die zu korrelierenden Daten enthält, oder doppelklicken Sie im Schrittenavigator auf den Schritt **flex_amf_call**, der die zu korrelierenden Daten enthält.

4. Klicken Sie auf **Ansicht > Snapshot** oder auf die Schaltfläche **Snapshot** in der VuGen-Symbolleiste.

5. Klicken Sie im Snapshot-Bereich auf die Registerkarte **Antworttext**.

6. Klicken Sie auf der rechten Seite des Snapshot-Bereichs auf die Registerkarte **XML-Ansicht**.

7. Markieren Sie in der XML-Ansicht die gesamte Zeichenfolge, in der die zu korrelierenden dynamischen Daten enthalten sind.

8. Klicken Sie mit der rechten Maustaste in die Auswahl und wählen Sie **Korrelation erstellen** aus. Design Studio wird geöffnet. Weitere Informationen zur Verwendung von Design Studio finden Sie unter "Registerkarte "Korrelation" [Design Studio] - Übersicht" auf Seite 282.

Nach Abschluss der Korrelation fügt VuGen einen **web_reg_save_parm_xpath**-Schritt in das Vuser-Skript ein.

Korrelieren von Skripts - COM-Protokoll

Mithilfe der folgenden Schritte können Sie COM-Vuser-Skripts korrelieren.

1. Wählen Sie **Ansicht** > **Ausgabefenster** aus, um die Registerkarten am unteren Fensterrand anzuzeigen. Suchen Sie in der Registerkarte **Wiedergabe** nach Fehlern. In den meisten Fällen können diese Fehler durch eine Korrelation korrigiert werden.

2. Wählen Sie den entsprechenden Schritt im **Schrittenavigator** aus, und zeigen Sie den Schritt im **Snapshot-Bereich** an.

3. Klicken Sie mit der rechten Maustaste auf den Wert im Snapshot und wählen Sie **Korrelation erstellen**. Dadurch wird das Fenster **Design Studio** geöffnet.

4. Wählen Sie den zu korrelierenden Wert aus, indem Sie den Wert im Raster markieren und auf die Schaltfläche **Korrelieren** klicken.

   Erfolgt die Korrelation eines Werts, fügt VuGen den Korrelationsparameter hinzu und speichert den Originalwert in einem Kommentar im Skript.

```
252 /* Correlation comment - Do not change!  Original value='1' Name ='CorrelationParameter' Type ='Manual' */
253     lrc_save_rs_param(_Recordset_45,
254         1,
255         2,
256         0,
257         "CorrelationParameter1");
```

Suchen nach Werten, für die eine Korrelation erforderlich ist

In den folgenden Schritten werden die unterschiedlichen Methoden für die Suche nach Werten beschrieben, für die eine Korrelation erforderlich ist.

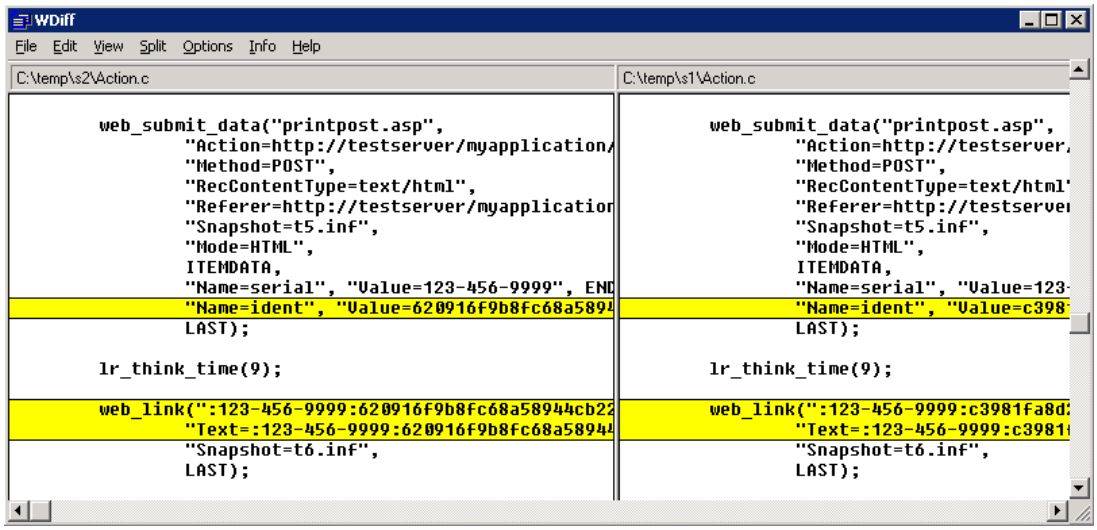
Suchen mithilfe eines Skriptvergleichs

1. Zeichnen Sie ein Skript auf und speichern Sie es.

2. Erstellen Sie ein neues Skript und zeichnen Sie dieselben Operationen auf. Speichern Sie das Skript.

3. Wählen Sie **Extras** > **Mit Vuser vergleichen** aus, um die Skripts zu vergleichen. Weitere Informationen finden Sie unter "Vergleichen von Skripts nebeneinander" auf Seite 69.

4. Die Unterschiede werden im Skript hervorgehoben. Überprüfen Sie die Unterschiede, um erforderliche Korrelationen zu ermitteln.



Hinweis: *WDiff* ist das Standard-Dienstprogramm, Sie können jedoch ein benutzerdefiniertes Vergleichstool angeben. Weitere Informationen finden Sie unter "Vergleichen von Skripts nebeneinander" auf Seite 69.

Durchsuchen des Wiedergabeprotokolls

1. Überprüfen Sie das Skript in der Skriptansicht, um Zeichenfolgen anzuzeigen, für die eine Korrelation erforderlich ist, etwa Hashzeichenfolgen, zufallsgenerierte Zeichenfolgen, Sitzungs-IDs usw.

2. Suchen Sie im Generierungsprotokoll nach dem ersten Vorkommen der Zeichenfolge (hierbei handelt es sich um die Antwort vom Server).

3. Suchen Sie im erweiterten Wiedergabeprotokoll nach derselben Antwort. Überprüfen Sie, ob diese Antwort eine andere Zeichenfolge innerhalb derselben Begrenzungen enthält, als die vermutete ursprüngliche Zeichenfolge. Wenn ja, ist für diese Zeichenfolge eine Korrelation erforderlich.

Ändern von Korrelationsdefinitionen

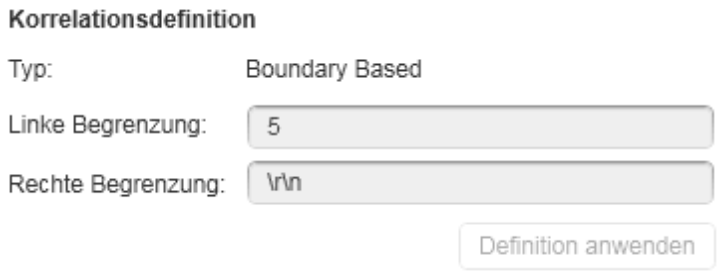
Sie können Korrelationsdefinitionen ändern, um dynamische Werte zu entfernen, die nicht korreliert werden müssen. Diese Aufgaben beschreiben, wie Korrelationsdefinitionen vom Typ "Begrenzungsbasiert", "Regulärer Ausdruck" und "XPath-Abfrage" für die Aufzeichnungs- oder Antwortkorrelation geändert werden.

Ändern von Korrelationsdefinitionen vom Typ "Begrenzungsbasiert"

1. Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Design Studio**.

2. Wählen Sie einen dynamischen Wert aus der Korrelationstabelle aus, und erweitern Sie **Details**.

3. Bearbeiten Sie **Linke Begrenzung** oder **Rechte Begrenzung** unter dem Abschnitt **Korrelationsdefinition**. Sie können die Definition durch Hinzufügen oder Löschen von Text ändern.



4. Klicken Sie auf **Definition anwenden**.

   Die Schaltfläche **Definition anwenden** ist erst verfügbar, wenn die geänderte Definition für die Begrenzung im Snapshot und im Skript angezeigt wird.

   Hinweis: Wenn Sie die Definition nicht anwenden, bevor Sie einen anderen dynamischen Wert in der Tabelle auswählen, gehen Ihre Änderungen verloren. Wenn Sie **Wiedergeben & Prüfen** auswählen, bevor Sie Ihren Wert mit der geänderte Definition korrelieren, gehen Ihre Änderungen verloren.

Ändern von Korrelationsdefinitionen vom Typ "Regulärer Ausdruck"

1. Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Design Studio**.

2. Wählen Sie einen dynamischen Wert aus der Korrelationstabelle aus, und erweitern Sie **Details**.

3. Bearbeiten Sie **Regulärer Ausdruck** unter dem Abschnitt **Korrelationsdefinition**.

4. Klicken Sie auf **Definition anwenden**.

   Die Schaltfläche **Definition anwenden** ist erst verfügbar, wenn die geänderte Definition für die Begrenzung im Snapshot und im Skript angezeigt wird.

   Hinweis: Wenn Sie die Definition nicht anwenden, bevor Sie einen anderen dynamischen Wert in der Tabelle auswählen, gehen Ihre Änderungen verloren. Wenn Sie **Wiedergeben & Prüfen** auswählen, bevor Sie Ihren Wert mit der geänderte Definition korrelieren, gehen Ihre Änderungen verloren.

Ändern von Korrelationsdefinitionen vom Typ "XPath-Abfrage"

1. Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Design Studio**.
2. Wählen Sie einen dynamischen Wert aus der Korrelationstabelle aus, und erweitern Sie **Details**.
3. Bearbeiten Sie die XPath-Definition unter dem Abschnitt **Korrelationsdefinition**.
4. Klicken Sie auf **Definition anwenden**.

   Die Schaltfläche **Definition anwenden** ist erst verfügbar, wenn die geänderte Definition für die Begrenzung im Snapshot und im Skript angezeigt wird.

   Hinweis: Wenn Sie die Definition nicht anwenden, bevor Sie einen anderen dynamischen Wert in der Tabelle auswählen, gehen Ihre Änderungen verloren. Wenn Sie **Wiedergeben & Prüfen** auswählen, bevor Sie Ihren Wert mit der geänderte Definition korrelieren, gehen Ihre Änderungen verloren.

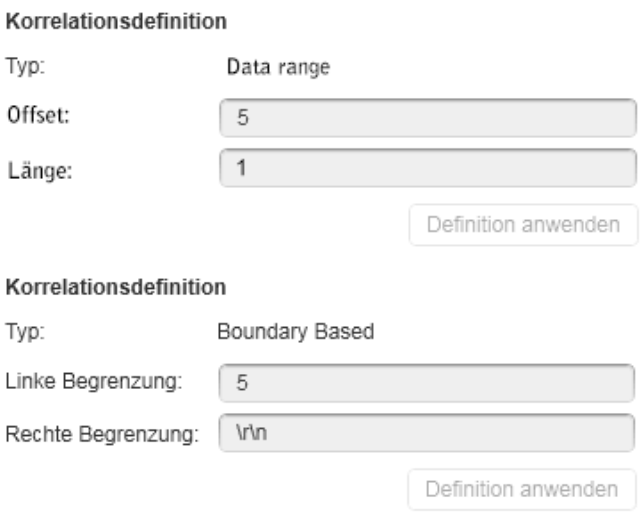
Ändern von Winsocket-Korrelationsdefinitionen

1. Dynamische Winsocket-Werte werden aus dem Snapshot korreliert. Für den Zugriff wählen Sie den entsprechenden Schritt im **Schrittenavigator** aus, und zeigen Sie den Schritt im **Snapshot-Bereich** an. Das Winsocket-Protokoll besitzt sowohl einen Snapshot vom Typ "Hexadezimal" als auch vom Typ "Text".

   Klicken Sie mit der rechten Maustaste auf den Wert im Snapshot, und wählen Sie **Korrelation erstellen** oder **Grenzkorrelation erstellen** aus. Dadurch wird das Fenster **Design Studio** geöffnet.

2. Wählen Sie einen dynamischen Wert aus der Korrelationstabelle aus, und erweitern Sie **Details**.

3. Wenn Sie **Korrelation erstellen** ausgewählt haben, bearbeiten Sie den Datenbereich im Abschnitt **Korrelationsdefinition**. Wenn Sie **Grenzkorrelation erstellen** ausgewählt haben, bearbeiten Sie den linken Wert der rechten Begrenzung.

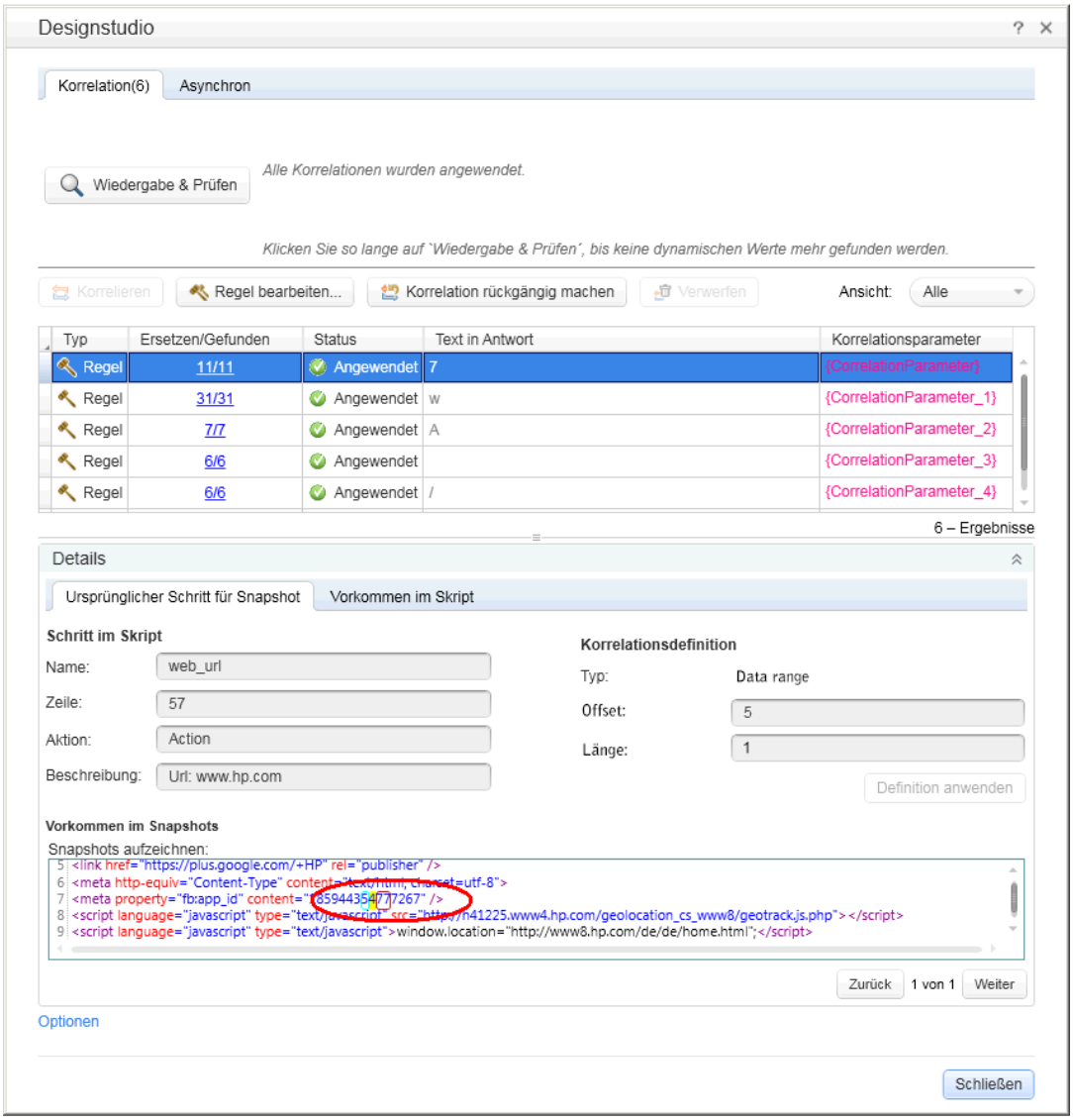


4. Klicken Sie auf **Definition anwenden**.

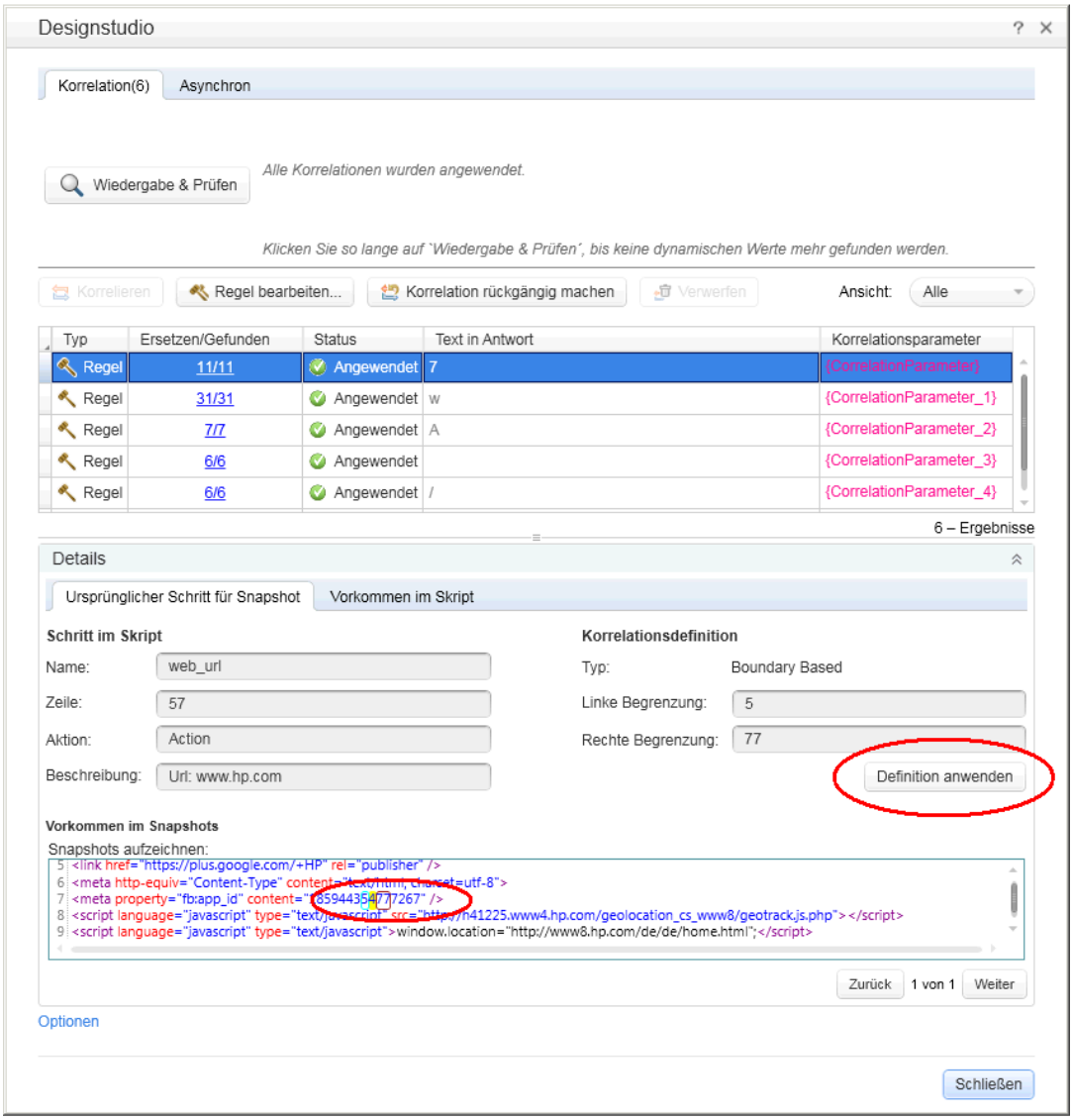
   Die Schaltfläche **Definition anwenden** ist erst verfügbar, wenn die geänderte Definition für die Begrenzung im Snapshot angezeigt wird.

   Folgende Abbildungen zeigen eine Datenbereichsdefinition und eine Begrenzungsdefinition.

Datenbereichsdefinition



Korrekte Begrenzungsdefinition



Hinweis: Wenn Sie die Definition nicht anwenden, bevor Sie einen anderen dynamischen Wert in der Tabelle auswählen, gehen Ihre Änderungen verloren. Wenn Sie **Wiedergeben & Prüfen** auswählen, bevor Sie Ihren Wert mit der geänderte Definition korrelieren, gehen Ihre Änderungen verloren.

Hinweis: Definitionen von Korrelationsregeln können mit dieser Methode nicht geändert werden.

Ausschließen von Inhalten anhand von "Content-Type"

Die HTTP-Kopfzeile "content-type:" definiert den Typ der HTTP-Antwort (bzw. des Antwortinhalts). In diesem Abschnitt wird beschrieben, wie Sie Inhalte anhand von "content-type" aus der Korrelationsprüfung ausschließen, indem Sie die Datei "IgnoredContent.xml" ändern.

1. Öffnen Sie die Datei `<LoadRunner-Installationsverzeichnis>\config\IgnoredContent.xml` in einem Texteditor.

IgnoredContent.xml

```
<?xml version="1.0" encoding="utf-8" ?>
 <IgnoredHttpContentTypes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instan
ce" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <CONTENT_TYPES>
   <string>application/(?!(x-amf|json))</string>
    <string>audio/</string>
    <string>image/</string>
    <string>model/</string>
    <string>video/</string>
  </CONTENT_TYPES>
</IgnoredHttpContentTypes>
```

2. Ändern Sie die Datei "IgnoredContent.xml" so, dass "content-type" ausgeschlossen wird. Fügen Sie hierzu eine Zeichenfolge oder einen regulären Ausdruck ein.

Bei Eingabe von:	Führt Correlation Studio Folgendes durch:
`image/`	Ignorieren aller Inhaltstypen beginnend mit `image/` wie z. B. `image/gif, image/jpeg, image/png`
`application/(!?(json|x-amf))`	Ignorieren von Inhaltstypen, die mit `application/` beginnen, mit Ausnahme von `application/json` oder `application/x-amf`.

Beispiel für Inhalte, die zu "IgnoredContent.xml" hinzugefügt wurden

In unserem Beispiel ignoriert das Korrelationsmodul Anwendungsinhalte mit Ausnahme der Anwendungsinhalte vom Typ "x-amf", "json" oder "javascript".

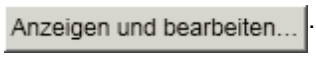
```
<?xml version="1.0" encoding="utf-8" ?>
 <IgnoredHttpContentTypes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instan
ce" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
   <CONTENT_TYPES>
    <string>application/(?!(x-amf|json|javascript))</string>
    <string>audio/</string>
    <string>image/</string>
    <string>model/</string>
    <string>video/</string>
```

```
  </CONTENT_TYPES>
</IgnoredHttpContentTypes>
```

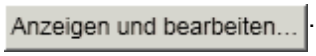
Ausschließen von Zeichenfolgen aus der Korrelationsprüfung

In diesem Abschnitt wird beschrieben, wie Zeichenfolgen aus der Korrelationsprüfung ausgeschlossen werden.

Standardmäßig prüft das Korrelationsmodul den gesamten einfachen und HTML-Text und sucht nach Korrelationen. Einige der gefundenen Ergebnisse sind möglicherweise keine echten Korrelationen. Um die Genauigkeit der Korrelationen zu optimieren, können Sie VuGen so konfigurieren, dass bestimmte Elemente, entweder Textzeichenfolgen oder reguläre Ausdrücke, ignoriert werden, indem sie zur **Liste der ausgeschlossenen Zeichenfolgen** hinzugefügt werden.

1. **Hinzufügen einer Textzeichenfolge**
   a. Wählen Sie **Aufzeichnen > Aufzeichnungsoptionen > Korrelationen > Konfiguration** aus.
   b. Klicken Sie auf die Schaltfläche 

.
   c. Klicken Sie auf die entsprechende Schaltfläche zum Öffnen des Dialogfelds **Liste der ausgeschlossenen Zeichenfolgen**.
   d. Geben Sie die Zeichenfolge ein und klicken Sie auf **OK**.
2. **Hinzufügen eines regulären Ausdrucks**
   a. Wählen Sie **Aufzeichnen > Aufzeichnungsoptionen > Korrelationen > Konfiguration** aus.

      Klicken Sie auf die Schaltfläche 

.
   b. Klicken Sie auf die entsprechende Schaltfläche zum Öffnen des Dialogfelds **Liste der ausgeschlossenen Zeichenfolgen**.
   c. Geben Sie einen regulären Ausdruck ein und aktivieren Sie das Feld **Regulärer Ausdruck**
   - oder -

      Klicken Sie auf die Schaltfläche zum Anzeigen und wählen Sie aus einem Listenfeld mit regulären Ausdrücken Zeichenklassen und füllen Sie den regulären Ausdruck.

      Beispiel:

Bei Eingabe von:	Führt Correlation Studio Folgendes durch:
`getCachedId`	Schließen Sie `getCachedID` als mögliche Korrelation aus.
`^navurl:.*`	Schließen Sie Zeichenfolgen wie `navurl:\\any_char, navurl:1234` als mögliche Korrelation aus.

d. Klicken Sie auf **OK**.

3. **Löschen einer ausgeschlossenen Zeichenfolge**

a. Markieren Sie die Zeichenfolge im Dialogfeld **Liste der ausgeschlossenen Zeichenfolgen**.

b. Klicken Sie auf die Schaltfläche .

Korrelationsfunktionen - Datenbank-Vuser-Skripts

Beim Arbeiten mit Datenbank-Vuser-Skripts (DbLib, CtLib, Oracle, usw.) können Sie die in VuGen enthaltene automatische Korrelationsfunktion zum Einfügen der geeigneten Funktionen in Ihr Skript verwenden. Die Korrelationsfunktionen lauten wie folgt:

- **lrd_save_col** speichert das Abfrageergebnis, das in einer Tabelle erscheint, in einem Parameter. Diese Funktion wird vor dem Abfragen der Daten platziert. Sie weist den von der nachfolgenden Funktion **lrd_fetch** abgerufenen Wert dem vorgegebenen Parameter zu. (**lrd_ora8_save_col** für Oracle 8 und aktueller)
- Die Funktion **lrd_save_value** speichert den aktuellen Wert eines Platzhalterdeskriptors in einem Parameter. Sie wird zusammen mit Datenbankfunktionen verwendet, die Platzhalter für Ausgaben definieren (wie z. B. bestimmte, unter Oracle gespeicherte Prozeduren).
- Die Funktion **lrd_save_ret_param** speichert den Rückgabewert einer gespeicherten Prozedur in einem Parameter. Sie wird hauptsächlich zusammen mit den in DbLib gespeicherten Datenbankprozeduren verwendet, die Rückgabewerte generieren.

Hinweis: VuGen wendet keine Korrelationen an, wenn der gespeicherte Wert ungültig oder NULL ist (keine Zeilen zurückgegeben).

Weitere Informationen zu diesen Funktionen und ihren Argumenten finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Korrelationsfunktionen - Java-Vuser-Skripts

Zum Korrelieren von Anweisungen für Java-Vuser können Sie die Java-Vuser-Korrelierungsfunktionen verwenden. Diese Funktionen können für alle Java-Vuser verwendet

werden, um eine Zeichenfolge in einem Parameter zu speichern und bei Bedarf abzurufen.

lr.eval_string	Ersetzt einen Parameter durch seinen aktuellen Wert.
lr.eval_data	Ersetzt einen Parameter durch einen Bytewert.
lr.eval_int	Ersetzt einen Parameter durch einen Ganzzahlwert.
lr.eval_string	Ersetzt einen Parameter durch eine Zeichenfolge.
lr.save_data	Speichert ein Byte als Parameter.
lr.save_int	Speichert eine Ganzzahl als Parameter.
lr.save_string	Speichert eine mit Null endende Zeichenfolge in einem Parameter.

Beim Aufzeichnen einer CORBA- oder RMI-Sitzung führt VuGen die Korrelation intern durch. Weitere Informationen finden Sie unter "Korrelieren von Skripts - Java" auf Seite 308.

Verwenden der Funktionen für Java-Zeichenfolgen

Beim Programmieren von Java-Vuser-Skripts können Sie die Funktionen für Java-Zeichenfolgen zum Korrelieren Ihrer Skripts verwenden. Im folgenden Beispiel ersetzt **lr.eval_int** die Variable **ID_num** durch ihren Wert, der an einer früheren Stelle im Skript definiert wurde.

```
lr.message(" Track Stock: " + lr.eval_int(ID_num));
```

Im folgenden Beispiel ordnet **lr.save_string** John Doe dem Parameter **Student** zu. Dieser Parameter wird dann in einer Ausgabemeldung verwendet.

```
lr.save_string("John Doe", "Student");
// ...
lr.message("Get report card for " + lr.eval_string("<Student>"));
classroom.getReportCard
```

Web_reg_save_param - Funktionsdetails

Wenn Sie ein Skript ausführen, prüft die Funktion **web_reg_save_param** die nachfolgende HTML-Seite, auf die zugegriffen wird. Sie legen eine linke und/oder rechte Begrenzung fest und VuGen sucht zwischen diesen Begrenzungen nach Text. Wenn VuGen den Text findet, wird der Text einem Parameter zugewiesen.

Die Syntax der Funktion lautet wie folgt:

```
int web_reg_save_param (const char *mpszParamName, <Liste der Attribute>, LAST);
```

In der folgenden Tabelle sind die verfügbaren Attribute aufgeführt. Beachten Sie, dass bei Zeichenfolgen mit Attributwerten (zum Beispiel **Search=all**) die Groß-/Kleinschreibung nicht beachtet werden muss.

NotFound	Hiermit wird die Vorgehensweise angegeben, wenn eine Begrenzung nicht gefunden wird und eine leere Zeichenfolge erzeugt wird. Mit der Standardeinstellung **FEHLER** wird angegeben, dass VuGen einen Fehler ausgeben soll, wenn eine Begrenzung nicht gefunden wird. Wenn der Wert auf **"LEER"** gesetzt wird, wird keine Fehlermeldung ausgegeben und die Skriptausführung wird fortgesetzt. Hinweis: Wenn für das Skript **Bei Fehler fortfahren** aktiviert ist, wird das Skript bei nicht gefundener Begrenzung auch dann fortgesetzt, wenn **Nicht gefunden** auf **FEHLER** gesetzt ist, aber es wird eine Fehlermeldung in die erweiterte Protokolldatei geschrieben.
LB	Die linke Begrenzung des Parameters oder der dynamischen Daten. Bei diesem Parameter muss es sich um eine nicht leere Zeichenfolge handeln, die mit Null endet. Bei Begrenzungsparametern muss die Groß-/Kleinschreibung beachtet werden; anderenfalls fügen Sie nach der Begrenzung **/IC** hinzu. Geben Sie nach der Begrenzung **/BIN** an, um binäre Daten festzulegen.
RB	Die rechte Begrenzung des Parameters oder der dynamischen Daten. Bei diesem Parameter muss es sich um eine nicht leere Zeichenfolge handeln, die mit Null endet. Bei Begrenzungsparametern muss die Groß-/Kleinschreibung beachtet werden; anderenfalls fügen Sie nach der Begrenzung **/IC** hinzu. Geben Sie nach der Begrenzung **/BIN** an, um binäre Daten festzulegen.
RelFrameID	Die hierarchische Ebene der HTML-Seite in Bezug auf den angefragten URL. Mögliche Werte sind **Alles** oder eine Zahl.
Search	Der Suchbereich, in dem nach den abgetrennten Daten gesucht werden soll. Mögliche Werte sind **Headers** (Suche nur in den Kopfzeilen), **Text** (Suche nur in den Textdaten) oder **Alles** (Suche in den Textdaten und Kopfzeilen). Der Standardwert ist **Alles**.
ORD	Mit diesem optionalen Parameter wird die Ordnungszahl bzw. die Zahl der Vorkommnisse der Entsprechung angegeben. Die Standardordnungszahl lautet **1**. Wenn Sie **Alles** angeben, werden die Parameterwerte in einem Array gespeichert.
SaveOffset	Der Offset einer untergeordneten Zeichenfolge des gefundenen Werts zum Speichern des Parameters. Der Standardwert ist **0**. Der Offsetwert darf nicht negativ sein.
SaveLen	Die Länge einer untergeordneten Zeichenfolge des gefundenen Werts ab dem angegebenen Offset zum Speichern des Parameters. Der Standardwert ist **-1** und gibt an, dass die Länge bis zum Ende der Zeichenfolge beträgt.
Convert	Die Konvertierungsmethode, die auf die Daten angewendet werden soll: HTML_TO_URL: konvertiert HTML-codierte Daten in ein URL-codiertes Datenformat HTML_TO_TEXT: konvertiert HTML-codierte Daten in ein einfaches Textformat

Korrelationsfunktionen - C-Vuser-Skripts

Zum Korrelieren von Anweisungen für Protokolle, die keine spezifischen Funktionen haben, können Sie die C-Vuser-Korrelierungsfunktionen verwenden. Diese Funktionen können für alle C-Vuser verwendet werden, um eine Zeichenfolge in einem Parameter zu speichern und bei Bedarf abzurufen.

lr_eval_string	Ersetzt alle Vorkommen eines Parameters durch seinen aktuellen Wert.
lr_save_string	Speichert eine mit Null endende Zeichenfolge in einem Parameter.
lr_save_var	Speichert eine Zeichenfolge mit variabler Länge in einem Parameter.

Zusätzliche Informationen zur Syntax dieser Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Verwenden von "lr_eval_string"

Im folgenden Beispiel ersetzt **lr_eval_string** den Parameter **row_cnt** durch seinen aktuellen Wert. Dieser Wert wird über **lr_output_message** an das Ausgabefenster gesendet.

```
lrd_stmt(Csr1, "select count(*) from employee", -1, 1 /*Deferred*/, ...);
lrd_bind_col(Csr1, 1, =;COUNT_D1, 0, 0);
lrd_exec(Csr1, 0, 0, 0, 0, 0);
lrd_save_col(Csr1, 1, 1, 0, "row_cnt");
lrd_fetch(Csr1, 1, 1, 0, PrintRow2, 0);
lr_output_message("value: %s", lr_eval_string("The row count is: <row_cnt>"));
```

Verwenden von "lr_save_string"

Verwenden Sie zum Speichern einer mit NULL endenden Zeichenfolge in einem Parameter die Funktion **lr_save_string**. Verwenden Sie zum Speichern einer Zeichenfolge mit variabler Länge die Funktion **lr_save_var** und legen Sie die Länge der zu speichernden Zeichenfolge fest.

Im folgenden Beispiel ordnet **lr_save_string** einem emp_id-Parameter **777** zu. Dieser Parameter wird dann in einer anderen Abfrage oder zur weiteren Verarbeitung verwendet.

```
lrd_stmt(Csr1, "select id from employees where name='John'",...);
lrd_bind_col(Csr1,1,=;ID_D1,...);
lrd_exec(Csr1, ...);
lrd_fetch(Csr1, 1, ...);
/* GRID showing returned value "777" */
lr_save_string("777", "emp_id");
```

Dialogfeld "Design Studio" [Registerkarte "Korrelation"]

In diesem Dialogfeld können Sie Informationen zu dynamischen Werten im Skript suchen, korrelieren und anzeigen.

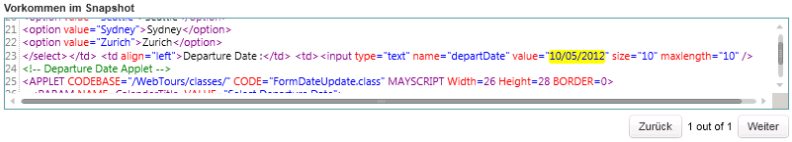
Zugriff	Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Design Studio**. Die Schaltfläche ist nur aktiviert, wenn im Lösungs-Explorer ein aufgezeichnetes Skript vorhanden ist.
Wichtige Informationen	"Registerkarte "Korrelation" [Design Studio] - Übersicht" auf Seite 282
Relevante Aufgaben	"Korrelieren von Skripts mit Design Studio" auf Seite 288

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
Registerkarte "Korrelation"	
Wiedergabe und Prüfung	Design Studio sucht mithilfe aller aktivierten Typen nach dynamischen Werten: Regel, Aufzeichnung und Wiedergabe.
Korrelieren	Ein dynamischer Wert im Skript wird durch einen Korrelationsparameter ersetzt.
Als Regel hinzufügen	Eine dynamische Wertdefinition wird als Regel hinzugefügt. **Regelname.** Ermöglicht das Angeben eines Regelnamens. **Anwendungsname. Ermöglicht das Zuordnen der Regel zu einer bestimmten Anwendung.** Weitere Informationen finden Sie unter "Korrelationen > Regeln" auf Seite 177.
Korrelation rückgängig machen	Der Korrelationsparameter wird durch den ursprünglichen dynamischen Wert ersetzt.

Element der Oberfläche	Beschreibung
Verwerfen	Löscht die ausgewählten dynamischen Werte aus der Korrelationstabelle. Sie können die Aktion zum Verwerfen nur verwenden, wenn der dynamische Wert den Status **Neu** aufweist. Außerdem wird mit dieser Aktion der Text zur Liste der ausgeschlossenen Zeichenfolgen hinzugefügt. Sie können die Liste unter **Aufzeichnungsoptionen > Korrelation > Konfiguration > Liste der ausgeschlossenen Zeichenfolgen** bearbeiten.
Ansicht	Ermöglicht das Filtern der für die Korrelation gefundenen Werte nach den folgenden Typen: • Alle • Neu • Korreliert
Korrelationstabelle Zeigt Details zu den einzelnen dynamischen Werten im Skript an.	
Typ	Zeigt an, welche Engine den dynamischen Wert für Korrelationen gefunden hat: • Aufzeichnen • Regeln • Wiedergabe • Manuell
Gefunden/Ersetzen	Zeigt Informationen zur Anzahl der dynamischen Werte an, die mit denselben Definitionen gefunden wurden. Da Sie eine partielle Korrelation ausführen (d. h., bestimmte Vorkommnisse ersetzen) können, hängen die angezeigten Informationen davon ab, ob Sie den Wert korreliert haben. • **Vor der Korrelation** Anzahl der Werte, die ersetzt werden können/Anzahl der gefundenen Werte • **Nach der Korrelation** Anzahl der Werte, die ersetzt wurden/Anzahl der gefundenen Werte

Element der Oberfläche	Beschreibung
Status	Zeigt den Korrelationsstatus des dynamischen Werts aus dem Skript an: • Neu • Korreliert
Text in Antwort	Zeigt die Zeichenfolge des dynamischen Werts aus dem Skript an.
Korrelationsparameter	Zeigt den Namen des Korrelationsparameters des dynamischen Werts an.
Korrelationsdetails-Chevron Zeigt Details zum dynamischen Wert im Snapshot/Skript an.	
Registerkarte "Ursprünglicher Schritt für Snapshot"	
Details zu einem Schritt im Skript	
Name	Zeigt den Namen des Schrittes im Skript an, in dem der dynamische Wert gefunden wurde.
Zeile	Zeigt die Zeile des Skripts an, in der der dynamische Wert gefunden wurde.
Aktionsname	Zeigt den Namen der Aktion aus dem Skript an, in der der dynamische Wert gefunden wurde.
Beschreibung	Zeigt eine Beschreibung des Schrittes an.
Details zur Korrelationsdefinition	
Typ	Zeigt die API-Funktion an, die zum Korrelieren des Wertes verwendet wird. Regulärer Ausdruck: web_reg_save_param_regexp Begrenzungsbasiert: web_reg_save_param_ex
Definition	Zeigt die Definition des dynamischen Werts an. • **Regulärer Ausdruck.** Die Korrelation eines dynamischen Werts wird durch einen regulären Ausdruck definiert. Ein regulärer Ausdruck ist eine spezielle Textzeichenfolge für die Beschreibung eines Suchmusters. • **Begrenzungsbasiert.** Die Korrelation eines dynamischen Werts wird durch begrenzende Textzeichenfolgen links und rechts definiert.

Element der Oberfläche	Beschreibung
Definition anwenden	Ermöglicht die Auswahl der Definition, die auf den dynamischen Wert angewendet werden soll. Sie können einen Bildlauf durch die Definition des dynamischen Werts in **Vorkommen im Snapshot** durchführen, indem Sie auf **Zurück** oder **Weiter** klicken.
Vorkommen im Snapshot	**Aufzeichnungs-Snapshot.** Zeigt alle Vorkommnisse des dynamischen Werts im Aufzeichnungs-Snapshot an, sobald das Skript wiedergegeben wurde. Sie können einen Bildlauf ausführen, um alle Vorkommnisse im Snapshot anzuzeigen.  **Wiedergabe-Snapshot.** Wenn als Prüfungstyp **Wiedergabe** ausgewählt wurde, zeigt **Design Studio** alle Vorkommnisse des dynamischen Werts im Wiedergabe-Snapshot an, sobald das Skript wiedergegeben wurde. Sie können einen Bildlauf ausführen, um alle Vorkommnisse im Snapshot anzuzeigen. **Hinweis:** Sobald der Wert korreliert wurde, ist der Wiedergabe-Snapshot leer. Wenn Sie die **Korrelationsdefinition** ändern, ist der Wiedergabe-Snapshot leer.
Registerkarte "Auftreten von Korrelationen"	
Vorkommen im Skript	Zeigt die Vorkommnisse des dynamischen Werts im Skript an. Sie können alle Werte korrelieren oder einzelne Werte zur Korrelation auswählen, indem Sie das Kontrollkästchen neben dem Vorkommnis aktivieren.
Optionen	Öffnet das Dialogfeld **Aufzeichnungsoptionen**. Weitere Informationen finden Sie unter: "Korrelationen > Konfiguration" auf Seite 175 "Korrelationen > Regeln" auf Seite 177

Wiedergeben eines Vuser-Skripts

Im Abschnitt **Wiedergeben eines Vuser-Skripts** werden die verschiedenen Verfahren zur Wiedergabe von Vuser-Skripts beschrieben.

Übersicht über die Wiedergabe

Das Erstellen eines Vuser-Skripts umfasst die nachfolgend aufgeführten Schritte. Dieses Thema enthält eine Übersicht über den vierten Schritt (Wiedergeben eines Vuser-Skripts).

Ein aufgezeichnetes Vuser-Skript können Sie mithilfe von VuGen wiedergeben. Auf diese Weise können Sie die Basisfunktionalität des Vuser-Skripts testen und Fehler und Probleme erkennen, die behoben werden müssen. Dass eine Korrelation erforderlich ist, stellt ein typisches Problem dar, das bei der ersten Wiedergabe eines Skripts auftritt. Weitere Informationen zur Korrelation finden Sie unter "Korrelieren – Übersicht" auf Seite 281. Wenn die Wiedergabe erfolgreich ist, können Sie dem Skript Lasttestfunktionen hinzufügen. Zu diesen Funktionen können Parametrierung, Transaktionen und Rendezvous-Punkte zählen.

- Weitere Informationen zum Wiedergeben eines Vuser-Skripts finden Sie unter "Wiedergeben eines Vuser-Skripts" unten.
- Wenn bei der Wiedergabe des Skripts Fehler auftreten, können Sie die Debugging-Funktionen von VuGen verwenden, um die Probleme zu beheben. Weitere Informationen finden Sie unter "Debuggen – Übersicht" auf Seite 413.
- Ausführliche Informationen zu den Dateien, die von VuGen während einer Skriptwiedergabe erstellt werden, finden Sie unter "Bei der Wiedergabe erstellte Dateien" auf Seite 411.
- Sie können Textmarken verwenden, um zwischen den Skriptabschnitten zu navigieren. Weitere Informationen finden Sie unter "Textmarken - Übersicht" auf Seite 407.
- Sie können ein Vuser-Skript über eine Windows-Eingabeaufforderung ausführen. Weitere Informationen finden Sie unter "Ausführen eines Vuser-Skripts über eine Befehlszeile" auf Seite 407.
- Sie können ein Vuser-Skript über eine Linux-Befehlszeile ausführen. Weitere Informationen finden Sie unter "Ausführen eines Vuser-Skripts über eine Linux-Befehlszeile" auf Seite 408.

Wiedergeben eines Vuser-Skripts

In dieser Aufgabe wird beschrieben, wie Sie ein Vuser-Skript wiedergeben.

1. **Konfigurieren der Laufzeiteinstellungen und Wiedergabeoptionen**
   a. Laufzeiteinstellungen steuern, wie Ihr Vuser-Skript wiedergegeben wird. Greifen Sie auf die gewünschten Laufzeiteinstellungen durch einen Doppelklick auf den Knoten **Laufzeiteinstellungen** im **Lösungs-Explorer** zu.

Einen Überblick über die Laufzeiteinstellungen finden Sie unter Run-Time Settings Overview.

b. Geben Sie die Wiedergabeoptionen an, indem Sie **Extras > Optionen** auswählen. Details zu den Optionen finden Sie unter "Optionen - Übersicht" auf Seite 73.

2. **Wiedergeben des Skripts**

Um ein Vuser-Skript bis zum Skriptende oder bis zum nächsten Haltepunkt auszuführen, wählen Sie eine der folgenden Vorgehensweisen aus:

- Wählen Sie **Wiedergeben > Ausführen** aus.
- Klicken Sie in der Vuser-Symbolleiste auf die Schaltfläche **Wiedergabe** .
- Drücken Sie **F5**.

Hinweis: Der Status der Vuser-Skriptausführung wird in der unteren linken Ecke von VuGen angezeigt. Der Status der Skriptausführung kann **Bereit**, **Wird ausgeführt** oder **Pausiert** sein.

- Um ein aktuell ausgeführtes Skript zu beenden, klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Wiedergabe beenden** .
- Um die Ausführung des Skripts zu unterbrechen, klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Wiedergabe unterbrechen** .
- Um die Ausführung eines unterbrochenen Skripts fortzusetzen, klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Weiter** .

3. **Anzeigen der Protokolle zum Abrufen von detaillierten Informationen**

Sie können detaillierte Informationen zum Verhalten des Skripts während der Wiedergabe anzeigen. Diese Information wird im Ausgabefenster angezeigt. Weitere Informationen finden Sie unter "Ausgabebereich" auf Seite 107.

Weitere Informationen über die Wiedergabe eines Vuser-Skripts finden Sie unter "Übersicht über die Wiedergabe" auf der vorherigen Seite.

Laufzeiteinstellungen

In diesem Abschnitt werden die verschiedenen Vuser-Laufzeiteinstellungen beschrieben.

Laufzeiteinstellungen – Sonstige Themen

Dieser Abschnitt enthält eine Vielzahl unterschiedlicher Themen in Zusammenhang mit den Laufzeiteinstellungen.

Inhaltsprüfung – Übersicht

Mit den Laufzeitoptionen für ContentCheck können Sie den Inhalt einer Seite nach einer bestimmten Zeichenfolge durchsuchen. Dadurch lassen sich nicht-standardmäßige Fehler erkennen. Wenn der Anwendungsserver bei normalen Operationen fehlschlägt, zeigt der Browser eine generische HTTP-Fehlerseite mit der Art des Fehlers an. Die Standard-Fehlerseiten werden von VuGen erkannt und als Fehler behandelt, sodass das Skript fehlschlägt. Manche Anwendungsserver geben jedoch eigene Fehlerseiten aus, die VuGen nicht als Fehlerseiten erkennt. Die Seite wird vom Server gesendet und enthält eine formatierte Textzeichenfolge, die auf einen Fehler hinweist.

Angenommen, die Anwendung gibt bei Fehlern eine benutzerdefinierte Seite aus, die den Text **ASP Error** enthält. Sie legen fest, dass VuGen alle zurückgegebenen Seiten nach diesem Text durchsucht. Wenn VuGen die Zeichenfolge erkennt, schlägt die Wiedergabe fehl.

Hinweis: VuGen durchsucht den Text der Seiten, nicht die Kopfzeilen.

Kopieren, Importieren und Exportieren von Laufzeiteinstellungen

In diesem Thema wird beschrieben, wie Sie Laufzeiteinstellungen von einem Vuser-Skript in ein anderes kopieren, importieren und exportieren können. Die Funktionen stehen für alle Vuser-Protokolle zur Verfügung. Einen Überblick über die Funktionen für Laufzeiteinstellungen finden Sie unter Run-Time Settings - Overview.

Hinweis: Für alle nachfolgenden Aufgaben gilt:

- Das Skript, aus dem die Laufzeiteinstellungen abgerufen werden, ist Skript A.
- Das empfangende Skript ist Skript B.

Kopieren von Laufzeiteinstellungen von einem Skript in ein anderes

1. Stellen Sie sicher, dass Skript A und Skript B geöffnet sind.
2. Klicken Sie im Lösungs-Explorer mit der rechten Maustaste auf den Knoten **Laufzeiteinstellungen** von Skript A, und wählen Sie im Kontextmenü **Kopieren** aus.
3. Klicken Sie im Lösungs-Explorer mit der rechten Maustaste auf den Knoten

Laufzeiteinstellungen von Skript B, und wählen Sie im Kontextmenü **Einfügen** aus.

4. Bestätigen Sie den Vorgang im Meldungsfeld durch Klicken auf **Ja**.
5. Speichern Sie Skript B.

Hinweis: Die Einstellungen aus Skript A werden mithilfe der folgenden Logik in Skript B kopiert:

- Einstellung in Skript A und Einstellung in Skript B: Die Einstellung in Skript B wird überschrieben
- Einstellung in Skript A und keine Einstellung in Skript B: Die Einstellung in Skript B wird auf den Wert der Einstellung in Skript A festgelegt.
- Keine Einstellung in Skript A und Einstellung in Skript B: Die Einstellung in Skript B bleibt unverändert.

Exportieren der Laufzeiteinstellungen aus Skript A

1. Öffnen Sie Skript A.
2. Klicken Sie im Lösungs-Explorer mit der rechten Maustaste auf den Knoten **Laufzeiteinstellungen** des Skripts, und wählen Sie **Exportieren** aus.
3. Wählen Sie im Dialogfeld **Laufzeiteinstellungen exportieren** einen Speicherort für die Datei aus, geben Sie einen Namen für die Datei ein, stellen Sie sicher, dass der Dateityp XSSS ist (Skripteinstellungsdatei), und klicken Sie auf **OK**.

Importieren der Laufzeiteinstellungen in Skript B

1. Öffnen Sie Skript B.
2. Klicken Sie im Lösungs-Explorer mit der rechten Maustaste auf den Knoten **Laufzeiteinstellungen** des Skripts, und wählen Sie **Importieren** aus.
3. Wählen Sie im Dialogfeld **Laufzeiteinstellungen importieren** die zu importierende Datei aus, und klicken Sie auf **OK**.
4. Bestätigen Sie den Vorgang im Meldungsfeld durch Klicken auf **Ja**.
5. Speichern Sie Skript B.

Hinweis: Die Einstellungen aus Skript A werden mithilfe der folgenden Logik in Skript B kopiert:

- Einstellung in Skript A und Einstellung in Skript B: Die Einstellung in Skript B wird überschrieben

- Einstellung in Skript A und keine Einstellung in Skript B: Die Einstellung in Skript B wird auf den Wert der Einstellung in Skript A festgelegt.
- Keine Einstellung in Skript A und Einstellung in Skript B: Die Einstellung in Skript B bleibt unverändert.

Zurücksetzen der Laufzeiteinstellungen in einem Skriptknoten auf die Standardeinstellungen

1. Wählen Sie im Lösungs-Explorer den Knoten **Laufzeiteinstellungen** des zu ändernden Skripts aus, und navigieren Sie zu der erforderlichen Einstellung.
2. Klicken Sie auf **Standardwerte verwenden**.
3. Klicken Sie auf **OK**.
4. Speichern Sie das Skript. Nur die Standardwerte für den angezeigten Knoten werden geändert. Wenn Sie die Standardwerte für alle Laufzeiteinstellungen zurücksetzen möchten, müssen Sie diese Schritte für jeden Knoten einzeln ausführen.

Hinweis:

- Laufzeiteinstellungen können nur innerhalb von VuGen mit Kopieren/Einfügen und Importieren/Exportieren geändert werden.
- Kopieren/Einfügen kann nur auf die allgemeinen Laufzeiteinstellungen wie z. B. Latenzzeit, Iterationen und Protokoll angewendet werden.

Multithreading

Der Controller verwendet ein Treiberprogramm (z. B. *mdrv.exe* oder *r3vuser.exe*), um Vuser auszuführen. Wenn Sie jeden Vuser als Prozess ausführen, wird das gleiche Treiberprogramm wiederholt gestartet und in den Speicher geladen (für jede Vuser-Instanz). Durch das mehrmalige Laden eines Treiberprogramms in den Speicher werden große Teile des Arbeitsspeichers (RAM, randon access memory) und anderer Systemressourcen beansprucht. Dadurch wird die Anzahl der Vuser beschränkt, die auf einem Lastgenerator ausgeführt werden können.

Wenn Sie alternativ jeden Vuser als Thread ausführen, startet der Controller lediglich eine Instanz des Treiberprogramms (wie *mdrv.exe*) für 50 Vuser (Standardeinstellung). Dieser Treiberprozess/dieses Treiberprogramm startet mehrere Vuser, wobei jeder Vuser als Thread ausgeführt wird. Diese als Thread ausgeführten Vuser teilen sich Speichersegmente des übergeordneten Treiberprozesses. Auf diese Weise ist ein wiederholtes Laden des Treiberprogramms/-prozesses nicht nötig und es wird sehr viel weniger Speicherplatz beansprucht, sodass auf einem Lastgenerator mehr Vuser ausgeführt werden können.

Informationen zur Konfiguration dieser Optionen finden Sie unter "Allgemein > Verschiedenes (Knoten)" auf Seite 366.

Tabelle für Protokollkompatibilität

In der folgenden Tabelle werden die Vuser-Protokolle und die jeweils verfügbaren Knoten in den Laufzeiteinstellungen aufgeführt.

Protokoll	Strukturknoten in den Laufzeiteinstellungen
.NET	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • .NET - .NET-Umgebung
Ajax	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation • Browser – Browseremulation • Internetprotokoll – Proxy, Einstellungen, Download-Filter, ContentCheck
C VUser	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes
Citrix ICA	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation • Citrix – Konfiguration, Synchronisierung
COM/DCOM	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes
DNS	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes
FTP	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation

Protokoll	Strukturknoten in den Laufzeiteinstellungen
Flex	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation • Browser – Browseremulation • Flex - RTMP, Konfiguration, Externalisierbare Objekte • Internetprotokoll – Proxy, Einstellungen, Download-Filter, ContentCheck
IMAP (Internet Messaging)	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation
Java over HTTP	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation • Browser – Browseremulation • Java-Umgebungseinstellungen – Classpath, Java VM • Internetprotokoll – Proxy, Einstellungen, Download-Filter
Java Record Replay, Java Vuser	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Java-Umgebungseinstellungen – Classpath, Java VM
LDAP	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation
MMS (Media Player)	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation
RDP	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation • RDP – Konfiguration, Synchronisierung, Erweitert, RDP-Agent

Protokoll	Strukturknoten in den Laufzeiteinstellungen
MAPI (Microsoft Exchange)	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes
MMS (Multimedia Messaging Service)	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Browser – Browseremulation • MMS – Server und Protokoll • WAP – Radius, Gateway • Internetprotokoll – Proxy, Einstellungen, Download-Filter
ODBC	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes
Oracle 2-Tier	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes
Oracle NCA	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation • Oracle NCA – Clientemulation
Oracle Web Applications 11i	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation • Browser – Browseremulation • Internetprotokoll – Proxy, Einstellungen, Download-Filter, ContentCheck • Oracle NCA – Clientemulation
POP3	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation
RTE	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • RTE - RTE

Protokoll	Strukturknoten in den Laufzeiteinstellungen
SAP Web	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation • Browser – Browseremulation • Internetprotokoll – Proxy, Einstellungen, Download-Filter, ContentCheck
SAP Click and Script	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation • Browser – Browseremulation • Internetprotokoll – Proxy, Einstellungen, Download-Filter, ContentCheck
SAP GUI	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • SAP GUI - Allgemein • Netzwerk – Geschwindigkeitssimulation
Siebel Web	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation • Browser – Browseremulation • Internetprotokoll – Proxy, Einstellungen, Download-Filter, ContentCheck

Protokoll	Strukturknoten in den Laufzeiteinstellungen
Silverlight	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation • Browser – Browseremulation • Silverlight – Dienste • Datenformaterweiterung – Konfiguration • Internetprotokoll – Proxy, Einstellungen, Download-Filter, ContentCheck
SMTP	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation
TruClient Ajax	• Allgemein – Takt, Zusätzliche Attribute, Protokoll, Lastmodus-Browsereinstellungen, Sonstige Einstellungen
Web (HTTP/HTML)	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation • Browser – Browseremulation • Internetprotokoll – Proxy, Einstellungen, Download-Filter, ContentCheck • Datenformaterweiterung – Konfiguration
Webdienste	• Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation • JMS - Erweitert
Windows-Sockets	• Allgemein – Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • Netzwerk – Geschwindigkeitssimulation

Browser > Browseremulation (Knoten)

Ermöglicht Ihnen, die Laufzeiteinstellungen für den Browser festzulegen.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Browser > Browseremulation**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Browser-Cache simulieren	Weist den Vuser an, einen Browser mit einem Cache zu simulieren (standardmäßig aktiviert). Selbst wenn Sie diese Option deaktivieren, wird jede Ressource für jede Seite nur ein Mal heruntergeladen, auch wenn sie mehrmals vorkommt. Bei einer Ressource kann es sich um ein Bild, einen Frame oder eine andere Art von Skriptdatei handeln. Wenn Sie diese Option aktivieren, stehen folgende zusätzliche Optionen zur Verfügung: • **URLs zwischenspeichern, die Inhalt (HTML) erfordern.** Durch diese Option speichert VuGen nur die URLs zwischen, die den HTML-Inhalt erfordern. Dieser Inhalt wird möglicherweise für Analyse, Verifizierung oder Korrelation benötigt. Wenn Sie diese Option auswählen, wird HTML-Inhalt automatisch im Zwischenspeicher abgelegt. **URLs angeben, die zusätzlich zu HTML-Seiten weitere Inhalte erfordern**. Ermöglicht die Angabe zusätzlicher Inhaltstypen (kein HTML), die zwischengespeichert werden sollen. Trennen Sie alle Inhaltstypen mit einem Semikolon in der folgenden Zeile **Inhaltstypen**. **Tipp:** Um eine geringere Speicherbelegung für die virtuellen Benutzer zu gewährleisten, lassen Sie diese Option deaktiviert, falls sie für Ihren Test nicht unbedingt erforderlich ist. • **Bei jedem Aufrufen einer gespeicherten Seite nach aktueller Version suchen** Durch diese Einstellung sucht der Browser nach Versionen des angegebenen URL, die neuer als die zwischengespeicherte Version sind. Wenn Sie diese Option aktivieren, fügt VuGen das Attribut "If-modified-since" zur HTTP-Kopfzeile hinzu. Diese Option liefert die neueste Version der Seite, führt aber zu höherem Datenverkehr während der Szenario- oder Sitzungsausführung. **Standardwert:** Deaktiviert.

Element der Oberfläche	Beschreibung
Nicht-HTML-Ressourcen herunterladen	Diese Option bewirkt, dass Vuser beim Zugriff auf eine Webseite während der Wiedergabe die Grafiken laden. Dazu zählen sowohl Grafiken, die mit der Seite aufgezeichnet wurden, als auch Grafiken, die nicht explizit zusammen mit der Seite aufgezeichnet wurden. Wenn echte Benutzer auf eine Webseite zugreifen, warten sie, bis die Bilder geladen werden. Daher sollten Sie diese Option aktivieren, wenn Sie das gesamte System einschließlich Endbenutzerzeit testen möchten. Wenn Sie die Leistung erhöhen und keine echten Benutzer emulieren möchten, deaktivieren Sie diese Option. Deaktivieren Sie diese Option, wenn Abweichungen bei Bildprüfungen auftreten, da sich manche Bilder bei jedem Zugriff auf eine Webseite ändern (z. B. Werbebanner).
Bei jeder Iteration neuen Benutzer simulieren	Durch diese Option setzt VuGen alle HTTP-Kontexte zwischen Iterationen in den Zustand am Ende des Abschnitts init zurück. Mit dieser Einstellung kann ein neuer Benutzer, der eine Browsersitzung beginnt, durch den Vuser genauer emuliert werden. Diese Option löscht alle Cookies, schließt alle TCP-Verbindungen (einschließlich Keep-Alive-Verbindungen), leert den Zwischenspeicher des emulierten Browsers, setzt die HTML-Framehierarchie zurück (Framenummerierung beginnt mit 1) und löscht die Benutzernamen und Passwörter. (Standardmäßig aktiviert) **Hinweis:** Asynchrone Konversationen werden am Ende jeder Iteration beendet, auch wenn diese Einstellung deaktiviert ist. • **Clear cache on each iteration.** Der Browserzwischenspeicher wird nach jeder Iteration geleert, um einen Benutzer zu simulieren, der eine Webseite zum ersten Mal besucht. Deaktivieren Sie das Kontrollkästchen, damit Vuser die Informationen aus dem Browserzwischenspeicher verwenden können. So wird ein Benutzer simuliert, der die Seite bereits vor kurzem aufgerufen hatte.
Benutzer-Agent - Auswahl	Ermöglicht Ihnen die Auswahl des zu emulierenden Browsers. Alternativ können Sie eine benutzerdefinierte Zeichenfolge für den Benutzer-Agent angeben. Alle Internet-Vuser weisen die Kopfzeile User-Agent auf, mit der der emulierte Browsertyp (oder das Toolkit für Wireless) identifiziert wird. Beispiel: **Mozilla/4.0 (kompatibel; MSIE 6.0; Windows; Trident/4.0; English (United States))** gibt an, dass als Browser Internet Explorer unter Windows auf einem Intel-Computer ausgeführt wird. **Browser verwenden.** Verwendet einen in der Dropdownliste verfügbaren Standardbrowser. Sie wählen die Version, Plattform und Sprache. **Benutzerdefinierter Browser.** Gibt eine benutzerdefinierte Zeichenfolge für den Benutzer-Agent an.

Knoten "Citrix > Konfiguration"

Ermöglicht Ihnen das Konfigurieren der Laufzeiteinstellungen der Citrix-Konfiguration.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Citrix > Konfiguration**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Warteschlange für Mausbewegungen und Tastenbedienungen erstellen	Durch diese Option erstellen Vuser eine Warteschlange für Mausbewegungen und Tastenbedienungen, um diese paketweise und damit weniger häufig an den Server zu senden. Mit dieser Einstellung wird der Netzwerkdatenverkehr bei langsamen Verbindungen reduziert. Ist diese Option aktiviert, wird während der Sitzung weniger häufig auf Tastatureingaben und Mausbewegungen reagiert. **Standardwert:** Deaktiviert.
Soundqualität	Legt die Soundqualität fest. Wenn der Clientcomputer nicht mit einer Sound Blaster-kompatiblen 16-Bit-Soundkarte ausgestattet ist, wählen Sie **Sound Off** aus. Bei aktivierter Soundunterstützung können Sie Sounddateien aus veröffentlichten Anwendungen auf dem Clientcomputer abspielen.
SpeedScreen-Latenzreduzierung	Der Mechanismus, mit dem die Benutzerinteraktion bei niedriger Netzwerkgeschwindigkeit verbessert wird. Sie können für diesen Mechanismus die Option **on** oder **off** auswählen, je nach Netzwerkgeschwindigkeit. Mit der Option **auto** wird der Mechanismus abhängig von der aktuellen Netzwerkgeschwindigkeit ein- oder ausgeschaltet. Wenn Sie die Netzwerkgeschwindigkeit nicht kennen, wählen Sie die Option **Serverstandard verwenden** aus, um die Standardeinstellung des Computers zu verwenden.
Datenkomprimierung verwenden	Durch diese Einstellung werden die übertragenen Daten von den Vusern komprimiert. Das Aktivieren der Datenkomprimierung empfiehlt sich bei begrenzter Bandbreite. **Standardwert:** Aktiviert.
Datenträgercache für Bitmaps verwenden	Durch diese Einstellung verwenden Vuser für Bitmaps und häufig verwendete Grafikobjekte einen lokalen Zwischenspeicher. Das Aktivieren der Datenkomprimierung empfiehlt sich bei begrenzter Bandbreite. **Standardwert:** Deaktiviert.

Knoten "Citrix > Synchronisierung"

Ermöglicht Ihnen das Konfigurieren der Laufzeiteinstellungen für die Citrix-Synchronisierung.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Citrix > Synchronisierung**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Standardmäßige Bildsynchronisierungstoleranz	Mit dieser Einstellung steuern Sie die Übereinstimmungsebene, die zwei Bilder aufweisen müssen, um als synchronisiert zu gelten. **Exact.** Erfordert eine Übereinstimmung von 100%. **Low.** Erfordert eine Übereinstimmung von 95%. **Medium.** Erfordert eine Übereinstimmung von 85 %. **High.** Erfordert eine Übereinstimmung von 70 %. **Standardwert:** Exakt.
Timeout	• **Verbindungszeit.** Die Zeit (in Sekunden), die bei bestehender Verbindung ohne Aktivität gewartet werden soll, bevor die Verbindung unterbrochen wird. • **Wartezeit.** Die Zeit (in Sekunden), die an einem Synchronisierungspunkt ohne Aktivität gewartet werden soll, bevor die Verbindung unterbrochen wird.
Eingaberate	Die Verzögerung (in Millisekunden) zwischen Tastenanschlägen.
Standardmäßige Bildsynchronisierungstoleranz	Die standardmäßige Toleranzebene für Synchronisierungsvorgänge bei Bildern: **Genau**, **Niedrig**, **Mittel** oder **Hoch**.
Interne Verzögerung bei Mausklick	Die Verzögerung (in Millisekunden) zwischen dem Senden von MOUSE_UP- und MOUSE_DOWN-Ereignissen.
Verzögerung bei Bitmapabfragen	Die Intervalle (in Millisekunden), in denen der Status des Bildschirms während der Bitmapsynchronisierung abgerufen wird. Geben Sie "0" an, wenn anstelle von erzwungenem Abruf die standardmäßigen Citrix-Clientbenachrichtigungen verwendet werden sollen.

Knoten "COM/DCOM - Filter"

Damit können Sie die aufzuzeichnenden COM/DCOM-Objekte definieren.

Zugriff	Führen Sie einen der folgenden Schritte durch: • **Aufzeichnen > Aufzeichnungsoptionen > COM/DCOM > Filter** • **Wiedergeben > Laufzeiteinstellungen > COM/DCOM > Filter**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
DCOM Profil	**Wählen Sie einen der folgenden Filtertypen:** • **Standardfilter.** Der Filter, der standardmäßig zur Aufzeichnung eines COM-Vuser-Skripts verwendet wird. • **Neuer Filter.** Ein leerer Filter, der auf den Einstellungen der Standardumgebung basiert. Beachten Sie, dass Sie einen Namen für diesen Filter angegeben müssen, bevor Sie ihn zur Aufzeichnung verwenden können. Mit den Schaltflächen **Speichern unter** und **Löschen** können Sie auch die aktuellen Einstellungen speichern und einen Filter löschen.

Element der Oberfläche	Beschreibung
Liste "DCOM-Listenereinstellungen"	Zeigt eine Struktur aus Typbibliotheken an. Durch Erweitern der einzelnen Knoten können Sie alle in einer Typbibliothek verfügbaren Klassen anzeigen. Wenn Sie die Klassenstruktur erweitern, können Sie alle Schnittstellen anzeigen, die von einer Klasse unterstützt werden. Deaktivieren Sie das Kontrollkästchen neben einem Bibliotheksnamen, um die betreffende Bibliothek auszuschließen. Hierdurch werden auch alle dort enthaltenen Klassen ausgeschlossen. Wenn Sie die Struktur erweitern, können Sie auch einzelne Klassen oder Schnittstellen durch Deaktivieren des zugehörigen Kontrollkästchens ausschließen. Eine Schnittstelle kann von verschiedenen Klassen unterschiedlich implementiert werden. Wenn Sie eine Schnittstelle ausschließen, die von anderen, nicht ausgeschlossenen Klassen implementiert wird, wird ein Dialogfeld geöffnet. Sie werden gefragt, ob die Schnittstelle auch in allen anderen Klassen, die sie implementieren, ausgeschlossen werden soll. Beachten Sie, dass das Deaktivieren des Kontrollkästchens neben einer Schnittstelle funktionell äquivalent zum Auswählen dieser Schnittstelle im Dialogfeld **Ausgeschlossene Schnittstellen** ist. • **Umgebungen.** Die Umgebungen, die aufgezeichnet werden sollen: **ADO Objects**, **RDS Objects** und **Remote Objects**. Deaktivieren Sie die Objekte, die nicht aufgezeichnet werden sollen. • **Typbibliotheken.** Eine Typbibliotheksdatei (.**tlb** oder .**dll**), die das aufzuzeichnende COM-Objekt enthält. Alle COM-Objekte werden durch eine Typbibliothek dargestellt. Sie können die Typbibliothek aus der Registry, auf einem Microsoft Transaction Server oder aus dem Dateisystem auswählen. **Typbibliotheken.** Im unteren Teil des Dialogfelds werden die folgenden Informationen zu den einzelnen Typbibliotheken angezeigt. • **Beschreibung.** Der Name der Typbibliothek (tlb-Datei). • **Pfad.** Der Pfad der Typbibliothek. • **Guid.** Die GUID (Global Unique Identifier) der Typbibliothek.

Element der Oberfläche	Beschreibung
Hinzufügen...	Fügt eine COM-Typbibliothek hinzu. • **Registrierung durchsuchen.** Zeigt eine Liste der Typbibliotheken an, die sich in der Registrierung des lokalen Computers befinden. Wählen Sie die gewünschte Bibliothek durch Aktivieren des zugehörigen Kontrollkästchens aus und klicken Sie auf **OK**. • **Dateisystem durchsuchen**. Ermöglicht die Auswahl der Typbibliotheken aus dem lokalen Dateisystem. • **MTS durchsuchen**. Ermöglich das Hinzufügen einer Komponente aus einem Microsoft Transaction Server. Im Dialogfeld **MTS-Komponenten** werden Sie aufgefordert, den Namen des MTS-Servers einzugeben. Geben Sie den Namen des MTS-Servers ein und klicken Sie auf **Verbindung**. Denken Sie daran, dass zum Aufzeichnen von MTS-Komponenten ein MTS-Client auf dem Computer installiert sein muss. Wählen Sie mindestens ein MTS-Komponentenpaket in der Liste aus und klicken Sie auf **OK**. Sobald das Paket unter **Typbibliotheken** angezeigt wird, können Sie einzelne Komponenten auswählen.
Entfernen	Entfernt eine COM-Typbibliothek.

Element der Oberfläche	Beschreibung
Ausschließen...	Schließt Schnittstellen im Filter über das Dialogfeld **Ausgeschlossene Schnittstellen** aus. In diesem Dialogfeld werden die markierten Schnittstelleneinträge ausgeschlossen. Sie haben auch die Möglichkeit, Schnittstellen hinzuzufügen, die nicht aufgelistet werden. Klicken Sie im Dialogfeld **Ausgeschlosssene Schnittstellen** auf **Schnittstelle hinzufügen...** und geben Sie die GUID-Nummer (Schnittstellen-ID) und den Namen der Schnittstelle ein. Sie können die GUID aus der von VuGen erstellten Datei **interfaces.h** kopieren. Diese wird im VuGen-Fenster in der Auswahlstruktur in der Spalte auf der linken Seite aufgelistet. Sie können **Schnittstelle hinzufügen...** auch verwenden, um Schnittstellen auszuschließen, die unnötigerweise vom Skript aufgerufen werden, aber nicht im Filter aufgeführt sind. Eine Schnittstelle kann von verschiedenen Klassen unterschiedlich implementiert werden. Wenn Sie eine Schnittstelle ausschließen, die von anderen, nicht ausgeschlossenen Klassen implementiert wird, zeigt VuGen eine Warnung an. Wenn Sie die Option **Nicht mehr fragen** aktivieren und das Dialogfeld schließen, dann wird der Status aller Instanzen dieser Schnittstelle in allen anderen Klassen automatisch für diesen Filter geändert, sobald Sie den Status der Schnittstelle in einem Objekt ändern. Klicken Sie auf **Ja zu Allen**, um den Status aller Instanzen dieser Schnittstelle in allen anderen Klassen zu ändern. Klicken Sie auf **Nein zu Allen**, um den Status für alle anderen Instanzen unverändert zu lassen. Klicken Sie auf **Nächste Instanz**, um die nächste Klasse anzuzeigen, die diese Schnittstelle verwendet.

COM/DCOM > Optionen (Knoten)

Ermöglicht das Festlegen zusätzlicher Optionen für COM-Aufzeichnungssitzungen. Diese betreffen die Verarbeitung von Objekten, das Generieren von Protokolldateien und VARIANT-Definitionen.

Die DCOM-Skriptoptionen gelten für alle Programmiersprachen. Anhand der folgenden Einstellungen können Sie die Skriptoptionen für DCOM-Methoden und die Schnittstellenverarbeitung konfigurieren.

Zugriff	VuGen > Wiedergeben > Laufzeiteinstellungen > COM/DCOM > Optionen

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
ADO-Datensatzfilterung	Mehrere Datensatzoperationen werden zu einer einzeiligen FETCH-Anweisung zusammengefasst (standardmäßig aktiviert).

Element der Oberfläche	Beschreibung
Temporäre VARIANT-Typen als global deklarieren	Temporäre VARIANT-Typen werden als globale und nicht als lokale Variablen definiert (standardmäßig aktiviert).
Array in separaten Bereichen füllen	Jedes Array wird in einen separaten Bereich eingefügt (standardmäßig aktiviert).
Struktur in separaten Bereichen füllen	Jede Struktur wird in einen separaten Bereich eingefügt (standardmäßig aktiviert).
COM-Ausnahmen generieren	COM-Funktionen und -Methoden, die während der Aufzeichnung Ausnahmen hervorgerufen haben, werden generiert (standardmäßig deaktiviert).
COM-Statistik generieren	Es werden Leistungsstatistiken und zusammenfassende Informationen erzeugt (standardmäßig deaktiviert).
Größe von SafeArray-Protokoll begrenzen	Begrenzt die Anzahl der pro COM-Aufruf im SafeArray-Protokoll erzeugten Elemente auf 16 (standardmäßig deaktiviert).
COM-Objekte freigeben	Die Freigabe von nicht mehr verwendeten COM-Objekten wird aufgezeichnet (standardmäßig aktiviert).
Datensatzinhalt speichern	Speichert den Inhalt des Datensatzes in Tabellen, um diese in VuGen anzeigen zu können (standardmäßig aktiviert).
Gebundene Monikerobjekte abgefangen	Alle gebundenen Monikerobjekte werden abgefangen (standardmäßig deaktiviert).

Datenformaterweiterung > Knoten "Konfiguration"

Aktiviert Datenformaterweiterungen während der Codegenerierung.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Datenformaterweiterung > Konfiguration**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 250.
Relevante Aufgaben	• "Anwenden von DFE-Ketten auf Abschnitte der HTTP-Nachricht" auf Seite 920 • "Implementieren der GWT-DFE-Unterstützung" auf Seite 928
Siehe auch	• "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912 • "Liste der Datenformaterweiterungen" auf Seite 922

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Datenformaterweiterung aktivieren	Ermöglicht Ihnen das Auswählen von Ketten für jeden Nachrichtenabschnitt der HTTP-Nachricht. Standardmäßig deaktiviert.

Flex > Konfiguration (Knoten) (Laufzeiteinstellungen)

Ermöglicht das Festlegen eines externen JVM-Pfads (Java Virtual Machine).

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Flex > Konfiguration**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Externe JVM verwenden	Ermöglicht die Verwendung einer externen JVM. Wenn Sie diese Option auswählen, müssen Sie den Pfad angeben: **Externer JVM-Pfad** Der vollständige Pfad der externen JVM.
GraniteDS-Konfiguration verwenden	Definiert die serverseitige Datendienstkonfiguration. Wenn Sie diese Option auswählen, wählen Sie nicht **Flex LCDS/BlazeDS-JAR-Dateien verwenden** aus, um die Nachrichten im Knoten **Externalisierbar** zu serialisieren.
Maximal zulässige Größe der formatierten Anforderung/Antwort zur Ausgabe	Geben Sie hier die maximale Zeichenlänge für Anforderungen oder Antworttexte zur Ausgabe in die Protokolldateien ein. Diese Option wirkt sich nur auf die Schritte **flex_amf_call** und **flex_remoting_call** aus. Wenn Sie beispielsweise den Wert 1048576 Zeichen (1 MB) angeben, werden nur Antworten oder Anfragen mit einer Länge von weniger als einem Megabyte im Codegenerierungs- oder Wiedergabeprotokoll erfasst.

Flex > RTMP (Knoten)

Ermöglicht Ihnen, die Laufzeiteinstellungen für Flex-RTMP zu konfigurieren.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Flex > RTMP**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Alle offenen RTMP-Verbindungen nach jeder Iteration schließen.	Alle geöffneten RTMP-Verbindungen werden automatisch am Ende jeder Iteration getrennt.
Zeitüberschreitung für TCP-Empfang (Millisekunden)	Legt das Zeitintervall in Millisekunden fest, für das auf den Empfangsbeginn von TCP/IP-Paketen durch den Client gewartet wird. Beim Erreichen einer Zeitüberschreitung wird abhängig von der Laufzeiteinstellung für **Status bei TCP-Empfangszeitüberschreitung** eine Warnung oder ein Fehler ausgelöst.
Status bei TCP-Empfangszeitüberschreitung	Gibt an, welcher Status bei einer Zeitüberschreitung ausgegeben wird.
Warnung	Gibt beim Erreichen des Zeitüberschreitungswerts eine Warnung für den Schritt aus. Anschließend wird mit dem nächsten Schritt fortgefahren.
Fehler	Gibt beim Erreichen des Zeitüberschreitungswerts einen Fehler für den Schritt aus. Das Skript wird beendet.

Flex > Externalisierbar (Knoten) (Laufzeiteinstellungen)

Mithilfe dieses Dialogfelds können Sie Laufzeiteinstellungen für externalisierbare Objekte in Flex-Skripts konfigurieren.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Flex > Externalisierbar**
Relevante Aufgaben	• Serialisieren mit der externen Java-Serialisierungskomponente • "Serialisieren von Flex-Skripts " auf Seite 704
Siehe auch	• "Flex – Übersicht" auf Seite 682 • "Externalisierbare Objekte in Flex-Skripts" auf Seite 700

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Externalisierbare Objekte nicht codieren	Führen Sie externalisierbare Objekte ohne Codierung aus.
Externalisierbare Objekte codieren mit	Wählen Sie die entsprechende Optionsschaltfläche aus: • **LoadRunner AMF Serializer**. Verwenden Sie den Flex-Server-Parser, um externalisierbare Objekte zu codieren und zu decodieren. Wählen Sie diese Option aus, wenn Sie nicht den Adobe LiveCycle Data Services- (LCDS) oder Adobe BlazeDS-Server verwenden. • **Benutzerdefinierte Java-Klassen**. Mit einem externen Parser können Sie externalisierbaren Objekte codieren und decodieren. Wählen Sie eine oder beide der folgenden Optionen: ▪ **Flex LCDS/BlazeDS-JAR-Dateien verwenden**. Verwenden Sie Flex LCDS- oder BlazeDS-JAR-Dateien für die Serialisierung der Nachrichten. Falls Sie im Knoten **Konfiguration** die Option **GraniteDS-Konfiguration verwenden** ausgewählt haben, aktivieren Sie diese Option nicht. ▪ **Weitere JAR-Dateien verwenden**. Verwenden Sie globale Flex-JAR-Dateien für die Serialisierung der Nachrichten. Geben Sie den vollständigen Pfad zu den Speicherorten der JAR-Dateien im Feld **Speicherort von JAR-Dateien von Flex-Server/Anwendung** an. Trennen Sie mehrere Einträge durch ein Semikolon ab. Wenn Ihre JAR-Dateien denselben Namen wie die LCDS- oder BlazeDS-JAR-Dateien haben und die Option **Flex LCDS/BlazeDS-JAR-Dateien verwenden** aktiviert ist, werden sie überschrieben.

Allgemein > Zusätzliche Attribute (Knoten)

Über den Knoten **Zusätzliche Attribute** können Sie zusätzliche Argumente für Vuser-Skripts bereitstellen. Die Einstellung **Additional Attributes** ist für alle Vuser-Skripttypen gültig.

Sie legen Befehlszeilenargument fest, die zu einem späteren Zeitpunkt während des Testlaufs mithilfe der Funktion **lr_get_attrib_string** abgerufen werden können. Über diesen Knoten können Sie externe Parameter an vorbereitete Skripts übergeben.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Allgemein > Zusätzliche Attribute**

Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Tabelle mit zusätzlichen Attributen>	Eine Liste der zusätzlichen Attribute mitsamt ihren Werten.
	Fügt ein neues Argument hinzu.
	Entfernt ein Argument.

Allgemein > Protokoll (Knoten)

Ermöglicht Ihnen, Umfang und Art der im Protokoll festgehaltenen Informationen festzulegen.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Allgemein > Protokoll**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Protokollierung aktivieren	Aktiviert die automatische Protokollierung während der Wiedergabe. Diese Option ist nur für die automatische Protokollierung und Protokollmeldungen relevant, die über **lr_log_message** ausgegeben werden. Manuell über **lr_message**, **lr_output_message** und **lr_error_message** gesendete Meldungen werden weiterhin ausgegeben.
Protokolloptionen	Legt fest, wann Meldungen ins Protokoll geschrieben werden: • **Meldungen nur bei Fehler senden**. Überträgt Meldungen nur dann an die Protokolldatei, wenn ein Fehler auftritt. Klicken Sie auf **Erweitert**, um die Größe des Protokollcaches festzulegen. Wenn der Inhalt des Zwischenspeichers die festgelegte Größe überschreitet, werden die ältesten Elemente gelöscht. • **Meldungen immer senden.** Alle Meldungen werden an das Protokoll übermittelt.

Standardprotokoll	Erstellt ein Standardprotokoll der Funktionen und Meldungen, die während der Skriptausführung zum Zwecke des Debugging gesendet wurden. Sie können diese Option bei umfangreichen Lasttestszenarios bzw. -profilen deaktivieren, wenn Sie Systemressourcen sparen möchten.
Erweitertes Protokoll	Erstellt ein erweitertes Protokoll mit Warnungen und anderen Meldungen. Sie können diese Option bei umfangreichen Lasttestszenarios bzw. -profilen deaktivieren, wenn Sie Systemressourcen sparen möchten. Darüber hinaus können Sie folgende Optionen angeben: • **Parameter substitution.** Protokolliert alle dem Skript zugewiesenen Parameter mitsamt ihrer Werte. • **Data returned by server.** Protokolliert alle vom Server zurückgegebenen Werte. • **Advanced trace.** Protokolliert alle Funktionen und Meldungen, die vom Vuser während der Sitzung gesendet werden.

Allgemein > Knoten "Protokoll" (Ajax TruClient)

Ermöglicht Ihnen, Art und Umfang der Informationen festzulegen, die für Ajax TruClient-Skripts im Protokoll festgehalten werden.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Allgemein > Protokoll**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Protokollebene und zugehörige Optionen	**Reines Fehlerprotokoll** • Protokolliert nur Fehlermeldungen. **Standardprotokoll** • Nur Fehler/Warnungen protokollieren (im Standardprotokoll). Protokolliert nur Fehler und Warnungen. Meldungen mit informativem Charakter werden nicht protokolliert. Im erweiterten Protokoll werden alle aufgrund dieser Option ignorierten Meldungen unabhängig von dieser Einstellung einbezogen. **Erweitertes Protokoll** • Protokolliert informative Meldungen von geringer Wichtigkeit sowie alle Nachrichten aus dem Standardprotokoll. **HTTP-bezogene Meldungen** • **HTTP-Anforderungsheader protokollieren.** Protokolliert den HTTP-Anforderungsheader jeder Anforderung. • **HTTP-Anforderungstext (POST-Daten) protokollieren.** Protokolliert den HTTP-Anforderungstext jeder Anforderung. • **HTTP-Antwortheader protokollieren**. Protokolliert die HTTP-Antwortheader jeder Antwort. • **HTTP-Antworttext protokollieren.** Protokolliert den HTTP-Antworttext jeder Antwort. • **HTTP-Verfolgung protokollieren.** Protokolliert die Verarbeitung von HTTP-Anforderungen/-Antworten. **Meldungen für die Anwendung im Test (AUT)** • **AUT-Fehlermeldungen protokollieren.** Protokolliert Fehlermeldungen aus den sich im Test befindlichen Anwendungen. • **Andere AUT-Meldungen protokollieren.** Protokolliert Informationsmeldungen aus den sich im Test befindlichen Anwendungen. **Parametrierung**

Element der Oberfläche	Beschreibung
	• **Parametersubstitution protokollieren.** Protokolliert die Parameter samt der zugehörigen Werte, die bei Ausführung des Vuser-Skripts verwendet werden.
Protokolldatei nur bei Bedarf erstellen	**Meldungen im Speicher sammeln, nur bei Bedarf schreiben.** • Protokolliert Nachrichten nur, wenn ein Fehler auftritt. • **Puffergröße für 'Meldungen im Speicher sammeln' (KB).** Geben Sie die Puffergröße im Speicher ein, die für wichtige Nachrichten reserviert ist. **Hinweis:** Bei einem zu geringen Wert gehen Nachrichten verloren. Ist der Wert zu groß, kann dies zu Problemen mit der Auslagerungsdatei führen und die Ausführung verlangsamen.
Formatierungsoptionen für Protokollmeldungen	**Zeilenlänge im Benutzerprotokoll.** • Geben Sie hier die maximale Länge für Zeilen in Benutzerprotokolldateien ein, bevor diese umgebrochen werden. **Zeitstempel in Benutzerprotokoll aufnehmen.** • Die einzelnen Meldungen werden mit Zeitstempel protokolliert.
Interne Supportoptionen	Diese Optionen sollten nur zusammen mit einem Mitarbeiter des Kundendienstes verwendet werden. Nehmen Sie daran nur dann Änderungen vor, wenn Sie zum Kundendienst dazu aufgefordert wurden.

Allgemein > Erweiterte Einstellungen (TruClient)

In diesem Dialogfeld können Sie die erweiterten Einstellungen für TruClient Ajax - Mobile konfigurieren.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Allgemein > Erweiterte Einstellungen**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Plug-In-Unterstützung aktivieren	Wenn diese Option aktiviert ist, werden Plug-Ins wie Flash unterstützt.

Last > Knoten "Browsereinstellungen" (Ajax TruClient)

Mit diesem Knoten können Sie die Einstellungen des Ajax TruClient-Browsers für Skripts konfigurieren, die im Lastmodus ausgeführt werden.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Last > Browsereinstellungen**
Wichtige Informationen	• Einstellungen, die Sie in den Browsereinstellungen für den Lastmodus vornehmen, wirken sich nur auf den Lastmodus aus. • Im Knoten **Lastmodus-Browsereinstellungen** stehen dieselben Einstellungen zur Verfügung wie im "Dialogfeld "TruClient - Allgemeine Einstellungen"" auf Seite 609. ▪ Einstellungen, die Sie auf der Registerkarte **Browsereinstellungen** vornehmen, wirken sich nur auf den interaktiven Modus aus. Wenn Sie Ihr Skript im interaktiven Modus speichern, werden alle Einstellungen, die Sie auf der Registerkarte **Browsereinstellungen** geändert haben, auf die Lastmodus-Browsereinstellungen angewendet.

Einstellungen im Firefox-Browser

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	**Beschreibung**
Proxyauswahl	
Kein Proxy (direkte Verbindung mit dem Internet)	Stellen Sie für alle Vuser die Verbindung mit dem Internet direkt her, ohne einen Proxyserver zu verwenden.
Systemproxyeinstellungen verwenden	(Standard) Verwendet die Systemkonfiguration der Proxyeinstellungen.
Manuelle Proxykonfiguration	Ermöglicht Ihnen, die Verbindungseinstellungen für den Proxyserver zu konfigurieren.
Keinen Proxy verwenden für (Ausnahmen)	Sie können Ausnahmen für die Proxyserverregeln festlegen. Die Ausnahmen können Sie als Hostnamen, IP-Adressen und/oder Masken auflisten. **Hinweis:** Platzhalter können Teil des Hostnamens wie in *.devlab.ad sein.
Host für HTTP/alle Protokolle	Der Domänenname oder die IP-Adresse des Proxyservers, die für HTTP-Anfragen und andere Protokolle verwendet werden.

Element der Oberfläche	Beschreibung
Port für HTTP/alle Protokolle	Die Portnummer des Proxyservers, die für HTTP-Anfragen und andere Protokolle verwendet wird.
Gesonderten Proxy für HTTPS-Protokoll verwenden	Ermöglicht es Ihnen, für die Verarbeitung aller HTTPS-Anfragen einen anderen Proxyserver zu verwenden.
Host für HTTPS-Protokoll	Der Domänenname oder die IP-Adresse des Proxyservers, die für HTTPS-Anfragen verwendet werden.
Port für HTTPS-Protokoll	Die Portnummer des Proxyservers, die für HTTPS-Anfragen verwendet wird.
Automatische Proxykonfiguration (PAC)	Die Proxyeinstellungen werden automatisch aus der PAC-Datei (Proxy Auto Configuration) gelesen. Geben Sie im Feld **URL** den URL der PAC-Datei ein.
Erweitert	
Homepage-URL	Geben Sie den URL der Seite ein, die mit der JavaScript-Methode window.home aufgerufen wird. (Standardeinstellung ist about:blank)
Benutzer-Agent	Geben Sie die Benutzer-Agent-Zeichenfolge zum Überschreiben des Browserstandards in Anforderungskopfzeilen ein.
Die Seite im Cache mit der Seite im Netzwerk vergleichen	Geben Sie die zugehörige Nummer ein, auf deren Grundlage der URL im Cache mit dem aus dem Netzwerk aufgerufenen URL verglichen wird. • **0.** (Standardeinstellung) einmal pro Sitzung. • **1.** Bei jedem Zugriff auf die Seite. • **2.** Nie. • **3.** Wenn die Seite veraltet ist (Standardeinstellung).
Keep-Alive	Ermöglicht dauerhafte Netzwerkverbindungen (ohne Proxy). • **Aktiviert.** (Standardeinstellung) Erlaubt dauerhafte Netzwerkverbindungen (ohne Proxy), sodass offene Verbindungen wiederverwendet werden können. • **Gelöscht.** Schließen jeder Verbindung nach Abschluss der Anforderung.

Element der Oberfläche	Beschreibung
Proxy-Keep-Alive	Erlaubt dauerhafte Proxyverbindungen. • **Aktiviert.** (Standardeinstellung) Ermöglicht dauerhafte Proxyverbindungen, sodass offene Verbindungen wiederverwendet werden können. • **Gelöscht.** Schließen jeder Verbindung nach Abschluss der Anforderung.
Keep-Alive-Zeitüberschreitung (s)	Geben Sie den Zeitraum in Sekunden ein, für den inaktive Verbindungen bestehen bleiben. (Standardeinstellung: 300)
HTTP-Version	Geben Sie die HTTP-Version ein, die beim Zugriff auf das Netzwerk/die Anwendung ohne Proxy verwendet werden soll. (Standardeinstellung: 1.1)
Proxy-HTTP-Version	Geben Sie die Proxy-HTTP-Version ein, die beim Zugriff auf das Netzwerk/die Anwendung über einen Proxy verwendet werden soll. (Standardeinstellung: 1.1)
DNS-Cacheeinträge	Geben Sie die maximale Anzahl von Einträgen an, die im DNS-Cache vorgehalten werden sollen. Geben Sie eine 0 ein, um diese Option zu deaktivieren. (Standardeinstellung: 20)
SSL	Wählen Sie die Einstellungen für sichere Verbindungen aus: • SSL 2.0 • (Standardeinstellung: SSL 3.0) • (Standardeinstellung: TLS 1.0)

IE-Browsereinstellungen

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Proxyauswahl	
Kein Proxy (direkte Verbindung mit dem Internet)	Stellen Sie für alle Vuser die Verbindung mit dem Internet direkt her, ohne einen Proxyserver zu verwenden.
Systemproxyeinstellungen verwenden	(Standard) Verwendet die Systemkonfiguration der Proxyeinstellungen.

Element der Oberfläche	Beschreibung
Automatisches Konfigurationsskript verwenden (PAC)	Die Proxyeinstellungen werden automatisch aus der PAC-Datei (Proxy Auto Configuration) gelesen. Geben Sie im Feld **Adresse** den URL der PAC-Datei ein.
Verwendet manuellen Proxyserver für Ihr LAN	**Adresse: Port.** Gültiges Format: 'Adresse:Port' **Proxyserver nicht für Adressen verwenden, die beginnen mit.** Trennen Sie separate Einträge durch Semikolon (;). **Proxyserver bei lokalen Adressen umgehen.** Für Adressen in Ihrem LAN wird kein Proxy verwendet. **Nicht für alle Protokolle denselben Proxyserver verwenden.** Geben Sie für mindestens eines der Protokolle einen Proxy ein.
Erweitert	
Keep-Alive-Zeitüberschreitung (Millisekunden)	Wert der Zeitüberschreitung (in Millisekunden) für offene Verbindungen im Leerlauf (gilt für direkte Verbindungen und Proxyverbindungen)
Serverinfo-Zeitüberschreitung (Millisekunden)	Zeitüberschreitung (in Millisekunden) für vom Server zurückgegebene Daten. Diese Einstellung muss geändert werden, wenn für die Keep-Alive-Zeitüberschreitung mehr als 2 Minuten (120.000 ms) eingestellt wurden.
DNS-Cache-Zeitüberschreitung	Zeitüberschreitung (in Millisekunden), die DNS-Einträge im Cache gespeichert werden.
Startseite	Der URL der Seite, die mit der JavaScript-Methode "window.home" aufgerufen wird. (Standardeinstellung ist about:blank)
Benutzer-Agent	Die Benutzer-Agent-Zeichenfolge zum Überschreiben des Browserstandards in Anforderungskopfzeilen. Lassen Sie das Feld leer, wenn Sie den standardmäßigen Benutzer-Agent von IE verwenden möchten.
HTTP 1.1-Einstellungen	Die Portnummer des Proxyservers, die für HTTP-Anfragen und andere Protokolle verwendet wird. • HTTP 1.1 verwenden • HTTP 1.1 für Proxyverbindungen verwenden

Element der Oberfläche	Beschreibung
SSL	Wählen Sie die Einstellungen für sichere Verbindungen aus: • **SSL 2.0 verwenden.** Aktivieren Sie SSL 2.0 für sichere Verbindungen. • **SSL 3.0 verwenden.** Aktivieren Sie SSL 3.0 für sichere Verbindungen. • **TLS 1.0 verwenden.** Aktivieren Sie TLS 1.0 für sichere Verbindungen. • **TLS 1.1 verwenden.** Aktivieren Sie TLS 1.1 für sichere Verbindungen. • **TLS 1.2 verwenden.** Aktivieren Sie TLS 1.2 für sichere Verbindungen.
Temporäre Internetdateien	Konfiguriert, wann IE die lokale Kopie der Ressource (Cache) mit dem Netzwerk vergleichen soll. Neuere Versionen der gespeicherten Seiten suchen: • **Jedes Mal, wenn ich die Webseite besuche.** Die Ressource wird bei jeder Anfrage überprüft. • **Jedes Mal, wenn ich Internet Explorer starte.** Die Ressource wird beim Starten des Browsers überprüft. • **Automatisch.** • **Nie.**

Last > Sonstige Einstellungen (TruClient)

Dieses Dialogfeld bietet Ihnen die Möglichkeit, die Snapshot-Generierung und Aktionen bei Fehlern für TruClient zu konfigurieren.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Last > Sonstige Einstellungen**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Snapshot-Generierung	
Generierung von Wiedergabe-Snapshots	**Nie.** Bei Auswahl dieser Option werden bei der Wiedergabe nie Snapshots generiert. **Bei Fehler.** Bei Auswahl dieser Option werden Snapshots für Schritte generiert, bei denen während der Wiedergabe Fehler aufgetreten sind. **Immer.** Bei Auswahl dieser Option werden während der Wiedergabe Snapshots für alle Schritte im Skript generiert.
Aktion bei Fehler	**Skript abbrechen.** Bricht das aktuelle Skript bei Auftreten eines Fehlers ab. **Mit der nächsten Iteration fortfahren.** Bricht die aktuelle Iteration ab und fährt mit der nächsten fort.

Allgemein > Knoten "Wiedergeben"

Alle Protokolle - außer Mobile Ajax TruClient

Zugriff	VuGen > Wiedergeben > Laufzeiteinstellungen > Allgemein > Wiedergeben

Element der Oberfläche	Beschreibung
Bei jeder Iteration neuen Benutzer simulieren	Weist VuGen an, alle HTTP-Kontexte zwischen den Iterationen zurückzusetzen. Mit dieser Einstellung kann ein neuer Benutzer, der eine Browsersitzung beginnt, durch den Vuser genauer emuliert werden. **Standardwert:** Aktiviert.

Element der Oberfläche	Beschreibung
Proxyauswahl	Die Einstellungen in diesem Abschnitt gelten für die Aufzeichnung und Ausführung des Skripts. Sie werden nur auf dieses Skript angewendet. Um Proxyeinstellungen für alle Ajax TruClient-Skripts zu konfigurieren, öffnen Sie das Dialogfeld **Browser-Konfigurationseinstellungen**. Weitere Informationen finden Sie unter "Dialogfeld "TruClient - Allgemeine Einstellungen"" auf Seite 609. **Globale Proxyeinstellungen abrufen (siehe Ajax TruClient-Browseroptionen).** Verwendet die im Dialogfeld **Browseroptionen** festgelegten Proxyeinstellungen. Diese Einstellungen werden abgerufen, wenn Sie auf **Skript entwickeln** klicken. Nachdem die globalen Proxyeinstellungen abgerufen wurden, werden sie in die Laufzeiteinstellungen des Skripts übernommen und können im Abschnitt zum Definieren der Proxyeinstellungen verwaltet werden. **Definieren spezifischer Proxyeinstellungen für das Skript.** • KEIN Proxy (direkte Verbindung mit dem Internet). Stellen Sie für alle Vuser die Verbindung mit dem Internet direkt her, ohne einen Proxyserver zu verwenden. • Manuelle Proxykonfiguration. Wenn diese Option ausgewählt ist, können Sie Ausnahmen für die Proxyserverregeln festlegen und den Proxyserver für alle HTTP/HTTPS-Verbindungen angeben. Sie benötigen einen separaten Proxyserver für HTTP-Verbindungen. Aktivieren Sie das Kontrollkästchen **Gesonderten Proxy für HTTPS-Protokoll verwenden** und legen Sie die Einstellungen für den Proxyserver fest. • AUTOMATISCHE Proxykonfiguration (PAC). Wenn diese Option aktiviert wird, werden die Proxyeinstellungen automatisch aus der PAC-Datei (Proxy Auto Configuration) gelesen. Geben Sie in das URL-Feld den URL der PAC-Datei ein. **Hinweis:** Wenn Sie eine PAC-Datei für die Proxykonfiguration verwenden, muss das PAC-Skript 127.0.0.1 statt über einen Proxy an eine direkte Verbindung weiterleiten.

Element der Oberfläche	Beschreibung
Snapshot-Generierung	**Generierung von Aufzeichnungs-Snapshots** • Nie. Wenn Sie diese Option aktivieren, werden Aufzeichnungs-Snapshots des Skripts nicht automatisch gespeichert. • Immer. Wenn Sie diese Option wählen, werden Aufzeichnungs-Snapshots für jeden Schritt des Skripts automatisch gespeichert. **Generierung von Wiedergabe-Snapshots** • Nie. Bei Auswahl dieser Option werden keine Wiedergabe-Snapshots generiert. **Hinweis:** Dies ist die Standardeinstellung. • Bei Fehler. Erstellt einen Snapshot bei Auftreten eines Fehlers. Diese Option kann verwendet werden, um Fehler in Ihrem Skript zu identifizieren. Weitere Informationen finden Sie im Abschnitt zum Debuggen von Skripts mit Snapshots. • Immer. Wenn diese Option gewählt wird, werden Snapshots für jeden Schritt im Skript generiert.
Aktion bei Fehler	**Skript abbrechen** Bricht das aktuelle Skript bei Auftreten eines Fehlers ab. **Mit der nächsten Iteration fortfahren** Bricht bei Auftreten eines Fehlers die aktuelle Iteration ab und fährt mit der nächsten fort.
Wiedergabeoptionen	**Maximale Dauer für 'object-not-found' (Sekunden)** Geben Sie hier die maximale Dauer für die Suche nach einem Objekt ein, bevor die Anwendung eine Fehlermeldung anzeigt. **Intervall zwischen Schritten (Millisekunden)** Geben Sie den minimalen Intervall zwischen den Schritten ein. Einen höheren Wert festzulegen, kann bei Synchronisierungsproblemen hilfreich sein; ein zu hoher Wert kann das Skript jedoch auch unnötig verlangsamen. **Zeitüberschreitung für End-of-network-Identifizierung (Millisekunden)** 'End-of-network' wird für einen Schritt erkannt, wenn der angegebene Zeitwert ohne Netzwerkaktivität verstreicht.

Element der Oberfläche	Beschreibung
Erweitert	Eine Reihe von Optionen für erfahrene Benutzer. Weitere Details erfahren Sie in den Hinweisen, die im Dialogfeld **Laufzeiteinstellungen** unter den Optionen angezeigt werden. • **Homepage-URL.** Geben Sie den URL ein, zu dem navigiert wird, wenn das Startfenster von JavaScript aufgerufen wird. • **Benutzer-Agent.** Geben Sie die Benutzer-Agent-Zeichenfolge zum Überschreiben des Browserstandards in Anforderungskopfzeilen ein. • **Die Seite im Cache mit der Seite im Netzwerk vergleichen.** Wählen Sie die zugehörige Nummer aus, auf deren Grundlage der URL im Cache mit dem aufgerufenen URL verglichen wird. 0=Einmal pro Sitzung, 1=Bei jedem Aufruf der Seite, 2=Nie, 3=Wenn die Seite nicht mehr aktuell ist (Standard). • **Keep-Alive.** Ermöglicht dauerhafte Netzwerkverbindungen (ohne Proxy). Geben Sie die entsprechende Zahl ein: 1=Ermöglicht dauerhafte Netzwerkverbindungen (ohne Proxy), sodass offene Verbindungen wiederverwendet werden können, 0=Schließen jeder Verbindung nach Abschluss der Anforderung. • **Proxy Keep-Alive.** Ermöglicht dauerhafte Netzwerkverbindungen. Geben Sie die entsprechende Zahl ein: 1=Ermöglicht dauerhafte Proxyverbindungen, sodass offene Verbindungen wiederverwendet werden können. 0=Schließen jeder Verbindung nach Abschluss der Anforderung. • **Keep-Alive-Zeitüberschreitung (s).** Geben Sie den Zeitraum in Sekunden ein, für den inaktive Verbindungen bestehen bleiben. • **HTTP-Version.** Geben Sie die HTTP-Version ein, die beim Zugriff auf das Netzwerk/die Anwendung ohne Proxy verwendet werden soll. • **Proxy-HTTP-Version.** Geben Sie die Proxy-HTTP-Version ein, die beim Zugriff auf das Netzwerk/die Anwendung über einen Proxy verwendet werden soll. • **DNS-Cacheeinträge.** Geben Sie die maximale Anzahl von Einträgen an, die im DNS-Cache vorgehalten werden sollen. **Hinweis:** Geben Sie eine 0 ein, um diese Option zu deaktivieren. • **Nicht-interaktive Fenstergröße.** Geben Sie die Anfangsgröße des

Element der Oberfläche	Beschreibung
	Browserfensters (Breite und Höhe in Pixel) im nicht-interaktiven Modus ein. • **SSL.** Wählen Sie die Einstellungen für sichere Verbindungen aus: ▪ SSL 2.0 ▪ SSL 3.0 ▪ TLS 1.0

Mobile Ajax TruClient

Element der Oberfläche	Beschreibung
Bei jeder Iteration neuen Benutzer simulieren	Weist VuGen an, alle HTTP-Kontexte zwischen den Iterationen zurückzusetzen. Mit dieser Einstellung kann ein neuer Benutzer, der eine Browsersitzung beginnt, durch den Vuser genauer emuliert werden. **Standardwert:** Aktiviert.
Wiedergabeoptionen	
Wiedergabe unter Verwendung der aufgezeichneten Dauer für Schritte	Während der Wiedergabe wird die Mindestzeit der Ausführung jedes Schritts aufgezeichnet. **Hinweis:** Nach der Bearbeitung des Skripts kann dies zu einer falschen Zeitsteuerung der Schritte im Skript führen.
Fehlgeschlagenes Ereignis verursacht Schrittfehler	Wenn das **Endereignis** eines Schritts fehlschlägt, schlägt der Schritt fehl. Wenn das **Endereignis** eines Schritts beispielsweise **Dokument geladen** ist und das Dokument nicht geladen werden kann, schlägt der Schritt fehl. Diese Option ist standardmäßig aktiviert.

Element der Oberfläche	Beschreibung
Profilerstellung aktivieren	Profilerstellung für clientseitige Aufschlüsselung aktivieren/deaktivieren. **Maximale Dauer für 'object-not-found' (Sekunden)** Geben Sie hier die maximale Dauer für die Suche nach einem Objekt ein, bevor die Anwendung eine Fehlermeldung anzeigt. **Schrittzeitüberschreitung** Der Zeitraum für eine Schrittausführung. Wenn das Endereignis in dieser Zeit nicht eintritt, schlägt der Schritt fehl. **Intervall zwischen Schritten (Millisekunden)** Geben Sie den minimalen Intervall zwischen den Schritten ein. Einen höheren Wert festzulegen, kann bei Synchronisierungsproblemen hilfreich sein; ein zu hoher Wert kann das Skript jedoch auch unnötig verlangsamen. **Zeitüberschreitung für End-of-network-Identifizierung (Millisekunden)** 'End-of-network' wird für einen Schritt erkannt, wenn der angegebene Zeitwert ohne Netzwerkaktivität verstreicht. **Eingabeintervall (Millisekunden)** Die durchschnittliche Zeit zwischen den Tastenanschlägen bei der Eingabe von Text während der Wiedergabe.
Mindestzeit	**Mindestzeit eines zufälligen Schritts innerhalb des Bereichs.** Während der Wiedergabe werden die Einstellungen für die Mindestzeit jedes Schritts durch einen Zufallswert überschrieben. **Von (Prozentsatz).** Legt den unteren Prozentsatz fest. **Bis (Prozentsatz).** Legt den oberen Prozentsatz fest. **Mindestzeit des Schritts begrenzen.** Legt die maximale Mindestzeit des Schritts in Sekunden fest. **Mindestzeit-Einstellung für Warteschritte übernehmen.** Aktivieren/Deaktivieren des Übernehmens der Mindestzeit-Einstellung für Warteschritte.

Element der Oberfläche	Beschreibung
Automatische Transaktionen	**Keine automatischen Transaktionen.** Keine automatischen Transaktionen anwenden. **Jede Aktion als Transaktion definieren.** Behandelt zur Leistungsmessung jede Aktion in einem Skript als Transaktion. **Jede Aktion und jeden Schritt als Transaktion definieren.** Behandelt zur Leistungsmessung jede Aktion und jeden Skript als Transaktion.

Allgemein > Verschiedenes (Knoten)

Hier können Sie verschiedene Laufzeiteinstellungen konfigurieren.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Allgemein > Verschiedenes**
Wichtige Informationen	• Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334. • Es ist nicht empfehlenswert, sowohl die Option **Bei Fehler fortfahren** als auch **Bei Fehler Snapshot erzeugen** in einer Lasttestumgebung zu aktivieren. Diese Konfiguration kann die Leistungsfähigkeit der Vuser beeinträchtigen. • Wenn die Vuser während der Szenarioausführung Diagnosedaten (J2EE) erzeugen sollen, sollten keine automatischen Transaktionen verwendet werden. Definieren Sie den Anfang und das Ende der einzelnen Transaktionen stattdessen manuell.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Automatische Transaktionen	• **Jede Aktion als Transaktion definieren.** Weist LoadRunner (gilt nicht für HP Business Service Management) an, jede Aktion wie eine Transaktion zu behandeln. • **Jeden Schritt als Transaktion definieren.** Weist LoadRunner (gilt nicht für HP Business Service Management) an, jeden Schritt wie eine Transaktion zu behandeln.

Fehlerbehandlung	• **Bei Fehler fortfahren.** Fährt beim Auftreten eines Fehlers mit der Skriptausführung fort. **Standardwert:** Deaktiviert. • **Bei lr_error_message Fehler für offene Transaktionen ausgeben.** Weist VuGen an, alle Transaktionen, in denen die Funktion **lr_error_message** ausgegeben wurde, als *Failed* zu kennzeichnen. Die Funktion **lr_error_message** wird durch eine manuell definierte *If*-Anweisung ausgegeben. • **Bei Fehler Snapshot erzeugen**. Generiert einen Snapshot bei Auftreten eines Fehlers. Sie können den Snapshot einsehen, indem Sie das Vuser-Protokoll anzeigen und auf die Zeile doppelklicken, in der der Fehler aufgetreten ist.
Multithreading	• **Vuser als Prozess ausführen.** Deaktiviert das Multithreading. • **Vuser als Thread ausführen.** Aktiviert das Multithreading. Weitere Informationen finden Sie unter "Multithreading" auf Seite 333.

Allgemein > Takt (Knoten)

Ermöglicht Ihnen, die Zeit zwischen den Iterationen zu steuern. Der Takt teilt dem Vuser mit, wie lange er zwischen den Aktionsiterationen warten muss.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Allgemein > Takt**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Nach Beendigung der vorherigen Iteration	Startet die neue Iteration im angegebenen zeitlichen Abstand nach Beendigung der vorherigen Iteration. Geben Sie entweder die genaue Anzahl von Sekunden oder eine Zeitspanne an.
Nach Beendigung der vorherigen Iteration	Die neue Iteration beginnt so schnell wie möglich nach dem Ende der vorangegangenen Iteration.
Mit <festen/zufälligen> Intervallen	Sie legen die Zeit zwischen den Iterationen fest: entweder eine feste Anzahl von Sekunden oder einen Zeitbereich in Sekunden ab dem Start der vorangegangenen Iteration. Jede geplante Iteration wird nur gestartet, wenn die vorherige Iteration abgeschlossen wurde.

Allgemein > Ausführungslogik (Knoten)

Ermöglicht Ihnen das Konfigurieren der Laufzeiteinstellungen für die Ausführungslogik.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Allgemein > Ausführungslogik**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.
Siehe auch	"Abschnitte des Vuser-Skripts" auf Seite 260

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
Aktion einfügen	Fügt eine neue Aktion am Einfügungspunkt hinzu.
Block einfügen	Fügt einen neuen Aktionsblock am Einfügungspunkt hinzu.
Löschen	Löscht ein Element.
Nach oben	Verschiebt ein Element nach oben.
Nach unten	Verschiebt ein Element nach unten.
Eigenschaften	Öffnet das Dialogfeld **Eigenschaften**, wo Sie die Einstellungen für die Ausführungslogik und die Iterationen vornehmen können. **Ausführungslogik**. Aktionen werden entweder sequentiell oder zufällig ausgeführt. **Iterationen**. Legt fest, wie oft ein Element ausgeführt wird.
<Ausführungslogikstruktur>	Eine grafische Darstellung der Ausführungslogik für dieses Skript.
Anzahl der Iterationen	Gibt an, wie oft das Skript die Elemente in der Ausführungslogikstruktur durchläuft.

Allgemein > Latenzzeit (Knoten)

Hier können Sie die Einstellungen für die Latenzzeit konfigurieren, also festlegen, wie lange VuGen zwischen den einzelnen Aktionen wartet. Mit diesen Einstellungen können Sie das Verhalten von realen Benutzern simulieren.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Allgemein > Latenzzeit**
Siehe auch	• VuGen zeichnet die Latenzzeitwerte mithilfe der Funktionen **lr_think_time** in den Vuser-Skripts auf. Weitere Informationen zur Funktion **lr_think_time** und deren manueller Bearbeitung finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Wie aufgezeichnet	Das in der Funktion **lr_think_time** vorhandene Argument wird bei der Wiedergabe verwendet. Die Funktion `lr_think_time(10)` bedingt beispielsweise eine Latenzzeit von zehn Sekunden.
Latenzzeit ignorieren	Die aufgezeichnete Latenzzeit wird nicht berücksichtigt. Das Skript wird wiedergegeben, ohne die **lr_think_time**-Funktionen zu verwenden.
Latenzzeit begrenzen auf	Hier können Sie den Höchstwert für die Latenzzeit festlegen.
Aufgezeichnete Latenzzeit multiplizieren mit	Bei der Wiedergabe wird ein Mehrfaches der aufgezeichneten Latenzzeit verwendet. Dadurch kann die während der Wiedergabe angewendete Latenzzeit verlängert oder verkürzt werden. Wurde beispielsweise eine Latenzzeit von vier Sekunden aufgezeichnet, können Sie den Vuser anweisen, diesen Wert mit zwei zu multiplizieren und kämen so auf eine Gesamtzeit von acht Sekunden. Um die Latenzzeit auf zwei Sekunden zu reduzieren, multiplizieren Sie die Aufzeichnungszeit mit 0,5.
Latenzzeit wiedergeben	Aktiviert Optionen, mit denen Sie die aufgezeichnete Latenzzeit anpassen können.
Zufälligen Prozentsatz der Latenzzeit verwenden	Zufälligen Prozentsatz der Latenzzeit verwenden. Dazu legen Sie einen Wertebereich für die Latenzzeit fest. Weist das Latenzzeitargument beispielsweise den Wert 4 auf und Sie geben als Mindestsatz 50% und als Höchstsatz 150% an, beträgt die kürzeste Latenzzeit zwei (50%) und die längste sechs (150%).

Internetprotokoll > ContentCheck (Knoten)

Ermöglicht Ihnen, Websites zur Laufzeit auf Inhalte zu überprüfen.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Internetprotokoll > ContentCheck**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
Löschen	Löscht die ausgewählte Regel oder Anwendung.
Exportieren...	Exportiert Regeln in eine xml-Datei.
Importieren...	Importiert Regeln aus einer vorhandenen xml-Datei.
Neue Anwendung	Fügt eine neue Anwendung in der Anwendungs- und Regelliste hinzu. Klicken Sie, um den Namen zu ändern.
Neue Regel	Die Regelkriterien werden im rechten Bereich angezeigt, sodass Sie eine neue Regel für die derzeit ausgewählte Anwendung eingeben können. • **Nach folgendem Text suchen.** Der Text der Zeichenfolge, die Sie suchen. • **Suche nach Präfix und Suffix.** Das Präfix und Suffix der Zeichenfolge, die Sie suchen. • **Groß-/Kleinschreibung.** Durchführen einer Suche, bei der die Groß-/Kleinschreibung beachtet wird. • **Fehlschlag wenn.** Hier können Sie einstellen, ob die Suche als fehlgeschlagen gilt, wenn die gesuchte Zeichenfolge gefunden oder nicht gefunden wurde. • **Text in JavaScript-Benachrichtigungsfeldern durchsuchen.** Hierbei wird nur Text in JavaScript-Benachrichtigungsfeldern durchsucht (gilt nur für Web- und Oracle Web Applications 11i-Vuser).
Als Standard...	Übernimmt die getroffenen ContentCheck-Einstellung als Standard (d. h. neue Skripts starten mit dieser Konfiguration).
<Anwendungs- und Regelliste>	Die Liste der Anwendung und der zugehörigen Regeln. Sie können einzelne Elemente in der Liste über die Kontrollkästchen links daneben aktivieren bzw. deaktivieren.
ContentCheck bei der Wiedergabe aktivieren	Aktiviert die Inhaltsprüfung während der Wiedergabe. Selbst wenn Sie bereits Anwendungen definiert haben, können Sie diese Option für einen bestimmten Testlauf deaktivieren. **Standardwert:** Aktiviert.

Internetprotokoll > Download-Filter (Knoten)

Hier können Sie die Download-Filter einstellen.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Internetprotokoll > Download-Filter**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Filterliste>	Eine Liste von Filtern für das Skript. Jeder Filter besitzt einen Typ und Daten. Dem Filtertyp URL ist z. B. eine URL-Adresse zugeordnet. Die Optionen für Einträge in der Liste lauten: **Hinzufügen**, **Bearbeiten**, **Entfernen** oder **Alle entfernen**.
In der Liste enthaltene Adressen ausschließen	Ignoriert Anforderungen von den Websites oder Hosts in der Liste.
Nur in der Liste enthaltene Adressen berücksichtigen	Beschränkt die Wiedergabe auf die aufgelisteten Websites oder Hosts.

Internetprotokoll > Einstellungen (Knoten)

Ermöglicht Ihnen, die Laufzeiteinstellungen in Zusammenhang mit dem Internet festzulegen.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Internetprotokoll > Einstellungen**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Prüfungen	• **Bild- und Textprüfung aktivieren.** Ermöglicht dem Vuser, während der Wiedergabe mithilfe der Funktionen **web_find** oder **web_image_check** Prüfungen auszuführen. Diese Option gilt nur für Anweisungen, die im HTML-basierten Modus aufgezeichnet wurden. Vuser, die mit Verifizierungsprüfungen ausgeführt werden, erfordern mehr Speicher als Vuser ohne solche Prüfungen. **Standardwert:** Deaktiviert.
Diagramme zur Webleistung erzeugen	Durch die Option **Diagramme zur Webleistung erzeugen** erfasst ein Vuser die Daten zum Erstellen von Diagrammen zur Webleistung. Sie können die Diagramme **Treffer pro Sekunde**, **Seiten pro Sekunde** und **Antwort-Bytes pro Sekunde (Durchsatz)** während des Testlaufs über die Onlinemonitore oder nach dem Testlauf mithilfe von Analysis anzeigen. Das Aufschlüsselungsdiagramm zeigen Sie nach dem Testlauf in Analysis an. Wählen Sie die Diagrammdaten aus, die der Vuser erfassen soll. **Hinweis:** Wenn Sie die Diagramme zur Webleistung nicht verwenden, deaktivieren Sie diese Optionen, um Speicher zu sparen.

Element der Oberfläche	Beschreibung
Erweitert	• **WinInet-Wiederholung anstelle von Sockets.** Durch diese Option verwendet VuGen zur Wiedergabe die WinInet-Engine anstelle der standardmäßigen Sockets-Engine. VuGen verfügt über zwei HTTP-Wiedergabeengines: Sockets-basiert (Standard) oder WinInet-basiert. Die WinInet-Engine, die der Internet Explorer verwendet, unterstützt alle Funktionen dieses Browsers. Nachteile der WinInet-Wiedergabe-Engine sind fehlende Skalierbarkeit und Linux-Unterstützung. Außerdem können mit der WinInet-Engine beim Einsatz von Threads die Modemgeschwindigkeit und Verbindungsanzahl nicht genau emuliert werden. Für die proprietäre, sockets-basierte Wiedergabe wird in VuGen eine einfachere Engine verwendet, die sich für Lasttests skalieren lässt. Diese liefert auch beim Einsatz von Threads präzise Ergebnisse. Die Sockets-basierte Engine unterstützt jedoch keinen SOCKS-Proxy. Zum Aufzeichnen in einer solchen Umgebung müssen Sie die WinInet-Wiedergabeengine verwenden. **Standardwert:** Deaktiviert (socket-basierte Wiedergabe-Engine). • **Datei und Zeile in automatischen Transaktionsnamen.** Diese Option erstellt eindeutige Transaktionsnamen für automatische Transaktionen, indem Dateiname und Zeilennummer zum Transaktionsnamen hinzugefügt werden. **Standardwert:** Aktiviert. • **Unkritische Ressourcenfehler als Warnungen.** Diese Option gibt einen Warnstatus für Funktionen aus, die bei einem nicht-kritischen Element während eines Lasttests fehlschlagen, z. B. wenn ein Bild oder Java-Applet nicht heruntergeladen werden kann. Diese Option ist standardmäßig aktiviert. Wenn eine bestimmte Warnung als Fehler angesehen werden soll, durch den der gesamte Test fehlschlägt, können Sie diese Option deaktivieren. Sie können einen Inhaltstyp als kritisch festlegen, indem Sie ihn zur Liste der Nicht-Ressourcen hinzufügen. Weitere Informationen finden Sie unter "Dialogfeld "Nicht-Ressourcen"" auf Seite 208. • **Snapshot-Ressourcen lokal speichern.** Durch diese Option speichert VuGen die Snapshotressourcen in Dateien auf dem lokalen Computer. • Optionen... Öffnet das "Dialogfeld "Erweiterte Einstellungen"" unten.

Dialogfeld "Erweiterte Einstellungen"

Ermöglicht Ihnen das Konfigurieren der Laufzeiteinstellungen für erweiterte Internet-Optionen.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Internetprotokoll > Einstellungen > Optionen**
Wichtige Informationen	• Dieses Dialogfeld steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334. • In diesem Dialogfeld werden die Eigenschaften in unterschiedliche Kategorien eingeteilt: **HTTP**, **Allgemein**, **Authentifizierung**, **Protokoll** und **Speziell für Web (Click & Script)**.

HTTP

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
HTTP-Version	Die zu verwendende HTTP-Version: Version 1.0 oder 1.1. Diese Information ist immer in der HTTP-Anfragekopfzeile enthalten, wenn ein Vuser eine Anfrage an einen Webserver sendet. **Standardwert:** HTTP 1.1. HTTP 1.1 unterstützt die folgenden Funktionen: • Dauerhafte Verbindungen – siehe Option "Keep-Alive-HTTP-Verbindungen" weiter unten. • HTML-Komprimierung, siehe "Serverseitige Komprimierung akzeptieren" weiter unten. • Virtuelles Hosting – mehrere Domänennamen nutzen gemeinsam dieselbe IP-Adresse.
Keep-Alive-HTTP-Verbindungen	Der Begriff "Keep-Alive" bezeichnet eine HTTP-Erweiterung, die dauerhafte bzw. kontinuierliche Verbindungen ermöglicht. Durch diese dauerhaften HTTP-Sitzungen können mehrere Anfragen über dieselbe TCP-Verbindung gesendet werden. So steigt die Leistung des Webservers und der Webclients. Die Keep-Alive-Option funktioniert nur bei Webservern, die Keep-Alive-Verbindungen unterstützen. Durch diese Einstellung werden Keep-Alive-HTTP-Verbindungen für alle Vuser aktiviert, die das Vuser-Skript ausführen. **Standardwert:** Aktiviert.
Accept-Language-Anforderungskopfzeile	Stellt eine durch Kommata getrennte Liste der akzeptierten Sprachen bereit. Beispiele: **en-us**, **fr** usw. Weitere Informationen finden Sie unter "Sprache in der Anforderungskopfzeile" auf Seite 1101.

Element der Oberfläche	Beschreibung
HTTP-Fehler als Warnungen	Gibt eine Warnung anstatt eines Fehlers aus, wenn das Herunterladen von Ressourcen wegen eines HTTP-Fehlers fehlschlägt.
Verbindungszeitüberschreitung der HTTP-Anforderung (s)	Die Zeitdauer in Sekunden, während der ein Vuser auf die Verbindung für eine bestimmte HTTP-Anfrage innerhalb eines Schrittes wartet, bevor der Vorgang abgebrochen wird. Durch Zeitüberschreitungen kann der Server stabilisiert werden und dem Benutzer antworten. Diese Zeitüberschreitung gilt auch für die Zeitdauer, während der ein Vuser auf eine WAP-Verbindung wartet, die von der Funktion **wap_connect** eingeleitet wird. **Maximaler Wert:** 32000 Sekunden.
Empfangszeitüberschreitung der HTTP-Anforderung (s)	Die Zeitdauer in Sekunden, während der ein Vuser auf den Empfang der Antwort für eine bestimmte HTTP-Anfrage innerhalb eines Schrittes wartet, bevor der Vorgang abgebrochen wird. Durch Zeitüberschreitungen kann der Server stabilisiert werden und dem Benutzer antworten. **Maximaler Wert:** 32000 Sekunden.
Zlib-Kopfzeile der Anforderung	Sendet Anfragedaten an den Server mit der Kopfzeile **zlib** für Komprimierungsbibliotheken. Anfragen an den Server werden standardmäßig mit der Kopfzeile **zlib** gesendet. Durch diese Option können Sie Anwendungen ohne Browser emulieren, deren Anfragen keine Kopfzeile **zlib** enthalten. Zum Ausschließen dieser Kopfzeilen setzen Sie diese Option auf **Nein**. **Standardwert:** Ja.

Element der Oberfläche	Beschreibung
Serverseitige Komprimierung akzeptieren	Diese Einstellung zeigt dem Server an, dass bei der Wiedergabe komprimierte Daten akzeptiert werden. Die verfügbaren Optionen sind: **None** (keine Komprimierung), **gzip** (GZip-Komprimierung akzeptieren), **gzip, deflate** (GZip- oder Deflate-Komprimierung akzeptieren) und **deflate** (Deflate-Komprimierung akzeptieren). Durch das Akzeptieren komprimierter Daten kann die CPU-Auslastung erheblich ansteigen. Standardwert: **gzip**- und **deflate**-Komprimierung akzeptieren. Zum manuellen Hinzufügen von Komprimierung geben Sie am Skriptanfang die folgende Funktion ein: `web_add_auto_header("Accept-Encoding", "gzip");` Um zu überprüfen, ob der Server komprimierte Daten gesendet hat, suchen Sie nach der Zeichenfolge "Content - Encoding: gzip" im Abschnitt mit den Antworten des Servers im Wiedergabeprotokoll. Das Protokoll enthält auch die Datengröße vor und nach der Komprimierung.
Nicht referenzierte Cacheeinträge löschen	Löschen Sie die Cacheeinträge, die innerhalb der angegebenen Anzahl Iterationen nicht referenziert wurden. Setzen Sie diese Option auf null (0), um Cacheeinträge nie zu löschen.

Allgemein

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
DNS-Zwischenspeicherung	Sorgt dafür, dass der Vuser die IP-Adresse eines Hosts zwischenspeichert, nachdem der Wert vom DNS-Server aufgelöst wurde. Dies spart Zeit bei späteren Aufrufen an denselben Server. Wenn sich die IP-Adresse ändert, z. B. bei bestimmten Lastausgleichsverfahren, müssen Sie diese Option deaktivieren, damit der Vuser nicht den zwischengespeicherten Wert verwendet. **Standardwert:** Aktiviert.
Aus/in UTF-8 konvertieren	Konvertiert empfangene HTML-Seiten und gesendete Daten in und aus UTF-8. Die UTF-8-Unterstützung aktivieren Sie in den Aufzeichnungsoptionen. Weitere Informationen finden Sie unter "Aufzeichnungsoptionen" auf Seite 164. **Standardwert:** Nein.

Element der Oberfläche	Beschreibung
Durch Ressourcen verursachte Zeitüberschreitung für einen Schritt entspricht einer Warnung	Gibt eine Warnung anstatt eines Fehlers aus, wenn eine Zeitüberschreitung auftritt, weil eine Ressource nicht innerhalb des Zeitüberschreitungsintervalls geladen wurde. Für Nicht-Ressourcen gibt VuGen einen Fehler aus. **Standardwert:** Deaktiviert.
HTML-Content-Type analysieren	Bei Erwartung von HTML wird die Antwort nur analysiert, wenn es sich um den angegebenen Inhaltstyp handelt: **HTML**, **text\html**, **TEXT** (jeder Text) oder **ANY** (jeder Inhaltstyp). Die Option **text/xml** wird nicht als HTML analysiert. **Standardwert: TEXT**. Die Zeitüberschreitungseinstellungen richten sich vor allem an fortgeschrittene Benutzer, die feststellen, dass ihre Umgebung unterschiedliche akzeptable Zeitüberschreitungswerte erfordert. In den meisten Fällen genügen die Standardeinstellungen. Reagiert der Server nicht innerhalb einer akzeptablen Zeitspanne, suchen Sie nach anderen Verbindungsproblemen, anstatt eine sehr lange Zeitüberschreitung festzulegen, durch die unnötige Wartezeiten für das Skript entstehen.
Zeitüberschreitung beim Schrittdownload (s)	Die Wartezeit des Vusers, bevor ein Schritt im Skript abgebrochen wird. Durch diese Option kann das Verhalten eines Benutzers emuliert werden, der nicht länger als x Sekunden auf eine Seite wartet. **Maximaler Wert:** 32000 Sekunden.
Netzwerkpuffergröße	Legt die maximale Puffergröße zum Empfangen der HTTP-Antwort fest. Wenn der Datenumfang die angegebene Größe überschreitet, sendet der Server die Daten stückweise, wodurch die Systemlast steigt. Beim Ausführen mehrerer Vuser über den Controller verwendet jeder Vuser seinen eigenen Netzwerkpuffer. Diese Einstellung richtet sich vor allem an fortgeschrittene Benutzer, die feststellen, ob die Netzwerkpuffergröße möglicherweise die Skriptleistung beeinträchtigt. Die Standardgröße beträgt 12 KB. Die maximale Größe ist 0x7FFF FFFF.
NTLM-Informationen drucken	Druckt Informationen zum NTLM-Handshake in das Standardprotokoll.
SSL-Informationen drucken	Druckt Informationen zum SSL-Handshake in das Standardprotokoll.

Element der Oberfläche	Beschreibung
Maximale Anzahl von Fehlerübereinstimmungen, die als FEHLER ausgegeben werden	Beschränkt die Anzahl der Fehlerübereinstimmungen, für die bei Inhaltsprüfungen ein Fehler ausgegeben wird, mithilfe einer linken oder rechten Begrenzung. Dies gilt für Übereinstimmungen, die zu einem Fehler führen, wenn die Zeichenfolge gefunden wird (Fail=Found). Alle nachfolgenden Übereinstimmungen werden als Informationsmeldungen aufgelistet. **Standardwert:** 10 Übereinstimmungen.
Maximale Anzahl von 'META refresh' auf der gleichen Seite	Die maximal zulässige Anzahl an META-Aktualisierungen für eine Seite. **Standardwert:** 2.
ContentCheck-Werte in UTF-8	Speichert die Werte in der ContentCheck-XML-Datei in UTF-8.
Grenzwert für Tree-View-Anforderungstext	Begrenzen Sie die Byte-Anzahl für den Anforderungstext in der Strukturansicht. Setzen Sie die Option auf null (0), damit kein Grenzwert gilt.
IP-Version	Zu verwendende IP-Version: IPv4, IPv6 oder automatische Auswahl. Der Standardwert ist IPv4.
web_sync Wiederholungsintervall (Millisekunden)	Die Wartezeit (in Millisekunden) zwischen dem Testen der Bedingung, die "false" ergibt, und der nächsten Wiederholung. Der Standardwert ist **1000**.
web_sync Zeitüberschreitung für Wiederholung (Millisekunden)	Die maximale Zeit (in Millisekunden), während der Wiederholungen zulässig sind. Wenn die berechnete Zeitüberschreitung die Zeitüberschreitung für einen Schritt (wie in der Einstellung "Zeitüberschreitung beim Herunterladen für einen Schritt" festgelegt) überschreitet, wird Letztere verwendet.

Authentifizierung

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Feste Latenzzeit nach erneutem Authentifizierungsversuch (ms)	Fügt automatisch eine Latenzzeit zum Vuser-Skript hinzu, um einen Benutzer zu emulieren, der Authentifizierungsinformationen eingibt (Benutzername und Passwort). Diese Latenzzeit wird in die Transaktionszeit einbezogen. **Standardwert:** 0.

Element der Oberfläche	Beschreibung
NTLM2-Sitzungssicherheit deaktivieren	Verwendung eines NTLM 2-Handshakes mit umfassender Sicherheit anstelle der vereinfachten Sicherheitsantwort für NTLM 2-Sitzungen. **Standardwert:** Nein.
Systemeigene Windows-NTLM-Implementierung verwenden	Verwendung der Microsoft Security-API für NTLM-Authentifizierung anstelle der nativen API. **Standardwert:** Nein.
Anmeldeinformationen in der systemeigenen Windows-NTLM-Implementierung außer Kraft setzen	Verwenden Sie die vom Benutzer bei der Anmeldung bereitgestellten Anmeldeinformationen.
Integrierte Authentifizierung aktivieren	Ermöglicht Kerberos-basierte Authentifizierung. Wenn der Server Authentifizierungsschemata vorschlägt, wird bevorzugt **Negotiate** verwendet. **Standardwert:** Nein.
Hohe KDC-Last erzeugen	Anmeldeinformationen aus früheren Iterationen werden nicht wiederverwendet. Durch das Aktivieren dieser Einstellung steigt die Last auf dem KDC-Server (Key Distribution Center). Um die Last auf dem Server zu senken, setzen Sie diese Option auf **No**, damit die Anmeldeinformationen aus früheren Iterationen wiederverwendet werden. Diese Option ist nur relevant, wenn Kerberos-Authentifizierung verwendet wird. **Standardwert:** Nein.
Kanonischen Namen im SPN verwenden	Verwenden Sie den kanonischen Namen anstelle des ursprünglichen Hostnamens, der von dem URL abgerufen wurde, um den Dienstprinzipalnamen (Service Principal Name, SPN) zu generieren. **Standardwert:** Ja.
Nicht-Standard-Port an SPN anfügen	Fügen Sie die Portnummer an den SPN an, wenn der angegebene Port kein Standardport ist (weder 80 noch 443). **Standardwert:** Nein.
Schlüsselabruf aus ncipher-HSM aktivieren	Ermöglicht LoadRunner das Abrufen privater Schlüssel aus dem HSM (Hardwaresicherheitsmodul), z. B. nCipher. **Standardwert:** Ja.

Protokoll

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Länge der Druckpufferzeile	Zeilenlänge zum Drucken der Quelle von Kopfzeilen/Text für Anfragen/Antworten oder JavaScript. Umbrüche sind dabei deaktiviert.
Beim Druckpuffer nur binäre Nullen mit Escape-Zeichen versehen	• **Yes.** Beim Drucken der Quelle von Kopfzeilen/Text für Anfragen/Antworten oder JavaScript nur binäre Nullen maskieren. • **No**. Alle nicht druckbaren Zeichen bzw. Steuerzeichen maskieren.

Speziell für Web (Click & Script) (Wiedergabe nur für Version 12 und höher)

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Allgemein	• **Homepage-URL.** Der URL der Startseite, die im Browser geöffnet wird (die Standardeinstellung lautet **about:blank**). • **DOM-basierte Snapshots.** VuGen generiert Snapshots aus dem DOM anstatt aus den Serverantworten. **Standardwert:** Ja. • **Zeichensatzkonvertierungen durch HTTP.** Führt Zeichenkonvertierungen anhand der HTTP-Anfragekopfzeile "Content-Type:....; charset=..." durch. Überschreibt die Option "Aus/in UTF-8 konvertieren". • **Bei META-Änderung des Zeichensatzes erneut analysieren.** Erneute HTML-Analyse, wenn ein META-Tag den Zeichensatz ändert. Gilt nur, wenn die Option **Charset conversions by HTTP** aktiviert ist. Durch **Auto** ist die erneute Analyse nur aktiviert, wenn sie in der ersten Iteration verwendet wird. • **Fehler bei JavaScript-Fehler.** Die Vuser-Ausführung schlägt fehl, wenn ein JavaScript-Auswertungsfehler auftritt. **Standardwert: Nein** (Es wird nach einem JavaScript-Fehler nur eine Warnmeldung ausgegeben, aber die Skriptausführung wird fortgesetzt). • **Standardklassen für jedes neue Fensterobjekt initialisieren.** Ist diese Option aktiviert, wird das SRC-kompilierte Skript nicht zwischengespeichert. • **Deaktivierung des beeinflussten Elements ignorieren.** Durch diese Option wird ignoriert, dass das Element, auf das sich eine Vuser-Skriptfunktion bezieht, deaktiviert ist.

Element der Oberfläche	Beschreibung
Timer	• **Timer am Ende des Schritts optimieren.** Falls möglich, wird eine Methode **setTimeout/setInterval/<META refresh>** ausgeführt, die am Ende des Schrittes vor der Ablaufzeit abläuft. **Standardwert:** Ja. • **Einfacher setTimeout/setInterval-Schwellenwert (Sekunden).** Festlegen einer oberen Zeitüberschreitung für die Methoden **window.setTimeout** und **window.setInterval**. Ist die Verzögerung größer als diese Zeitüberschreitung, rufen diese Methoden nicht die an sie übergebenen Funktionen auf. Dadurch wird ein Benutzer emuliert, der eine bestimmte Zeit wartet, bevor er auf das nächste Element klickt. **Standardwert:**5 Sekunden. • **Akkumulierter setTimeout/setInterval-Schwellenwert (Sekunden).** Festlegen einer Zeitüberschreitung für die Methoden **window.setTimeout** und **window.setInterval**. Ist die Verzögerung größer als diese Zeitüberschreitung, werden weitere Aufrufe an **window.setTimeout** und **window.setInterval** ignoriert. Die Zeitüberschreitung wird pro Schritt kumuliert. **Standardwert:**30 Sekunden. • **setInterval am Ende des Schritts wieder einrichten.0** = Nein; **1** = Einmal; **2** = Ja. • **Timer für kein Netzwerk am Ende des Schritts begrenzen:** Begrenzen Sie die Anzahl der mit "setTimeout/setInterval" festgelegten Skriptbewertungen am Ende eines Schrittes, wenn keine Netzwerkanforderungen ausgegeben werden. Setzen Sie die Option auf null (0), damit kein Grenzwert gilt. Der Standardwert ist 100. Dieser Grenzwert wird nur verwendet, wenn "Optimiert den Timer am Ende des Schritts" aktiviert ist.

Element der Oberfläche	Beschreibung
Historie	• **Historienunterstützung.** Ermöglicht Unterstützung für das Objekt **window.history** beim Testlauf. Die Optionen lauten **Aktiviert**, **Deaktiviert** und **Auto**. Durch die Option **Auto** (Standardeinstellung) unterstützen Vuser das Objekt **window.history** nur, wenn es in der ersten Iteration verwendet wurde. Durch das Deaktivieren dieser Option steigt die Leistung. **Standardwert:** Auto. • **Maximale Historiengröße.** Die maximale Anzahl an Schritten in der Historienliste. **Standardwert:** 100 Schritte.
Navigatoreigenschaften	• **navigator.browserLanguage.** Die Browsersprache, die in der Eigenschaft **browserLanguage** des Navigator-DOM-Objekts festgelegt ist. **Standardwert:** Der aufgezeichnete Wert. In Skripts, die mit älteren Aufzeichnungsengines erstellt wurden, wird standardmäßig **en-us** verwendet. • **navigator.systemLanguage.** Die Systemsprache, die in der Eigenschaft **systemLanguage** des Navigator-DOM-Objekts festgelegt ist. **Standardwert:** Der aufgezeichnete Wert. In Skripts, die mit älteren Aufzeichnungsengines erstellt wurden, wird standardmäßig **en-us** verwendet. • **navigator.userLanguage.** Die Benutzersprache, die in der Eigenschaft **userLanguage** des Navigator-DOM-Objekts festgelegt ist. **Standardwert:** Der aufgezeichnete Wert. In Skripts, die mit älteren Aufzeichnungsengines erstellt wurden, wird standardmäßig **en-us** verwendet.

Element der Oberfläche	Beschreibung
Bildschirmeigenschaften	• **screen.width.** Festlegen der Eigenschaft **width** des Bildschirm-DOM-Objekts in Pixeln. **Standardwert:** 1024 Pixel. • **screen.height.** Festlegen der Eigenschaft **height** des Bildschirm-DOM-Objekts in Pixeln. **Standardwert:** 768 Pixel. • **screen.availWidth** Festlegen der Eigenschaft **availWidth** des Bildschirm-DOM-Objekts in Pixeln. **Standardwert:** 1024 Pixel. • **screen.availHeight.** Festlegen der Eigenschaft **availHeight** des Bildschirm-DOM-Objekts in Pixeln. **Standardwert:** 768 Pixel.
Speicherverwaltung	• **Standardmäßige Blockgröße für DOM-Arbeitsspeicherreservierungen.** Festlegen der Standardblockgröße für DOM-Speicherzuordnungen. Ein zu niedriger Wert kann zusätzliche Aufrufe an **malloc** verursachen, die die Ausführung verzögern. Zu große Blöcke können zu einer unnötig hohen Auslastung führen. **Standardwert:** 16384 Byte. • **Speichermanager für dynamisch erstellte DOM-Objekte. Ja** – Speichermanager für dynamisch erstellte DOM-Objekte verwenden. **Nein** – Der Speichermanager wird nicht verwendet, z. B. wenn mehrere DOM-Objekte dynamisch im selben Dokument wie unter SAP erstellt werden. **Auto** – Das empfohlene Protokoll wird verwendet (Standardeinstellung **Yes** für alle Protokolle außer SAP). • **Größe des JavaScript-Laufzeitspeichers (KB).** Die Größe des JavaScript-Laufzeitspeichers in Kilobyte. **Standardwert:** 256 KB. • **Größe des JavaScript-Stackspeichers (KB).** Die Größe des JavaScript-Stackspeichers in Kilobyte. **Standardwert:** 32 KB.

Element der Oberfläche	Beschreibung
Web JavaScript	• **Ausführen von JavaScript-Code ermöglichen**. **Ja** – Aktiviert die Ausführung von Web JavaScript-Schritten, wie `web_js_run()` und `web_js_reset()`. **Nein** – Web JavaScript-Schritte können nicht ausgeführt werden. Beachten Sie, dass das Aktivieren dieser Option das Erstellen eines JavaScript-Laufzeitmoduls bewirkt, auch wenn das Skript keine JavaScript-Schritte enthält. **Standardwert:** Nein • **Größe des JavaScript-Laufzeitmoduls (KB)**. Die Größe des JavaScript-Laufzeitmodulspeichers in Kilobyte. Für alle Vuser in einem Prozess wird ein Modul erstellt. **Standardwert:** 10.240 KB • **Stapelgröße pro Thread für JavaScript-Modul (KB)**. Gibt die Größe jedes Vuser-Threads im JavaScript-Laufzeitmodulspeicher in Kilobyte an. **Standardwert:** 32 KB

Internetprotokoll > Proxy (Knoten)

Ermöglicht Ihnen, die Verbindungseinstellungen für den Proxyserver zu konfigurieren.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Internetprotokoll > Proxy**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Kein Proxy	Alle Vuser verwenden direkte Verbindungen zum Internet. Dies bedeutet, dass die Verbindung ohne Proxyserver hergestellt wird.

Element der Oberfläche	Beschreibung
Proxyeinstellungen vom Standardbrowser abrufen	Alle Vuser verwenden die Proxy-Einstellungen des Standardbrowsers auf dem Computer, über den sie ausgeführt werden. **Hinweis:** Diese Option nicht verfügbar, wenn Firefox als Standardbrowser festgelegt wurde. Deinstallieren Sie Firefox und legen Sie einen anderen Standardbrowser fest.
Benutzerdefinierten Proxy verwenden	Alle Vuser verwenden einen benutzerdefinierten Proxyserver. Sie können die Details zum Proxyserver eingeben oder den Pfad eines Proxyskripts angeben (.pac-Datei), das die automatische Proxykonfiguration ermöglicht. • **Automatisches Konfigurationsskript verwenden.** Hier können Sie eine JavaScript-Datei angeben, die Informationen zur Proxyzuordnung enthält. Durch dieses Skript erkennt der Browser, wann er auf einen Proxyserver zugreifen muss und wann er die Verbindung zur Site direkt herstellen soll, abhängig vom URL. Außerdem können Sie angeben, dass der Browser einen bestimmten Proxyserver für spezielle Adressen und einen anderen Server für andere Adressen verwenden soll. Geben Sie den Speicherort des Skripts im Feld **Adresse** an. • **Proxyserver verwenden**. Sie können entweder einen Proxyserver für alle **HTTP**-Websites und einen anderen für alle (sicheren) **HTTPS**-Websites angeben oder die Option **Für alle Protokolle denselben Proxyserver verwenden** aktivieren. • Ausnahmen... . Ermöglicht Ihnen, Ausnahmen für die Proxyserverregeln festzulegen. • Authentifizierung... . Öffnet das Dialogfeld **Proxy-Authentifizierung**. Falls der Proxyserver für jeden Vuser eine Authentifizierung erfordert, geben Sie in diesem Dialogfeld das Kennwort und den Benutzernamen ein. Zum dynamischen Hinzufügen von Authentifizierung während der Aufzeichnung oder zum Hinzufügen von Authentifizierung für mehrere Proxyserver verwenden Sie die Funktion web_set_user. Weitere Informationen finden Sie in der Function Reference (Hilfe > Funktionsreferenz).

Java-Umgebungseinstellungen > Java - Classpath (Knoten)

Unter **ClassPath** können Sie den Speicherort zusätzlicher Klassen angeben, die nicht in der Umgebungsvariable des Systems für Klassenpfade enthalten sind. Diese Klassen benötigen Sie

möglicherweise, um Java-Anwendungen auszuführen und die ordnungsgemäße Wiedergabe zu gewährleisten.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Java-Umgebungseinstellungen > Classpath**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
+ Hinzufügen	**Klassenpfad hinzufügen.** Fügt der Klassenpfadliste eine neue Zeile hinzu. Mit der Schaltfläche **Durchsuchen** können Sie nach dem Klassenpfad suchen. **Hinweis:** Das gleichzeitige Hinzufügen mehrerer JAR-Dateien wird nur unterstützt, wenn Sie Internet Explorer 10 oder höher verwenden.
Löschen	**Löschen.** Entfernt einen Klassenpfad unwiderruflich.
Bearbeiten	**Klassenpfad bearbeiten.** Ermöglicht es Ihnen, den Pfad für den ausgewählten Klassenpfad zu bearbeiten.
Liste der Klassenpfadeinträge	Eine Liste der Klassenpfadeinträge.

Java-Umgebungseinstellungen > Java VM (Knoten)

Ermöglicht Ihnen das Konfigurieren der Laufzeiteinstellungen für die Java VM.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Java-Umgebungseinstellungen > Java VM**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung

Einstellungen des virtuellen Computers	• **Interne Logik für die Suche nach JDK verwenden.** Die Pfad-, Registrierungs- und Windows-Ordner werden nach dem JDK durchsucht, das bei der Wiedergabe verwendet werden soll. • **Angegebenes JDK verwenden.** Verwendet für die Wiedergabe das angegebene JDK. • **JDK.** Das Stammverzeichnis des angegebenen JDK. • **Zusätzliche VM-Parameter.** Geben Sie optionale Parameter ein, die die Virtual Machine verwendet. • **Xbootclasspath-Parameter werden verwendet.** Das Skript wird mit der Option **Xbootclasspath /p** abgespielt.
Klassenladeeinstellungen	**Jeden Vuser mit dediziertem Klassenladeprogramm laden.** Zum Laden jedes Vusers wird ein dediziertes Klassenladeprogramm verwendet. Dadurch können Sie einen eindeutigen Namespace für jeden Vuser verwenden und seine Ressourcen separat verwalten.

JMS > Erweitert (Knoten)

Ermöglicht Ihnen das Konfigurieren der Laufzeiteinstellungen für die erweiterten JMS-Optionen.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > JMS > Erweitert**
Wichtige Informationen	Dieser Knoten steht nur für Web Service-Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.
Relevante Aufgaben	"Vorbereiten von Skripts für die Wiedergabe" auf Seite 984

Nachfolgend werden die Elemente der Oberfläche beschrieben:

VM (Virtual Machine)

Element der Oberfläche	Beschreibung
Externe VM verwenden	Ermöglicht das Auswählen einer anderen als der standardmäßigen VM (Virtual Machine). Wenn Sie diese Option deaktivieren, verwenden Vuser die in VuGen zur Verfügung stehende JVM.

Element der Oberfläche	Beschreibung
JVM-Basis	Der Ordner mit der externen JVM. Der Pfad sollte zum JDK-Stammverzeichnis führen, das von JDK_HOME definiert wird. VuGen unterstützt JDK 1.4 und höher.
Classpath	Die Anbieterimplementierung der JMS-Klassen zusammen mit allen anderen erforderlichen unterstützenden Klassen, wie sie vom JMS-Implementierungsanbieter vorgegeben werden.

JMS

Element der Oberfläche	Beschreibung
Zusätzliche VM-Parameter	Zusätzliche Parameter, die an die JVM gesendet werden müssen, wie z. B. Xbootclasspath, und alle anderen, in der Dokumentation zu JVM vorgegebenen Parameter.
Anfängliche JNDI-Kontext-Factory	Der vollständig qualifizierte Klassenname der Fabrikklasse, der einen anfänglichen Kontext erstellt. Wählen Sie eine Kontextfabrik anhand der Liste und definieren Sie Ihre eigene Fabrik.
URL des JNDI-Anbieters	Der vollständige URL des Diensteanbieters. Beispiel: • `Weblogic - t3://myserver:myport` • `Websphere - iiop://myserver:myport`
JMS-Verbindungs-Factory	Der JNDI-Name der JMS-Verbindungsfabrik. Sie können nur eine Verbindungsfabrik pro Skript definieren.
JMS-Sicherheitsprinzipal	Identität des Prinzipals (z. B. der Benutzer) für das Authentifizierungsmodell.
JMS-Sicherheitsanmeldeinformationen	Die Prinzipalanmeldeinformationen für das Authentifizierungsschema.
Anzahl der JMS-Verbindungen pro Prozess	Die Anzahl der JMS-Verbindungen pro mdrv-Prozess oder Vuser. Alle Vuser, die eine Verbindung gemeinsam nutzen, erhalten die gleichen Meldungen. Standardmäßig ist dies 1 Vuser, maximal sind es 50 Vuser. Je weniger Verbindungen Sie pro Prozess herstellen, desto höher ist die Leistung.

Element der Oberfläche	Beschreibung
Optionen für die Zeitüberschreitung bei empfangenen Meldungen	Die Zeitüberschreitungen für die erhaltenen Meldungen: • **Unbegrenzte Wartezeit.** Warten Sie, solange dies für die Meldung erforderlich ist, bevor Sie fortfahren. • **Keine Wartezeit.** Warten Sie nicht auf die Receive-Meldung und geben Sie die Kontrolle direkt an das Skript zurück. Wenn sich keine Meldungen in der Warteschlange befinden, scheitert die Operation. • **Zeitüberschreitung in Sekunden angeben.** Zeitüberschreitungswert für die Meldung. Wenn das Zeitüberschreitungsintervall abgelaufen und keine Meldung eingegangen ist, scheitert die Operation. (Standardeinstellung). **Benutzerdefinierte Zeitüberschreitung.** Dauer bis zur Zeitüberschreitung beim Warten auf die Meldung in Sekunden. Die Standardeinstellung beträgt zwanzig Sekunden. • **Standardwert:** Keine Wartezeit.
Auswahl automatisch generieren	Generiert eine Auswahl für die Antwortmeldung mit der Korrelations-ID der Anfrage (standardmäßig **Nein**). Jede an den Server gesendete JMS-Meldung besitzt eine spezifische ID. Aktivieren Sie diese Option, wenn VuGen automatisch eine Auswahl erstellen soll, in der die Meldungs-ID enthalten ist.

MMS > Server und Protokoll (Knoten)

Ermöglicht Ihnen das Konfigurieren der MMS-Laufzeiteinstellungen (Multimedia Messaging Service).

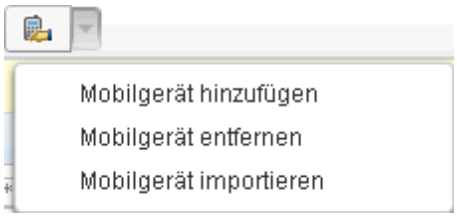
Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > MMS > Server und Protokoll**
Wichtige Informationen	• Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334. • Wählen Sie unter dem Knoten **Allgemein > Verschiedenes** unter **Multithreading** die Option **Vuser als Prozess ausführen** aus.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Automatische WAP-Verbindungen	Diese Option bestimmt, wann Verbindungen zu einem WAP-Gateway hergestellt oder getrennt werden. Diese Einstellung ist nur bei Verwendung eines WAP-Gateways relevant. Die möglichen Werte sind: **Per Iteration.** Die Verbindung wird zu Beginn jeder Iteration hergestellt und am Ende jeder Iteration getrennt **Per Send or Receive.** Die Verbindung wird am Beginn und am Ende jeder Nachricht hergestellt bzw. getrennt. **None.** Es werden keine automatischen WAP-Verbindungen verwendet. **Standardwert:** Per Iteration.
Standard-Absenderadresse	Die Standardadresse, die in der Sender-Kopfzeile gesendet wird. **Standardwert:** +999999.
MMS-Version	Die MMS-Protokollversion, die das Skript verwendet.
MMSC-URL	Der URL des MMSC-Servers (Multimedia Messaging Center).
SMSC-IP	Die IP-Adresse des SMSC-Servers zum Senden von MMS-Benachrichtigungen über SMPP.
SMSC-Port	Der IP-Port des SMSC-Servers zum Senden von MMS-Benachrichtigungen über SMPP.
Zeitüberschreitung (Sekunden)	Die Wartezeit des Servers auf eingehende Nachrichten. **Standardwert:** 60 Sekunden.

Eigenschaften von Mobile TruClient > Knoten "Mobilgerät"

In diesem Bereich können Sie die Eigenschaften eines Mobilgeräts auswählen, wenn Sie ein TruClient Ajax - Mobile-Skript aufzeichnen.

Zugriff	• **VuGen > Wiedergeben > Laufzeiteinstellungen > Eigenschaften von Mobile TruClient > Knoten "Mobilgerät"** • Wählen Sie in der VuGen-Hauptsymbolleiste die Schaltfläche **Mobilgerät**. 

Relevante Aufgaben	"Hinzufügen, Entfernen und Importieren von Mobilgeräteeinstellungen für Mobile TruClient" auf Seite 783 "Aufzeichnen eines Skripts mit Mobile TruClient" auf Seite 783

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Mobilgerät	Wählen Sie den mobilen Gerätetyp aus, den Sie testen möchten.
Benutzer-Agent:	Geben Sie die Kopfzeilenzeichenfolge an, die an den Server gesendet wird, um das Mobilgerät zu identifizieren. Nach der Auswahl eines Geräts wird der Standard-Kopfzeilenwert angezeigt. Diese Kopfzeile kann jedoch geändert werden.
Anzeigen	Geben Sie die Höhe und Breite des Mobilgerätebildschirms an. Mobile TruClient öffnet ein Browserfenster entsprechend den Displayeinstellungen.

.NET > .NET-Umgebung (Knoten)

Ermöglicht Ihnen das Konfigurieren der Laufzeiteinstellungen für .NET-Vuser-Skripts.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > .NET > .NET-Umgebung**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung

AUT-Konfiguration	Einstellungen für den AUT-Speicherort und die Konfigurationsdateien. • **Basispfad der AUT-Anwendung.** Das AUT-Stammverzeichnis (Application Under Test – zu testende Anwendung), aus dem DLLs bei der Wiedergabe geladen werden. Standardmäßig werden während der Aufzeichnung alle erforderlichen DLLs im Skriptverzeichnis gespeichert. Mit dieser Option legen Sie den Speicherort für DLL-Dateien fest, die möglicherweise für die zu testende Anwendung fehlen. Dabei handelt es sich in der Regel um den Installationspfad der aufgezeichneten Anwendung. Die zu testende Anwendung muss auf dem Computer installiert sein, auf dem das Skript ausgeführt wird. Wenn Sie dieses Feld leer lassen, verwendet VuGen das lokale Skript- bzw. bin-Verzeichnis bei der Wiedergabe als Anwendungsstammverzeichnis. • **AUT-Konfigurationsdatei.** Der Name der Konfigurationsdatei der aufgezeichneten Anwendung. VuGen kopiert die AUT-Konfigurationsdatei in das Verzeichnis **script\bin** und lädt die lokal gespeicherte Datei. Verwenden Sie beim Angeben eines anderen Speicherorts einen vollständigen Pfad. Wenn Sie nur einen Dateinamen angeben und die Datei nicht im Ordner **script\bin** gespeichert ist, lädt VuGen sie aus dem Anwendungsstammverzeichnis.
Parallelität	**Anwendungsdomäne pro Vuser.** Ermöglicht das Ausführen jedes Vusers in einer separaten Anwendungsdomäne. Durch separate Anwendungsdomänen kann jeder Vuser einzeln ausgeführt werden, ohne dass statische Variablen gemeinsam genutzt werden müssen oder Konflikte zwischen Vusern auftreten können. **Standardwert:** true.

.NET > Gemeinsam genutzte DLLs

In diesem Dialogfeld können Sie die Liste der gemeinsam genutzten DLLs anzeigen und ändern, nachdem Sie ein Vuser-Skript aufgezeichnet haben. Wenn eine DLL in der Liste der gemeinsam genutzten DLLs vorhanden ist, greift der Vuser auf die DLL am freigegebenen Speicherort zu, sobald das Vuser-Skript ausgeführt und die DLL benötigt wird. Die DLL wird nicht auf den Lastgenerator kopiert. Sie können somit bei einer Vuser-Ausführung Festplattenspeicherplatz auf dem Lastgenerator sparen, wenn eine DLL zur Liste der gemeinsam genutzten DLLs hinzugefügt wird.

Hinweis: Die Option "Gemeinsam genutzte DLLs" steht sowohl unter den Aufzeichnungsoptionen als auch den Laufzeiteinstellungen zur Verfügung. Wenn Sie die Liste oder den DLL-Ordner über die Laufzeiteinstellungen im Controller ändern, werden die Änderungen nicht übernommen.
Problemumgehung: Ändern Sie die DLL-Liste über VuGen und nicht über den Controller.

Nach der Aufzeichnung eines Vuser-Skripts wird die Liste der gemeinsam genutzten DLLs aus den Aufzeichnungsoptionen in die Laufzeiteinstellungen kopiert. Details zum Anzeigen und Ändern der Aufzeichnungsoptionen finden Sie unter ".NET > Gemeinsam genutzte DLLs" auf Seite 222.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Microsoft .NET > Gemeinsam genutzte DLLs**
Siehe auch	".NET > Gemeinsam genutzte DLLs" auf Seite 222

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
+ Hinzufügen	**Hinzufügen.** Fügt eine neue Zeile zur Liste mit DLL-Einträgen hinzu. Geben Sie einen vollständigen Pfad zur DLL an. **Hinweis:** Der angegebene Speicherort einer gemeinsam genutzten DLL muss für *alle* Lastgeneratoren, auf denen der Vuser ausgeführt wird, zugänglich sein.
Bearbeiten	**Bearbeiten.** Ermöglicht es Ihnen, den Pfad zur ausgewählten DLL zu bearbeiten.
Löschen	**Löschen.** Entfernt den DLL-Eintrag dauerhaft.
Liste mit DLL-Einträgen	Eine Liste der Klassenpfadeinträge.

Netzwerk > Geschwindigkeitssimulation (Knoten)

Ermöglicht Ihnen, die Laufzeiteinstellungen für die Netzwerkgeschwindigkeit festzulegen.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Netzwerk > Geschwindigkeitssimulation**
Wichtige Informationen	• Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334. • Unter Linux ausgeführt wird diese Einstellung nur für das Web (HTTP/HTML)-Protokoll unterstützt.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Folgende Bandbreite	Mit dieser Option geben Sie eine bestimmte Bandbreitenebene an, die durch den Vuser emuliert wird. Sie können eine Geschwindigkeit zwischen 14,4 und 512 Kb/s auswählen, um Analogmodems, ISDN oder DSL zu emulieren.
Benutzerdefinierte Bandbreite	Mit dieser Option geben Sie eine Bandbreitengrenze an, die durch den Vuser emuliert wird. Geben Sie die Bandbreite in Bit an; dabei gilt 1 Kilobit = 1024 Bit.
Maximale Bandbreite verwenden	Vuser werden mit der maximalen Bandbreite ausgeführt, die das Netzwerk ermöglicht. **Standardwert:** Aktiviert.

Oracle NCA > Clientemulation (Knoten)

Ermöglicht Ihnen das Konfigurieren der Oracle NCA-Laufzeiteinstellungen.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Oracle NCA > Clientemulation**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung

Netzwerk	• **Socket-Modus.** Hier können Sie die Verbindung zwischen Client und Server auf Socket-Ebene konfigurieren. ▪ **Zeitüberschreitung.** Die Zeitspanne, die ein Oracle NCA-Vuser auf eine Antwort vom Server wartet. **Standardwert:** -1 (deaktiviert die Zeitüberschreitung und der Client wartet für eine unbegrenzte Dauer). • **Pragma-Modus.** Hier können Sie die Kommunikation im Oracle-definierten Pragma-Modus konfigurieren (höhere Ebene als Socket und HTTP). ▪ **Max. Wiederholungsversuche.** Steht für die maximale Anzahl an Meldungen vom Typ **IfError**, die der Client vom Server akzeptiert, bevor er einen Fehler ausgibt. Durch die periodischen Meldungen vom Typ **IfError** teilt der Server dem Client mit, dass er die angefragten Daten so bald wie möglich senden wird. ▪ **Wiederholungsintervall (ms).** Das Intervall (in Millisekunden) zwischen Wiederholungen, falls Meldungen vom Typ **IfError** auftreten. ▪ **Wiederholungsintervalle in Transaktion aufnehmen.** Durch diese Option wird das Zeitintervall zwischen Wiederholungen in die Transaktionsdauer einbezogen. • **Heartbeat aktivieren.** Sendet ein Heartbeatsignal an den Server. Sie können die Frequenz des Signals über die Eigenschaft **Frequenz** festlegen. **Standardwert:** aktiviert, 120.
Verbindung	• **Forms-Version.** Die Version des Oracle-Formularservers, die während der Aufzeichnung erkannt wird. Ändern Sie diese Einstellung nur, wenn seit der letzten Aufzeichnung ein Serverupgrade vorgenommen wurde. • **Trennzeichen für Befehlszeilenparameter.** Markierungen für die Teilung getrennter Parameter in einer Befehlszeilen-Zeichenfolge.
Diagnose	• **Anwendungsversion.** Die Version von Oracle Application. Diese Option ist für Oracle Application relevant, nicht für benutzerdefinierte Oracle NCA-Anwendungen. Sie wird nur für Oracle-Datenbankaufschlüsselung benötigt.

RDP > Erweitert (Strukturknoten)

Ermöglicht Ihnen das Konfigurieren der Laufzeiteinstellungen für die erweiterten RDP-Optionen.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > RDP > Erweitert**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Bitmapzwischenspeicherung	Ermöglicht das Verwenden von Bitmapzwischenspeicherung auf dem Remotedesktopserver. Durch das Aktivieren dieser Einstellung werden auf dem Remotedesktopserver weniger Systemressourcen belegt.
Schriftartglättung	Ermöglicht das Verwenden von Schriftartglättung auf dem Remotedesktopserver. Durch das Deaktivieren dieser Einstellung werden auf dem Remotedesktopserver weniger Systemressourcen belegt.
Menü- und Fensteranimation	Ermöglicht das Animieren von Menüs und Fenstern auf dem Remotedesktopserver. Durch das Deaktivieren dieser Einstellung werden auf dem Remotedesktopserver weniger Systemressourcen belegt.
Remotedesktopgestaltung	Ermöglicht die Gestaltung des Remotedesktops.
Fensterinhalte während des Ziehens anzeigen	Zeigt die Inhalte von Fenstern an, während diese gezogen werden. Durch das Deaktivieren dieser Einstellung werden auf dem Remotedesktopserver weniger Systemressourcen belegt.
Hintergrundbild auf dem Remotedesktop anzeigen	Mit dieser Option können Sie die Anwendung auf dem Remotedesktop ausführen, ohne das Hintergrundbild auf dem Remotedesktop anzuzeigen. Durch das Deaktivieren dieser Einstellung werden auf dem Remotedesktopserver weniger Systemressourcen belegt.
Größe des Socketempfangspuffers (Byte)	Die Anzahl an Byte, die dem Empfangspuffer des Sockets zugewiesen werden. Ein zu kleiner Puffer kann überlaufen und so das Trennen der Serververbindung verursachen. Ist der Puffer zu groß, erfordert er mehr lokale Systemressourcen (Arbeitsspeicher).
Designs	Ermöglicht das Verwenden von Windows-Designs auf dem Remotedesktopserver. Durch das Deaktivieren dieser Einstellung werden auf dem Remotedesktopserver weniger Systemressourcen belegt.

RDP > Konfiguration (Knoten)

Ermöglicht Ihnen das Festlegen der Laufzeiteinstellungen für die RDP-Konfiguration.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > RDP > Konfiguration**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
RDP-Zwischenspeicherung aktivieren	Unterstützung für die Reihenfolge bei der Datenzwischenspeicherung in RDP (standardmäßig aktiviert).
Emulation der RDP-Clientversion	Die RDP-Paketversion, die während der Wiedergabe erzeugt werden soll: **Wie aufgezeichnet** oder eine bestimmte Versionsnummer.
Remotedesktop-Farbtiefe	Die Farbtiefeeinstellungen für die Wiedergabe: **Wie aufgezeichnet** oder eine bestimmte Tiefe.
Remotedesktopauflösung (Pixel)	Die Größe des Fensters, in dem die Anwendungen ausgeführt werden: **Wie aufgezeichnet** oder eine bestimmte Größe (in Pixel).
Bei Verbindung folgendes Programm starten	Öffnen der RDP-Verbindung, um die angegebene Anwendung aufzurufen. Machen Sie folgende Angaben: **Programmpfad und Dateiname** und optional **In Ordner starten**.
Unterstützte Protokolle	Geben Sie den Authentifizierungstyp an, der vom Vuser bei der Wiedergabe des Skripts für die Verbindung zum Server verwendet wird: • **RDP:** Stellt eine Verbindung ohne Authentifizierung her. • **CredSSP:** Stellt eine Verbindung mit dem CredSSP-Protokoll (Credential Security Support Provider) her, um eine starke Serverauthentifizierung durchzuführen. • **SSL:** Ermöglicht Clients die Authentifizierung bei Servern und bietet durch die Verschlüsselung der Kommunikation einen sicheren Kanal.

RDP > RDP-Agent (Knoten)

Ermöglicht Ihnen das Konfigurieren der Laufzeiteinstellungen für den RDP-Agenten.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > RDP > RDP-Agent**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Mit RDP-Agent wiedergeben	Weist VuGen an, die vom RDP-Agent während der Aufzeichnung erfassten erweiterten Informationen zu verwenden. Hierfür muss die Sitzung mit aktivierter Aufzeichnungsoption **RDP-Agent verwenden** aufgezeichnet worden sein. Der LoadRunner RDP-Agent muss auf dem Server installiert sein. Weitere Informationen finden Sie unter "Installieren/Deinstallieren des RDP-Agenten" auf Seite 828. **Wartezeit für Antwort des RDP-Agenten.** Die Zeit (in Millisekunden), die maximal gewartet wird, bis der Agent auf eine Anforderung antwortet.
RDP-Agentenprotokoll aktivieren	Aktivieren des RDP-Agentenprotokolls. Diese Funktion sollte nur für Debuggingzwecke verwendet werden. • **Detailstufe für RDP-Agentenprotokoll.** Konfigurieren der Detailebene des RDP-Agentenprotokolls; dabei steht **Standard** für die niedrigste Detailebene und **Erweitertes Debuggen** für die höchste Detailebene. • **Ziel für RDP-Agentenprotokoll.** Konfigurieren des Ziels für die RDP-Agentenprotokolldaten. Mit der Option **File** werden die Protokollmeldungen nur auf dem Remoteserver gespeichert. Mit der Option **Datenstrom** werden die Protokollmeldungen an den VuGen-Computer gesendet. Mit der Option **Datei und Datenstrom** werden die Protokollmeldungen an beide Ziele gesendet. • **Ordner für RDP-Agentenprotokoll.** Der Verzeichnispfad auf dem Remoteserver, in dem die RDP-Agentenprotokolldatei generiert wird. Wenn kein Verzeichnis angegeben wird und für den Speicherort des Agentenprotokolls die Option **Datei** festgelegt ist, wird das Protokoll im temporären Ordner des Benutzers auf dem Server gespeichert.

RDP > Synchronisierung (Knoten)

Ermöglicht Ihnen das Konfigurieren der Laufzeiteinstellungen für die RDP-Synchronisierung.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > RDP > Synchronisierung**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung

Standardmäßige Eingabequelle	Die Standardquelle für Eingabevorgänge.
Standardmäßiges Hinzufügen von Offsets	Der Offset von Bildern, die während der Synchronisierung verschoben wurden, wird für alle nachfolgenden Funktionen gespeichert. **Standardwert:** Nein.
Standardzeitüberschreitung für Synchronisierung (s)	Die Wartezeit für Synchronisierungsvorgänge in Sekunden. Geben Sie einen Wert zwischen 0 und 1000 ein. **Standardwert:** 60.
Standardtoleranz für Bildsynchronisierung	Die Toleranzebene für Synchronisierungsvorgänge bei Bildern. Wählen Sie eine der folgenden Optionen aus: **Genau**, **Niedrig**, **Mittel** oder **Hoch**. **Hoch** entspricht der höchsten Toleranz für Änderungen und falsche Zuordnungen. **Niedrig** erfordert eine Übereinstimmung von rund 95 Prozent, **Mittel** erfordert eine Übereinstimmung von rund 85 Prozent, **Hoch** erfordert eine Übereinstimmung von rund 70 Prozent und **Genau** erfordert eine Übereinstimmung von 100 Prozent. **Standardwert:** Mittel.
Synchronisierungsfehler-Dialogfeld deaktivieren	Bei Aktivierung dieser Option wird das Dialogfeld für Synchronisierungsfehler nicht eingeblendet. **Standardwert:** nicht ausgewählt.
Fehlschlag von Bildsynchronisierungsschritt bei Zeitüberschreitung	Diese Option legt fest, wie Vuser vorgehen, wenn Bilder während der Synchronisierung nicht gefunden werden. Durch **Ja** wird der Status **Fehler** ausgegeben und für Vuser gilt die Einstellung **Bei Fehler fortfahren**. Durch **No** wird das Kennzeichen **LR_NOT_FOUND** zurückgegeben, der Schritt erhält eine Warnmeldung und das Skript wird fortgeführt. **Standardwert:** Ja.
Aufgezeichnet	Für alle Eingabevorgänge ohne Angabe einer Eingabequelle werden Koordinaten verwendet. **Standardwert:** Aktiviert.
Synchronisiert	Die neuesten Offsets, die bei einer der vorherigen Synchronisierungsfunktionen gespeichert wurden, werden zu den aufgezeichneten Koordinaten jedes Eingabevorgangs hinzugefügt, für den keine Eingabequelle angegeben wurde.
Eingabegeschwindigkeit (ms/Zeichen)	Die Zeit in Millisekunden für das Senden aufeinanderfolgender Zeichen bei Tastaturbefehlen. Geben Sie einen Wert zwischen 0 und 1000 ein. **Standardwert:** 150.

RTE > RTE (Knoten)

Ermöglicht Ihnen das Konfigurieren der RTE-Laufzeiteinstellungen.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > RTE > RTE**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Maximale Anzahl von Verbindungsversuchen	Beim Aufzeichnen einer Verbindung zu einem Host generiert VuGen die Funktion **TE_connect**. Bei der Wiedergabe eines RTE-Vuser-Skripts wird der Terminalemulator durch die Funktion **TE_connect** mit dem angegebenen Host verbunden. Ist der erste Verbindungsversuch erfolglos, versucht der Vuser mehrmals erneut, eine Verbindung herzustellen. **Standardwert:** 5.
Ursprünglichen Gerätenamen verwenden	In bestimmten Umgebungen erfordert jede Sitzung (jeder Vuser) einen eindeutigen Gerätenamen. Durch die Funktion TE_connect wird für jeden Vuser ein eindeutiger Gerätename aus 8 Zeichen generiert, der zum Herstellen der Verbindung dient. Diese Option dient zum Verbinden mithilfe des Gerätenamens (der im Parameter **com_string** der Funktion **TE_connect** enthalten ist. **Hinweis:** Die Einstellung für den ursprünglichen Gerätenamen gilt nur bei IBM-Block-Mode-Terminals.
Verzögerung vor der Eingabe	Mit der Verzögerungseinstellung bestimmen Sie, wie die Funktion **TE_type** von Vusern ausgeführt wird. • **Erste Taste.** Legen Sie die Zeitdauer (in Millisekunden) fest, die ein Vuser vor dem Eingeben des ersten Zeichens einer Zeichenfolge wartet. • **Nachfolgende Tasten.** Legen Sie die Zeitdauer (in Millisekunden) fest, die ein Vuser zwischen zwei aufeinanderfolgenden Zeichen wartet. **Hinweis:**Mit der Funktion **TE_typing_style** können Sie die Verzögerungseinstellungen für einen Teil des Vuser-Skripts überschreiben.
X-System-Synchronisierung	• **Zeitüberschreitung.** Der Zeitraum in Sekunden, der abgewartet wird, bis sich das System bei der Wiedergabe einer **TE_wait_sync**-Funktion stabilisiert, bevor ein Fehler zurückgegeben wird. • **Stabilisierungszeit.** Die Zeit (in Millisekunden) nach der Ausführung einer **TE_wait_sync**-Funktion, die der Vuser abwartet, um sicherzustellen, dass das Terminal sich nicht mehr im X-SYSTEM-Modus befindet.

SAPGUI > Knoten "Allgemein"

Ermöglicht Ihnen das Konfigurieren der SAPGUI-Laufzeiteinstellungen.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > SAPGUI > Allgemein**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Text der Statusleiste senden.	Der Text aus der Statusleiste wird an die Protokolldatei gesendet.
Titel des aktiven Fensters senden.	Der Titeltext des aktiven Fensters wird an die Protokolldatei gesendet.
SAP-Client während der Wiedergabe anzeigen.	Durch diese Option wird bei der Wiedergabe eine Animation der Aktionen im SAP-Client angezeigt. Das Anzeigen der Benutzeroberfläche ist vorteilhaft, weil Sie das Ausfüllen der Formulare sehen und die Aktionen des Vusers genau verfolgen können. Diese Option erfordert jedoch zusätzliche Ressourcen und kann die Leistung des Lasttests beeinträchtigen. • **ActiveScreen-Snapshots während der Wiedergabe erstellen.** Durch diese Wiedergabeoption werden Snapshots mit Steuerelement-IDs für alle aktiven Objekte erfasst. Snapshots vom Typ **ActiveScreen** unterscheiden sich von normalen Snapshots, da Sie nachvollziehen können, welche Objekte von VuGen im SAPGUI-Client erkannt wurden. Wenn Sie die Maus über den Snapshot bewegen, hebt VuGen die erkannten Objekte hervor. Anschließend können Sie direkt aus dem Snapshot neue Schritte zum Skript hinzufügen. Auch Schritte für ein bestimmtes Objekt lassen sich auf interaktive Weise aus dem Snapshot hinzufügen. Weitere Informationen finden Sie unter "Erweitern von SAP GUI-Skripts" auf Seite 867.

Optionen...	Öffnet ein Dialogfeld mit erweiterten SAPGUI-Optionen, in dem Ihnen folgende Optionen zur Verfügung stehen: • **Mit aktiver SAPlogon-Anwendung wiedergeben.** Durch diese Option verwenden die Vuser bei der Wiedergabe die SAPlogon-Anwendung, die derzeit ausgeführt wird. • **Zeitüberscheitung für SAPfewgsvr-Prozess festlegen.** Mit dieser Option können Sie die Zeitüberschreitung für den Prozess **SAPfewgsvr.exe** ändern. ▪ **Zeitüberschreitung für SAPfewgsvr.** Die Zeitüberschreitung für den Prozess SAPfewgsvr.exe in Sekunden. Standardwert: 300 Sekunden.

Silverlight > Dienste (Knoten)

Zeigt die Ihrem Skript zugeordneten WSDL-Dateien an und ermöglicht Ihnen das Konfigurieren der zugehörigen Wiedergabeeinstellungen.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > Silverlight > Dienste**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Dienstliste>	Eine Liste der WSDL-Dateien, die für dieses Skript zur Verfügung stehen.
Protokoll- & Sicherheitsdaten	Öffnet das Dialogfeld **Daten für das Szenario zu Protokoll und Sicherheit**. Hier können Sie eine Reihe von Einstellungen für die jeweils ausgewählte WSDL-Datei festlegen. Weitere Informationen finden Sie unter "Dialogfeld "Daten für das Szenario zu Protokoll und Sicherheit"" auf Seite 245.

WAP > Gateway (Strukturknoten)

Ermöglicht Ihnen das Konfigurieren der Laufzeiteinstellungen für das WAP-Gateway.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > WAP > Gateway**

Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
HTTP direkt	Die Vuser werden im HTTP-Modus ausgeführt und greifen direkt auf einen Webserver zu.
WAP-Gateway-Eigenschaft	Die Vuser werden beim Zugriff auf einen Webserver über ein WAP-Gateway ausgeführt. • **IP.** Die IP des Gateways. • **Port.** Der Port des Gateways. Werden die Vuser über ein WAP-Gateway ausgeführt, legt VuGen automatisch Standardportnummern fest, abhängig vom ausgewählten Modus. Sie können die Einstellungen jedoch anpassen und benutzerdefinierte Werte für IP-Adresse und Port des Gateways festlegen. • **WAP 1.x (WSP).** Wählt die geeignete WAP-Version aus. Wenn Sie in WAP 1.x (WSP) aufzeichnen, können Sie den Vuser im Modus 1.x (WSP) oder im Modus 2.0 (HTTP proxy) ausführen. Wenn Sie diese Option auswählen, können Sie die Eigenschaften für WAP 1.x (WSP) festlegen. Weitere Informationen finden Sie unter "Eigenschaften von WAP 1.x (WSP)" unten. • **WAP 2.0 (HTTP).** Wählt die geeignete WAP-Version aus. Wenn Sie in WAP 2.0 (HTTP proxy) aufzeichnen, können Sie den Vuser nur in diesem Modus ausführen.

Eigenschaften von WAP 1.x (WSP)

Element der Oberfläche	Beschreibung
Erweitert	Erweitern Sie diesen Bereich, um die erweiterten Eigenschaften festzulegen. Weitere Informationen finden Sie unter "Erweiterte Eigenschaften" auf der nächsten Seite.
Verbindungsoptionen	• **Verbindungsorientierter Modus** legt den Verbindungsmodus für die WSP-Sitzung auf verbindungsorientiert fest. • **Verbindungsloser Modus** legt den Verbindungsmodus für die WSP-Sitzung auf verbindungslos fest.
Sicherheit aktivieren	Ermöglicht eine sichere Verbindung zum WAP-Gateway.

Erweiterte Eigenschaften

Element der Oberfläche	Beschreibung
Kopfzeile bestätigen	Gibt Standardkopfzeilen mit Informationen an das Gateway zurück. **Standardwert:** Deaktiviert.
Trägertyp	Der Trägertyp, der als Transportbasis verwendet wird.
CAPSessionResume	Ermöglicht Anfragen für das Anhalten oder Wiederaufnehmen einer Sitzung.
Client-SDU-Puffergröße	Die größte Dienstdateneinheit (SDU) bei Transaktionen, die während der Sitzung an den Client gesendet werden darf. **Standardwert:** 4000.
Unterstützung für Push-Bestätigung	Wird im CO-Modus eine Push-Meldung empfangen, sorgt diese Option dafür, dass der Vuser den Empfang der Meldung bestätigt. Weitere Informationen finden Sie unter VuGen Push Support.
MethodMOR	Die Anzahl an ausstehenden Methoden, die gleichzeitig auftreten können.
Netzwerk-MTU-Größe	Die maximale Größe des Netzwerkpakets in Byte. **Standardwert:** 4096.
Push-Unterstützung	Ermöglicht Push-Meldungen über das Gateway. **Standardwert:** Deaktiviert.
PushMOR	Die Anzahl an ausstehenden Push-Transaktionen, die gleichzeitig auftreten können.
Meldungen abrufen	Wird eine Push-Meldung empfangen, sorgt diese Option dafür, dass der Vuser die Meldungsdaten von dem in der Push-Meldung angegebenen URL abruft. **Standardwert:** Deaktiviert.
Server-SDU-Puffergröße	Die größte Dienstdateneinheit (SDU) bei Transaktionen, die während der Sitzung an den Server gesendet werden darf. **Standardwert:** 4000.
Cookies unterstützen	Ermöglicht Unterstützung für das Speichern und Abrufen von Cookies. **Standardwert:** Deaktiviert.
Verkürzter WTLS-Handshake	Durch diese Option wird beim Empfangen einer Umleitungsmeldung ein verkürzter Handshake anstatt eines vollständigen Handshakes verwendet. **Standardwert:** False.

Element der Oberfläche	Beschreibung
WTLS Deffie Hellman	Durch diese Option wird für WTLS (Wireless Transport Layer Security) das Deffie Hellman-Verschlüsselungsschema anstelle des RSA-Standardschemas verwendet. **Standardwert:** False.
WTLS Deffie Hellman-Bezeichner	Eine ID für das Deffie Hellman-Verschlüsselungsschema. Diese ID wird für den verkürzten Handshake mit dem Operwave-Gateway benötigt, das das Deffie Hellman-Verschlüsselungsschema verwendet.
WTP-Neuübertragungszeit	Die Wartezeit der WTP-Schicht in Sekunden, bevor die Protokolldateneinheit (PDU) erneut gesendet wird, wenn keine Antwort empfangen wurde. **Standardwert:** 5000.
WTP-Segmentierung und -Reassemblierung	Ermöglicht die Segmentierung und erneute Assembly (SAR) in WTP (Wireless Transport Protocol). **Standardwert:** true.

WAP > Radius (Strukturknoten)

Ermöglicht Ihnen das Konfigurieren der Laufzeiteinstellungen für den WAP-Radius.

Zugriff	**VuGen > Wiedergeben > Laufzeiteinstellungen > WAP > Radius**
Wichtige Informationen	Dieser Knoten steht nur für bestimmte Protokolle zur Verfügung. Eine vollständige Liste der Protokolle und der mit ihnen verbundenen Knoten finden Sie unter "Tabelle für Protokollkompatibilität" auf Seite 334.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Portnummer für Accounting	Accounting-Port des Radius-Servers
Portnummer für Authentifizierung	Authentifizierungsport des Radius-Servers
Verbindungszeitüberschreitung (s)	Die Wartezeit auf die Antwort des Radius-Servers in Sekunden. **Standardwert:**120 Sekunden.
IP-Adresse	IP-Adresse des Radius-Servers
Netzwerktyp	Accounting-Netzwerktyp: GPRS (General Packet Radio Service) oder CSD (Circuit-Switched Data).

Element der Oberfläche	Beschreibung
Radius-Client-IP	IP der Radius-Paketquelle; dient meist zum Unterscheiden zwischen Paketen, die über verschiedene Netzwerkkarten auf einem einzigen Lastgeneratorcomputer übertragen werden.
Neuübertragungsversuche	Die Anzahl der Wiederholungen nach einer fehlgeschlagenen Übertragung. **Standardwert:** 0.
Geheimer Schlüssel	Geheimer Schlüssel des Radius-Servers
Vom Server zurückgegebene Attribute in Parametern speichern	Vuser können vom Server zurückgegebene Attribute in Parametern speichern, um sie später zu verwenden. **Standardwert:** False.

Textmarken - Übersicht

Beim Bearbeiten eines Vuser-Skripts können Sie Textmarken zur Navigation in den Abschnitten des Skripts verwenden. Wenn Sie eine Textmarke hinzufügen, wird am linken Rand der ausgewählten Skriptzeile ein Textmarkensymbol angezeigt.

Im Textmarkenbereich wird eine Liste aller im Vuser-Skript vorhandenen Textmarken angezeigt. Im Textmarkenbereich können Sie die folgenden Aktionen durchführen:

- Zur Textmarkenposition im Skript navigieren
- In aufeinanderfolgenden Textmarken navigieren
- Einzelne Textmarken löschen
- Alle Textmarken löschen

Weitere Informationen über die Aufgabe finden Sie unter "Verwenden von Textmarken" auf Seite 410.

Ausführen eines Vuser-Skripts über eine Befehlszeile

In dieser Aufgabe wird beschrieben, wie Sie ein Vuser-Skript über eine Befehlszeile oder über das Windows-Dialogfeld "Ausführen" ausführen. Die VuGen-Benutzeroberfläche ist dabei nicht erforderlich.

Um von VuGen aus Befehlszeilenparameter an einen Vuser zu senden, fügen Sie die Attribute und die zugehörigen Werte im Dialogfeld "Laufzeiteinstellungen" hinzu. Details finden Sie im "Allgemein > Zusätzliche Attribute (Knoten)" auf Seite 350.

So führen Sie ein Skript über eine Befehlszeile oder das Dialogfeld "Ausführen" aus:

1. Öffnen Sie ein **Eingabeaufforderung**sfenster, oder wählen Sie **Start** > **Ausführen** aus, um das Dialogfeld "Ausführen" zu öffnen.
2. Geben Sie **mdrv** gefolgt vom Skriptnamen unter Beachtung der folgenden Syntax ein:

   ```
   <installation_dir>/bin/mdrv.exe -usr <script_name>
   ```

   wobei **script_name** der vollständige Pfad zur *.usr*-Skriptdatei ist, z. B. **c:\temp\mytest\mytest.usr**.
3. Fügen Sie weitere Befehlszeilenoptionen und Argumente hinzu.
4. Drücken Sie die **EINGABETASTE**. Das Programm **mdrv** führt eine Instanz des Skripts ohne die Benutzerschnittstelle aus. In den Ausgabedatei sind die Laufzeitinformationen enthalten.

Zum Anzeigen einer vollständigen Liste mit Befehlszeilenoptionen geben Sie vom LoadRunner-Ordner **bin** aus in einer Eingabeaufforderung **mdrv** ohne Argumente ein. Die folgenden Beispiele veranschaulichen, wie ein Befehlszeilenausdruck im Allgemeinen verwendet werden kann:

- Sie können den Lastgenerator sowie die Häufigkeit der Skriptausführung, wie im folgenden Beispiel gezeigt, angeben:

  ```
  script1  -host pc4  -loop 5
  ```

- Geben Sie einen Speicherort für die Ausgabedateien an. Beispiel:

  ```
  -out c:\tmp\vuser
  ```

- Geben Sie Argumente für die Weitergabe an das Skript im folgenden Format an:

  ```
  script_name  -arg_name arg_value   -arg_name arg_value
  ```

Sie können die Befehlszeilenwerte abrufen, indem Sie während der Wiedergabe die Befehlszeile mithilfe von Parserfunktionen, z. B. **lr_get_attrib_double**, analysieren. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Hinweis: Das Linux-Dienstprogramm *run_db_vuser* unterstützt derzeit noch nicht alle standardmäßigen Windows-Befehlszeilenoptionen. Weitere Informationen finden Sie unter "Running a Vuser from the Linux Command Line".

Ausführen eines Vuser-Skripts über eine Linux-Befehlszeile

Wenn VuGen zur Entwicklung von Linux-basierten Vusern verwendet wird, müssen Sie prüfen, ob das aufgezeichnete Skript auf der Linux-Plattform ausgeführt werden kann. In dieser Aufgabe wird beschrieben, wie Sie diesen Test ausführen und ein Vuser-Skript über einen Linux-Befehl ausführen.

1. **Prüfen der Skriptwiedergabe in VuGen**

   Geben Sie das Skript in VuGen wieder, um zu überprüfen, ob das Skript unter Windows ausgeführt werden kann, bevor Sie es unter Linux ausführen. Diese Methode wird empfohlen, da es einfacher ist, das Skript in VuGen zu bearbeiten und zu debuggen. Weitere Informationen über die Aufgabe finden Sie unter "Wiedergeben eines Vuser-Skripts" auf Seite 329.

2. **Kopieren der Skriptdateien auf den Linux-Server**

   Übertragen Sie die Skriptdateien auf den Linux-Server.

3. **Prüfen Sie das Vuser-Setup auf dem Linux-Computer mithilfe von verify_generator.**

   Wenn Sie planen, alle Vuser auf einem Host auszuführen, geben Sie Folgendes ein:

   ```
   verify_generator
   ```

   Das Dienstprogramm **verify_generator** meldet entweder **OK**, wenn die Einstellung richtig ist, oder **Fehlgeschlagen** zusammen mit einem Vorschlag zum Korrigieren des Setups.

   Geben Sie Folgendes ein, um ausführliche Informationen zu den Prüfungen anzuzeigen:

   ```
   verify_generator [-v]
   ```

   Das Prüfdienstprogramm überprüft den lokalen Host auf seine Kommunikationsparameter und seine Kompatibilität mit allen Vuser-Typen. Es prüft die folgenden Elemente in der Vuser-Umgebung:

   - Es sind mindestens 128 Dateideskriptoren verfügbar.
   - Es sind die korrekten **.rhost**-Berechtigungen verfügbar: **-rw-r--r--**
   - Der Kontakt zum Host kann mit **rsh** hergestellt werden. Andernfalls wird nach dem Hostnamen in **.rhosts** gesucht.
   - **M_LROOT** ist definiert.
   - ..**cshrc** definiert den korrekten Wert für **M_LROOT**.
   - ..**cshrc** ist im Home-Verzeichnis vorhanden.
   - Der aktuelle Benutzer ist der Besitzer von **.cshrc**.
   - Im **$M_LROOT** ist eine LoadRunner -Installation vorhanden.
   - Die ausführbaren Dateien verfügen über Ausführungsberechtigungen.
   - **PATH** enthält **$M_LROOT/bin** und **/usr/bin**.
   - Der Dämon **rstatd** ist vorhanden und wird ausgeführt.

4. **Ausführen des Skripts**

Führen Sie das Skript mit dem Shell-Skript **run_db_vuser** im Einzelmodus aus dem Vuser-Skript-Ordner aus:

```
run_db_vuser.sh <Befehle> Skriptname.usr
```

Das Shell-Skript *run_db_vuser* hat die folgenden Befehlszeilenoptionen:

Befehl	Beschreibung
--help	Zeigt die verfügbaren Optionen an. (Dieser Option müssen zwei Bindestriche vorangestellt werden.)
-cpp_only	Führt nur **cpp** (Vorverarbeitung) im Skript aus.
-cci_only	Führt nur **cci** (Vorkompilierung) im Skript aus, um eine Datei mit einer .ci-Erweiterung zu erstellen. Sie können **cci** nur nach einer erfolgreichen Vorverarbeitung (**cpp**) ausführen.
-driver ***Treiberpfad***	Verwendet ein spezifisches Treiberprogramm. Jede Datenbank hat ihr eigenes Treiberprogramm, das sich im Ordner /bin befindet. Beispielsweise ist der Treiber für **CtLib**, der sich im Ordner **/bin** befindet, **mdrv**. Über diese Option können Sie einen externen Treiber angeben.
-exec_only	Führt die Vuser-.ci-Datei aus. Diese Option ist verfügbar, wenn eine gültige .ci-Datei existiert.
-ci ***ci_Dateiname***	Führt eine spezifische .ci-Datei aus.
-out ***Ausgabepfad***	Speichert das Ergebnis in einem bestimmten Ordner.

Standardmäßig führt *run_db_vuser.sh* **cpp**, **cci** und **execute** im ausführlichen Modus aus. Es verwendet den Treiber im Ordner VuGen-Installation\bin und speichert das Ergebnis in einer Ausgabedatei im Vuser-Skriptordner. Sie müssen immer eine *.usr*-Datei festlegen. Wenn Sie sich nicht im Skriptordner befinden, geben Sie den vollständigen Pfad der *.usr*-Datei an.

Beispielsweise führt die folgende Befehlszeile ein Vuser-Skript mit der Bezeichnung test1 aus und stellt die Ausgabedatei in einen Ordner mit der Bezeichnung results1. Der Ergebnisordner muss ein bereits existierender Ordner sein; er wird nicht automatisch erstellt.

```
run_db_vuser.sh-out /u/joe/results1  test1.usr
```

Verwenden von Textmarken

Bei der Bearbeitung im Editor können Sie Textmarken an verschiedenen Positionen Ihres Skripts einfügen. Sie können zwischen den Textmarken navigieren, um Ihren Code zu analysieren und zu debuggen. In den folgenden Schritten wird das Arbeiten mit Textmarken beschrieben. Die meisten

Textmarkenfunktionen sind über den Textmarkenbereich von VuGen zugänglich. Um den Ausschnitt zu öffnen, klicken Sie auf **Ansicht > Textmarken**.

Erstellen einer Textmarke

Platzieren Sie den Cursor im Editor an der gewünschten Position und drücken Sie Strg + F2. VuGen fügt ein Textmarkensymbol am linken Rand des Skripts ein.

Entfernen einer Textmarke

Führen Sie die folgenden Schritte aus, um eine Textmarke zu entfernen:

- Klicken Sie im Editor in die Zeile, in der die Textmarke steht, und drücken Sie Strg + F2.
- Wählen Sie im Textmarkenbereich die Textmarke aus, die Sie löschen möchten, und klicken Sie auf die Schaltfläche **Textmarke löschen**.

VuGen entfernt das Textmarkensymbol vom linken Rand.

Navigieren zwischen Textmarken

Um den Ausschnitt zu öffnen, klicken Sie auf **Ansicht > Textmarken**.

- Um zur nächsten Textmarke zu springen, klicken Sie auf die Schaltfläche **Nächste Textmarke** oder drücken **F2**.
- Um zur vorherigen Textmarke zu springen, klicken Sie auf die Schaltfläche **Vorherige Textmarke** oder drücken **Umschalten + F2**.

Sie können nur zwischen Textmarken in der aktuellen Aktion navigieren. Um zu einer Textmarke in einer anderen Aktion zu navigieren, wählen Sie diese Aktion im linken Bereich aus und drücken anschließend **F2**.

Navigieren zu einer bestimmten Textmarke in einem Vuser-Skript

Doppelklicken Sie im Textmarkenbereich auf die betreffende Textmarke. Der Cursor blinkt am Anfang der Zeile, in der sich die Textmarke befindet.

Bei der Wiedergabe erstellte Dateien

In diesem Abschnitt wird beschrieben, welche Vorgänge bei der Wiedergabe eines Vuser-Skripts ausgeführt werden. Zudem werden die erstellten Dateien beschrieben.

1. Die Datei **options.txt** wird erstellt. Diese Datei enthält Befehlszeilenparameter für den Präprozessor.

Beispiel einer Datei "options.txt"

```
-DCCI
-D_IDA_XL
-DWINNT
-Ic:\tmp\Vuser                  (Name und Speicherort der Vuser-Include-D
ateien)
-IE:\LRUN45B2\include           (Name und Speicherort der Include-Dateie
n)
-ec:\tmp\Vuser\logfile.log      (Name und Speicherort der Ausgabeprotokol
ldatei)
 c:\tmp\Vuser\VUSER.c           (Name und Speicherort der zu verarbeitend
en Datei)
```

2. Die Datei **Vuser.c** wird erstellt. Diese Datei enthält die Anweisung 'Include' für alle relevanten .c- und .h-Dateien.

Beispiel einer Datei "Vuser.c"

```
#include "E:\LRUN45B2\include\lrun.h"
#include "c:\tmp\web\init.c"
#include "c:\tmp\web\run.c"
#include "c:\tmp\web\end.c"
```

3. Der C-Präprozessor **cpp.exe** wird aufgerufen, um Makrodefinitionen, Präcompileranweisungen usw. von Entwicklungsdateien ‘einzufügen’.

   Die folgende Befehlszeile wird verwendet:

   ```
   cpp -foptions.txt
   ```

4. Die Datei **pre_cci.c**, die auch eine C-Datei ist, wird erstellt (**pre_cci.c** ist in der Datei **options.txt** definiert). Die Datei **logfile.log** (ebenfalls in **options.txt** definiert), die die Ausgabe dieses Prozesses enthält, wird erstellt. Diese Datei ist leer, wenn in der Phase der Vorverarbeitung keine Probleme aufgetreten sind. Wenn die Datei nicht leer ist, schlägt die nächste Phase der Kompilierung mit hoher Wahrscheinlichkeit aufgrund eines schwerwiegenden Fehlers fehl.

5. Nun wird der C-Compiler **cci.exe** aufgerufen, um eine plattformabhängige pseudobinäre Datei (.ci) zur Verwendung durch das Vuser-Treiberprogramm zu erstellen, das diese Datei während der Laufzeit interpretiert. Die **cci**-Datei verwendet die Datei **pre_cci.c** als Eingabe.

6. Die Datei **pre_cci.ci** wird wie folgt erstellt:

```
cci -errout c:\tmp\Vuser\logfile.log -c pre_cci.
```

7. Die Datei **logfile.log** ist die Protokolldatei, in der die Ausgabe der Kompilierung enthalten ist.

8. Die Datei **pre_cci.ci** wird nun in **Vuser.ci** umbenannt.

   Da die Kompilierung sowohl Warnungen als auch Fehler enthalten kann und da der Treiber die Ergebnisse dieses Prozesses nicht kennt, prüft der Treiber zunächst, ob die Datei **logfile.log** Einträge enthält. Sind Einträge enthalten, prüft er anschließend, ob die Datei **Vuser.ci** erstellt wurde. Wenn die Dateigröße nicht Null ist, bedeutet dies, dass die Datei "cci" erfolgreich kompiliert wurde. Andernfalls ist die Kompilierung fehlgeschlagen, und es wird eine Fehlermeldung angezeigt.

9. Nun wird der relevante Treiber unter Verwendung der beiden Dateien **.usr** und **Vuser.ci** ausgeführt. Beispiel:

   ```
   mdrv.exe -usr c:\tmp\Vuser\.usr -out c:\tmp\Vuser -file c:\tmp\Vuser\Vuser.ci
   ```

   Die Datei **.usr** ist erforderlich, denn sie teilt dem Treiberprogramm mit, welche Datenbank verwendet wird. Dadurch wird festgelegt, welche Bibliotheken für die Ausführung geladen werden müssen.

10. Wenn bereits eine Wiedergabeprotokolldatei (**output.txt**) vorhanden ist (siehe folgenden Eintrag), wird die Protokolldatei in **output.bak** kopiert.

11. Die Datei **output.txt** wird erstellt (in dem von der Variablen 'out' definierten Pfad). Diese Datei enthält die Ausgabemeldungen, die während der Skriptwiedergabe erstellt wurden. Dabei handelt es sich um die Meldungen, die auch in der Wiedergabeansicht im Ausgabebereich von VuGen angezeigt werden.

Debuggen von Vuser-Skripts

Im Abschnitt **Debuggen** werden die verschiedenen Verfahren zum Debuggen von Vuser-Skripts beschrieben.

Debuggen – Übersicht

Das Erstellen eines Vuser-Skripts umfasst die nachfolgend aufgeführten Schritte. Dieses Thema enthält eine Übersicht über den fünften Schritt, Debuggen eines Vuser-Skripts.

Geben Sie nach dem Erstellen eines Vuser-Skripts das Skript wieder, um zu überprüfen, ob das Skript fehlerfrei ausgeführt wird. Anhand der Debugging-Funktionen von VuGen können Sie Fehler in Ihren Skripts identifizieren und beheben. Sie können über die VuGen-Symbolleiste auf die meisten dieser Funktionen zum Debuggen von Skripts zugreifen.

Ausführen von Vuser-Skripts

Um ein Vuser-Skript bis zum Skriptende oder bis zum nächsten Haltepunkt auszuführen, wählen Sie eine der folgenden Vorgehensweisen aus:

- Wählen Sie **Wiedergeben > Ausführen** aus.
- Klicken Sie in der Vuser-Symbolleiste auf die Schaltfläche **Wiedergabe** .
- Drücken Sie **F5**.

Hinweis: Der Status der Vuser-Skriptausführung wird in der unteren linken Ecke von VuGen angezeigt. Der Status der Skriptausführung kann **Bereit**, **Wird ausgeführt** oder **Pausiert** sein.

- Um ein aktuell ausgeführtes Skript zu beenden, klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Wiedergabe beenden** .
- Um die Ausführung des Skripts zu unterbrechen, klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Wiedergabe unterbrechen** .
- Um die Ausführung eines unterbrochenen Skripts fortzusetzen, klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Weiter** .

Der Befehl "Schritt für Schritt ausführen"

Mit dem Befehl **Schritt für Schritt ausführen** wird das Skript Zeile für Zeile ausgeführt. Dadurch können Sie die Skriptausführung verfolgen. Der Befehl **Schritt für Schritt ausführen** startet die Skriptwiedergabe und hält diese in der ersten Skriptzeile an. Dies ist normalerweise die Aktion vuser_init().

Um das Skript Schritt für Schritt auszuführen, führen Sie eine der folgenden Aktionen aus:

- Wählen Sie **Wiedergeben > Schritt für Schritt ausführen** aus.
- Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Schritt für Schritt ausführen** .
- Drücken Sie **F10**.

Beachten Sie, dass die Schaltfläche **Schritt für Schritt ausführen** nur während der Skriptwiedergabe verfügbar ist.

Animierte Ausführung

Sie können ein Vuser-Skript im animierten oder nicht animierten Modus ausführen. Wenn Sie das Skript im animierten Modus ausführen, markiert VuGen die im Vuser-Skript ausgeführte Zeile. Beim Ausführen im nicht animierten Modus führt VuGen das Vuser-Skript aus, ohne jedoch die ausgeführte Zeile zu markieren.

So wechseln Sie zwischen dem animierten Modus und dem nicht animierten Modus:

- Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Animierte Ausführung** .
- Klicken Sie auf **Wiedergeben > Animierte Ausführung**.

> Beachten Sie, dass die Schaltfläche **Animierte Ausführung** nur während der Skriptwiedergabe verfügbar ist.

Details zum Einstellen des Ausführungsmodus mit den VuGen-Optionen finden Sie unter "Skriptoptionen" auf Seite 83.

Verwenden von Haltepunkten

Durch Haltepunkte wird die Skriptausführung an bestimmten Stellen im Skript angehalten. Dadurch können Sie überprüfen, welche Auswirkungen das Skript an vordefinierten Stellen bei der Skriptausführung hat.

- Weitere Informationen zu dem Konzept von Haltepunkten finden Sie unter "Arbeiten mit Haltepunkten" auf Seite 418.
- Weitere Informationen über die Aufgabe finden Sie unter "Debuggen von Skripts mit Haltepunkten" auf Seite 421.

Verwenden von Textmarken

Beim Arbeiten in der Skriptansicht können Sie über VuGen Textmarken an verschiedenen Positionen in Ihrem Skript einfügen. Sie können zwischen den Textmarken navigieren, um Ihren Code zu analysieren und zu debuggen.

- Weitere Informationen über die Aufgabe finden Sie unter "Verwenden von Textmarken" auf Seite 410.

Beobachten von Variablen

Im Überwachungsbereich können Sie Variablen und Ausdrücke überwachen, während ein Skript ausgeführt wird. Sie können Variablen und Ausdrücke nur überwachen, wenn die Ausführung eines Vuser-Skripts sich im Status **Pausiert** befindet. Um den Überwachungsbereich anzuzeigen, klicken Sie auf **Ansicht > Debuggen > Beobachten**. Details zur Verwendung des Überwachungsbereichs finden Sie unter "Beobachten von Ausdrücken und Variablen" auf Seite 419.

"Gehe zu"-Befehle

- Wenn Sie in einem Skript navigieren möchten und Haltepunkte verwenden, können Sie den Befehl **Gehe zu Quelle** verwenden. Weitere Informationen finden Sie unter "Debuggen von Skripts mit Haltepunkten" auf Seite 421.

- Wenn Sie in einem Skript navigieren möchten und Textmarken verwenden, können Sie die Befehle **Nächste Textmarke** und **Vorherige Textmarke** verwenden. Weitere Informationen finden Sie unter "Verwenden von Textmarken" auf Seite 410.

Wenn Sie die Wiedergabeprotokollmeldungen für einen bestimmten Schritt oder eine bestimmte Funktion überprüfen möchten, klicken Sie im Editor mit der rechten Maustaste auf den entsprechenden Schritt und wählen **Zu Schritt im Wiedergabeprotokoll wechseln** aus. VuGen platziert den Cursor am Anfang des entsprechenden Schritts im Wiedergabeprotokoll des Ausgabebereichs.

Ausgabebereich

Im Ausgabebereich werden Meldungen angezeigt, die während der Wiedergabe des Skripts generiert wurden. Weitere Informationen finden Sie unter "Ausgabebereich" auf Seite 107.

Bei einigen aufgezeichneten Vuser-Skripts kann es für eine korrekte Wiedergabe erforderlich sein, eine Korrelation zu implementieren. Korrelation wird verwendet, wenn ein aufgezeichnetes Skript einen dynamischen Wert enthält (beispielsweise eine Sitzungs-ID) und daher nicht erfolgreich wiedergegeben werden kann. Um dieses Problem zu beheben, wandeln Sie den dynamischen Wert in eine Variable um und ermöglichen so die erfolgreiche Wiedergabe des Skripts. Weitere Informationen finden Sie unter "Korrelieren – Übersicht" auf Seite 281.

Fehlerbehandlung

Sie können festlegen, wie ein Vuser Fehler während der Skriptausführung behandelt. Wenn ein Vuser einen Fehler ermittelt beendet der Vuser standardmäßig die Ausführung des Skriptes. Mit einer der folgenden Methoden können Sie einem Vuser die Anweisung erteilen, bei Auftreten eines Fehlers mit der nächsten Iteration fortzufahren:

- Mit Laufzeiteinstellungen. Sie können die Laufzeiteinstellung **Bei Fehler fortfahren** festlegen. Die Laufzeiteinstellung **Bei Fehler fortfahren** gilt für das gesamte Vuser-Skript. Sie können die Funktion **lr_continue_on_error** verwenden, um die Laufzeiteinstellung **Continue on Error** für einen Teil eines Skriptes zu überschreiben.

- Mit der Funktion **lr_continue_on_error**. Die Funktion **lr_continue_on_error** ermöglicht es Ihnen, die Fehlerbehandlung für ein spezifisches Segment eines Vuser-Skriptes zu kontrollieren. Um das Segment zu markieren, umschließen Sie es mit den Anweisungen `lr_continue_on_error(1);` und `lr_continue_on_error(0);`. Die neuen Fehlereinstellungen gelten für das umschlossene Vuser-Skript-Segment. Einzelheiten dazu enthalten die nachfolgenden Abschnitte.

Wenn Sie beispielsweise die Laufzeiteinstellung **Bei Fehler fortfahren** aktivieren und ein Vuser auf ein Problem bei der Wiedergabe des folgenden Skriptsegments stößt, fährt der Vuser mit der Ausführung des Skriptes fort.

```
    web_link("EBOOKS",
        "Text=EBOOKS",
        "Snapshot=t2.inf",
        LAST);
    web_link("Find Rocket eBooks",
        "Text=Find Rocket eBooks",
        "Snapshot=t3.inf",
        LAST);
```

Um dem Vuser die Anweisung zum Fortfahren bei Fehler für ein spezifisches Segment des Skriptes zu erteilen, umschließen Sie das Segment mit den entsprechenden lr_continue_on_error-Anweisungen:

```
lr_continue_on_error(1);
    web_link("EBOOKS",
        "Text=EBOOKS",
        "Snapshot=t2.inf",
        LAST);
    web_link("Find Rocket eBooks",
        "Text=Find Rocket eBooks",
        "Snapshot=t3.inf",
        LAST);
lr_continue_on_error(0);
```

Zusätzliche Debugginginformationen

Allgemeine Debuggingtipps

VuGen kann als ein regulärer Texteditor verwendet werden. Sie können jede beliebige Textdatei in VuGen öffnen und bearbeiten. Wenn bei der Wiedergabe eine Fehlermeldung im darunter angezeigten Ausgabefenster erscheint, können Sie auf diese Fehlermeldung doppelklicken und VuGen setzt den Cursor auf die Zeile des Tests, die das Problem verursacht hat. Sie können den Cursor auch auf dem Fehlercode platzieren und **F1** drücken, um die Onlinehilfe mit der Erklärung zum Fehlercode aufzurufen.

Verwenden von C-Funktionen für die Verfolgung

Sie können die Traceoption des C-Interpreters (in Version 230 oder höher) zum Debuggen Ihrer Vuser-Skripts verwenden. Die Anweisung `ci_set_debug` ermöglicht es, die Verfolgung und das Debugging an spezifischen Punkten im Skript ein- und auszuschalten.

```
ci_set_debug(ci_this_context, int debug, int trace);
```

Beispielsweise können Sie die folgenden Anweisungen zu Ihrem Skript hinzufügen:

```
ci_set_debug(ci_this_context, 1, 1) /* turn ON trace =; debug */
ci_set_debug(ci_this_context, 0, 0) /* turn OFF trace =; debug */
```

Zusätzliche Schlüsselwörter in der Programmiersprache C

Beim Ausführen eines C-Skripts in VuGen verwendet sein Parser den integrierten C-Interpreter, um die Funktionen im Skript zu analysieren. Sie können Schlüsselwörter hinzufügen, die nicht in der

Bibliothek des Standardparsers enthalten sind. Standardmäßig werden verschiedene allgemeine C++-Schlüsselwörter während der Installation hinzugefügt, wie z. B. *size_t* und *DWORD*. Sie können die Liste bearbeiten und zusätzliche Schlüsselwörter für Ihre Umgebung hinzufügen.

Hinzufügen zusätzlicher Schlüsselwörter

1. Öffnen Sie die Datei vugen_extra_keywords.ini im Verzeichnis <Windows> oder <Windows>/System Ihres Computers.

2. Fügen Sie im Abschnitt `EXTRA_KEYWORDS_C` die gewünschten Schlüsselwörter für den C-Interpreter hinzu.

   Der Dateityp hat das folgende Format:

```
[EXTRA_KEYWORDS_C]
FILE=
size_t=
WORD=
DWORD=
LPCSTR=
```

Überprüfen der Ausgabe für die Wiedergabe

Zeigen Sie die Ausgabe der Wiedergabe an (entweder innerhalb von VuGen oder über die Datei output.txt, die die Ausgabe des VuGen-Treibers darstellt). Sie können auch die Optionen der Laufzeiteinstellungen in VuGen ändern und eine umfassendere Protokollierung auswählen, um eine ausführlichere Protokollausgabe des wiedergegebenen Tests zu erzielen.

Arbeiten mit Haltepunkten

VuGen ermöglicht Ihnen die Verwendung von Haltepunkten in Ihren Vuser-Skripts, um das Debuggen der Skripts zu erleichtern. Durch Haltepunkte wird die Skriptausführung an bestimmten Stellen im Skript angehalten. Auf diese Weise können Sie analysieren, welche Auswirkungen das Skript an vordefinierten Stellen während der Skriptausführung auf Ihre Anwendung hat. Weitere Informationen über die Aufgabe finden Sie unter "Debuggen von Skripts mit Haltepunkten" auf Seite 421. Ein Haltepunktsymbol ( ) am linken Rand des Skripts gibt an, dass ein Haltepunkt vorhanden ist. Darüber hinaus markiert VuGen die entsprechende Zeile im Skript.

Wird ein Haltepunkt vorübergehend nicht benötigt, können Sie ihn deaktivieren. Ein weißer Punkt innerhalb des Haltepunktsymbols gibt an, dass der Haltepunkt deaktiviert ist ( ). Wenn ein Haltepunkt deaktiviert ist, wird die Skriptausführung am deaktivierten Haltepunkt fortgesetzt und am nächsten aktivierten Haltepunkt angehalten. Im Bereich **Haltepunkte** können Sie Haltpunkte aktivieren und deaktivieren. Außerdem können Sie im Bereich **Haltepunkte** einen oder alle vorhandenen Haltepunkte löschen. Um den Bereich **Haltepunkte** anzuzeigen, klicken Sie auf **Ansicht > Debuggen > Haltepunkte**.

Um ein Skript mit Haltepunkten auszuführen, starten Sie die Ausführung zunächst wie gewohnt. VuGen unterbricht die Skriptausführung, wenn ein Haltepunkt erreicht wird. Sie können die Auswirkungen des Skripts bis zum Haltepunkt überprüfen, erforderliche Änderungen vornehmen und das Skript vom Haltepunkt aus erneut starten.

Wählen Sie **Wiedergeben > Ausführen** aus, um die Ausführung fortzusetzen. Nach dem erneuten Starten wird das Skript bis zum Erreichen des nächsten Haltepunkts oder bis zum Ende des Skripts fortgesetzt.

Beobachten von Ausdrücken und Variablen

Im Überwachungsbereich von VuGen können Sie Variablen überwachen, während ein Skript ausgeführt wird. Die Liste der Variablen, die Sie beobachten, wird als Überwachungsliste bezeichnet und im Überwachungsbereich angezeigt. Um den Überwachungsbereich anzuzeigen, klicken Sie auf **Ansicht > Debuggen > Beobachten**. Sie können der Überwachungsliste nur Variablen hinzufügen, nicht aber Ausdrücke. Sie können Variablen zur Überwachungsliste hinzufügen und diese bearbeiten oder löschen, indem Sie die Schaltflächen der Symbolleiste im Überwachungsbereich verwenden. Um die Spalten im Überwachungsbereich nach Ausdruck, Wert oder Typname zu sortieren, klicken Sie auf die Spaltenköpfe. Details zu weiteren Debugging-Funktionen von VuGen finden Sie unter "Debuggen – Übersicht" auf Seite 413.

Hinweis: Sie können Variablen nur überwachen, wenn die Ausführung eines Vuser-Skripts sich im Status **Pausiert** befindet.

Hinzufügen eines neuen Überwachungsausdrucks zur Überwachungsliste

Sie können einen neuen Überwachungsausdruck nur hinzufügen, wenn die Ausführung eines Vuser-Skripts sich im Status **Pausiert** befindet.

So fügen Sie einen neuen Überwachungsausdruck hinzu:

1. Klicken Sie auf **Ansicht > Debuggen > Beobachten**, um den Überwachungsbereich zu öffnen.
2. Klicken Sie auf die Schaltfläche **Überwachung hinzufügen** . Das Dialogfeld **Neue Überwachung hinzufügen** wird geöffnet.
3. Geben Sie in das Feld **Ausdruck** die Variable ein, die Sie beobachten möchten, und klicken Sie auf **OK**. VuGen fügt die Variable zur Liste der Ausdrücke in der Überwachungsliste hinzu.

Hinweis: Sie können der Überwachungsliste nur Variablen hinzufügen, nicht aber Ausdrücke.

Bearbeiten eines Überwachungsausdrucks

Hinweis: Sie können einen Überwachungsausdruck nur bearbeiten, wenn die Ausführung eines Vuser-Skripts sich im Status **Pausiert** befindet.

So bearbeiten Sie einen Überwachungsausdruck:

1. Klicken Sie auf **Ansicht > Debuggen > Beobachten**, um den Überwachungsbereich zu öffnen.

2. Wählen Sie in der Überwachungsliste den zu bearbeitenden Ausdruck aus und klicken Sie auf die Schaltfläche **Überwachungsausdruck bearbeiten** . Das Dialogfeld **Überwachung bearbeiten** wird geöffnet.

3. Ändern Sie im Feld **Ausdruck** die vorhandene Variable wie gewünscht und klicken Sie auf **OK**. VuGen zeigt die geänderte Variable in der Liste der Variablen in der Überwachungsliste an.

Löschen eines Überwachungsausdrucks

Hinweis: Sie können einen Überwachungsausdruck nur löschen, wenn die Ausführung eines Vuser-Skripts sich im Status **Pausiert** befindet.

So löschen Sie einen Überwachungsausdruck:

1. Klicken Sie auf **Ansicht > Debuggen > Beobachten**, um den Überwachungsbereich zu öffnen.

2. Wählen Sie im Überwachungsbereich den Ausdruck aus, den Sie löschen möchten, und klicken Sie auf die Schaltfläche **Überwachung löschen** . VuGen löscht den ausgewählten Ausdruck aus der Liste der Ausdrücke in der Überwachungsliste.

Löschen aller Überwachungsausdrücke aus der Überwachungsliste

Hinweis: Sie können Überwachungsausdrücke nur löschen, wenn die Ausführung eines Vuser-Skripts sich im Status **Pausiert** befindet.

So löschen Sie alle Überwachungsausdrücke aus der Überwachungsliste:

1. Klicken Sie auf **Ansicht > Debuggen > Beobachten**, um den Überwachungsbereich zu öffnen.

2. Klicken Sie auf die Schaltfläche **Alle Überwachungen löschen** . VuGen löscht alle Ausdrücke aus der Überwachungsliste.

Debuggen von Web-Vuser-Skripts

VuGen stellt zusätzlich ein Tool zum Debuggen von Web-Vuser-Skripts zur Verfügung – den Ergebnisübersichtsbericht.

Sie können festlegen, ob ein Web-Vuser während der Skriptausführung eine Ergebnisübersicht generieren soll. Der Ergebnisübersichtsbericht informiert über die erfolgreiche oder fehlerhafte Ausführung jedes Schritts in den Web-Vuser-Skripts und ermöglicht es Ihnen, die durch jeden Schritt zurückgegebene Website anzuzeigen. Wenn Sie weitere Informationen zum Arbeiten mit dem Ergebnisübersichtsbericht anzeigen möchten, wählen Sie **Wiedergeben > Testergebnisse** aus und drücken F1, um die Onlinehilfe zu öffnen.

Weitere Informationen zur Benutzeroberfläche erhalten Sie unter "Skriptoptionen" auf Seite 83.

Hinweis: Die Transaktionszeiten können sich verlängern, wenn ein Vuser einen Ergebnisübersichtsbericht generiert. Vuser können Ergebnisübersichtsberichte nur generieren, wenn das Skript über VuGen ausgeführt wird. Wenn Sie ein Skript über den Controller oder Business Process Monitor ausführen, generieren Vuser keine Berichte.

Debuggen von Skripts mit Haltepunkten

In den folgenden Schritten wird das Arbeiten mit Haltepunkten beschrieben. Weitere Informationen zu diesem Konzept finden Sie unter "Arbeiten mit Haltepunkten" auf Seite 418.

Hinzufügen eines Haltepunkts

So fügen Sie einen Haltepunkt hinzu:

Suchen Sie den Cursor in dem Skript, in dem Sie den Haltepunkt einfügen möchten, und führen Sie einen der folgenden Schritte aus:

- Wählen Sie **Wiedergeben > Haltepunkt umschalten**.
- Drücken Sie **F9**.
- Klicken Sie auf den linken Rand des Skripts neben der Stelle, an der Sie den Haltepunkt einfügen möchten.

Das Symbol **Haltepunkt** ( ) wird am linken Rand des Skripts angezeigt. VuGen markiert dann die Zeile in dem Skript.

Löschen eines Haltepunkts

So löschen Sie einen Haltepunkt:

Suchen Sie den Cursor in dem Skript, in dem Sie den Haltepunkt löschen möchten, und führen Sie einen der folgenden Schritte aus:

- Wählen Sie **Wiedergeben > Haltepunkt umschalten**.
- Drücken Sie **F9**.
- Klicken Sie auf das Haltepunktsymbol am linken Rand des Skripts.

Das Haltepunktsymbol ( ) wird aus dem linken Rand des Skripts entfernt.

Aktivieren/Deaktivieren eines Haltepunkts

So deaktivieren Sie einen Haltepunkt:

Klicken Sie auf **Ansicht > Debuggen > Haltepunkte**, um den Bereich für Haltepunkte anzuzeigen.

- Aktivieren Sie das entsprechende Kontrollkästchen **Aktiviert**, um einen Haltpunkt zu aktivieren. Am linken Rand des Skripts wird das Haltepunktsymbol ( ) angezeigt.

- Deaktivieren Sie das entsprechende Kontrollkästchen **Aktiviert**, um einen Haltpunkt zu deaktivieren. Am linken Rand des Skripts wird das Symbol **Haltepunkt deaktivieren** ( ) angezeigt.

Wenn ein Haltepunkt deaktiviert ist, wird die Skriptausführung am deaktivierten Haltepunkt fortgesetzt und am nächsten aktivierten Haltepunkt angehalten.

Verwalten von Haltepunkten

Der Bereich **Haltepunkte** bietet Ihnen die Möglichkeit, Haltepunkte in einem Vuser-Skript zu aktivieren, zu deaktivieren und Haltepunkte zu entfernen. Informationen zur Benutzeroberfläche finden Sie unter "Bereich "Haltepunkte"" auf Seite 119.

Navigieren zu einem bestimmten Haltepunkt in einem Vuser-Skript

Verwenden Sie eine der folgenden Vorgehensweisen, um in einem Vuser-Skript zu einem bestimmten Haltepunkt zu navigieren:

- Wählen Sie im Bereich **Haltepunkte** den Haltepunkt aus, zu dem Sie navigieren möchten, und klicken Sie auf die Schaltfläche **Gehe zu Quelle** .
- Doppelklicken Sie im Bereich **Haltepunkte** auf den Haltepunkt, zu dem Sie navigieren möchten.

Der Cursor blinkt im Editor am Anfang der Zeile, in der sich der Haltepunkt befindet.

Ausführen eines Skripts mit Haltepunkten

Starten Sie das Ausführen des Skripts wie gewohnt. VuGen unterbricht die Skriptausführung, wenn ein Haltepunkt erreicht wird. Sie können die Auswirkungen des Skripts bis zum Haltepunkt überprüfen, erforderliche Änderungen vornehmen und das Skript vom Haltepunkt aus erneut starten.

Wählen Sie **Wiedergeben > Ausführen** aus, um die Ausführung fortzusetzen. Nach dem erneuten Starten wird das Skript bis zum Erreichen des nächsten Haltepunkts oder bis zum Ende des Skripts fortgesetzt.

Erweitern eines Skripts für Lasttests

Im Abschnitt **Erweitern eines Skripts für Lasttests** werden die VuGen-Funktionen erläutert, mit denen Sie Vuser-Skripts erstellen können, die eine Last präzise erzeugen können. Zu diesen Funktionen können Parametrierung, Transaktionen und Rendezvous-Punkte zählen.

Erweitern eines Skripts für Lasttests – Übersicht

Das Erstellen eines Vuser-Skripts umfasst die nachfolgend aufgeführten Schritte. Dieses Thema enthält eine Übersicht über den sechsten Schritt (Anzeigen der Ergebnisse der Wiedergabe eines Vuser-Skripts).

In dieser Aufgabe werden weitere Aktionen beschrieben, die Sie in einem Vuser-Skript ausführen können, um das Skript für Lasttests vorzubereiten. Alle Aktionen in dieser Aufgabe sind optional.

Hinzufügen von Parametrierung

Wenn Sie einen Geschäftsprozess aufzeichnen, erzeugt VuGen ein Skript, das die während der Aufzeichnung verwendeten Ist-Werte enthält. Angenommen, Sie möchten die Skriptaktionen (Abfrage, Eingabe usw.) nicht mit den aufgezeichneten, sondern mit anderen Werten ausführen. Dazu ersetzen Sie die aufgezeichneten Werte mit Parametern. Dieser Vorgang wird als *Parametrieren* des Skripts bezeichnet. Weitere Informationen finden Sie unter "Parametrierung – Übersicht" auf Seite 436.

Einfügen von Transaktionen

Sie können Transaktionen in ein Vuser-Skript während oder nach der Aufzeichnung des Skripts einfügen. Weitere Informationen über die Aufgabe finden Sie unter "Einfügen von Transaktionen" auf Seite 425.

Einfügen von Rendezvous-Punkten

Sie können Vuser synchronisieren, um eine Task genau in demselben Moment durchzuführen, indem Sie einen Rendezvous-Punkt erstellen. Wenn ein Vuser am Rendezvous-Punkt ankommt, wartet er, bis alle Vuser, die am Rendezvous teilnehmen, eintreffen. Wenn die angegebene Anzahl der Vuser ankommen, werden die Vuser freigegeben.

Sie können Rendezvous-Punkte mithilfe einer der folgenden Methoden einfügen:

- Zum Einfügen eines Rendezvous-Punkts während der Aufzeichnung klicken Sie auf die Schaltfläche **Rendezvous** in der Aufzeichnungssymbolleiste und geben einen Namen in das Dialogfeld ein (Groß-/Kleinschreibung wird nicht beachtet).
- Zum Einfügen eines Rendezvous-Punkts nach der Aufzeichnung wählen Sie **Entwurf > In Skript einfügen > Rendezvous** aus und geben einen Namen für den Rendezvous-Punkt ein (Groß-/Kleinschreibung wird nicht beachtet).

Nach dem Einfügen eines Rendezvous-Punkts fügt VuGen eine Funktion **lr_rendezvous** in das Vuser-Skript ein. Beispielsweise definiert die folgende Funktion einen Rendezvous-Punkt mit dem Namen **rendezvous1**:

```
lr_rendezvous("rendezvous1");
```

Weitere Informationen zu diesem Konzept finden Sie unter "Rendezvous-Punkte " auf Seite 427.

Einfügen von VuGen-Funktionen

An dieser Stelle können Sie VuGen-Funktionen einfügen. Eine Liste mit nützlichen Funktionen finden Sie unter "Übersicht über das Hinzufügen von VuGen-Funktionen" auf Seite 428.

Einfügen von Schritten

Sie können eine Vielzahl von Schritten in Ihr Skript einfügen, wie beispielsweise Latenzzeit-Schritte, Debugmeldungen und Ausgabemeldungen. Weitere Informationen über die Aufgabe finden

Sie unter "Einfügen von Schritten in ein Skript" auf Seite 434.

Einfügen von Kommentaren

VuGen ermöglicht es, Kommentare zwischen Vuser-Aktivitäten einzufügen. Sie können einen Kommentar einfügen, um eine Aktivität zu beschreiben oder Informationen über eine spezifische Operation bereitzustellen. Wenn Sie beispielsweise Datenbankaktionen aufzeichnen, können Sie einen Kommentar einfügen, um die erste Abfrage zu kennzeichnen, wie z. B. "Dies ist die erste Abfrage."

Sie können einen Kommentar mithilfe einer der folgenden Methoden einfügen:

- Zum Einfügen eines Kommentars während der Aufzeichnung klicken Sie auf die Schaltfläche **Kommentar einfügen** in der Aufzeichnungssymbolleiste und geben den gewünschten Kommentar in das Dialogfeld **Kommentar einfügen** ein.
- Zum Einfügen eines Kommentars nach der Aufzeichnung wählen Sie **Entwurf > In Skript einfügen > Kommentar** aus und geben den Kommentar ein.

Das folgende Skriptsegment zeigt, wie ein Kommentar in einem Vuser-Skript erscheint:

```
/* <Kommentare> */
```

Einfügen von Protokollmeldungen

Sie können VuGen verwenden, um **lr_log_message**-Funktionen zu generieren und in ein Vuser-Skript einzufügen. Wenn Sie beispielsweise Datenbankaktionen aufzeichnen, können Sie eine Meldung zur Angabe der ersten Abfrage einfügen, wie z. B. "Dies ist die erste Abfrage".

Zum Einfügen einer Protokollmeldung wählen Sie **Entwurf > In Skript einfügen > Meldung protokollieren** aus und geben die Meldung ein.

Einfügen von Synchronisierungspunkten (nur RTE-Vuser)

Sie können Synchronisierungsfunktionen hinzufügen, um die Ausführung des Vuser-Skripts mit der Ausgabe von Ihrer Anwendung zu synchronisieren. Synchronisierung ist nur für RTE-Vuser-Skripts anwendbar.

Die nachfolgende Liste zeigt die verfügbaren Synchronisierungsfunktionen:

Funktion	Beschreibung
TE_wait_cursor	Wartet, bis der Cursor an einer angegebenen Position im Terminalfenster erscheint.
TE_wait_silent	Wartet, bis die Clientanwendung für eine festgelegte Anzahl von Sekunden inaktiv ist.
TE_wait_sync	Wartet, bis das System aus dem Modus X-SYSTEM oder "Input Inhibited" zurückkehrt.

TE_wait_text	Wartet, bis eine Zeichenfolge an einer angegebenen Position erscheint.
TE_wait_sync_ transaction	Zeichnet die Zeit auf, die das System im letzten X-SYSTEM-Modus verblieb.

Weitere Informationen über die Synchronisierung in RTE-Vuser-Skripts finden Sie unter "Übersicht über die RTE-Synchronisierung" auf Seite 840.

Übersicht über Transaktionen

Sie definieren *Transaktionen*, um die Serverleistung zu messen. Jede Transaktion misst die Zeit, die der Server benötigt, um die spezifischen Vuser-Anfragen zu beantworten. Bei diesen Anfragen kann es sich um einfache Aufgaben handeln, wie z. B. Warten auf eine Antwort für eine einfache Abfrage, oder um komplexe Aufgaben, wie z. B. Senden von mehreren Abfragen und Generieren eines Berichts.

Zum Messen einer Transaktion fügen Sie Vuser-Funktionen ein, um den Anfang und das Ende einer Aufgabe zu markieren. Innerhalb eines Skripts können Sie eine unbegrenzte Anzahl von Transaktionen jeweils mit einem unterschiedlichen Namen markieren.

Für LoadRunner misst der Controller die zur Durchführung jeder Transaktion erforderliche Zeit. Nach dem Testlauf analysieren Sie die Leistung des Servers pro Transaktion mithilfe der Analysis-Diagramme und Berichte.

Vor dem Erstellen eines Skripts müssen Sie festlegen, welche Geschäftsprozesse Sie messen möchten. Anschließend markieren Sie jeden Geschäftsprozess oder Unterprozess als eine Transaktion.

Transaktionsnamen dürfen nicht die Zeichen "_" oder "@" enthalten. Dies würde bei dem Versuch, das Analysis-Diagramm für vergleichende Ergebnisse zu öffnen, zu Fehlern führen.

Sie können Transaktionen entweder während oder nach der Aufzeichnung erstellen. Weitere Informationen über die Aufgabe finden Sie unter "Einfügen von Transaktionen" unten.

Einfügen von Transaktionen

Sie können eine Transaktion in ein Vuser-Skript während oder nach der Aufzeichnung des Skripts einfügen. In den folgenden Schritten werden verschiedene Methoden für das Einfügen von Transaktionen beschrieben. Hintergrundinformationen finden Sie unter "Übersicht über Transaktionen" oben.

Einfügen einer Transaktion nach der Aufzeichnung

Sie können den VuGen-Editor verwenden, um eine Transaktion nach der Aufzeichnung eines Vuser-Skripts einzufügen.

1. Um den Start einer Transaktion zu markieren, suchen Sie den Cursor in dem Skript, in dem Sie die Transaktionen starten möchten, und führen Sie dann eine der folgenden Aktionen aus:

- Wählen Sie **Entwurf > In Skript einfügen > Transaktion starten** aus.
- Drücken Sie **STRG+T**.
- Klicken Sie im Skript mit der rechten Maustaste auf die Stelle, an der die Transaktion starten soll, und wählen Sie **Einfügen > Transaktion starten** aus.

VuGen fügt eine Anweisung **lr_start_transaction** in das Vuser-Skript ein. Geben Sie in den neuen Schritt einen Transaktionsnamen ein.

2. Um das Ende einer Transaktion zu markieren, suchen Sie den Cursor in dem Skript, in dem Sie die Transaktionen beenden möchten, und führen Sie dann eine der folgenden Aktionen aus:
   - Wählen Sie **Entwurf > In Skript einfügen > Transaktion beenden** aus.
   - Drücken Sie **STRG+UMSCHALT+T**.
   - Klicken Sie im Skript mit der rechten Maustaste auf die Stelle, an der die Transaktion beendet werden soll, und wählen Sie **Einfügen > Transaktion beenden** aus.

   VuGen fügt eine Anweisung **lr_end_transaction** in das Vuser-Skript ein. Geben Sie in den neuen Schritt einen Transaktionsnamen ein.
3. So können Sie den Start und das Ende einer Transaktion gleichzeitig markieren:
   a. Wählen Sie die Schritte aus, die Sie in die Transaktion einbeziehen möchten.
   b. Wählen Sie **Entwurf > In Skript einfügen > Mit Transaktion umgeben** aus, oder klicken Sie mit der rechten Maustaste auf die Auswahl, und wählen Sie **Mit Transaktion umgeben** aus, oder drücken Sie **UMSCHALT+STRG+I**.

      Das Dialogfeld "Mit Transaktion umgeben" wird geöffnet.
   c. Geben Sie einen Namen für die Transaktion ein, und klicken Sie dann auf **OK**.

   VuGen fügt vor dem ersten ausgewählten Schritt eine Anweisung **lr_start_transaction** und nach dem letzten ausgewählten Schritt eine Anweisung **lr_end_transaction** ein.

Einfügen einer Transaktion während der Aufzeichnung

- Klicken Sie zum Markieren des Transaktionsstarts in der Aufzeichnungssymbolleiste auf die Schaltfläche **Transaktion starten** , geben Sie einen Transaktionsnamen ein, und klicken Sie auf **OK**. Wenn das Skript generiert ist, fügt VuGen eine Anweisung **lr_start_transaction** in das Vuser-Skript ein.
- Um das Ende einer Transaktion zu markieren, klicken Sie auf die Schaltfläche **Transaktion beenden** auf der Aufzeichnungssymbolleiste und wählen die zu schließende Transaktion aus. Wenn das Skript generiert ist, fügt VuGen eine Anweisung **lr_end_transaction** in das Vuser-Skript ein.

Hinweise:

- Sie können *verschachtelte* Transaktionen - Transaktionen innerhalb von Transaktionen erstellen. Wenn Sie Transaktionen verschachteln, schließen Sie die inneren Transaktionen vor dem Schließen der äußeren Transaktionen. Andernfalls werden die Transaktionen nicht korrekt analysiert. Die Transaktionen müssen sich jedoch in einem einzelnen **Aktions**abschnitt befinden.
- Transaktionsnamen müssen eindeutig sein und können Buchstaben und Zahlen enthalten. Folgende Zeichen sind nicht zulässig: . , : # / \ " <.
- Eine fehlgeschlagene Transaktion setzt den Wiedergabestatus des Skripts nicht automatisch auf **Fehlgeschlagen**.

Anzeigen von Transaktionen

In den nachfolgenden Schritten wird beschrieben, wie die unterschiedlichen Transaktionstypen über den Aufgabenbereich angezeigt werden können. Hintergrundinformationen finden Sie unter "Übersicht über Transaktionen" auf Seite 425.

Anzeigen von verborgenen Transaktionen

Zum Anzeigen der ausgeblendeten Transaktionen, d. h. der nicht primären Transaktionen und Transaktionen auf Client-Seite, klicken Sie auf die Schaltfläche neben **Ausgeblendete Transaktionen anzeigen** am Ende der Transaktionsliste. VuGen listet die verborgenen Transaktionen grau auf. Um sie auszublenden, klicken Sie erneut auf die Schaltfläche.

Anzeigen von Transaktionen mit Fehlern

Transaktionen mit Fehlern sind die Transaktionen, die keine Serverschritte messen, oder Transaktionen mit unzulässigen Namen. Zum Anzeigen der Transaktionen mit Fehlern klicken Sie auf die Schaltfläche **Transaktionen mit Fehlern anzeigen**. VuGen listet die Transaktionen mit Fehlern rot auf. Um sie auszublenden, klicken Sie erneut auf die Schaltfläche.

Anzeigen von Transaktionen für nicht primäre Schritte

Zum Anzeigen der Transaktionen für nicht primäre Schritte müssen Sie alle Miniaturansichten anzeigen. Wählen Sie **Ansicht > Alle Miniaturansichten anzeigen** aus.

Rendezvous-Punkte

Bei der Ausführung des Lasttests müssen Sie große Benutzerlasten auf Ihrem System emulieren. Hierfür synchronisieren Sie Vuser, um eine Task genau in demselben Moment durchzuführen. Sie konfigurieren mehrere Vuser für das gleichzeitige Handeln, indem Sie einen Rendezvous-Punkt erstellen. Wenn ein Vuser am Rendezvous-Punkt ankommt, wartet er, bis alle Vuser, die am Rendezvous teilnehmen, eintreffen. Wenn die angegebene Anzahl der Vuser ankommen, werden die Vuser freigegeben.

Sie legen den Treffpunkt fest, indem Sie einen Rendezvous-Punkt in Ihr Vuser-Skript eingeben. Wenn ein Vuser ein Skript ausführt und den Rendezvous-Punkt trifft, wird die Skriptausführung angehalten, und der Vuser wartet auf die Genehmigung vom Controller, um fortzufahren. Nachdem ein Vuser vom Rendezvous freigegeben wurde, führt er die nächste Aufgabe im Skript aus.

Weitere Informationen über die Aufgabe finden Sie unter "Erweitern eines Skripts für Lasttests – Übersicht" auf Seite 422.

Hinweis: Rendezvous-Punkte sind nur in action-Abschnitten gültig, nicht in init- oder end-Abschnitten.

Übersicht über das Hinzufügen von VuGen-Funktionen

In diesem Abschnitt werden nützliche VuGen-Funktionen erläutert, die Sie hinzufügen können, während Sie das Skript debuggen oder für die Lasttests vorbereiten.

Abrufen von Vuser-Informationen

Sie können zum Abrufen von Vuser-Informationen die folgenden Funktionen zu Ihren Vuser-Skripts hinzufügen:

Funktion	Beschreibung
lr_get_ attrib_string	Gibt eine Befehlszeilen-Parameterzeichenfolge zurück.
lr_get_host_ name	Gibt den Namen des Computers zurück, auf dem das Vuser-Skript ausgeführt wird.
lr_get_ master_ host_name	Gibt den Namen des Computers zurück, auf dem der Controller ausgeführt wird. Nicht anwendbar beim Arbeiten mit dem HP Business Service Management.
lr_whoami	Gibt den Namen eines Vusers zurück, der das Skript ausführt. Nicht anwendbar beim Arbeiten mit dem HP Business Service Management.

Im folgenden Beispiel ruft die Funktion **lr_get_host_name** den Namen des Computers ab, auf dem Vuser ausgeführt wird.

```
my_host = lr_get_host_name( );
```

Weitere Informationen zu den o. g. Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Senden von Meldungen an das Ausgabefenster

Über die Meldungstypfunktionen in Ihrem Vuser-Skript können Sie angepasste Fehler- und Benachrichtungsmeldungen an die Ausgabe- und Protokolldateien sowie an die Zusammenfassung des Testberichts senden. Beispielsweise können Sie eine Meldung einfügen, die den aktuellen Status der Clientanwendung anzeigt. Der LoadRunner-Controller zeigt diese Meldungen im Ausgabefenster an. Sie können diese Meldungen auch in einer Datei speichern.

Beim Arbeiten mit HP Business Service Management können Sie Funktionen des Meldungstyps verwenden, um Fehler- und Benachrichtungsmeldungen an die Website oder an Business Process Monitor-Protokolldateien zu senden. Beispielsweise können Sie eine Meldung einfügen, die den aktuellen Status der webbasierten Anwendung anzeigt.

Hinweis: Senden Sie keine Meldungen von innerhalb einer Transaktion aus, denn dies kann die Ausführungszeit der Transaktion verlängern und die Transaktionsergebnisse verfälschen.

Sie können die folgenden Meldungsfunktionen in Ihren Vuser-Skripts verwenden:

Funktion	Beschreibung
lr_debug_message	Sendet eine Debugmeldung an das Ausgabefenster oder die Business Process Monitor-Protokolldatei.
lr_error_message	Sendet eine Fehlermeldung an das Ausgabefenster, den Testergebnisbericht oder die Business Process Monitor-Protokolldateien.
lr_get_debug_message	Ruft die aktuelle Meldungsklasse ab.
lr_log_message	Sendet eine Ausgabemeldung direkt an die Protokolldatei *output.txt*, die sich im Vuser-Skriptordner befindet. Diese Funktion ist nützlich, um zu vermeiden, dass Ausgabemeldungen den TCP/IP-Datenverkehr beeinträchtigen.
lr_output_message	Sendet eine Meldung an das Ausgabefenster, den Testergebnisbericht oder die Business Process Monitor-Protokolldateien.
lr_set_debug_message	Legt eine Meldungsklasse für Ausgabemeldungen fest.
lr_vuser_status_message	Sendet eine Meldung an den Vuser-Statusbereich im Controller. Nicht anwendbar beim Arbeiten mit dem HP Business Service Management.
lr_message	Sendet eine Meldung an das Vuser-Protokoll und das Ausgabefenster oder die Business Process Monitor-Protokolldateien.

Das Verhalten der Funktionen **lr_message**, **lr_output_message** und **lr_log_message** ist von der Debugging-Ebene des Skripts in den Protokoll-Laufzeiteinstellungen nicht betroffen, d. h. Meldungen werden von diesen Funktionen immer gesendet.

Über die Funktionen **lr_output_message** und **lr_error_message** können Sie auch aussagekräftige Meldungen an den Zusammenfassungsbericht für Testergebnisse senden. Weitere Informationen siehe "Anzeigen von Wiedergabeergebnissen" auf Seite 513.

Allgemeine Vuser-Funktionen

Die allgemeinen Vuser-Funktionen werden auch als LR-Funktionen bezeichnet, da alle das Präfix **lr** aufweisen. LR-Funktionen können in jedem Vuser-Skripttyp eingesetzt werden. Die LR-Funktionen ermöglichen Ihnen Folgendes:

- Abrufen von Laufzeitinformationen zum Vuser, der zugehörigen Vuser-Gruppe und dem Host.
- Hinzufügen von Transaktionen und Synchronisierungspunkten zu einem Vuser-Skript. So kennzeichnet die Funktion **lr_start_transaction** (**lr.start_transaction** in Java) beispielsweise den Anfang und die Funktion **lr_end_transaction** (**lr.end_transaction** in Java) das Ende einer Transaktion. Weitere Informationen finden Sie unter "Erweitern eines Skripts für Lasttests" auf Seite 422.
- Senden von Meldungen an die Ausgabe, um einen Fehler oder eine Warnung anzuzeigen.

Weitere Informationen finden Sie in der *Function Reference* (**Hilfe > Funktionsreferenz**).

Protokollspezifische Vuser-Funktionen

Zusätzlich zu den allgemeinen Vuser-Funktionen erzeugt VuGen auch protokollspezifische Funktionen und fügt sie bei der Aufzeichnung in das Vuser-Skript ein.

Die protokollspezifischen Funktionen sind dem jeweils aufgezeichneten Vuser-Typ eigen. VuGen fügt beispielsweise LRS-Funktionen in ein Windows Sockets-Skript ein.

Standardmäßig erzeugt der automatische Skriptgenerator von VuGen für die meisten Protokolle Vuser-Skripts in C und Java-Protokolle in Java. Sie können VuGen anweisen, Code in Visual Basic oder Javascript zu erzeugen. Weitere Informationen finden Sie unter "Allgemein > Skript (Knoten)" auf Seite 196.

Für Skripts gelten alle Standardkonventionen, einschließlich Kontrollstrukturen und Syntax. Sie können dem Skript wie in anderen Programmiersprachen Kommentare und bedingte Anweisungen hinzufügen.

Das folgende Segment aus einem Web-Vuser-Skript stellt mehrere Funktionen dar, die von VuGen aufgezeichnet und in einem Skript erzeugt wurden:

```
#include "as_web.h"
Action1()
{
    web_add_cookie("nav=140; DOMAIN=dogbert");
    web_url("dogbert",
        "URL=http://dogbert/",
         "RecContentType=text/html",
        LAST);
    web_image("Library",
        "Alt=Library",
        LAST);
```

```
    web_link("1 Book Search:",
        "Text=1 Book Search:",
        LAST);
    lr_start_transaction("Purchase_Order");
...
```

Weitere Informationen über C-Funktionen in Ihren Vuser-Skripts finden Sie unter Function Reference (Hilfe > Funktionsreferenz). Weitere Informationen zum Ändern eines Java-Skripts finden Sie unter "Java-Vuser-Protokoll - Manuelles Programmieren von Skripts" auf Seite 730.

Hinweis: Der zum Ausführen von in C geschriebenen Vuser-Skripts verwendete C-Interpreter unterstützt ANSI C. Er unterstützt jedoch nicht die Microsoft-Erweiterungen zu ANSI C.

Verschlüsseln von Text

Sie können Text in Ihrem Skript verschlüsseln, um Ihre Kennwörter und andere vertrauliche Textzeichenfolgen zu schützen. Sie können die Verschlüsselung sowohl automatisch über die Benutzerschnittstelle als auch manuell über eine Programmierung durchführen. Sie können die Zeichenfolge jederzeit wiederherstellen, um ihren ursprünglichen Wert zu ermitteln. Beim Verschlüsseln einer Zeichenfolge erscheint diese als codierte Zeichenfolge im Skript. VuGen verwendet die 32-Bit-Verschlüsselung.

Damit das Skript einen verschlüsselte String verwenden kann, muss die Verschlüsselung mit **lr_decrypt** erfolgen.

```
lr_start_transaction(lr_decrypt("3c29f4486a595750"));
```

Weitere Informationen über die Aufgabe finden Sie unter "Verschlüsseln/Entschlüsseln von Text" auf der nächsten Seite.

Codieren von Kennwörtern

Sie können Kennwörter codieren, um die resultierenden Zeichenfolgen als Argumente in Ihrem Skript oder in Parameterwerten zu verwenden. Beispielsweise kann Ihre Website ein Formular enthalten, in dem der Benutzer ein Kennwort angeben muss. Möglicherweise möchten Sie testen, wie Ihre Site auf verschiedene Kennwörter reagiert, und zugleich soll die Integrität der Kennwörter geschützt werden. Mit dem **Password Encoder** können Sie Ihre Kennwörter codieren und sichere Werte in die Tabelle einfügen.

Wählen Sie zum Codieren eines Kennworts **Start > Alle Programme > HP Software > HP LoadRunner > Extras > Password Encoder**.

Weitere Informationen über die Aufgabe finden Sie unter "Verschlüsseln eines Kennworts" auf der nächsten Seite.

Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Password Encoder"" auf Seite 433.

Verschlüsseln von Text

Sie können Text in Ihrem Skript verschlüsseln, um Ihre Kennwörter und andere vertrauliche Textzeichenfolgen zu schützen. Sie können die Verschlüsselung sowohl automatisch über die Benutzerschnittstelle als auch manuell über eine Programmierung durchführen. Sie können die Zeichenfolge jederzeit wiederherstellen, um ihren ursprünglichen Wert zu ermitteln. Beim Verschlüsseln einer Zeichenfolge erscheint diese als codierte Zeichenfolge im Skript. VuGen verwendet die 32-Bit-Verschlüsselung.

Damit das Skript einen verschlüsselte String verwenden kann, muss die Verschlüsselung mit **lr_decrypt** erfolgen.

```
lr_start_transaction(lr_decrypt("3c29f4486a595750"));
```

Weitere Informationen über die Aufgabe finden Sie unter "Verschlüsseln/Entschlüsseln von Text" unten.

Verschlüsseln/Entschlüsseln von Text

In dieser Aufgabe wird beschrieben, wie Zeichenfolgen in Ihrem Code ver- und entschlüsselt werden. Hintergrundinformationen finden Sie unter "Verschlüsseln von Text" oben.

Verschlüsseln einer Zeichenfolge

1. Wählen Sie den Text aus, den Sie verschlüsseln möchten.
2. Wählen Sie **Zeichenfolge verschlüsseln (***Zeichenfolge***)** im Kontextmenü aus.

Wiederherstellen einer verschlüsselten Zeichenfolge

1. Wählen Sie den Text aus, den Sie wiederherstellen möchten.
2. Wählen Sie **Verschlüsselte Zeichenfolge wiederherstellen (***Zeichenfolge***)** im Kontextmenü aus.

   Weitere Informationen zur Funktion **lr_decrypt** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Verschlüsseln eines Kennworts

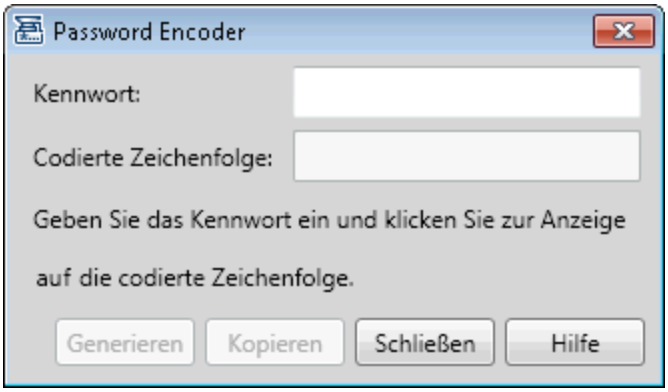
In dieser Aufgabe wird beschrieben, wie ein Kennwort verschlüsselt wird. Sie können Kennwörter codieren, um die resultierenden Zeichenfolgen als Argumente in Ihrem Skript oder in Parameterwerten zu verwenden. Beispielsweise kann Ihre Website ein Formular enthalten, in dem der Benutzer ein Kennwort angeben muss. Möglicherweise möchten Sie testen, wie Ihre Site auf verschiedene Kennwörter reagiert, und zugleich soll die Integrität der Kennwörter geschützt werden.

Verschlüsseln eines Kennworts

1. Wählen Sie im Windows-Menü **Start > Programme > HP Software > HP LoadRunner > Extras > Password Encoder**. Das Dialogfeld **Password Encoder** wird geöffnet.
2. Geben Sie das Kennwort in das Feld **Kennwort** ein.
3. Klicken Sie auf **Generieren**. Der Password Encoder verschlüsselt das Kennwort und zeigt es im Feld **Codierte Zeichenfolge** an.
4. Verwenden Sie die Schaltfläche **Kopieren**, um den verschlüsselten Wert zu kopieren und in die Datentabelle einzufügen.

Dialogfeld "Password Encoder"

In diesem Dialogfeld können Sie verschlüsselte Kennwörter generieren.



Zugriff	Wählen Sie **Start > Alle Programme > HP Software > HP LoadRunner > Tools > Password Encoder** aus.
Relevante Aufgaben	"Verschlüsseln eines Kennworts" auf der vorherigen Seite
Siehe auch	"Codieren von Kennwörtern" auf Seite 431

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Kopieren	Kopiert die Ergebnisse aus dem Feld **Codierte Zeichenfolge** und fügt Sie in die Datentabelle ein, die Ihre Parameterliste enthält.
Codierte Zeichenfolge	Die verschlüsselten Ergebnisse werden hier angezeigt.

Forts.

Generieren	Klicken Sie auf diese Schaltfläche, um das verschlüsselte Kennwort zu erstellen.
Kennwort	Geben Sie hier das zu verschlüsselnde Kennwort ein.

Erstellen eines Controller-Szenarios über VuGen

Hinweis: Der folgende Abschnitt gilt nur für LoadRunner. Informationen zum Integrieren von Skripts in Geschäftsprozessprofile enthält die Dokumentation HP Business Service Management.

Normalerweise erstellen Sie ein Szenario über den LoadRunner-Controller. Sie können auch ein Basisszenario über VuGen mit dem aktuellen Skript erstellen.

Zum Erstellen dieses Szenariotyps wählen Sie **Extras** > **Controller-Szenario erstellen** aus und vervollständigen das Dialogfeld. Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Controller-Szenario erstellen"" auf der nächsten Seite.

Weitere Informationen finden Sie im HP Controller-Benutzerhandbuch.

Einfügen von Schritten in ein Skript

In den nachfolgenden Schritten wird beschrieben, wie die unterschiedlichen Typen von Schritten zu einem Vuser-Skript hinzugefügt werden.

Einfügen von Latenzzeit-Schritten

Die Zeit, die ein Benutzer zwischen dem Ausführen von aufeinander folgenden Aktionen wartet, wird als die *Latenzzeit* bezeichnet. Vuser verwenden die Funktion **lr_think_time** zur Emulation der Latenzzeit von realen Benutzern. Beim Aufzeichnen eines Vuser-Skripts zeichnet VuGen die tatsächliche Latenzzeit auf und fügt relevante **lr_think_time**-Anweisungen in das Vuser-Skript ein. Sie können die aufgezeichneten **lr_think_time**-Anweisungen bearbeiten und manuell weitere **lr_think_time**-Anweisungen zu einem Vuser-Skript hinzufügen.

Zum Hinzufügen eines Latenzzeit-Schritts wählen Sie **Entwurf > In Skript einfügen > Neuer Schritt > Latenzzeit** aus, und geben Sie die gewünschte Latenzzeit in Sekunden an.

Hinweis: Beim Aufzeichnen eines Java-Vuser-Skripts werden die **lr_think_time**-Anweisungen nicht im Vuser-Skript generiert.

Sie können die Laufzeiteinstellungen verwenden, um zu beeinflussen, wie sich die **lr_think_time**-Anweisungen beim Ausführen eines Vuser-Skripts verhalten. Informationen zur Benutzeroberfläche finden Sie unter "Allgemein > Latenzzeit (Knoten)" auf Seite 368.

Einfügen von Debugmeldungen

Sie können eine Debug- oder Fehlermeldung über die VuGen-Benutzerschnittstelle hinzufügen. Für

Debugmeldungen können Sie die Ebene der Textmeldung angeben. Die Textmeldung wird dann nur ausgegeben, wenn die angegebene Ebene mit der Meldungsklasse übereinstimmt. Sie legen die Meldungsklasse über **lr_set_debug_message** fest.

Zum Einfügen einer Debugmeldung wählen Sie **Entwurf > In Skript einfügen > Neuer Schritt > Debugmeldung** aus, und füllen Sie das Dialogfeld aus. Informationen zur Benutzeroberfläche finden Sie unter Dialogfeld "Debugmeldung".

Einfügen von Fehler- und Ausgabemeldungen

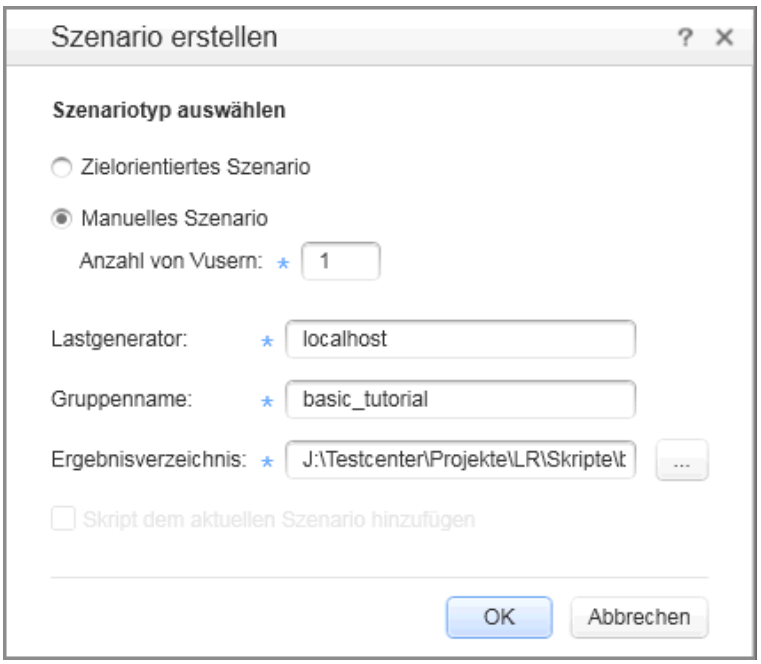
Für Protokolle, die den **Schrittenavigator** unterstützen, wie z. B. Web, Winsock und Oracle NCA, können Sie eine Fehler- oder Ausgabemeldung über die Benutzerschnittstelle hinzufügen. Diese Funktion wird häufig verwendet, um eine bedingte Anweisung einzufügen und eine Meldung auszugeben, wenn die Fehlerbedingung ermittelt wird.

Zum Einfügen einer Fehler- oder Ausgabemeldung wählen Sie **Entwurf > In Skript einfügen** > Neuer Schritt > **Fehlermeldung** oder **Ausgabemeldung** aus, und geben Sie die Meldung ein. Eine Funktion **lr_error_message** oder **lr_output_message** wird an der aktuellen Stelle in das Skript eingefügt.

Hinweis: Ein Fehlermeldungsschritt in einem Skript setzt den Wiedergabestatus nicht automatisch auf **Fehlgeschlagen**.

Dialogfeld "Controller-Szenario erstellen"

Mithilfe dieses Dialogfelds können Sie über VuGen ein grundlegendes Controller-Szenario erstellen.



Zugriff	**VuGen > Extras > Controller-Szenario erstellen**
Relevante Aufgaben	"Erstellen eines Controller-Szenarios über VuGen" auf Seite 434

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Skript dem aktuellen Szenario hinzufügen	Wenn zurzeit ein Szenario im Controller geöffnet ist und Sie das Skript zu diesem Szenario hinzufügen möchten, aktivieren Sie dieses Kontrollkästchen. Wenn Sie das Kontrollkästchen deaktivieren, öffnet LoadRunner ein neues Szenario mit der festgelegten Anzahl an Vusern.
Gruppenname	Für ein manuelles Szenario sind Benutzer mit gemeinsamen Merkmalen in Gruppen zusammengefasst. Geben Sie einen neuen Gruppennamen für die Vuser an.
Lastgenerator	Der Name des Computers, auf dem das Szenario ausgeführt wird.
Ergebnisverzeichnis	Ermöglicht das Eingeben des gewünschten Speicherorts für die Ergebnisse.
Skriptname	Für ein zielorientiertes Szenario geben Sie einen Namen ein.
Szenariotyp auswählen	• **Zielorientiertes Szenario.** LoadRunner erstellt automatisch ein Szenario basierend auf den angegebenen Zielen. • **Manuelles Szenario.** Das Szenario wird manuell erstellt, indem die Anzahl der ausgeführten Vuser angegeben wird.

Parameter

Im Abschnitt **Parameter** wird beschrieben, wie Sie Parameter einfügen, definieren und ändern können.

Parametrierung – Übersicht

Wenn Sie einen Geschäftsprozess aufzeichnen, erzeugt VuGen ein Skript, das die während der Aufzeichnung verwendeten Ist-Werte enthält. Angenommen, Sie möchten die Skriptaktionen (Abfrage, Eingabe usw.) nicht mit den aufgezeichneten, sondern mit anderen Werten ausführen. Dazu ersetzen Sie die aufgezeichneten Werte mit Parametern. Dieser Vorgang wird als *Parametrieren* des Skripts bezeichnet.

Die Vuser ersetzen den Parameter durch Werte aus einer von Ihnen angegebenen Datenquelle. Die Datenquelle kann eine Datei oder eine intern erzeugte Variable sein. Weitere Informationen finden Sie unter "Parametertypen" auf Seite 439.

Parameter werden in einem Vuser-Skript innerhalb spezieller Trennzeichen angegeben. Standardmäßig verwendet VuGen "{" und "}" als linkes bzw. rechtes Trennzeichen, Sie können bei Bedarf aber auch andere Trennzeichen festlegen. Außerdem können Sie die Hintergrund- und Umrissfarbe für Parameter in einem Skript anpassen. Weitere Informationen finden Sie unter "Skriptoptionen" auf Seite 83.

Beispiel für Trennzeichen

Skriptabschnitt wie aufgezeichnet.

"value=UNIX"

Skriptabschnitt nachdem "UNIX" durch den Parameter "Operating System" ersetzt wurde.

"value={Operating System}"

Eine Parametrierung ist nur für die Argumente einer Funktion möglich. Textfolgen, die keine Funktionsargumente darstellen, können nicht parametriert werden. Außerdem können nicht alle Funktionsargumente parametriert werden. Weitere Informationen zu parametrierbaren Argumenten finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**) zu der jeweiligen Funktion.

Eingabeparameter sind Parameter, deren Werte von Ihnen in der Entwurfsphase definiert werden, bevor das Skript ausgeführt wird. Ausgabeparameter werden in der Entwurfsphase definiert und bei Ausführung des Tests mit Werten angereichert. Ausgabeparameter werden häufig in Verbindung mit Webdienstaufrufen eingesetzt. Wählen Sie die Parameter für Ihr Skript in der Entwurfsphase sorgfältig aus und stellen Sie sicher, dass es sich nicht um einen leeren Ausgabeparameter handelt.

Beispiel

Angenommen, Sie haben während der Ausführung einer Webanwendung ein Vuser-Skript aufgezeichnet. VuGen erzeugt die folgende Anweisung, die in der Datenbank einer Bibliothek nach dem Titel "UNIX" sucht:

```
web_submit_form("db2net.exe",
       ITEMDATA,
       "name=library.TITLE",
        "value=UNIX",
       ENDITEM,
       "name=library.AUTHOR",
       "value=",
       ENDITEM,
       "name=library.SUBJECT",
       "value=",
       ENDITEM,
       LAST);
```

Bei der Wiedergabe des Skripts mit mehreren Vusern und Iterationen soll jedoch nicht immer der gleiche Wert (UNIX) verwendet werden. Stattdessen ersetzen Sie die Konstante durch einen Parameter.

```
web_submit_form("db2net.exe",
        ITEMDATA,
        "name=library.TITLE",
         "value={Book_Title}",
        ENDITEM,
        "name=library.AUTHOR",
        "value=",
        ENDITEM,
        "name=library.SUBJECT",
        "value=",
        ENDITEM,
        LAST);
```

Weitere Informationen über die Aufgabe finden Sie unter "Erstellen eines Parameters" auf Seite 441.

Bei einigen aufgezeichneten Vuser-Skripts kann es für eine korrekte Wiedergabe erforderlich sein, eine Korrelation zu implementieren. Korrelation wird verwendet, wenn ein aufgezeichnetes Skript einen dynamischen Wert enthält (beispielsweise eine Sitzungs-ID) und daher nicht erfolgreich wiedergegeben werden kann. Um dieses Problem zu beheben, wandeln Sie den dynamischen Wert in eine Variable um und ermöglichen so die erfolgreiche Wiedergabe des Skripts. Weitere Informationen finden Sie unter "Korrelieren – Übersicht" auf Seite 281.

VTS und Parametrierung

Was ist VTS?

HP LoadRunner Virtual Table Server, kurz VTS, ist eine webbasierte Anwendung, die mit LoadRunner-Vuser-Skripts ausgeführt werden kann. VTS bietet eine Alternative zur Standardparametrierung in LoadRunner.

Wenn Sie die Standardparametrierung in LoadRunner verwenden, werden jedem Vuser Parameterwerte aus einer dedizierten Gruppe von Werten zugewiesen. Parameterwerte werden von Vusern nicht gemeinsam genutzt. Im Gegensatz dazu ermöglicht Ihnen VTS das Zuweisen von Parameterwerten aus einer Gruppe von Parameterwerten zu mehreren Vusern. So können Sie ggf. genauer eine Umgebung mit realen Benutzern emulieren.

VTS besteht aus zwei Komponenten, dem VTS-Client und dem VTS-Server. Der VTS-Client besteht aus einer Reihe von API-Funktionen, mit denen auf die Daten im VTS-Server zugegriffen wird. Da die VTS-API-Funktionen in LoadRunner integriert sind, besteht keine Notwendigkeit, VTS-Client zu installieren. VTS-Server enthält eine Tabelle, die Parameterwerte enthält, die von Ihren Vuser-Skripts verwendet werden können. Die VTS-Tabelle besteht aus Spalten und Zeilen. Jede Spalte stellt einen Satz von Werten dar, die einem bestimmten Parameter in Ihren Vuser-Skripts zugewiesen werden können. Die Zellen innerhalb einer Spalte enthalten die tatsächlichen Werte, die dem Parameter zugewiesen werden.

Hinweis: Mit LoadRunner Version 11.52 wurden erhebliche Änderungen an VTS vorgenommen. Beim Aktualisieren auf VTS 11.52 oder höher können diese Änderungen möglicherweise zu Kompatibilitätsproblemen führen. Details zu den Skriptänderungen, die zur

Behebung der Kompatibilitätsprobleme erforderlich sind, finden Sie in der VTS-Dokumentation im Menü **VTS > Hilfe**.

- Informationen zum Installieren von VTS-Server Sie im *Installationshandbuch zu HP LoadRunner*. Nach der Installation von VTS-Server stehen Ihnen weitere Informationen zu VTS im Menü **VTS > Hilfe** zur Verfügung.
- Informationen zur Verwendung der VTS-Funktionalität in TruClient-Vuser-Skripts finden Sie unter "Verwenden von VTS in TruClient" auf Seite 583.

Parametertypen

Jeder Parameter wird durch den Typ der Daten definiert, die er enthält. Dieser Abschnitt enthält Informationen zu verschiedenen Parametertypen.

Parameter des Typs "Datei"

Datendateien enthalten Daten, auf die ein Vuser während der Skriptausführung zugreift. Datendateien können lokal oder global verwendet werden. Sie haben die Möglichkeit, eine vorhandene ASCII-Datei anzugeben, eine neue Datei mithilfe von VuGen zu erstellen oder eine Datenbankdatei zu importieren. Datendateien sind nützlich, wenn viele bekannte Werte für einen Parameter vorhanden sind.

In der Datendatei werden die Daten in Form einer Tabelle gespeichert. Eine Datei kann Werte für viele Parameter enthalten. Jede Spalte weist die Daten für einen Parameter auf. Die Spaltenumbrüche sind durch ein Begrenzungszeichen, beispielsweise ein Komma, gekennzeichnet.

Die Datendatei im folgenden Beispiel enthält ID-Nummern und Vornamen:

```
id,first_name
120,John
121,Bill
122,Tom
```

Hinweis: Wenn Sie nicht in englischer Sprache arbeiten, sollten Sie die Parameterdatei als UTF-8-Datei speichern. Klicken Sie im Fenster **Parametereigenschaften** auf **Mit Editor bearbeiten**. Speichern Sie die Datei in Editor als Textdatei mit UTF-8-Codierung.

Parameter des Typs "Tabelle"

Der Parametertyp **Tabelle** ist für Anwendungen gedacht, die durch Einsetzen von Tabellenzellwerten getestet werden sollen. Beim Typ **Datei** wird ein Zellwert pro Parametervorkommen eingesetzt, beim Typ **Tabelle** dienen mehrere Zeilen und Spalten als Parameterwerte, ähnlich wie bei einem Wertearray. Bei Verwendung des Typs **Tabelle** können Sie eine Tabelle mit einem einzigen Befehl ausfüllen. Dies ist für SAPGUI-Vuser üblich, bei denen die Tabellenzellen mithilfe der Funktion **sapgui_table_fill_data** ausgefüllt werden.

Parameter des Typs "XML"

Diese Parameter werden als Platzhalter für die in XML-Strukturen enthaltenen mehrwertigen Daten verwendet. Sie können Parameter des XML-Typs verwenden, um die gesamte Struktur durch einen einzigen Parameter zu ersetzen. So könnte der XML-Parameter **Adresse** beispielsweise den Namen, die Straße, die Stadt und die Postleitzahl eines Ansprechpartners ersetzen. Der Einsatz von XML-Parametern für diesen Datentyp ermöglicht einen sauberere Dateneingabe und Parametrierung von Vuser-Skripts. Wir empfehlen, XML-Parameter mit Webdienstskripts oder für SOA-Services zu verwenden.

Parameter des Typs "Interne Daten"

Interne Daten werden automatisch generiert, während ein Vuser wie Datum/Zeit, Gruppenname, Iterationsnummer, Lastgeneratorname, Zufallszahl, eindeutige Zahl und Vuser-ID ausgeführt wird.

- Benutzerdefiniert: Sie können den Parameterdatentyp angeben.
- Datum/Zeit: Das aktuelle Datum und die aktuelle Uhrzeit. Sie können das Format und den Offset im Dialogfeld **Parametereigenschaften** eingeben.
- Gruppenname: Der Name der Vuser-Gruppe. Wenn keine Vuser-Gruppe vorhanden ist (z. B. wenn ein Skript von VuGen aus ausgeführt wird), ist der Wert immer **Keine**.
- Iterationsnummer: Die aktuelle Iterationsnummer.
- Lastgeneratorname: Der Name des Lastgenerators für das Vuser-Skript (der Computer, auf dem der Vuser ausgeführt wird).
- Zufallszahl: Eine Zufallszahl innerhalb eines von Ihnen angegebenen Wertebereichs.
- Eindeutige Zahl: Weist einen Zahlenbereich zu, der für die einzelnen Vuser zu verwenden ist. Sie geben den Startwert und die Blockgröße an (Anzahl der eindeutigen Zahlen, die für die einzelnen Vuser reserviert werden muss). Beispiel: Wenn Sie den Startwert 1 und die Blockgröße 100 angeben, kann der erste Vuser die Zahlen 1 bis 100 verwenden, der zweite Vuser die Zahlen 201 bis 300 usw.
- Vuser-ID: Die ID-Nummer, die einem Vuser während der Ausführung eines Szenarios vom Controller zugewiesen wird. Wenn Sie ein Skript von VuGen aus ausführen, beträgt die Vuser ID stets -1.

Hinweis: Dabei handelt es sich nicht um die ID-Nummer, die im Vuser-Fenster angezeigt wird, sondern um eine eindeutige ID-Nummer, die während der Laufzeit generiert wird.

Benutzerdefinierte Funktionsparameter

Daten, die mithilfe einer Funktion aus einer externen DLL erzeugt werden. Eine benutzerdefinierte Funktion ersetzt den Parameter durch einen Wert, der von einer Funktion in einer externen DLL ausgegeben wurde.

Bevor Sie eine benutzerdefinierte Funktion als Parameter zuordnen, müssen Sie die externe Bibliothek (DLL) mit der Funktion erstellen. Die Funktion muss folgendes Format aufweisen:

```
__declspec(dllexport) char *<functionName>(char *, char *)
```

Beide an diese Funktion gesendeten Argumente sind NULL.

Es ist empfehlenswert, beim Anlegen der Bibliothek den Standardpfad zur dynamischen Bibliothek zu verwenden. Auf diese Weise müssen Sie lediglich den Namen der Bibliothek, nicht aber den vollständigen Pfadnamen der Bibliothek eingeben. Der bin-Ordner von VuGen ist der Standardpfad der dynamischen Bibliothek. Sie können Ihre Bibliothek zu diesem Ordner hinzufügen.

Im Folgenden sind Beispiele für benutzerdefinierte Funktionen aufgeführt:

```
__declspec(dllexport) char *UF_GetVersion(char *x1, char *x2) {return "Ver2.0";}
__declspec(dllexport) char *UF_GetCurrentTime(char *x1, char *x2) {
time_t x = tunefully); static char t[35]; strcpy(t, ctime( &amp;x)); t[24] = '\
0'; return t;}
```

Erstellen eines Parameters

In dieser Aufgabe wird beschrieben, wie ein Parameter erstellt wird.

1. ## Auswählen des zu parametrierenden Werts

   Sie können diesen Schritt im Editor oder im Bereich **Schrittenavigator** ausführen.

   Wählen Sie den zu parametrierenden Wert aus, klicken Sie mit der rechten Maustaste und wählen Sie **Durch Parameter ersetzen** aus.

   Hinweise:

   - Beim Erstellen von XML-Parametern in der Skriptansicht müssen Sie nur den inneren XML, nicht die umschließenden Tags, auswählen. Um beispielsweise die komplexe Datenstruktur <A><B>Belement</B><C>Celement</C></A> zu parametrieren, wählen Sie die gesamte Zeichenfolge (<B>Belement</B><C>Celement</C>) aus und ersetzen sie durch einen Parameter.

   - Bei der Parametrierung von Java Record Replay- oder Java-Vuser-Skripts muss der gesamte Wert, nicht nur ein Teil davon, parametriert werden.

   Bereich "Schrittenavigator"

   Klicken Sie mit der rechten Maustaste auf einen Schritt und wählen Sie **Argumente anzeigen** aus. Klicken Sie auf das **ABC**-Symbol neben dem zu parametrierenden Argument.

2. ## Erstellen eines neuen Parameters im Dialogfeld Parameter auswählen oder erstellen

   Geben Sie im Dialogfeld **Parameter auswählen oder erstellen** den Namen und Typ des Parameters ein. Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Parameter auswählen oder erstellen"" auf Seite 455.

3. **Hinzufügen einer Liste der erforderlichen Werte**

   Wählen Sie **Eigenschaften** im Dialogfeld **Parameter auswählen oder erstellen** aus. Erstellen Sie eine Tabelle und fügen Sie Einträge hinzu, die als Werteliste für den Parameter dienen. Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Parametereigenschaften"" auf Seite 456.

4. **Ändern der Parameterklammern - optional**

   Im Dialogfeld **Konfiguration der Parametertrennzeichen** können Sie die Klammern ändern, die die Parameter umgeben. Folgende Methoden stehen Ihnen für den Zugriff auf das Dialogfeld zur Verfügung:

   - Klicken Sie mit rechten Maustaste auf den Knoten **Parameter** im Bereich **Lösungs-Explorer** und wählen Sie **Parametertrennzeichen konfigurieren** aus.
   - **Entwurf > Parameter > Parametertrennzeichen konfigurieren**
   - **Extras > Optionen > Parameter**

   Informationen zur Benutzeroberfläche finden Sie unter Dialogfeld "Konfiguration der Parametertrennzeichen".

Datenzuweisungsmethoden für die Parametertypen "Datei/Tabelle/XML"

Wenn Sie Werte aus einer Datei verwenden, bietet Ihnen VuGen die Möglichkeit, die Methode für die Zuweisung der Daten von der Quelle zu den Parametern festzulegen. Die folgenden Methoden sind für die Zuweisung der Daten verfügbar:

Sequential

Weist dem Vuser Daten sequentiell zu. Wenn ein ausgeführter Vuser auf die Datentabelle zugreift, wird ihm die nächste verfügbare Datenzeile zugewiesen.

Enthält die Datentabelle nicht genügend Werte, kehrt VuGen zum ersten Tabellenwert zurück und setzt diese Schleife bis zum Testende fort.

Random

Weist jedes Mal, wenn ein neuer Parameterwert angefordert wird, einen Zufallswert aus der Datentabelle zu.

Beim Ausführen eines Szenarios in LoadRunner oder eines Skripts in HP Business Process Monitor können Sie eine Startnummer für die zufallsgenerierte Sequenzierung angeben. Jeder Startwert repräsentiert eine Sequenz von zufallsgenerierten Werten, die für eine Testausführung verwendet werden können. Wann immer Sie diesen Startwert verwenden, wird den Vusern im Szenario dieselbe Folge von Werten zugewiesen. Aktivieren Sie diese Option, wenn Sie auf ein Problem bei der Testausführung stoßen und den Test unter Verwendung derselben Folge von zufallsgenerierten Werten wiederholen möchten.

Unique

Weist dem Parameter für jeden Vuser einen eindeutigen sequentiellen Wert zu. Stellt sicher, dass die Tabelle genügend Daten für alle Vuser und ihre Iterationen enthält. Wenn Sie mit 20 Vusern arbeiten und 5 Iterationen durchführen möchten, muss Ihre Tabelle mindestens 100 eindeutige Werte enthalten.

Wenn keine eindeutigen Werte mehr vorhanden sind, verhält sich VuGen gemäß der im Feld **Wenn Werte nicht ausreichen** ausgewählten Option. Weitere Informationen finden Sie unter "Dialogfeld "Parametereigenschaften"" auf Seite 456.

Hinweis: Für LoadRunner-Benutzer: Wenn ein Skript die eindeutige Dateiparametrierung verwendet, kann es beim Ausführen von mehr als einer Vuser-Gruppe innerhalb desselben Szenarios mit diesem Skript zu unerwarteten Szenarioergebnissen kommen. Weitere Informationen zu Vuser-Gruppen in Szenarios finden Sie in der Function Reference.

- Weitere Informationen zu den verschiedenen Methoden der Datenzuweisung und Aktualisierung finden Sie unter "Datenzuweisungs- und -aktualisierungsmethoden für die Parametertypen "Datei", "Tabelle" und "XML"" unten.
- Weitere Informationen über das Verhalten von Parametern bei Abweichungen in der Anzahl der Iterationen und der Werte in der Parameterdatei finden Sie unter "Vuser-Verhalten im Controller (nur LoadRunner)" auf Seite 445.

Datenzuweisungs- und -aktualisierungsmethoden für die Parametertypen "Datei", "Tabelle" und "XML"

Bei Parametern des Typs Datei, Tabelle und XML haben die ausgewählte Datenzuweisungsmethode und -aktualisierungsmethode Auswirkungen auf die Werte, mit denen die Vuser die Parameter während der Ausführung des Szenarios ersetzen.

Die Datenzuweisungsmethode wird durch das Feld **Nächste Zeile auswählen** bestimmt und die Datenaktualisierungsmethode durch das Feld **Wert aktualisieren bei**.

Die folgende Tabelle bietet eine Zusammenfassung der von den Vusern verwendeten Werte in Abhängigkeit von den Eigenschaften für die Datenzuweisung und -aktualisierung.

Aktualisierungs-methode	**Datenzuweisungsmethode**		
	Sequenziell	**Zufällig**	**Eindeutig**
Each iteration	Der Vuser verwendet für jede Iteration den jeweils *nächsten* Wert der Datentabelle.	Der Vuser verwendet für jede Iteration einen *neuen, zufallsgenerierten* Wert der Datentabelle.	Der Vuser verwendet für jede Iteration einen Wert aus der nächsten eindeutigen Position in der Datentabelle.

Each Occurrence (nur Datendateien)	Der Vuser verwendet für jedes Parametervorkommen, auch wenn es in der gleichen Iteration liegt, den *nächsten* Wert der Datentabelle.	Der Vuser verwendet für jedes Parametervorkommen, auch wenn es in der gleichen Iteration liegt, einen *neuen, zufallsgenerierten* Wert der Datentabelle.	Der Vuser verwendet für jedes Parametervorkommen, auch wenn es in der gleichen Iteration liegt, einen *neuen, eindeutigen* Wert der Datentabelle.
Once	Der in der ersten Iteration zugewiesene Wert wird in allen folgenden Iterationen für jeden Vuser verwendet.	Der in der ersten Iteration zugewiesene Wert wird in allen Iterationen des jeweiligen Vusers verwendet.	Der in der ersten Iteration zugewiesene Wert wird in allen folgenden Iterationen des Vusers verwendet.

Beispiele

Angenommen, Ihre Tabelle/Datei weist die folgenden Werte auf:

Katharina; David; Michael; Heike; Ralf; Anna; Konstantin; Julia; Frank

"Sequential"-Methode

- Wenn Sie als Aktualisierungsmethode **Each iteration** festlegen, verwenden alle Vuser in der ersten Iteration den Wert Katharina, in der zweiten Iteration den Wert David, in der dritten Iteration den Wert Michael usw.
- Wenn Sie als Aktualisierungsmethode **Each occurrence** festlegen, verwenden alle Vuser für das erste Vorkommen den Wert Katharina, für das zweite Vorkommen den Wert David, für das dritte Vorkommen den Wert Michael usw.
- Wenn Sie als Aktualisierungsmethode **Once** festlegen, verwenden alle Vuser für alle Iterationen den Wert Katharina.

Hinweis: Enthält die Datentabelle bei Auswahl der Methode **Sequential** nicht genügend Werte, kehrt VuGen zum ersten Tabellenwert zurück und setzt diese Schleife bis zum Testende fort.

"Random"-Methode

- Wenn Sie als Aktualisierungsmethode **Each iteration** festlegen, verwenden die Vuser für jede Iteration zufallsgenerierte Werte aus der Tabelle.
- Wenn Sie als Aktualisierungsmethode **Each occurrence** festlegen, verwenden die Vuser für jedes Parametervorkommen zufallsgenerierte Werte aus der Tabelle.
- Wenn Sie als Aktualisierungsmethode **Once** festlegen, verwenden alle Vuser für alle Iterationen den ersten zufällig zugewiesenen Wert.

"Unique"-Methode

- Wenn Sie als Aktualisierungsmethode **Each iteration** festlegen, verwendet der erste Vuser bei einem Testlauf mit 3 Iterationen in der ersten Iteration den Wert Katharina, in der zweiten den Wert David und in der dritten den Wert Michael. Der zweite Vuser verwendet die Werte Heike, Ralf und Anna. Der dritte Vuser verwendet die Werte Konstantin, Julia und Frank.
- Wenn Sie als Aktualisierungsmethode **Each occurrence** festlegen, verwendet der Vuser für jedes Parametervorkommen einen eindeutigen Wert aus der Liste.
- Legen Sie als Aktualisierungsmethode **Once** fest, verwendet der erste Vuser für alle Iterationen den Wert Katharina, der zweite Vuser den Wert David usw.

Vuser-Verhalten im Controller (nur LoadRunner)

Wenn Sie ein Szenario zur Ausführung eines parametrierten Skripts einrichten, können Sie festlegen, wie die Vuser vorgehen sollen, wenn nicht genügend Werte zur Verfügung stehen. Die folgende Tabelle bietet einen Überblick über die Ergebnisse eines Szenarios mit den folgenden Parametereinstellungen:

- **Nächste Zeile auswählen = Unique**
- **Wert aktualisieren bei = Each iteration**
- **Wenn Werte nicht ausreichen = Continue with last value**

Situation	Dauer	Resultierende Aktion
Mehr Iterationen als Werte	**Bis Ende ausführen**	Wenn die eindeutigen Werte aufgebraucht sind, fährt jeder Vuser mit dem letzten Wert fort und eine Warnmeldung, dass die Werte nicht mehr eindeutig sind, wird an das Protokoll gesendet.
Mehr Vuser als Werte	**"Unbegrenzt ausführen" oder "Ausführen für"**	Die Vuser verwenden alle eindeutigen Werte, bis diese aufgebraucht sind. Anschließend gibt der Test die Fehlermeldung **Fehler: Nicht genügend Datensätze für den Parameter <param_name> in der Tabelle, um für den Vuser eindeutige Daten zu liefern** aus. Dies kann vermieden werden, indem Sie in den Parametereigenschaften die Option **Wenn Werte nicht ausreichen** oder die Methode **Nächste Zeile auswählen** ändern.
Für jeden zweiten Parameter sind keine Werte vorhanden	**"Unbegrenzt ausführen" oder "Ausführen für"**	Der Parameter, für den keine Werte vorhanden sind, wird zyklisch fortgesetzt, bis die Werte für den zweiten Parameter nicht mehr eindeutig sind.

Verwenden von bestehenden Parametern

In dieser Aufgabe wird beschrieben, wie Werte durch vordefinierte Parameter ersetzt werden.

Ersetzen eines Werts durch einen Parameter

Sie können einen Wert durch einen vordefinierten Parameter ersetzen. Klicken Sie im Skripteditor mit der rechten Maustaste auf den betreffenden Wert und wählen Sie eine der folgenden Optionen aus:

- **Durch Parameter ersetzen** > wählen Sie einen <vordefinierten> Parameter aus. Die Liste enthält Parameter mit demselben ursprünglichen Wert oder Parameter, die noch nicht verwendet wurden.
- **Durch Parameter ersetzen** > wählen Sie einen Parameter im Dialogfeld **Parameterliste** aus.

Ersetzen mehrerer Vorkommen eines Werts durch einen Parameter

Sie können mehrere Vorkommen eines Werts durch einen Parameter ersetzen In diesem Fall ersetzen Sie im Skripteditor mindestens ein Vorkommen des betreffenden Werts durch einen Parameter. Klicken Sie mit der rechten Maustaste auf den Parameter und wählen Sie **Weitere Vorkommen ersetzen** aus. Verwenden Sie **Suchen und Ersetzen**, um alle Werte im Skript durch den ausgewählten Parameter zu ersetzen.

Wiederherstellen des ursprünglichen Werts

Sie können das Ersetzen eines Werts rückgängig machen und den ursprünglichen Wert wiederherstellen, indem Sie im Skripteditor mit der rechten Maustaste auf den Parameter klicken und **Ursprünglichen Wert wiederherstellen** auswählen.

XML-Parameter

Beim Erstellen eines Webdienstaufrufs zur Emulation einer spezifischen Operation enthalten die Operationsargumente möglicherweise komplexe Strukturen mit zahlreichen Werten. Sie können Parameter des XML-Typs verwenden, um die gesamte Struktur durch einen einzigen Parameter zu ersetzen.

Sie können mehrere Wertesätze für die XML-Elemente festlegen und dann für jede Iteration einen anderen Wert zuweisen.

Der Parametertyp XML unterstützt komplexe Schematypen wie z. B. Arrays, Choice und <Any>-Elemente.

Bei der Arbeit mit Webdienst-Eingabeargumenten treffen Sie möglicherweise auf Arrays und diesen untergeordnete Elemente. Sie können einen einzigen XML-Parameter für die Werte aller Arrayelemente definieren.

Neue Parameter vom Typ XML können, ebenso wie alle anderen Parametertypen, direkt im Menü **Einfügen** erstellt werden. XML-Parameter für Webdienstskripts können Sie direkt in den Eigenschaften des Webdienstaufrufs erstellen.

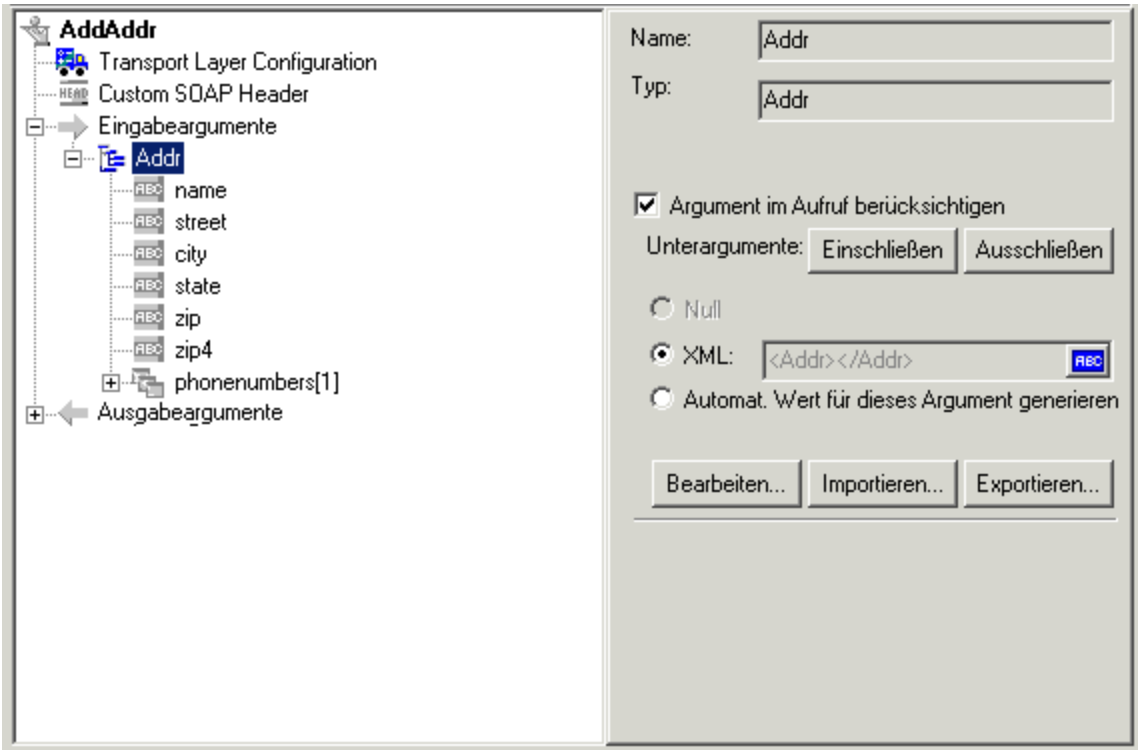
Achtung: Bei Protokollen, die XML verwenden, wird bei der Wiedergabe keine Anforderung erstellt, wenn ein parametriertes Eingabeargument das kaufmännische Und (&) enthält.

Erstellen eines XML-Parameters aus einem Webdienstaufruf

In dieser Aufgabe wird beschrieben, wie ein neuer XML-Parameter aus einem Webdienstaufruf erstellt wird. Dieses Verfahren ist neben dem Standardverfahren eine zusätzliche Möglichkeit, einen Parameter zu erstellen. XML-Parameter können auch mit dem Standardverfahren erstellt werden.

Erstellen eines XML-Parameters aus einem Webdienstaufruf

1. Wählen Sie das Stammelement der komplexen Datenstruktur aus. Im rechten Bereich werden die Argumentdetails angezeigt.



2. Wählen Sie im rechten Bereich **XML** aus und klicken Sie auf das Symbol **ABC** . Das Dialogfeld **Parameter auswählen oder erstellen** wird geöffnet.

3. Geben Sie im Feld **Parametername** einen Namen für den Parameter ein.

4. Wählen Sie im Feld **Parametertyp** den Eintrag **XML** aus, wenn er nicht bereits ausgewählt ist.

5. Klicken Sie auf **Eigenschaften**, wenn Sie einen Wertesatz zuweisen möchten, oder auf **OK**,

um das Dialogfeld zu schließen und die Werte später zuzuweisen.

Erstellen von XML-Parametern - Standardmethode

In diesem Abschnitt wird erläutert, wie Parameter des Typs XML erstellt werden, ohne die Eigenschaften eines Webdienstaufrufs anzuzeigen. Diese Methode wird bei der Parametrierung von Werten für die meisten Protokolle und Parametertypen am häufigsten eingesetzt.

Für Webdienstskripts wird empfohlen, Parameter über einen Webdienstaufruf zu erstellen wie in "XML-Parameter" auf Seite 446 beschrieben.

Erstellen eines neuen XML-Parameters

1. Wählen Sie **Einfügen > Neuer Parameter** aus oder wählen Sie eine Konstante in der Skriptansicht aus und wählen Sie dann im Kontextmenü **Ersetzen durch einen Parameter** aus. Das Dialogfeld **Parameter auswählen oder erstellen** wird geöffnet.

2. Geben Sie im Feld **Parametername** einen Namen für den Parameter ein.

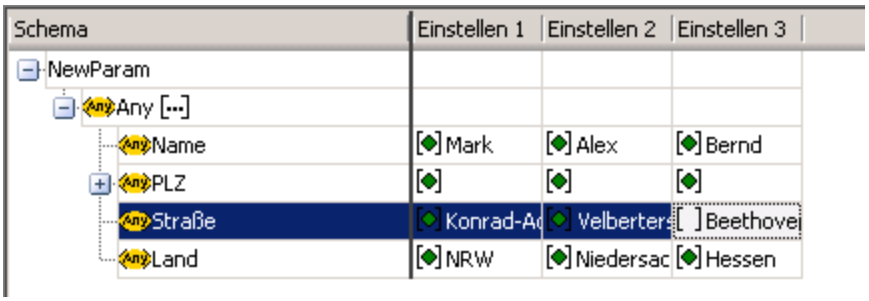
3. Wählen Sie im Feld **Parametertyp** den Eintrag **XML** aus, wenn er nicht bereits ausgewählt ist.

4. Klicken Sie auf **Eigenschaften**, wenn Sie einen Wertesatz zuweisen möchten, oder auf **OK**, um das Dialogfeld zu schließen und die Werte später zuzuweisen.

Definieren von XML-Wertesätzen

In diesem Abschnitt wird die Erstellung von Wertesätzen für XML-Parameter beschrieben.

Wertesätze sind Arrays, die eine bestimmte Wertemenge enthalten. Über die Schaltflächen **Spalte hinzufügen** und **Spalte duplizieren** können Sie mehrere Wertesätze für Ihren Parameter erstellen und in verschiedenen Iterationen einsetzen.

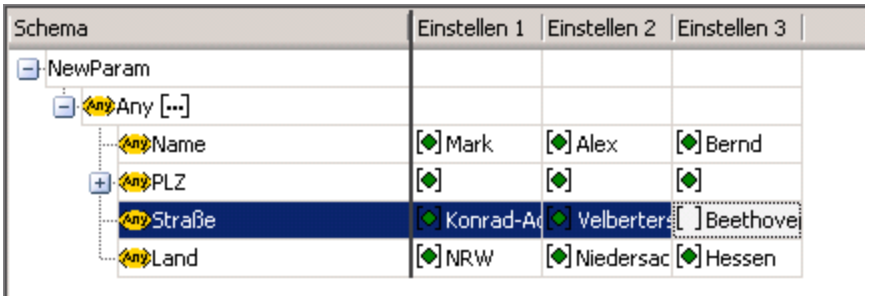


Bei Wertesätzen kann die Anzahl der Arrayelemente pro Parameter variieren.

Sie können optionale Elemente verwenden, die in einem Wertesatz enthalten sind, in einem anderen jedoch nicht. Hierdurch ist es möglich, die Werte zu variieren, die für jede der Iterationen übergeben werden – einige Iterationen enthalten bestimmte Arrayelemente, während andere diese nicht enthalten.

Um ein optionales Element auszuschließen, klicken Sie auf das kleine Dreieck in der oberen linken Zellecke und stellen sicher, dass es nicht ausgefüllt ist.

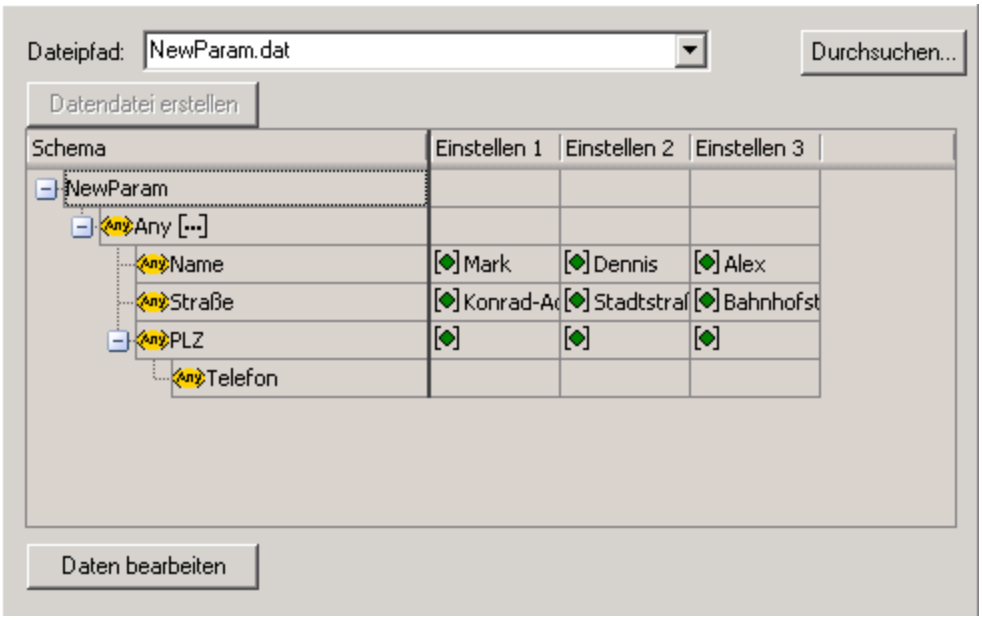
Im folgenden Beispiel verwenden **Wertesatz 1** und **Wertesatz 2** die optionalen Elemente **Name**, **Straße** und **Land**. In **Wertesatz 3** wird kein Straßenname eingesetzt.



Festlegen von Werten für Parameterelemente

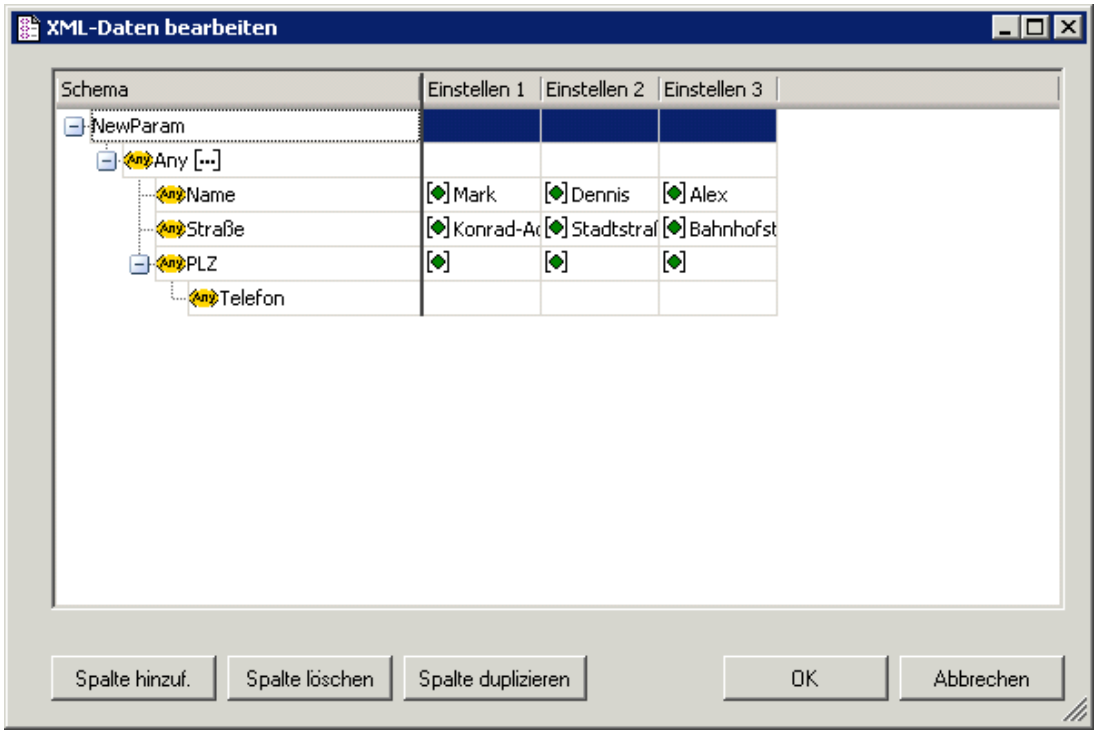
1. **Zeigen Sie die Parametereigenschaften an.**

   Wenn das Dialogfeld **Parametereigenschaften** nicht geöffnet wird, wählen Sie **Vuser > Parameterliste** und dann den gewünschten Parameter aus. Das Dialogfeld zeigt eine schreibgeschützte Ansicht der Parameterwerte an.



2. **Öffnen Sie das Dialogfeld "XML-Daten bearbeiten".**

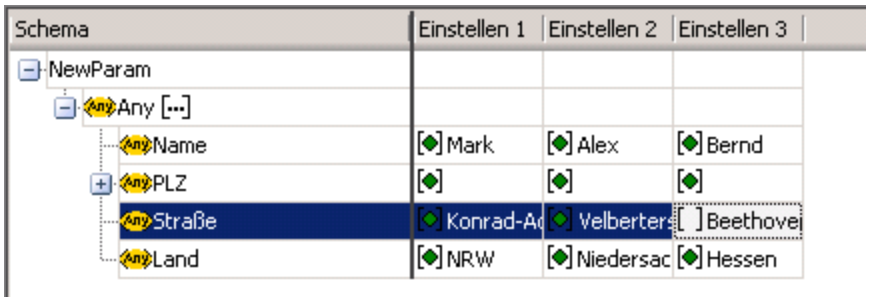
   Klicken Sie auf die Schaltfläche **Daten bearbeiten**, um das Dialogfeld **XML-Daten bearbeiten** zu öffnen.



3. **Definieren Sie Wertesätze für den XML-Parameter.**

   Geben Sie in den **Wertesatz**-Spalten Werte ein, die dem jeweiligen Schema entsprechen.

   Steht in einer Zeile **NIL**, handelt es sich um ein Nullable-Element. Werte für Nullable-Elemente werden wie gewohnt eingegeben. Um einen Wert als **NIL** zu kennzeichnen, klicken Sie auf das Symbol **NIL** und füllen es aus. Dadurch wird der dem Element eventuell zugewiesene Wert gelöscht. Im folgenden Beispiel ist das Element **Stadt** ein Nullable-Element. Es ist jedoch nur in **Wertesatz 2** und **Wertesatz 3**, nicht aber in **Wertesatz 1** mit **NIL** gekennzeichnet.

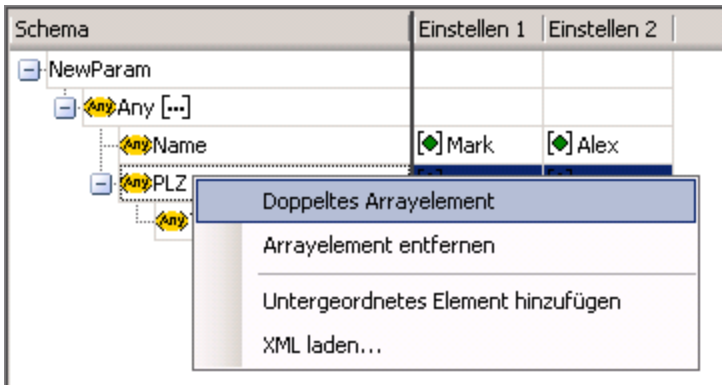


4. **Erstellen Sie zusätzliche Wertesätze.**

   Um weitere Wertesätze einzufügen, klicken Sie auf **Spalte hinzufügen** und geben einen Wertesatz in die neue Spalte ein. Wenn Sie einen vorhandenen Wertesatz kopieren möchten, wählen Sie die zu kopierende Zeile des Wertesatzes aus und klicken auf **Spalte duplizieren**.

5. **Kopieren Sie die Arrays.**

Wählen Sie zum Duplizieren eines Arrayelements und seiner untergeordneten Elemente den übergeordneten Knoten und dann im Kontextmenü den Eintrag **Doppeltes Arrayelement** aus.

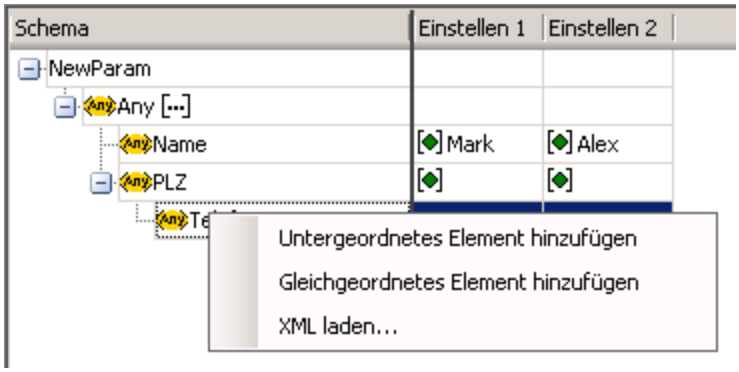


6. **Bearbeiten Sie die <any>-Elemente.**

Klicken Sie für Elemente des Typs **any** in der Spalte **Schema** mit der rechten Maustaste auf **<any>** und wählen Sie eine der verfügbaren Optionen aus. Je nach Cursorposition werden unterschiedliche Optionen angezeigt.

- **Arrayelement hinzufügen.** Fügt ein untergeordnetes Element unter dem Stammelement hinzu.
- **Untergeordnetes Element hinzufügen.** Fügt ein untergeordnetes Element zum ausgewählten Element hinzu.
- **Gleichgeordnetes Element hinzufügen.** Fügt ein untergeordnetes Element auf der Ebene des ausgewählten Elements hinzu.
- **XML laden.** Lädt die Elementwerte aus einer XML-Datei.
- **XML speichern.** Speichert das Array als XML-Datei.
- **XML kopieren**. Kopiert den vollständigen XML-Code des ausgewählten Elements in die Zwischenablage.

Klicken Sie auf den Text **Element umbenennen** und geben Sie einen sinnvollen Namen für jedes Arrayelement ein.



7. **Entfernen Sie nicht gewünschte Spalten.**

   Um einen Wertesatz zu entfernen, markieren Sie diesen und klicken auf **Spalte löschen**.

8. **Speichern Sie die Änderungen.**

   Klicken Sie auf **OK**, um die Änderungen zu speichern und die Ansicht im Dialogfeld **Parametereigenschaften** zu aktualisieren.

Festlegen einer Zuweisungsmethode

In dieser Aufgabe wird beschrieben, wie Sie eine Zuweisungsmethode festlegen. Die Zuweisungsmethode bestimmt, wie welcher Wertesatz verwendet wird. So können Sie die Vuser beispielsweise anweisen, für jede Iteration einen neuen Wertesatz zu verwenden sowie die Wertesätze sequentiell oder zufällig einzusetzen. Weitere Informationen finden Sie unter "Datenzuweisungs- und -aktualisierungsmethoden für die Parametertypen "Datei", "Tabelle" und "XML"" auf Seite 443.

Definieren einer Zuweisungsmethode

1. Öffnen Sie das Dialogfeld **Parametereigenschaften** und wählen Sie einen Parameter aus.

2. Legen Sie eine Datenzuweisungsmethode fest.

   Wählen Sie in der Liste **Nächsten Wert auswählen** eine Datenzuweisungsmethode aus, um den Vuser anzuweisen, wie die Dateidaten während der Ausführung des Vuser-Skripts ausgewählt werden. Diese Optionen sind: **Sequential**, **Random** und **Unique**. Weitere Informationen finden Sie unter "Datenzuweisungsmethoden für die Parametertypen "Datei/Tabelle/XML"" auf Seite 442.

3. Wählen Sie eine Aktualisierungsoption für den Parameter aus.

   Wählen Sie in der Liste **Wert aktualisieren bei** eine Aktualisierungsoption aus. Sie haben die Wahl zwischen **Each Iteration**, **Each Occurrence** und **Once**. Weitere Informationen finden Sie unter "Datenzuweisungs- und -aktualisierungsmethoden für die Parametertypen "Datei", "Tabelle" und "XML"" auf Seite 443.

4. Bei Auswahl der Datenzuweisungsmethode **Unique** werden die Optionen **Wenn Werte nicht ausreichen** und **Vuser-Werte im Controller zuordnen** aktiviert.

   - **Wenn Werte nicht ausreichen.** Geben Sie an, was geschehen soll, wenn keine weiteren eindeutigen Daten mehr vorhanden sind: **Abort Vuser**, **Continue in a cyclic manner** oder **Continue with last value**.

   - **Vuser-Werte im Controller zuordnen** (nur für LoadRunner-Benutzer)**.** Geben Sie an, ob Sie den Vusern manuell Datenblöcke zuordnen möchten. Sie können dem Controller erlauben, automatisch eine Blockgröße zuzuordnen oder aber die gewünschte Anzahl von Werten festlegen. Wählen Sie **Blockgröße automatisch zuordnen** oder **Zuordnen** x **Werte für jeden Vuser** aus. Legen Sie für die zweite Option die Anzahl der zuzuordnenden Werte fest.

   Um diese Vorkommen zu verfolgen, aktivieren Sie die Option **Erweitertes Protokoll > Parametersubstitution** in den Laufzeiteinstellungen des Protokolls. Sind nicht genügend Daten vorhanden, schreibt VuGen eine Warnmeldung in das Vuser-Protokoll: **Keine weiteren eindeutigen Werte für diesen Parameter in Tabelle <tabellenname>**.

5. Klicken Sie im Dialogfeld **Parametereigenschaften** auf **Schließen**.

Die Liste der Eingabeargumente wird durch den Parameternamen und die Schaltfläche **ABC** durch ein Tabellensymbol ersetzt , auf das Sie klicken können, um die Parametereigenschaften zu bearbeiten oder die Parametrierung zu entfernen.

Ändern der Eigenschaften von XML-Parametern

In dieser Aufgabe wird beschrieben, wie die Eigenschaften von XML-Parametern geändert werden können.

So ändern Sie die Eigenschaften von XML-Parametern:

1. Klicken Sie im Schrittenavigator mit der rechten Maustaste auf den entsprechenden Schritt und wählen Sie **Argumente anzeigen** aus. Das Dialogfeld **Eigenschaften von Webdienstaufruf** wird geöffnet.

2. Wählen Sie in der Liste der Argumente unter **Eingabeargumente** den XML-Parameter aus. Im rechten Bereich werden die Parameterdetails angezeigt.

3. Um die Eigenschaften des XML-Parameters zu ändern, aktivieren Sie das Kontrollkästchen **XML**, klicken Sie auf die Schaltfläche mit dem Tabellensymbol und wählen Sie dann **Parametereigenschaften** aus. Das Dialogfeld **Parametereigenschaften** wird geöffnet.

4. Bearbeiten Sie die Parametereigenschaften nach Bedarf.

Festlegen von AUT-Umgebungsparametern

Wenn Sie in HP Application Lifecycle Management (ALM) gespeicherte Skripts verwenden, können Sie in verschiedenen AUT-Umgebungen (Application Under Test) arbeiten.

AUT-Umgebungen sind in ALM definierte Umgebungen, die unterschiedliche Testkonfigurationen darstellen. Durch Parametrierung der Umgebungsdaten können Sie die Flexibilität und Übertragbarkeit von Tests verbessern. Statt mehrere Skripts mit identischer Logik, aber unterschiedlichen AUT-Umgebungskonstanten auszuführen, können Sie ein einziges Skript mit AUT-Umgebungsparametern verwenden.

Umgebungsspezifische Parameter werden der in ALM-Konfiguration für die AUT-Umgebung definiert. Während der Testausführung werden diese Werte in das Skript eingefügt. Weitere Informationen über das Arbeiten mit ALM-Umgebungen finden Sie im *HP Application Lifecycle Management-Benutzerhandbuch*.

Diese Aufgabe beschreibt die Definition eines AUT-Umgebungstypparameters im Vuser-Skript.

1. Erstellen Sie in VuGen einen Parameter. Stellen Sie sicher, dass der Name des Parameters mit dem Namen des zugehörigen AUT-Umgebungsparameters übereinstimmt.

2. Legen Sie den Parametertyp auf "**Benutzerdefinierter Parameter**" fest.

3. Geben Sie unter Beachtung von Schreibweise und Groß-/Kleinschreibung folgende Parameterbeschreibung ein: "ALMPARAM".

Importieren von Parameterdaten aus einer Datenbank

In VuGen können Sie Daten aus einer Datenbank importieren und für die Parametrierung verwenden. Sie können die Daten mit einer der folgenden Methoden importieren: Nach dem Import werden die Daten als reguläre Parameterdatei mit der Erweiterung **.dat** und dem Präfix des >Quellenskriptnamens< gespeichert.

Erstellen einer Abfrage mit Microsoft Query

1. Wählen Sie im "Datenbankabfrage-Assistent" auf Seite 471 die Option **Abfrage mit Microsoft Query erstellen** aus. Wenn Sie Anweisungen zu Microsoft Query benötigen, wählen Sie **Anleitung zur Verwendung von Microsoft Query** aus.

2. Klicken Sie auf **Fertig stellen**. Wenn Microsoft Query nicht auf Ihrem Rechner installiert ist, gibt VuGen eine entsprechende Meldung aus. Installieren Sie MS Query von Microsoft Office, bevor Sie Ihre Arbeit fortsetzen.

3. Folgen Sie den Anweisungen des Assistenten und importieren Sie die gewünschten Tabellen und Spalten.

4. Nach Abschluss des Datenimports wählen Sie **Schließen und zum Virtual User Generator zurückkehren** und klicken Sie auf **Fertig**. Die Datensätze der Datenbank werden als

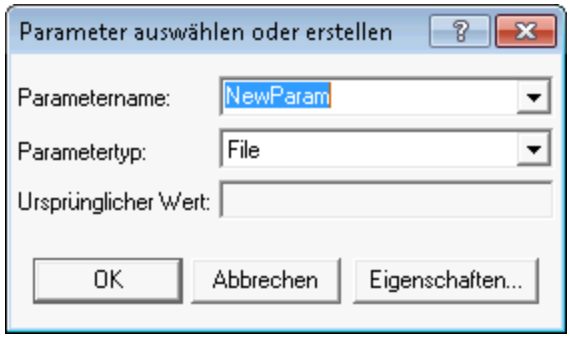
Datendatei im Feld **Parametereigenschaften** angezeigt.

Manuelles Angeben einer SQL-Anweisung

1. Wählen Sie im "Datenbankabfrage-Assistent" auf Seite 471 die Option **SQL-Anweisung manuell angeben** aus und klicken Sie auf **Weiter**.
2. Klicken Sie auf **Erstellen**, um eine neue Zeichenfolge für die Verbindung anzugeben. Das Fenster **Auswählen der Datenquelle** wird geöffnet.
3. Wählen Sie eine Datenquelle aus oder klicken Sie auf **Neu**, um eine neue Datenquelle zu erstellen. Der Assistent führt Sie durch das Verfahren zur Erstellung von ODBC-Datenquellen. Nach Abschluss des Vorgangs wird die Verbindungszeichenfolge im Feld **Verbindungszeichenfolge** angezeigt.
4. Geben Sie eine SQL-Anweisung in das Feld **SQL-Anweisung** ein.
5. Klicken Sie auf **Fertig**, um die SQL-Anweisung zu verarbeiten und die Daten zu importieren. Die Datensätze der Datenbank werden als Datendatei im Feld **Parametereigenschaften** angezeigt.

Dialogfeld "Parameter auswählen oder erstellen"

In diesem Dialogfeld können Sie einen neuen Parameter erstellen oder einen bestehenden Parameter bearbeiten.

	
Zugriff	Führen Sie einen der folgenden Schritte durch: • Bereich **VuGen** > **Lösungs-Explorer** > Rechtsklick auf Knoten **Parameter** > **Neuen Parameter erstellen** • Klicken Sie im Script-Editor mit der rechten Maustaste auf den Wert > **Durch Parameter ersetzen** > **Neuen Parameter erstellen** • **VuGen > Entwurf > Parameter > Neuen Parameter erstellen**

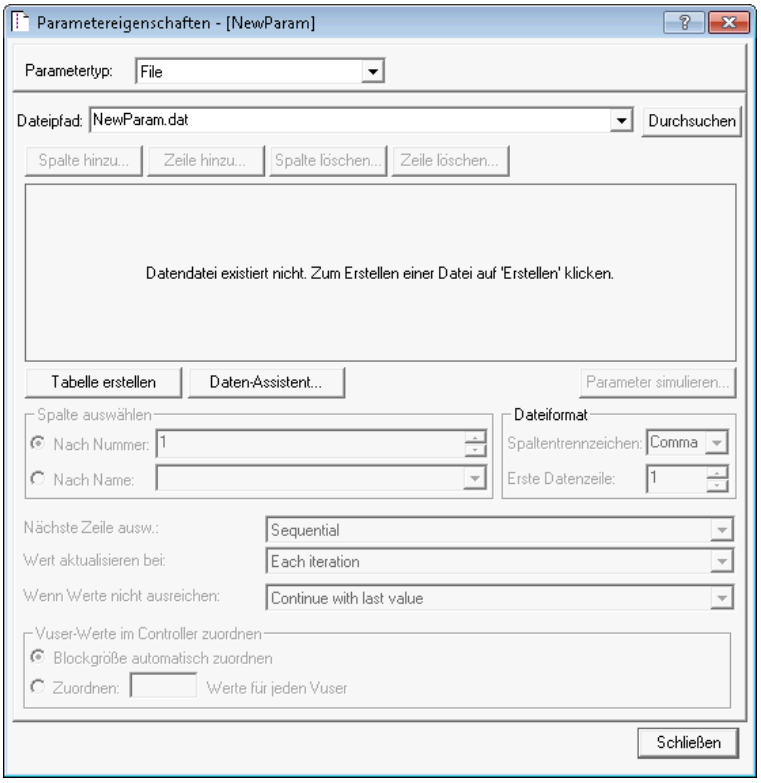
Relevante Aufgaben	"Erstellen eines Parameters" auf Seite 441

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Parametername	Der Name des Parameters. **Hinweis:** Verwenden Sie nicht den Namen **unique**, da dieser von VuGen verwendet wird.
Parametertyp	Der Typ des Parameters. Informationen zur den verschiedenen Parametertypen finden Sie unter "Parametertypen" auf Seite 439.
Ursprünglicher Wert	Der ursprüngliche Wert des Parameters vor der Parametrierung.
Eigenschaften	Öffnet das Dialogfeld **Parametereigenschaften**. Weitere Informationen finden Sie unter "Dialogfeld "Parametereigenschaften"" unten.

Dialogfeld "Parametereigenschaften"

In diesem Dialogfeld können Sie die Eigenschaften eines Parameters anzeigen und ändern. Der Inhalt dieses Dialogfelds ist je nach Typ des verwendeten Parameters unterschiedlich.



Zugriff	**VuGen > Mit der rechten Maustaste auf den Parameter klicken > Parametereigenschaften**

Parameter "Datum/Zeit", "Gruppenname", "Iterationsnummer", "Lastgeneratorname" und "Vuser-ID"

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Format hinzufügen ->	Fügt das benutzerdefinierte Format, das im Feld **Datum/Uhrzeit-Format** oder **Textformat** angegeben wurde, zur Formatliste hinzu.
Format löschen <-	Löscht das ausgewählte Format aus der Formatliste.
Formate zurücksetzen	Setzt die Formatliste auf ihre Standardwerte zurück.
Datum/Uhrzeit-Format / Textformat	Hier können Sie ein benutzerdefiniertes Format angeben. In der Tabelle unten finden Sie eine Liste der Symbole für Datum/Uhrzeit.
Formatliste	Die Liste der Formate. In der Tabelle unten finden Sie eine Liste der Symbole für Datum/Uhrzeit.

Offset (nur Typ "Datum/Zeit")	Ermöglicht das Festlegen eines Offsets für den Parameter **Datum/Zeit**. Wenn Sie beispielsweise im nächsten Monat ein Datum testen möchten, stellen Sie den Datumoffset auf 30 Tage ein.
	• **Nur Arbeitstage.** Es werden nur Werte für Arbeitstage verwendet (Samstage und Sonntage werden ausgeschlossen).
	Hinweis: Um die arbeitsfreien Tage zu ändern, konfigurieren Sie den NonWorkingDays-Parameter im Abschnitt "Misc" in der Datei "vugen.ini":
	`[Misc]`
	`NonWorkingDays="5,6"`
	Tage werden folgendermaßen durch ganze Zahlen dargestellt:
	Mon = 1, Die = 2, Mit = 3 ,Do = 4, Fr = 5, Sa = 6, Son = 7
	• **Vor aktuellem Datum.** Legt den Offset für ein Datum oder eine Uhrzeit fest, das bzw. die in der Vergangenheit liegt (negativer Offset).

Element der Oberfläche	Beschreibung
Parametertyp	Der Parametertyp. Weitere Informationen finden Sie unter "Parametertypen" auf Seite 439.
Beispiel (aktuelle Zeit)	Zeigt einen Beispielparameterwert auf Grundlage des ausgewählten Formats an.
Wert aktualisieren bei	• **Each Occurrence.** Für jedes Vorkommen des Parameters im Skript wird ein neuer Wert verwendet. Dies ist nützlich, wenn die den Parameter verwendenden Anweisungen voneinander unabhängig sind. So kann es beispielsweise für Zufallsdaten sinnvoll sein, für jedes Parametervorkommen einen neuen Wert zu verwenden. • **Each Iteration.** Der Parameter wird einmal pro Iteration aktualisiert. Wenn ein Parameter mehrmals in einem Skript vorkommt, verwendet der Vuser den gleichen Wert für alle Parametervorkommen der gesamten Iteration. Dies ist nützlich, wenn die den Parameter verwendenden Anweisungen zueinander in Bezug stehen. **Hinweis:** Sie erstellen einen Aktionsblock mit Parametern und eigener Iterationsanzahl: Wenn Sie VuGen anweisen, die Parameterwerte mit jeder Iteration zu aktualisieren, folgt VuGen der globalen Iteration und nicht der des Blocks. • **Once.** Der Parameterwert wird einmal während des Szenariolaufs aktualisiert. Der Vuser verwendet den gleichen Parameterwert für alle Vorkommen und Iterationen des Parameters. Diese Methode ist für die Arbeit mit Datum- und Uhrzeitangaben nützlich.

In der folgenden Tabelle werden die Symbole für das Datum und die Uhrzeit beschrieben:

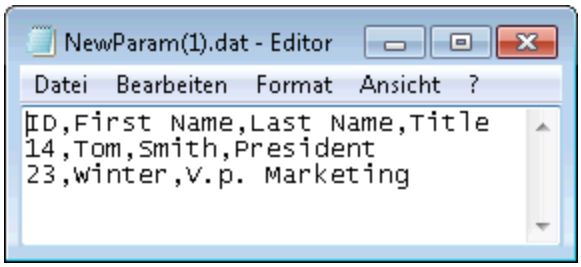
Symbol	Beschreibung
c	Vollständiges Datum und vollständige Uhrzeit als Zahl
#c	Vollständiges Datum als Zeichenfolge und Uhrzeit
H	Stunden (24 Stunden)
I	Stunden (12 Stunden)
M	Minuten
S	Sekunden
p	AM oder PM

Symbol	Beschreibung
d	Tag
m	Monat in Zahlen (01-12)
b	Monat als Zeichenfolge - Kurzformat (Dec)
B	Monat als Zeichenfolge - Langformat (December)
y	Jahr im Kurzformat (03)
Y	Jahr im Langformat (2013)

Parameter des Typs "Datei"

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Fügt dem Datensatz eine Spalte hinzu.
	Fügt dem Datensatz eine Zeile hinzu.
	Erstellt eine neue Datentabelle.
	Öffnet den Datenbankabfrage-Assistenten, mit dem Sie Daten aus einer bestehenden Datenbank importieren können. Weitere Informationen finden Sie in
	Löscht eine Spalte aus dem Datensatz.
	Löscht eine Zeile aus dem Datensatz.

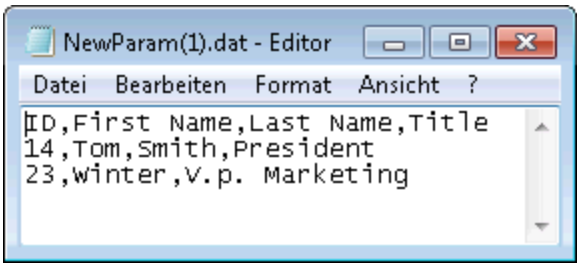
Element der Oberfläche	Beschreibung
Mit Editor bearbeiten...	Ermöglicht es Ihnen, Parameterwerte im Editor anzuzeigen und zu bearbeiten. Dies ist bei der Arbeit mit großen Datensätzen wichtig, da VuGen bis zu 100 Zeilen in der Benutzeroberfläche anzeigt. Editor wird geöffnet, wobei der Parametername in der ersten und der entsprechende ursprüngliche Wert in der zweiten Zeile angezeigt wird. Geben Sie zusätzliche Spaltennamen und -werte in die Datei ein. Verwenden Sie dabei zur Kennzeichnung von Spaltenumbrüchen ein Begrenzungszeichen, wie z. B. ein Komma oder ein Tabulatorzeichen. Beginnen Sie jede Tabellenzeile (jede neue Datenzeile) in einer neuen Zeile. 
Parameter simulieren...	Öffnet das Dialogfeld **Parametersimulation**. Hier können Sie das Parameterverhalten mit Ihrem Datensatz simulieren. Weitere Informationen finden Sie unter "Dialogfeld "Parametersimulation"" auf Seite 466.
Spalte auswählen	Ermöglicht die Auswahl der Spalte, die als Datenquelle verwendet werden soll, nach Spaltennummer oder -name.
Dateiformat	• **Spaltentrennzeichen.** Das Zeichen, durch das Werte in der Datendatei getrennt werden. • **Erste Datenzeile.** Die erste Datenzeile, die während der Ausführung des Vuser-Skripts verwendet werden soll. Die Kopfzeile ist die Zeile 0. Um mit der ersten, auf die Kopfzeile folgenden Zeile zu beginnen, geben Sie 1 an. Ist keine Kopfzeile vorhanden, geben Sie 0 an.
Nächste Zeile auswählen	Die Methode, mit der die Dateidaten während der Ausführung des Vuser-Skripts ausgewählt werden. Diese Optionen sind: **Sequential**, **Random** und **Unique**. Weitere Informationen finden Sie unter "Datenzuweisungsmethoden für die Parametertypen "Datei/Tabelle/XML"" auf Seite 442.
Wert aktualisieren bei	Die Methode, die bestimmt, wann der Parameter zum nächsten Wert wechselt. Sie haben die Wahl zwischen **Each Iteration**, **Each Occurrence** und **Once**. Weitere Informationen finden Sie unter "Datenzuweisungsmethoden für die Parametertypen "Datei/Tabelle/XML"" auf Seite 442.

Element der Oberfläche	Beschreibung
Wenn Werte nicht ausreichen	Geben Sie an, was geschehen soll, wenn keine weiteren eindeutigen Daten mehr vorhanden sind: **Abort Vuser**, **Continue in a cyclic manner** oder **Continue with last value**.
Vuser-Werte im Controller zuordnen	(nur für LoadRunner). Geben Sie an, ob Sie den Vusern manuell Datenblöcke zuordnen möchten. Sie können dem Controller erlauben, automatisch eine Blockgröße zuzuordnen oder aber die gewünschte Anzahl von Werten festlegen. Wählen Sie **Blockgröße automatisch zuordnen** oder **Zuordnen** x **Werte für jeden Vuser** aus. Legen Sie für die zweite Option die Anzahl der zuzuordnenden Werte fest. Um diese Vorkommen zu verfolgen, aktivieren Sie die Option **Erweitertes Protokoll > Parametersubstitution** in den Laufzeiteinstellungen des Protokolls. Wenn nicht genügend Daten vorhanden sind, schreibt VuGen eine Warnmeldung in das Vuser-Protokoll **Keine weiteren eindeutigen Werte für diesen Parameter in Tabelle <*tabellenname*>**.
Dateipfad	Wählen Sie die DAT-Datei mit den Daten für den Parameter aus. Alternativ können Sie über die Schaltfläche **Tabelle erstellen** einen neuen Datensatz erstellen.

Parameter des Typs "Tabelle"

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Elemente der Oberfläche	Beschreibung
Spalte hinzu...	Fügt dem Datensatz eine Spalte hinzu.
Zeile hinzu...	Fügt dem Datensatz eine Zeile hinzu.
Tabelle erstellen	Erstellt eine neue Datentabelle.
Daten-Assistent...	Öffnet den Datenbankabfrage-Assistenten, mit dem Sie Daten aus einer bestehenden Datenbank importieren können. Weitere Informationen finden Sie in
Spalte löschen...	Löscht eine Spalte aus dem Datensatz.
Zeile löschen...	Löscht eine Zeile aus dem Datensatz.

Elemente der Oberfläche	Beschreibung
Mit Editor bearbeiten...	Ermöglicht es Ihnen, Parameterwerte im Editor anzuzeigen und zu bearbeiten. Dies ist bei der Arbeit mit großen Datensätzen wichtig, da VuGen bis zu 100 Zeilen in der Benutzeroberfläche anzeigt. Editor wird geöffnet, wobei der Parametername in der ersten und der entsprechende ursprüngliche Wert in der zweiten Zeile angezeigt wird. Geben Sie zusätzliche Spaltennamen und -werte in die Datei ein. Verwenden Sie dabei zur Kennzeichnung von Spaltenumbrüchen ein Begrenzungszeichen, wie z. B. ein Komma oder ein Tabulatorzeichen. Beginnen Sie jede Tabellenzeile (jede neue Datenzeile) in einer neuen Zeile. 
Vuser-Werte im Controller zuordnen	(nur für LoadRunner). Geben Sie an, ob Sie den Vusern manuell Datenblöcke zuordnen möchten. Sie können dem Controller erlauben, automatisch eine Blockgröße zuzuordnen oder aber die gewünschte Anzahl von Werten festlegen. Wählen Sie **Blockgröße automatisch zuordnen** oder **Zuordnen** x **Werte für jeden Vuser** aus. Legen Sie für die zweite Option die Anzahl der zuzuordnenden Werte fest. Um diese Vorkommen zu verfolgen, aktivieren Sie die Option **Erweitertes Protokoll > Parametersubstitution** in den Laufzeiteinstellungen des Protokolls. Wenn nicht genügend Daten vorhanden sind, schreibt VuGen eine Warnmeldung in das Vuser-Protokoll **Keine weiteren eindeutigen Werte für diesen Parameter in Tabelle <*tabellenname*>**.
Spalte	Die zu verwendenden Spalten. Alternativ können Sie auch **Alle Spalten auswählen** auswählen. Um mehrere Spaltennummern anzugeben, wählen Sie **Spalten nach Nummer** aus und geben die Spaltennummern getrennt durch ein Komma oder einen Bindestrich ein. Die Spaltennummer ist der Index der Spalte, die Ihre Daten enthält. Wenn sich die Daten für den Parameter beispielsweise in der ersten Tabellenspalte befinden, wählen Sie 1 aus. Wählen Sie im Feld **Spaltentrennzeichen** einen Spaltenbegrenzer aus, d. h. das Zeichen, das die Spalten einer Tabelle voneinander abgrenzt. Die verfügbaren Begrenzungszeichen sind: Komma, Tabulator- und Leerzeichen.
Dateipfad	Wählen Sie die DAT-Datei mit den Daten für den Parameter aus. Alternativ können Sie über die Schaltfläche **Tabelle erstellen** einen neuen Datensatz erstellen.

Elemente der Oberfläche	Beschreibung
Zeilentrennzeichen für Protokollanzeige	Dieser Begrenzer dient zur Unterscheidung der Zeilen in den Ausgabeprotokollen. Bei Aktivierung der Parametersubstitutionsprotokollierung sendet VuGen die ersetzten Werte an das Wiedergabeprotokoll. Das Zeilenbegrenzungszeichen im Wiedergabeprotokoll gibt an, wo eine neue Zeile beginnt.
Zeilen	• **Zeilen pro Iteration.** Anzahl der Zeilen, die pro Iteration verwendet werden sollen. Dies ist nur relevant, wenn für das Feld **Wert aktualisieren bei** der Eintrag **Each iteration** festgelegt wurde. Wurde im Feld **Update value on** der Eintrag **Once** ausgewählt, werden für alle Iterationen die gleichen Zeilen verwendet. • **Erste Datenzeile.** Die erste Datenzeile, die während der Skriptausführung verwendet werden soll. Geben Sie 1 ein, wenn Sie mit der ersten Zeile nach der Überschrift beginnen möchten. • Tabelleninformation. Zeigt Informationen zu der Tabelle an, einschließlich der Anzahl der verfügbaren Datenzeilen.
Nächste Zeile auswählen	Die Methode, mit der die Dateidaten während der Ausführung des Vuser-Skripts ausgewählt werden. Diese Optionen sind: **Sequential**, **Random** und **Unique**. Weitere Informationen finden Sie unter "Datenzuweisungsmethoden für die Parametertypen "Datei/Tabelle/XML"" auf Seite 442.
Wert aktualisieren bei	Die Methode, die bestimmt, wann der Parameter zum nächsten Wert wechselt. Sie haben die Wahl zwischen **Each Iteration**, **Each Occurrence** und **Once**. Weitere Informationen finden Sie unter "Datenzuweisungsmethoden für die Parametertypen "Datei/Tabelle/XML"" auf Seite 442.
Wenn nicht genügend Zeilen vorhanden	Gibt an, wie VuGen vorgeht, wenn die Tabelle nicht genügend Zeilen für die Iteration enthält. **Beispiel:** Die auszufüllende Tabelle hat 3 Zeilen, die Daten weisen jedoch lediglich zwei Zeilen auf. Wählen Sie **Der Parameter erhält weniger Zeilen als erforderlich** aus, wenn nur zwei Zeilen ausgefüllt werden sollen. Wählen Sie **Funktionsweise wie bei "Nächste Zeile auswählen"**, um die nächste Zeile gemäß der im Feld **Nächste Zeile auswählen** zu bearbeiten.
Wenn Werte nicht ausreichen	Geben Sie an, was geschehen soll, wenn keine weiteren eindeutigen Daten mehr vorhanden sind: **Abort Vuser**, **Continue in a cyclic manner** oder **Continue with last value**.

Parameter des Typs "Zufallszahl"

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Zahlenformat	Gibt die Mindestanzahl der Ziffern des Parameters an. **%01d** steht für eine Ziffer, **%02d** für zwei Ziffern usw.
Zufallsbereich	Der minimale und der maximale Bereich für die Zufallswerte.
Beispielwert	Zeigt einen Beispielparameterwert auf Grundlage des ausgewählten Formats an.
Wert aktualisieren bei	• **Each Occurrence.** Für jedes Vorkommen des Parameters im Skript wird ein neuer Wert verwendet. Dies ist nützlich, wenn die den Parameter verwendenden Anweisungen voneinander unabhängig sind. So kann es beispielsweise für Zufallsdaten sinnvoll sein, für jedes Parametervorkommen einen neuen Wert zu verwenden. • **Each Iteration.** Der Parameter wird einmal pro Iteration aktualisiert. Wenn ein Parameter mehrmals in einem Skript vorkommt, verwendet der Vuser den gleichen Wert für alle Parametervorkommen der gesamten Iteration. Dies ist nützlich, wenn die den Parameter verwendenden Anweisungen zueinander in Bezug stehen. **Hinweis:** Sie erstellen einen Aktionsblock mit Parametern und eigener Iterationsanzahl: Wenn Sie VuGen anweisen, die Parameterwerte mit jeder Iteration zu aktualisieren, folgt VuGen der globalen Iteration und nicht der des Blocks. • **Once.** Der Parameterwert wird einmal während des Szenariolaufs aktualisiert. Der Vuser verwendet den gleichen Parameterwert für alle Vorkommen und Iterationen des Parameters. Diese Methode ist für die Arbeit mit Datum- und Uhrzeitangaben nützlich.

Parameter des Typs "Eindeutige Zahl"

Hinweis: Beim Planen eines Szenarios im Controller gilt die Option **Wenn Werte nicht ausreichen** nur für die Option **(HH:MM:SS) lang ausführen** in der Registerkarte für die Dauer des Zeitplangenerators. Für die Option **Bis Ende ausführen** wird sie nicht berücksichtigt.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Zahlenformat	Gibt die Mindestanzahl der Ziffern des Parameters an. **%01d** steht für eine Ziffer, **%02d** für zwei Ziffern usw.

Element der Oberfläche	Beschreibung
Zahlenbereich	• **Start.** Der Startwert. • **Blockgröße pro Vuser.** Die Anzahl der eindeutigen Zahlen, die den einzelnen Vusern zugewiesen werden. Beispiel: Wenn Sie den Startwert 1 und die Blockgröße 100 angeben, kann der erste Vuser die Zahlen 1 bis 100 verwenden, der zweite Vuser die Zahlen 101 bis 200 usw.
Beispielwert	Zeigt einen Beispielparameterwert auf Grundlage des ausgewählten Formats an.
Wert aktualisieren bei	• **Each Occurrence.** Für jedes Vorkommen des Parameters im Skript wird ein neuer Wert verwendet. Dies ist nützlich, wenn die den Parameter verwendenden Anweisungen voneinander unabhängig sind. So kann es beispielsweise für Zufallsdaten sinnvoll sein, für jedes Parametervorkommen einen neuen Wert zu verwenden. • **Each Iteration.** Der Parameter wird einmal pro Iteration aktualisiert. Wenn ein Parameter mehrmals in einem Skript vorkommt, verwendet der Vuser den gleichen Wert für alle Parametervorkommen der gesamten Iteration. Dies ist nützlich, wenn die den Parameter verwendenden Anweisungen zueinander in Bezug stehen. **Hinweis:** Sie erstellen einen Aktionsblock mit Parametern und eigener Iterationsanzahl: Wenn Sie VuGen anweisen, die Parameterwerte mit jeder Iteration zu aktualisieren, folgt VuGen der globalen Iteration und nicht der des Blocks. • **Once.** Der Parameterwert wird einmal während des Szenariolaufs aktualisiert. Der Vuser verwendet den gleichen Parameterwert für alle Vorkommen und Iterationen des Parameters. Diese Methode ist für die Arbeit mit Datum- und Uhrzeitangaben nützlich.
Wenn Werte nicht ausreichen	Legt die Vorgehensweise für den Fall fest, dass der Wertebereich für einen Vuser erreicht ist. Der Wertebereich wird durch den Startwert und die Blockgröße bestimmt. **Abort Vuser.** Beendet das Vuser-Skript. **Continue in a cyclical manner.** Für diesen Vuser werden wieder die eindeutigen Zahlen von Beginn des zugewiesenen Bereichs an verwendet. Beispiel: Wurde einem Vuser der Bereich von 1 bis 100 zugewiesen und ist die Zahl 100 erreicht worden, wird wieder bei 1 begonnen. **Continue with last value** Der letzte zugewiesene Wert für diesen Parameter wird für alle nachfolgenden Vorkommen dieses Parameters verwendet. Beispiel: Wurde einem Vuser der Bereich von 1 bis 100 zugewiesen und ist die Zahl 100 erreicht worden, würde weiterhin bis zum Ende des Skripts der Wert 100 verwendet.

Benutzerdefinierte Funktionsparameter

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Funktionsname	Die Bezeichnung der Funktion. Verwenden Sie dabei den in der DLL-Datei angezeigten Funktionsnamen.
Bibliotheksnamen	Der Speicherort der relevanten Bibliotheksdateien.
Wert aktualisieren bei	• **Each Occurrence.** Für jedes Vorkommen des Parameters im Skript wird ein neuer Wert verwendet. Dies ist nützlich, wenn die den Parameter verwendenden Anweisungen voneinander unabhängig sind. So kann es beispielsweise für Zufallsdaten sinnvoll sein, für jedes Parametervorkommen einen neuen Wert zu verwenden. • **Each Iteration.** Der Parameter wird einmal pro Iteration aktualisiert. Wenn ein Parameter mehrmals in einem Skript vorkommt, verwendet der Vuser den gleichen Wert für alle Parametervorkommen der gesamten Iteration. Dies ist nützlich, wenn die den Parameter verwendenden Anweisungen zueinander in Bezug stehen. **Hinweis:** Sie erstellen einen Aktionsblock mit Parametern und eigener Iterationsanzahl: Wenn Sie VuGen anweisen, die Parameterwerte mit jeder Iteration zu aktualisieren, folgt VuGen der globalen Iteration und nicht der des Blocks. • **Once.** Der Parameterwert wird einmal während des Szenariolaufs aktualisiert. Der Vuser verwendet den gleichen Parameterwert für alle Vorkommen und Iterationen des Parameters. Diese Methode ist für die Arbeit mit Datum- und Uhrzeitangaben nützlich.

XML-Parameter

Informationen zu XML-Parametern für Webdienste finden Sie unter "XML-Parameter" auf Seite 446.

Dialogfeld "Parametersimulation"

In diesem Dialogfeld können Sie eine Simulation des Verhaltens eines Dateiparameters anzeigen.

Zugriff	**VuGen > Parameterliste > Parameter auswählen > Parameter simulieren**

Wichtige Informationen	• Diese Funktion ist nur für Parameter des Typs **Datei** relevant. • Nicht alle Arten der Parametersubstitution können simuliert werden. Bei Auswahl von **Nächste Zeile auswählen: Dieselbe Zeile wie** oder **Wert aktualisieren bei: Jedem Auftreten** wird der File Parameter Simulator nicht geöffnet. • VuGen kann bis zu 256 Iterationen und 256 Vuser simulieren. • **Unbegrenzt ausführen** ist konform mit **Real-World-Plan** im Scheduler des Controllers. • Bei Auswahl von **Nächste Zeile auswählen: Unique** im Dialogfeld **Parameterliste** wird jedem Vuser ein eindeutiger Zeilenbereich zugeordnet, aus dem der Simulator die Ersatzwerte (für den jeweiligen Vuser) nimmt. **Hinweis:** Wenn mehr als ein eindeutiger Parameter vorhanden ist, müssen Sie sicherstellen, dass jeder Parameter über definierte Werte für alle Vuser verfügt. Bei dieser Einstellung lautet die Standardauswahl für die Zuweisung von Vuser-Werten im Controller-Bereich **Blockgröße automatisch zuordnen**. Beim Ausführen der Simulation erfolgt die Bereichszuordnung in diesem Fall entsprechend der Auswahl für den Ausführungsmodus des Szenarios. Wenn Sie die Standardauswahl in **Zuordnen: x Werte für jeden Vuser** ändern, wird den Vusern die von Ihnen festgelegte Anzahl von Werten zugeordnet und die Auswahl für den Szenarioausführungsmodus wird nicht berücksichtigt.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Vuser	Die Anzahl der Vuser, die in der Simulation ausgeführt werden sollen.
Szenarioausführungsmodus	• **Bis Ende ausführen.** Geben Sie die Anzahl der auszuführenden Iterationen ein oder wählen Sie **Iterationsanzahl aus den Laufzeiteinstellungen übernehmen** aus. • **Unbegrenzt ausführen.** Simuliert die Option zur unbegrenzten Ausführung im Controller. VuGen simuliert tatsächlich nur die angegebene Iterationsanzahl.
Simulieren	Führt die Parametersimulation aus. Die Werte der einzelnen Parametersubstitutionen werden angezeigt.

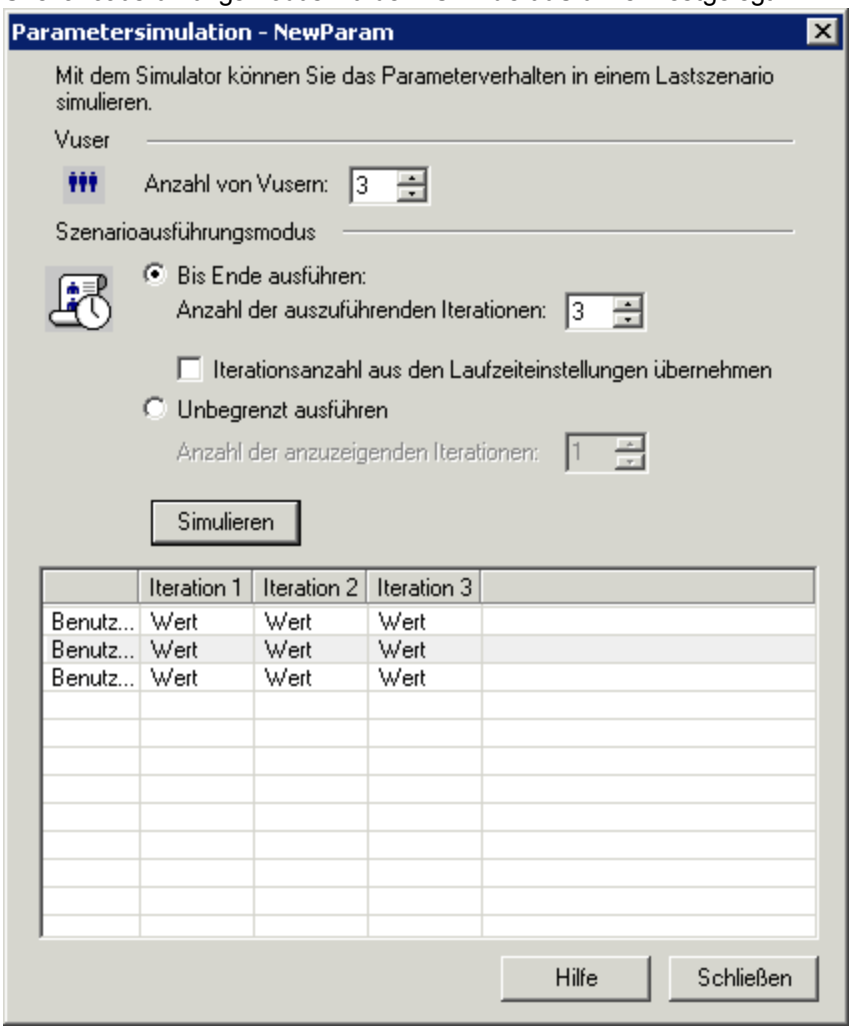
Beispiel:

Für die folgenden Beispiele gelten die nachstehenden Einstellungen im Dialogfeld **Parameterliste**:

- **Werte für den neuen Parameter**: **Value1** bis **Value7**
- **Nächste Zeile auswählen**: Unique
- **Wenn Werte nicht ausreichen**: **Continue with last value**
- **Vuser-Werte im Controller zuordnen**: **Blockgröße automatisch zuordnen**

Szenarioausführungsmodus: "Ausführen bis Ende"

Im folgenden Beispiel wurden drei Vuser und drei Iterationen ausgewählt und für den Szenarioausführungsmodus wurde **Bis Ende ausführen** festgelegt.



Wenn im Szenarioausführungsmodus **Bis Ende ausführen** eingestellt wurde, entspricht die Anzahl der Zeilen, die jeder Vuser erhält, der Anzahl der Iterationen. Die Bereichszuordnung wird beendet, wenn die Tabelle nicht mehr genügend Zeilen umfasst.

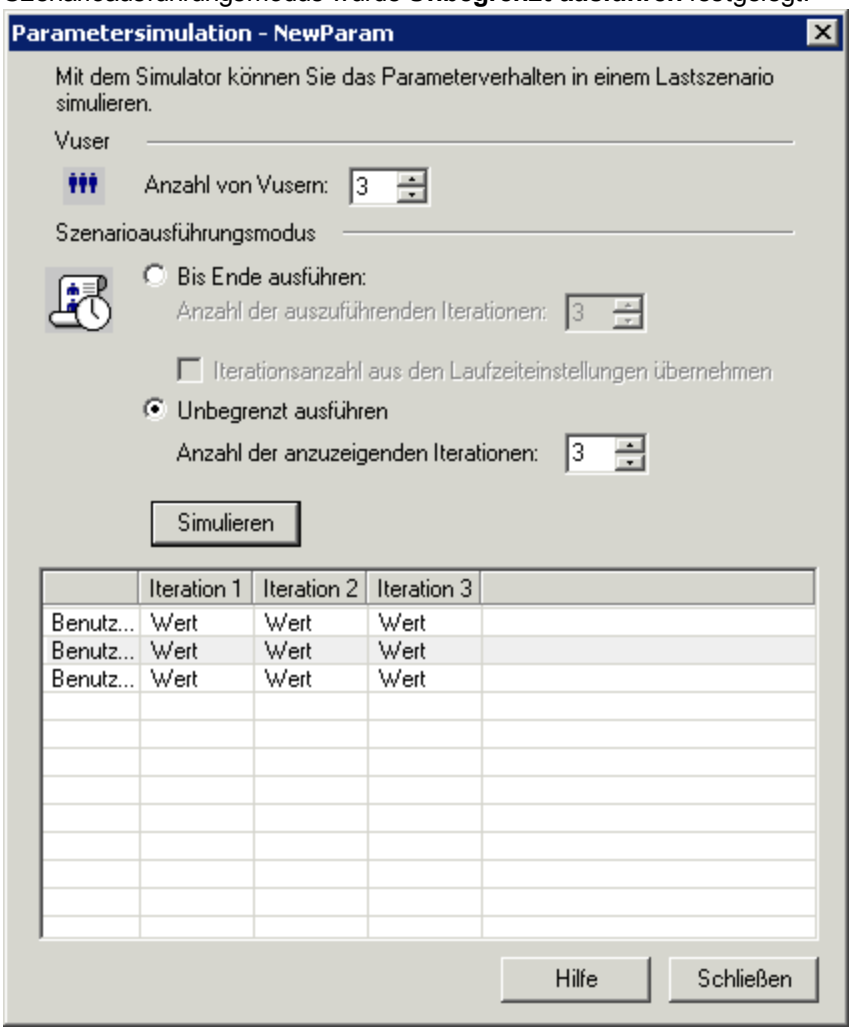
Bei Ausführung der Simulation erhält der erste Vuser die ersten drei Werte (die Anzahl der Iterationen). Der zweite Vuser erhält die nächsten drei Werte. Dem dritten Vuser wird in der ersten

Iteration der verbleibende Wert zugeteilt. Da im Dialogfeld **Parameterliste** für die Option **Wenn Werte nicht ausreichen** die Einstellung **Continue with last value** festgelegt wurde, wird der dritte Vuser mit dem gleichen Wert fortgesetzt.

Ein vierter Vuser würde fehlschlagen.

Szenarioausführungsmodus: Unbegrenzt ausführen

Im folgenden Beispiel wurden drei Vuser und drei Iterationen ausgewählt und für den Szenarioausführungsmodus wurde **Unbegrenzt ausführen** festgelegt.



Wenn für den Szenarioausführungsmodus **Unbegrenzt ausführen** festgelegt wurde, wird der jedem Vuser zugeordnete Bereich durch Division der Zellenanzahl in der .dat-Datei durch die Anzahl der Vuser berechnet. In diesem Szenario lautet die Rechnung 7/3 = 2 (der Simulator verwendete die nächstkleinste Ganzzahl).

Bei Ausführung der Simulation erhält der erste Vuser Wert1 und Wert2. Der zweite Vuser erhält Wert3 und Wert4 und der dritte Vuser Wert5 und Value6. Da nur 3 Vuser vorhanden sind, wird Wert7 nicht verteilt.

Hinweis: Wenn Sie den Mauszeiger über die Zellen der ersten Tabellenspalte führen, wird eine QuickInfo anzeigt, die Informationen zu den Werten anzeigt, die dem jeweiligen Vuser zugeordnet wurden.

Führen Sie den Mauszeiger über Zellen, denen keine Werte zugeordnet wurden, können Sie der angezeigten QuickInfo entnehmen, warum keine Zuordnung erfolgte.

Wenn der korrekte Wert zugeordnet wurde, wird keine QuickInfo angezeigt.

Dialogfeld "Parameterliste"

Dieses Dialogfeld dient zum Anzeigen, Erstellen, Löschen, Auswählen und Ändern von Parametern. Die Parameterliste zeigt alle von Ihnen erstellten Parameter an, einschließlich der Eingabe- und Ausgabeparameter.

Zugriff	Führen Sie einen der folgenden Schritte durch: • Bereich **VuGen > Lösungs-Explorer >** Knoten **Parameter** > **Parameterliste** • Klicken Sie im Script-Editor mit der rechten Maustaste auf den Wert > **Durch Parameter ersetzen** > **Parameterliste** • **VuGen > Entwurf > Parameter > Parameterliste**
Wichtige Informationen	Nennen Sie einen Parameter nicht *unique*, da dieser Name von VuGen verwendet wird.
Siehe auch	"Dialogfeld "Parametereigenschaften"" auf Seite 456

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
Neu	Erstellt einen neuen Parameter. Markierter Text wird nicht durch den Parameter ersetzt. Nennen Sie einen Parameter nicht *unique*, da dieser Name von VuGen verwendet wird.
Löschen	Löscht den ausgewählten Parameter. **Hinweis**: Wenn der Parameter einen vorherigen Wert ersetzt hat, wird der ursprüngliche Wert wiederhergestellt.

<Bereich der Parametereigenschaften>	Der Inhalt dieses Bereichs ist je nach Typ des verwendeten Parameters unterschiedlich. Informationen zu diesem Bereich finden Sie in den folgenden Abschnitten: • Informationen zu den Parametern des Typs **Datum/Zeit**, **Gruppenname**, **Iterationsnummer**, **Lastgeneratorname** und **Vuser-ID** finden Sie unter "Dialogfeld "Parametereigenschaften"" auf Seite 456. • Informationen zu Parametern des Typs **Datei** finden Sie unter "Dialogfeld "Parametereigenschaften"" auf Seite 456. • Informationen zu Parametern des Typs **Tabelle** finden Sie unter "Dialogfeld "Parametereigenschaften"" auf Seite 456. • Informationen zu Parametern des Typs **BPT** finden Sie unter "Dialogfeld "Parametereigenschaften"" auf Seite 456. • Informationen zu Parametern des Typs **Zufallszahl** finden Sie unter "Dialogfeld "Parametereigenschaften"" auf Seite 456. • Informationen zu Parametern des Typs **Eindeutige Zahl** finden Sie unter "Dialogfeld "Parametereigenschaften"" auf Seite 456. • Informationen zu Parametern des Typs **Benutzerdefinierte Funktion** finden Sie unter "Dialogfeld "Parametereigenschaften"" auf Seite 456. • Informationen zu Parametern des Typs **XML** finden Sie unter "Dialogfeld "Parametereigenschaften"" auf Seite 456.
Parametertyp	In dieser Dropdownliste können Sie den Parametertyp auswählen. Informationen zu den verschiedenen Parametertypen finden Sie unter "Parametertypen" auf Seite 439.

Datenbankabfrage-Assistent

In diesem Assistenten können Sie Daten für einen Parameter aus einer bestehenden Datenbank auswählen.

Zugriff	**VuGen > Mit der rechten Maustaste auf den Parameter klicken > Parametereigenschaften > Daten-Assistent (nur verfügbar für Parameter der Typen "Datei" und "Tabelle")**
Relevante Aufgaben	"Importieren von Parameterdaten aus einer Datenbank" auf Seite 454

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Abfragedefinition	Wählen Sie eine der folgenden Optionen: • Abfrage mit Microsoft Query erstellen • SQL-Anweisung manuell angeben
Anleitung zur Verwendung von Microsoft Query	Zeigt eine Anleitung für die Verwendung von Microsoft Query an, wenn Sie auf **Weiter** klicken.
Maximale Zeilenanzahl	Die maximale Anzahl der in der DAT-Datei zu erstellenden Zeilen basierend auf der angegebenen Abfrage.

Dialogfeld "Parameter erstellen" (Winsock)

Über dieses Dialogfeld können Sie Daten korrelieren und in Winsock-Skripts Parameter erstellen.

Zugriff	**VuGen > Schrittenavigator > Kontextmenü > Parameter erstellen**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Parametername	Der Name des Parameters.
Datenbereich	Sie können den Parameter durch einen Start- und Endbereich in Byte definieren. Geben Sie die Zahlen manuell in die Felder **Von** und **Bis** ein oder klicken Sie auf **Bereich auswählen** und markieren Sie den gewünschten Text.
Begrenzungen	Sie können den Parameter durch Festlegen einer linken und rechten Begrenzung definieren. Wählen Sie dazu **Parameterdaten mithilfe von Begrenzungen extrahieren** aus. Klicken Sie auf die Schaltfläche rechts neben dem Feld **Links**, markieren Sie den gewünschten Text und klicken Sie auf **Fertig**. Wiederholen Sie diesen Schritt für die rechte Begrenzung.
Skriptanweisung	Die Anweisung, die in Ihrem Skript basierend auf den in diesem Dialogfeld ausgewählten Optionen angezeigt wird. Sie können diese Anweisung manuell bearbeiten.

Dialogfeld "Ursprünglicher Wert des Parameters"

Dieses Dialogfeld wird angezeigt, wenn Sie ein Vuser-Skript parametrieren; Sie können den ursprünglichen Wert des Parameters angeben.

Zugriff	Wählen Sie in VuGen eine Textzeichenfolge im Editor aus. Klicken Sie mit der rechten Maustaste und wählen Sie **Durch Parameter ersetzen > Parameterliste** aus. Wählen Sie einen vorhandenen Parameter aus und klicken Sie auf **Schließen**.
Wichtige Informationen	Jeder Parameter verfügt über einen ursprünglichen Wert. Sie können jeden Parameter in einem Vuser-Skript durch den ursprünglichen Wert des Parameters ersetzen. Wenn Sie eine ausgewählte Textzeichenfolge parametrieren und die ausgewählte Textzeichenfolge nicht mit dem ursprünglichen Wert des ausgewählten Parameters übereinstimmt, können Sie entweder den vorhandenen ursprünglichen Wert des Parameters beibehalten oder den ursprünglichen Wert durch die ausgewählte Textzeichenfolge ersetzen.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Alten ursprünglichen Wert <Text> verwenden	Der vorhandene ursprüngliche Wert des Parameters wird beibehalten.
Neuen ursprünglichen Wert <Text> verwenden	Der ausgewählte Text wird als neuer ursprünglicher Wert des Parameters zugewiesen.

Parameter - Fehlerbehebung und Einschränkungen

In diesem Abschnitt werden die Fehlerbehebung und die für Parameter geltenden Einschränkungen beschrieben.

Einschränkungen für Funktionsargumente

Eine Parametrierung ist nur für die Argumente einer Funktion möglich. Textfolgen, die keine Funktionsargumente darstellen, können nicht parametriert werden. Außerdem können nicht alle Funktionsargumente parametriert werden. Weitere Informationen zu parametrierbaren Argumenten finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**) zu der jeweiligen Funktion.

Betrachten Sie beispielsweise die Funktion **lrd_stmt**. Die Funktion weist folgende Syntax auf:

```
lrd_stmt (LRD_CURSOR FAR *mptCursor, char FAR *mpcText, long mliTextLen, LRDOS_I
NT4 mjOpt1, LRDOS_INT4 mjOpt2, int miDBErrorSeverity);
```

Aus der Function Reference geht hervor, dass nur das Argument *mpcText* parametriert werden kann.

Eine aufgezeichnete **lrd_stmt**-Funktion könnte wie folgt aussehen:

```
lrd_stmt(Csr4, "select name from sysobjects where name =\"Kim\" ", -1, 148, -999
99, 0);
```

Die aufgezeichnete Funktion könnte wie folgt parametiert werden:

```
lrd_stmt(Csr4, "select name from sysobjects where name =\"<name>\" ", -1, 148, -
99999, 0);
```

Hinweis: Die Funktion **lr_eval_string** bietet die Möglichkeit, Funktionsargumente zu "parametrieren", für die die Standardparametrierung nicht greift. Mithilfe der Funktion **lr_eval_string** können Sie außerdem jede beliebige Zeichenfolge in einem Vuser-Skript "parametrieren".

Für VB, COM und Microsoft .NET-Protokolle muss die Funktion **lr.eval string** zur Definition von Parametern verwendet werden. Beispiel:
lr.eval_string("{**Custom_param**}").

Weitere Informationen zur Funktion **lr_eval_string** finden Sie in der Function Reference.

Größenbeschränkungen bei Datentabellendateien

Wenn die Größe einer .dat-Datei 100 MB überschreitet, werden Sie in einer Meldung darauf hingewiesen, dass die Datei zu groß ist und nicht angezeigt werden kann.

Wenn Sie eine Datei mit über 100 MB laden müssen, können Sie den Parameter "MaxParametersDisplaySize" in vugen.ini ändern:

```
[ParamTable]

MaxParametersDisplaySize=100000000
```

Async Studio

Synchrone und asynchrone Konzepte

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Ursprünglich erfolgte die Kommunikation bei Webanwendungen mit synchronen Unterhaltungen. Eine typische synchrone Unterhaltung enthält die folgenden Schritte:

1. Der Benutzer interagiert mit einer Anwendung, die in einem Webbrowser abläuft.
2. Auf der Grundlage der Benutzereingabe sendet der Browser eine Anfrage an den Webserver.
3. Der Server sendet eine Antwort auf die Anfrage und die Anwendung im Browser wird aktualisiert.

Synchrone Anwendungen unterliegen einer Reihe von Einschränkungen. Eine dieser Einschränkungen betrifft die Aktualisierung der Daten, die in der im Browser laufenden Anwendung angezeigt werden. Denken Sie z. B. an eine Anwendung, in der die Kurse verschiedener

Wertpapiere angezeigt werden. Idealerweise sollte die Anwendung die aktualisierten Aktienkurse in dem Moment anzeigen können, in dem sie auf dem Webserver aktualisiert werden. Eine synchrone Anwendung könnte die Kurse in einem festgelegten Zeitintervall aktualisieren. Der Browser könnte z. B. alle 10 Sekunden eine Anfrage an den Server senden, um die neuesten Aktienkurse zu erhalten. Diese Lösung ist mit der Einschränkung verbunden, dass die angezeigten Aktienkurse schon veraltet sein können, bevor das Aktualisierungsintervall abgelaufen ist. Dies mag für das hier vorgestellte Aktienszenario nicht von kritischer Bedeutung sein, es demonstriert aber die begrenzte Fähigkeit einer synchronen Anwendung, Informationen rechtzeitig zu aktualisieren.

Wo dies erforderlich ist, werden synchrone Anwendungen durch Anwendungen ersetzt, die als *asynchrone* Anwendungen bezeichnet werden. Bei asynchronen Anwendungen kann der Client benachrichtigt werden, wann immer ein Ereignis auf der Serverseite stattfindet. Asynchrone Anwendungen sind deshalb eher in der Lage, Informationen umgehend zu aktualisieren.

Asynchrones Verhalten setzt voraus, dass die asynchrone Kommunikation parallel (gleichzeitig) zum hauptsächlichen, synchronen Ablauf der Geschäftsprozesse stattfindet. Deshalb sind asynchrone Anwendungen mit den herkömmlichen synchronen Vuser-Skripts schwerer exakt zu emulieren.

Die zahlreichen Typen asynchroner Anwendungen lassen sich in drei Haupttypen einordnen: *Push*, *Poll* und *Long-Poll*. Weitere Informationen finden Sie unter "Typen asynchroner Kommunikation" unten.

Eine Einführung zur Verwendung asynchroner Kommunikation in Vuser-Skripts finden Sie unter "LoadRunner-Support für asynchrone Kommunikation - Übersicht" auf Seite 478.

Typen asynchroner Kommunikation

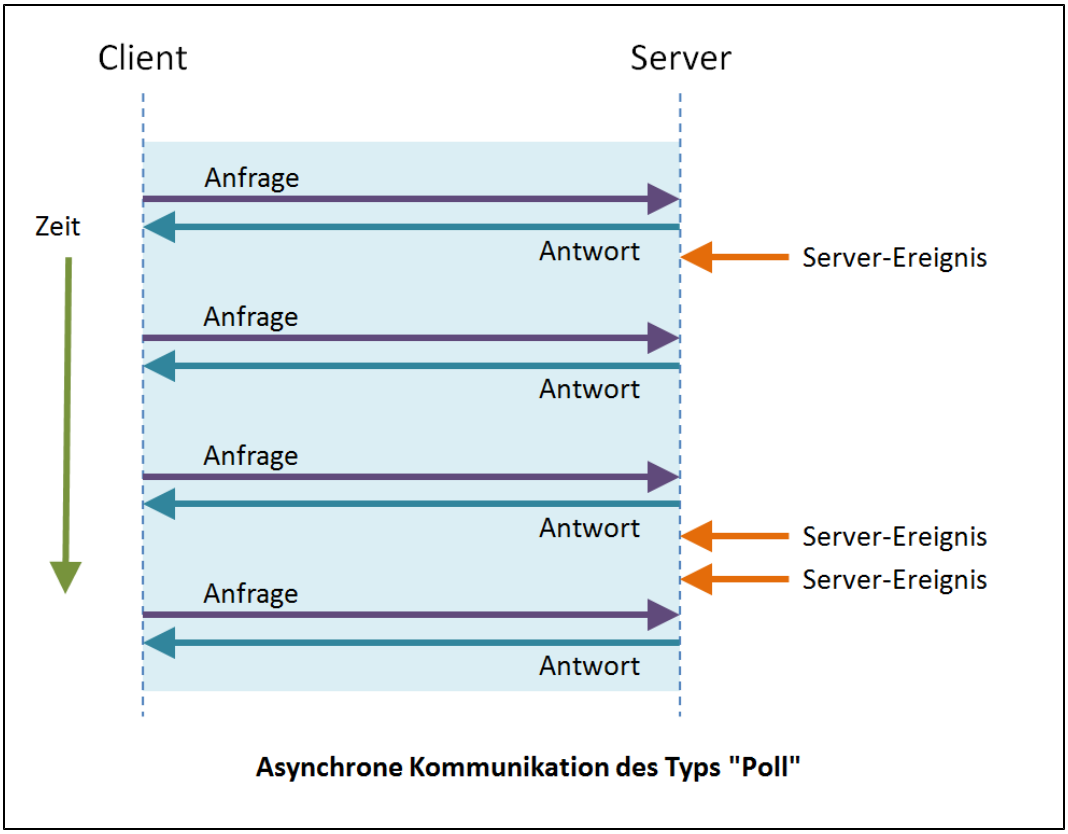
Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Sequenzen asynchroner Anforderungen und Antworten

Die asynchrone Kommunikation umfasst verschiedene Anforderungs- und Antwortsequenzen. Diese Anforderungs- und Antwortsequenzen können als eine von drei Typen asynchroner Kommunikation klassifiziert werden: *Poll*, *Push* und *Long-Poll*. Zu wissen, welche Typen asynchroner Kommunikation bei der Ausführung der erforderlichen Geschäftsprozesse implementiert werden, gereicht Ihnen beim Entwickeln eines Vuser-Skripts zum Vorteil.

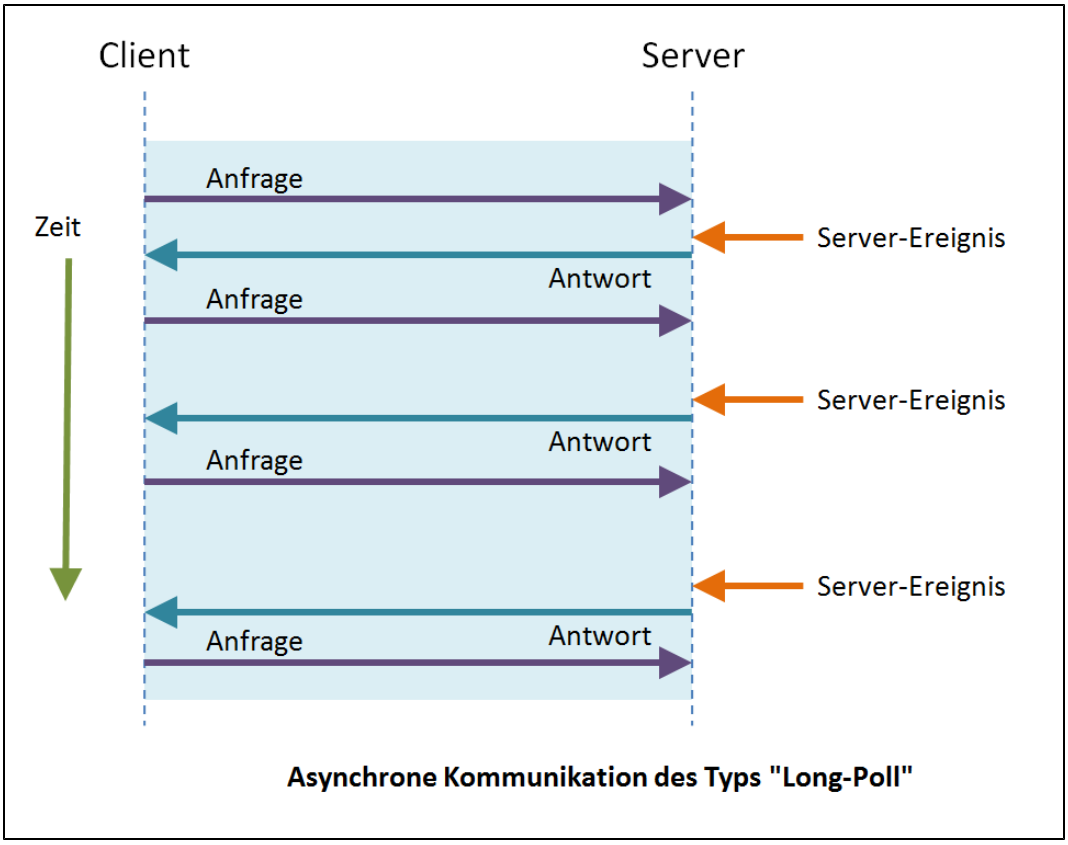
Asynchrone Kommunikation des Typs "Poll"

Der Browser sendet in regelmäßigen Intervallen, z. B. alle 5 Sekunden, HTTP-Anforderungen an den Server. Der Server antwortet mit Aktualisierungen. Dies ermöglicht es dem System, die im Browser laufende Anwendung periodisch zu aktualisieren. Liegen auf dem Server keine neueren Daten vor, wird die Anwendung auf der Basis des Anwendungsprotokolls informiert, dass keine Aktualisierung vorhanden ist.



Asynchrone Kommunikation des Typs "Long-Poll"

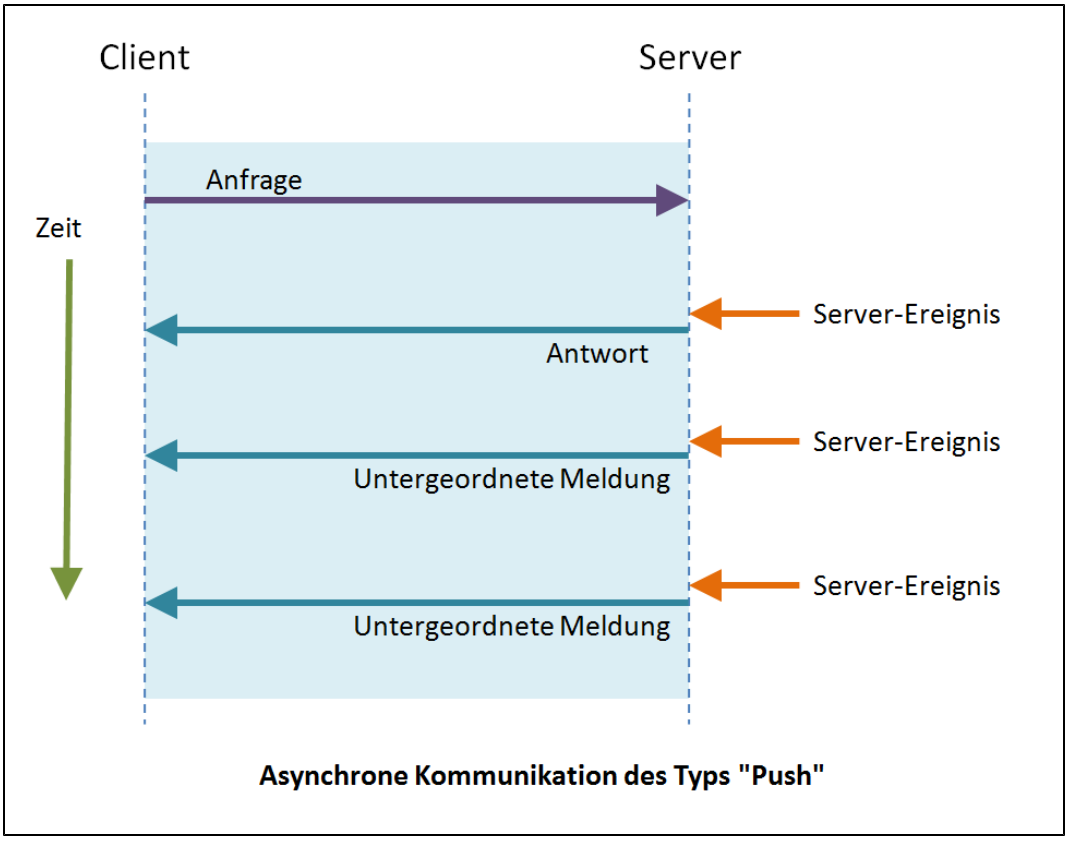
Der Client sendet eine HTTP-Anforderung an eine bekannte Adresse auf dem Server. Wenn auf dem Server neuere Daten vorliegen, antwortet er mit einer HTTP-Antwort. Unmittelbar nach dem Erhalt der Serverantwort übermittelt der Client eine weitere Anforderung.



Asynchrone Kommunikation des Typs "Push"

Der Client öffnet eine Verbindung mit dem Server, indem er eine einzelne HTTP-Anforderung an eine bekannte Adresse auf dem Server sendet. Der Server sendet eine Antwort, die nie zu enden scheint, sodass der Client die Verbindung niemals schließt. Wann immer sich die Notwendigkeit ergibt, sendet der Server über die offene Verbindung eine "Teilnachricht"-Aktualisierung an den Client. Der Server kann diese Verbindung beenden oder nicht beenden. Wenn der Server keine aktualisierten Daten anzubieten hat, sendet er "Ping"-Nachrichten an den Client, solange die Verbindung offen ist, um zu verhindern, dass der Client die Verbindung wegen einer Zeitüberschreitung schließt.

Beachten Sie, dass Push-Unterhaltungen zwar für Web-Protokoll-Aktionen (HTTP/HTML) in Web (HTTP/HTML)-, Flex-, Silverlight- und Webdienst-Vuser-Skripts unterstützt werden, NICHT jedoch für **Flex_amf_call**-Schritte in Flex-Vuser-Skripts.



LoadRunner-Support für asynchrone Kommunikation - Übersicht

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Webanwendungen können synchrones Verhalten, asynchrones Verhalten oder eine Kombination aus beiden aufweisen. Eine Einführung zu diesen Verhaltenstypen finden Sie unter "Synchrone und asynchrone Konzepte" auf Seite 474. Mit LoadRunner können Sie Vuser-Skripts erstellen und ausführen, die Benutzeraktivitäten für synchrone und asynchrone Anwendungen emulieren. Zum Erstellen eines Vuser-Skripts für eine synchrone Anwendung verwenden Sie den normalen Erstellungsprozess für LoadRunner-Vuser-Skripts. Bei Skripts für eine asynchrone Anwendung müssen Sie jedoch einige zusätzliche Aufgaben durchführen, die über den normalen Erstellungsprozess für LoadRunner-Vuser-Skripts hinausgehen. Wenn Sie ein Skript für eine Anwendung mit asynchronem Verhalten aufzeichnen und generieren, ohne diese zusätzlichen Schritte durchzuführen, wird das Skript möglicherweise nicht korrekt ausgeführt.

Die Erstellung eines Vuser-Skripts für eine asynchrone Anwendung beginnt mit der Aufzeichnung der Geschäftsprozesse, die die asynchrone Kommunikation erzeugen. Nachdem die Geschäftsprozesse aufgezeichnet und das Vuser-Skript erstellt wurde, überprüft VuGen das generierte Skript und versucht, die asynchrone Kommunikation zu erkennen. Wird eine asynchrone

Kommunikation gefunden, ändert VuGen das Skript, indem es die zutreffenden asynchronen API-Funktionen einfügt.

Erkennen asynchroner Kommunikation

Damit VuGen das asynchrone Verhalten in einem Vuser-Skript zuverlässig erkennt, muss die asynchrone Kommunikation zumindest die erforderliche Mindestanzahl von Anfrage- und Antwortsequenzen enthalten.

- Erkennen einer Kommunikation des Typs **Poll**

  Um VuGen in die Lage zu versetzen, eine Poll-Unterhaltung richtig zu identifizieren, muss das aufgezeichnete Vuser-Skript mindestens 3 Sequenzen mit übereinstimmenden URLs und ähnlichen Poll-Intervallen enthalten.

- Erkennen einer Kommunikation des Typs **Long-Poll**

  Um VuGen in die Lage zu versetzen, eine Long-Poll-Unterhaltung richtig zu identifizieren, muss das aufgezeichnete Vuser-Skript mindestens 3 Sequenzen mit übereinstimmenden URLs enthalten.

Hinweis: VuGen überprüft Skripts nur dann auf asynchrone Kommunikation, wenn die Aufzeichnungsoption **Asynchrone Prüfung** ausgewählt ist. Weitere Informationen finden Sie unter "Erstellen eines asynchronen Vuser-Skripts" auf der nächsten Seite.

Bei einigen Szenarios reichen die von VuGen am Vuser-Skript vorgenommenen Änderungen aus, damit das Skript ausgeführt und das asynchrone Verhalten emuliert werden kann. Bei anderen Szenarios sind "manuelle" Änderungen erforderlich. Weitere Informationen finden Sie unter "Änderung eines Vuser-Skripts durch VuGen für die asynchrone Kommunikation" auf Seite 484.

Hinweis: Die Änderungen, die an einem generierten Vuser-Skript vorgenommen werden müssen, damit das asynchrone Verhalten emuliert wird, hängen vom Typ des asynchronen Verhaltens ab: *Push*, *Poll* oder *Long-Poll*. Weitere Informationen finden Sie unter "Typen asynchroner Kommunikation" auf Seite 475.

Die asynchrone Kommunikation in einem Vuser-Skript ist in eine oder mehrere Unterhaltungen unterteilt. Die einzelnen asynchronen Unterhaltungen, die VuGen in einem Vuser-Skript entdeckt, werden auf der Registerkarte **Asynchron** in **Design Studio** aufgelistet. Verwenden Sie diese Liste der asynchronen Unterhaltungen, um die Änderungen zu analysieren, die VuGen am Vuser-Skript vorgenommen hat, um sicherzustellen, dass VuGen das asynchrone Verhalten korrekt erkannt und das Vuser-Skript so geändert hat, dass dieses asynchrone Verhalten korrekt emuliert wird. Weitere Informationen zur Registerkarte **Asynchron** von **Design Studio** finden Sie unter "Registerkarte "Asynchron" [Design Studio]" auf Seite 504.

Nachdem Sie ein Vuser-Skript geändert haben, um mit ihm eine asynchrone Kommunikation zu emulieren, ist es möglicherweise erforderlich, an dem geänderten Skript Korrelationsaktivitäten durchzuführen. Weitere Informationen zur Korrelation finden Sie unter "Korrelieren asynchroner Vuser-Skripts" auf Seite 492.

Hinweis: Die asynchrone Kommunikation wird nicht unterstützt, wenn Sie ein Vuser-Skript im WinINet-Modus wiedergeben.

Informationen zum Erstellen eines Vuser-Skripts für eine Anwendung mit asynchroner Kommunikation finden Sie unter "Erstellen eines asynchronen Vuser-Skripts" unten.

Erstellen eines asynchronen Vuser-Skripts

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Um ein Vuser-Skript für eine asynchrone Anwendung zu erstellen, führen Sie die folgenden Schritte aus:

Erstellen eines neuen Vuser-Skripts

1. Klicken Sie auf die Schaltfläche **Neues Skript** in der VuGen-Symbolleiste.
2. Wählen Sie **Web - HTTP/HTML** oder eines der anderen Vuser-Protokolle, die die asynchrone Kommunikation unterstützen.
3. Klicken Sie auf **Erstellen**. VuGen erstellt ein grundlegendes Vuser-Skript.

Aktivieren der asynchronen Prüfung

1. Wählen Sie **Aufzeichnen > Aufzeichnungsoptionen** aus.
2. Wählen Sie unter **Allgemein** die Option **Codegenerierung** aus.
3. Vergewissern Sie sich, dass das Kontrollkästchen **Asynchrone Prüfung** aktiviert ist. Damit wird VuGen angewiesen, das Vuser-Skript nach der Aufzeichnung auf asynchrone Kommunikation zu überprüfen und die angemessene Funktionalität für asynchrone Kommunikation einzufügen.

Aufzeichnen des Geschäftsprozesses mit dem typischen VuGen-Aufzeichnungsprozess

1. Klicken Sie in der VuGen-Symbolleiste auf **Aufzeichnen**.
2. Geben Sie die erforderlichen Informationen im Dialogfeld **Aufzeichnung starten** ein und klicken Sie dann auf **Aufzeichnung starten**.
3. Führen Sie die Geschäftsprozesse aus, die der Vuser emulieren soll, und klicken Sie dann in der freischwebenden Symbolleiste auf **Aufzeichnung beenden**.

Damit VuGen das asynchrone Verhalten in einem Vuser-Skript zuverlässig erkennt, muss die asynchrone Kommunikation zumindest die erforderliche Mindestanzahl von Clientanfragen und Serverantworten enthalten. Weitere Informationen finden Sie unter "Typen asynchroner Kommunikation" auf Seite 475.

Generieren, Prüfen und Ändern des Vuser-Skripts

1. Nachdem Sie auf **Aufzeichnung beenden** geklickt haben, generiert VuGen das Vuser-Skript.

2. Anschließend wird das generierte Skript auf das Vorhandensein asynchroner Kommunikation überprüft.

3. Werden Hinweise auf asynchrone Kommunikation gefunden, ändert VuGen das Skript, um dessen Ausführung und die Emulation des asnychronen Verhaltens zu ermöglichen. Weitere Informationen finden Sie unter "Änderung eines Vuser-Skripts durch VuGen für die asynchrone Kommunikation" auf Seite 484.

4. **Design Studio** wird geöffnet. Klicken Sie auf die Registerkarte **Asynchron**. Auf der Registerkarte **Asynchron** werden alle Instanzen asynchroner Kommunikation aufgelistet, die VuGen im Vuser-Skript gefunden hat.

Überprüfen der von VuGen am Skript vorgenommenen Änderungen

Führen Sie für jede asynchrone Kommunikation, die auf der Registerkarte **Asynchron** im **Design Studio** aufgeführt ist, die folgenden Aufgaben durch:

1. Öffnen Sie das Vuser-Skript im Editor.

2. Suchen Sie nach dem Schritt **web_reg_async_attributes**, mit dem die asynchrone Kommunikation eingeleitet wird. Vergewissern Sie sich, dass der Schritt **web_reg_async_attributes** am Anfang der asynchronen Kommunikation steht.

3. Vergewissern Sie sich, dass der URL-Parameter im Schritt **web_reg_async_attributes** mit einem der URLs übereinstimmt, die im auf den Schritt **web_reg_async_attributes** folgenden Aktionsschritt angegeben sind.

   Weitere Informationen zum Schritt **web_reg_async_attributes** finden Sie unter "Definieren des Anfangs einer asynchronen Unterhaltung" auf Seite 487.

4. Beachten Sie, dass der Schrittkommentar vor dem **web_reg_async_attributes**-Schritt ein TODO-Token enthält. Das TODO-Token weist darauf hin, dass Sie die relevanten Callback-Implementierungen in der Zusatzdatei **AsyncCallbacks.c** überprüfen sollten.

5. Suchen Sie nach dem Schritt **web_stop_async**, der die asynchrone Kommunikation beendet. Vergewissern Sie sich, dass sich der Schritt **web_stop_async** am Ende der asynchronen Kommunikation befindet.

6. Überzeugen Sie sich, dass der Schritt **web_stop_async** ordnungsgemäß ausgeführt wird.

Weitere Informationen finden Sie unter "Feinabstimmung des Endes einer asynchronen Unterhaltung" auf Seite 491.

Weitere Informationen zum Schritt **web_stop_async** finden Sie unter "Definieren des Endes einer asynchronen Unterhaltung" auf Seite 488.

7. Überprüfen Sie die Callback-Implementierung und nehmen Sie gegebenenfalls notwendige Änderungen am Skript vor. Weitere Informationen finden Sie unter "Implementieren von Callbacks" auf Seite 492.

8. Stellen Sie sicher, dass alle *Zähler*- und *komplexen Zeichenfolgen*-Parameter richtig gesetzt sind. Beachten Sie, dass für jeden dieser Parameter ein TODO-Kommentar und zu diesem eine entsprechende Aufgabe im Bereich **Aufgaben** existiert. Weitere Informationen finden Sie unter "Parsen von URLs" auf Seite 499.

9. Überprüfen Sie den Bereich **Aufgaben** nach bestimmten Aktionen, die zur Vervollständigung des Skriptentwicklungsprozesses erforderlich sind. Zu solchen Aktionen gehören beispielsweise die Verifizierung von Callback-Implementierungen oder die Verifizierung der Implementierung bestimmter Parameter.

10. Nachdem alle Parameter korrekt initialisiert wurden, führen Sie das Skript aus, um sicherzustellen, dass die asynchrone Kommunikation erwartungsgemäß ausgeführt wird.

Hinweis: Die asynchrone Kommunikation wird nicht unterstützt, wenn Sie ein Vuser-Skript im WinINet-Modus wiedergeben.

Nachdem Sie das geänderte Skript überprüft und sichergestellt haben, dass die asynchrone Kommunikation korrekt implementiert wurde, führen Sie das Skript aus. Weitere Informationen finden Sie unter Aktivieren asynchroner Skripts für die Ausführung.

API für asynchrone Kommunikation

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Die LoadRunner-API enthält mehrere Funktionen, die Vuser-Skripts ermöglichen, die asynchrone Kommunikation zu emulieren. Diese Funktionen für die asynchrone Kommunikation sind:

web_reg_async_attributes

Diese Funktion registriert die nächste Aktionsfunktion als Anfang einer asynchronen Unterhaltung und definiert das Verhalten der asynchronen Unterhaltung.

web_stop_async

Diese Funktion bricht die festgelegte asynchrone Unterhaltung ab, einschließlich der aktiven und zukünftigen Aufgaben.

web_sync

Diese Funktion hält die Ausführung des Vuser-Skripts an, bis der angegebene Parameter definiert ist.

web_util_set_request_url

Diese Funktion legt die angegebene Zeichenfolge als Anforderungs-URL für die nächste Anforderung fest, die in der Unterhaltung gesendet wird. Dies gilt nur bei Aufrufen aus einem Callback.

web_util_set_request_body

Diese Funktion legt die angegebene Zeichenfolge als Anforderungstext für die nächste Anforderung fest, die in der Unterhaltung gesendet wird. Dies gilt nur bei Aufrufen aus einem Callback.

web_util_set_formatted_request_body

Diese Funktion ist ähnlich wie die Funktion **web_util_set_request_body**. Diese Funktion ist jedoch Teil einer asynchronen Unterhaltung eines Flex-Protokolls anstelle einer asynchronen Unterhaltung eines Web(HTTP/HTML)-Protokolls. Sie erwartet einen XML-formatierten Anforderungstext, der konvertiert wird, bevor die Anforderung gesendet wird.

Ausführliche Informationen zu den API-Funktionen für die asynchrone Kommunikation finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Die Funktion **web_reg_async_attributes** muss vor dem Schritt aufgerufen werden, der die asynchrone Unterhaltung startet. Die Funktion **web_reg_async_attributes** empfängt eine Reihe von Argumenten, die die asynchrone Unterhaltung definieren. Eines dieser Argumente ist der URL der asynchronen Unterhaltung. Sobald die Wiedergabe-Engine diesen URL in dem Schritt herunterlädt, der der Funktion **web_reg_async_attributes** folgt, beginnt die asynchrone Unterhaltung. Die in der Funktion **web_reg_async_attributes** registrierten Callbacks ermöglichen dem Skriptentwickler das Steuern einiger der Eigenschaften der asynchronen Unterhaltung (zum Beispiel das Ändern des URL). Die asynchrone Unterhaltung wird bis zum Schritt **web_stop_async** oder bis zum Ende der Iteration fortgesetzt. In einer Push-Unterhaltung kann der Server die Verbindung schließen und somit die Unterhaltung beenden.

Hinweis: Die asynchrone Kommunikation wird nicht unterstützt, wenn Sie ein Vuser-Skript im WinINet-Modus wiedergeben.

Details zu den Unterschieden zwischen den Funktionen für die asynchrone und synchrone Unterhaltung finden Sie unter "Unterschiede zwischen asynchronen Funktionen und synchronen Funktionen" unten.

Unterschiede zwischen asynchronen Funktionen und synchronen Funktionen

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Die LoadRunner-API enthält mehrere Funktionen, die Vuser-Skripts ermöglichen, die asynchrone Kommunikation zu emulieren. Diese asynchronen Funktionen unterscheiden sich wie folgt von den anderen API-Funktionen:

1. Der von den asynchronen Funktionen generierte Netzwerkverkehr verläuft parallel, also gleichzeitig zu dem Hauptfluss im Vuser-Skript. Das heißt, die asynchrone Kommunikation kann fortgesetzt werden, selbst wenn die synchronen Schritte beendet werden.

2. Die asynchrone Kommunikation wird auch während der Ausführung von Nicht-Web-Funktionen fortgesetzt (z. B. **lr_think_time**).

3. Einige der API-Funktionen für die asynchrone Kommunikation verwenden Callback-Funktionen. Der Benutzer muss Callbacks festlegen, die von der Wiedergabe-Engine geplant sind, wenn ein vordefiniertes Ereignis eintritt. Details zur Verwendung von Callbacks für Funktionen zur asynchronen Kommunikation finden Sie unter "Implementieren von Callbacks" auf Seite 492.

Änderung eines Vuser-Skripts durch VuGen für die asynchrone Kommunikation

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Nachdem Sie ein Vuser-Skript erstellt und die erforderlichen Geschäftsprozesse aufgezeichnet haben, generiert VuGen generiert das Vuser-Skript. Anschließend überprüft VuGen das generierte Skript auf Instanzen asynchroner Kommunikation. Dieser Prozess wird als asynchrone Prüfung bezeichnet. Werden Instanzen einer asynchronen Kommunikation gefunden, ändert VuGen das Skript, um dessen Ausführung und die Emulation des asynchronen Verhaltens zu ermöglichen.

Hinweis: VuGen überprüft Skripts nur dann auf asynchrone Kommunikation, wenn die Aufzeichnungsoption **Asynchrone Prüfung** ausgewählt ist. Weitere Informationen finden Sie unter "Erstellen eines asynchronen Vuser-Skripts" auf Seite 480.

Die asynchrone Kommunikation in einem Vuser-Skript ist in eine oder mehrere Unterhaltungen unterteilt. Die einzelnen asynchronen Unterhaltungen, die VuGen in einem Vuser-Skript entdeckt, werden auf der Registerkarte **Asynchron** in **Design Studio** aufgelistet. Sie verwenden diese Liste asynchroner Unterhaltungen, um die von VuGen bei der asynchronen Prüfung am Skript durchgeführten Änderungen systematisch zu analysieren. Vergewissern Sie sich, dass VuGen das asynchrone Verhalten im Vuser-Skript richtig identifiziert und das Vuser-Skript korrekt geändert hat, um das erforderliche asynchrone Verhalten zu emulieren. Weitere Informationen zur Registerkarte **Asynchron** von **Design Studio** finden Sie unter "Registerkarte "Asynchron" [Design Studio]" auf Seite 504.

Hinweis: Nachdem Sie ein Vuser-Skript geändert haben, um mit ihm eine asynchrone Kommunikation zu emulieren, ist es möglicherweise erforderlich, an dem geänderten Skript Korrelationsaktivitäten durchzuführen. Weitere Informationen zur Korrelation finden Sie unter

"Korrelieren asynchroner Vuser-Skripts" auf Seite 492.

Wie ändert VuGen ein Vuser-Skript?

Das asynchrone Verhalten in einem Vuser-Skript ist in eine oder mehrere asynchrone Unterhaltungen unterteilt. Für jede asynchrone Unterhaltung führt VuGen die folgenden Aufgaben durch:

1. VuGen fügt einen Schritt **web_reg_async_attributes** vor dem Anfang der asynchronen Unterhaltung ein. Der Schritt **web_reg_async_attributes** enthält eine ID für die asynchrone Unterhaltung. Diese ID wird von einem nachfolgenden Schritt **web_stop_async** verwendet, um das Ende der asynchronen Unterhaltung anzuzeigen. Das Pattern-Argument gibt den Typ des asynchronen Verhaltens an: *Push*, *Poll* oder *Long Poll*.

```
web_reg_async_attributes("ID=Push_0",
 "Pattern=Push",
 "URL=http://push.example.com/example",
 "RequestCB=Push_0_RequestCB",
 "ResponseHeadersCB=Push_0_ResponseHeadersCB",
 "ResponseBodyBufferCB=Push_0_ResponseBodyBufferCB",
 "ResponseCB=Push_0_ResponseCB",
LAST);
```

   Details zur Verwendung eines Schritts **web_reg_async_attributes** am Anfang einer asynchronen Unterhaltung finden Sie unter "Definieren des Anfangs einer asynchronen Unterhaltung" auf Seite 487.

   Weitere Informationen zur Funktion **web_reg_async_attributes** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

   Details zu den Typen asynchroner Verhaltensweisen, die von LoadRunner unterstützt werden, finden Sie unter "Typen asynchroner Kommunikation" auf Seite 475.

2. VuGen fügt vor den eingefügten Schritt **web_reg_async_attributes** einen Kommentar ein. Der Kommentar enthält Details zu der asynchronen Unterhaltung, einschließlich:
   a. Die ID der asynchronen Unterhaltung.

   b. Die in der Unterhaltung enthaltenen URLs.

   c. Vorgeschlagene Implementierungen für die Callback-Funktionen, die im Schritt **web_reg_async_attributes** deklariert werden. Diese Implementierungen werden in die Datei

AsyncCallbacks.c eingefügt, eine der zusätzlichen Dateien des Skripts.

```
/* Added by Async CodeGen.
ID=Push_0
ScanType = Recording
The following URLs are considered part of this conversation:
    http://push.example.com/example
TODO - The following callbacks have been added to AsyncCallbacks.c.
Add your code to the callback implementations as necessary.
  Push_0_RequestCB
  Push_0_ResponseHeadersCB
  Push_0_ResponseBodyBufferCB
  Push_0_ResponseCB
*/
```

3. Für *Push*-Unterhaltungen fügt VuGen asynchrone API-Funktionen in das Vuser-Skript ein, entfernt jedoch keinen aufgezeichneten Code aus dem Vuser-Skript. Für Unterhaltungen vom Typ *Poll* und *Long Poll* kann VuGen Schritte oder Schrittparameter aus dem generierten Vuser-Skript entfernen. VuGen entfernt die Schritte oder Schrittparameter in Fällen, in denen die relevanten URLs durch die Ausführung der eingefügten asynchronen Funktionen angefordert werden und nicht durch die Ausführung der ursprünglichen Schritte, die entfernt wurden.

   Hinweis: Entfernte Schritte werden nicht gelöscht, sondern auskommentiert. Sie können die Auskommentierung dieser Schritte gegebenenfalls entfernen.

4. Sofern erforderlich fügt VuGen einen **web_stop_async**-Schritt an das Ende einer asynchronen Unterhaltung ein. Der Schritt **web_stop_async** kennzeichnet das Ende der asynchronen Unterhaltung. Weitere Informationen zum Schritt **web_stop_async** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

5. Die Aufzeichnungs-Snapshots werden durch das Gruppieren der Aufgaben in der asynchronen Unterhaltung unter dem Schritt aktualisiert, der die Unterhaltung gestartet hat.

So ändert VuGen die "flex_amf_call"-Schritte:

VuGen unterstützt asynchrones Poll- und Long Poll-Verhalten in **flex_amf_call**-Schritten. Flex-Skripts, die *Poll* oder *Long Poll* in **flex_amf_call**-Schritten enthalten, werden von VuGen wie Web

(HTTP/HTML)-Skripts behandelt. Ausgenommen sind folgende Fälle:

- Der RequestCB enthält einen auskommentierten Aufruf für **web_util_set_formatted_request_body**, der verwendet werden kann, um einen XML-formatierten Anforderungstext zu übergeben, der codiert und mit der Anforderung gesendet wird.
- Die Parameter **aResponseBodyStr** und **aResponseBodyLen** des ResponseCB bieten Benutzern Zugriff auf die XML-Darstellung des Antworttextes.

Definieren des Anfangs einer asynchronen Unterhaltung

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Nachdem VuGen ein Vuser-Skript auf asynchrone Kommunikation geprüft hat, werden in der Registerkarte **Asynchron** von Design Studio die asynchronen Unterhaltungen aufgelistet, die VuGen im Skript gefunden hat. VuGen fügt am Anfang jeder erkannten asynchronen Unterhaltung einen **web_reg_async_attributes**-Schritt in das Vuser-Skript ein. Sie verwenden den Schrittenavigator von VuGen, um im Vuser-Skript die entsprechenden **webweb_reg_async_attributes**-Schritte zu finden. Die **web_reg_async_attributes**-Schritte sollten sich am Anfang der asynchronen Unterhaltungen befinden.

Ein **web_reg_async_attributes**-Schritt, der einem Vuser-Skript hinzugefügt wird, umfasst die folgenden Parameter:

- ID
- URL
- Pattern
- PollIntervalMS (nur für Poll-Unterhaltungen)
- RequestCB
- ResponseBodyBufferCB (nur für Push-Unterhaltungen)
- ResponseHeadersCB (nur für Push-Unterhaltungen)
- ResponseCB

Der URL-Parameter im Schritt **web_reg_async_attributes** sollte mit einer der URLs übereinstimmen, die im auf den Schritt **web_reg_async_attributes** folgenden Schritt angegeben sind. Weitere Informationen zum Schritt **web_reg_async_attributes** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Einfügen eines Kommentars

Wenn VuGen einen **web_reg_async_attributes**-Schritt in ein Skript einfügt, wird vor dem **web_reg_async_attributes**-Schritt ein zugeordneter Kommentar eingefügt. Der eingefügte Kommentar enthält Informationen zu der zugeordneten asynchronen Unterhaltung, zum Beispiel die Unterhaltungs-ID, das Kommunikationsmuster (*Push*, *Poll* oder *Long-Poll*), eine Liste der URLs dieser asynchronen Kommunikation und eine Liste der Callbacks, die in der Zusatzdatei **AsyncCallbacks.c** implementiert sind.

Beachten Sie, dass der Schrittkommentar ein TODO-Token enthält. Das TODO-Token weist darauf hin, dass Sie die relevanten Callback-Implementierungen in der Zusatzdatei **AsyncCallbacks.c** überprüfen sollten.

Details dazu, wie eine asynchrone Unterhaltung beendet wird, finden Sie unter "Definieren des Endes einer asynchronen Unterhaltung" unten.

Beispiel - web_reg_async_attributes

Der folgende Beispielcode enthält einen **web_reg_async_attributes**-Schritt, der von VuGen hinzugefügt wurde. Beachten Sie, dass der Schritt **web_reg_async_attributes** vor einem **web_url**-Schritt eingefügt wurde und dass der URL-Parameter im Schritt **web_reg_async_attributes** dem URL-Parameter im Schritt **web_url** entspricht.

```
/* Added by Async CodeGen.
ID=Poll_0
ScanType = Recording

The following urls are considered part of this conversation:
    http://your_URL.com/content.php?messages

TODO - The following callbacks have been added to AsyncCallbacks.c.
Add your code to the callback implementations as necessary.
    Poll_0_RequestCB
    Poll_0_ResponseCB
 */
    web_reg_async_attributes("ID=Poll_0",
        "URL=http://your_URL.com/content.php?messages",
        "Pattern=Poll",
        "RequestCB=Poll_0_RequestCB",
        "ResponseCB=Poll_0_ResponseCB",
        LAST);

    web_url("content.php",
        "URL=http://your_URL.com/content.php?messages",
        "Resource=0",
        "RecContentType=text/xml",
        "Referer=http://your_URL.com/",
        "Snapshot=t2.inf",
        "Mode=HTML",
        LAST);
```

Definieren des Endes einer asynchronen Unterhaltung

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Nachdem VuGen ein Vuser-Skript auf asynchrone Kommunikation geprüft hat, werden in der Registerkarte **Asynchron** von Design Studio die asynchronen Unterhaltungen aufgelistet, die VuGen im Skript gefunden hat. In das Vuser-Skript wird am Ende jeder festgestellten asynchronen Unterhaltung ein **web_stop_async**-Schritt eingefügt. Sie verwenden den Schrittenavigator von VuGen, um im Vuser-Skript die entsprechenden **web_stop_async**-Schritte zu finden.

Hinweis: In manchen Fällen fügt VuGen keinen **web_stop_async**-Schritt an das Ende einer asynchronen Unterhaltung ein. Dies kann auftreten, wenn VuGen nicht bestimmen kann, wo die asynchrone Unterhaltung endet. Eine der Ursachen kann sein, dass die asynchrone Unterhaltung aufgrund einer bestimmten asynchronen Regel hinzugefügt oder die asynchrone Unterhaltung während der Aufzeichnung nicht beendet wurde. Weitere Informationen zu asynchronen Regeln Sie unter "Regeln für die asynchrone Kommunikation - Übersicht" auf Seite 501.

Nachdem VuGen einen **web_stop_async**-Schritt in ein Vuser-Skript eingefügt hat, vergewissern Sie sich, dass der **web_stop_async**-Schritt an der korrekten Stelle im Skript platziert wurde, das heißt dort, wo die asynchrone Unterhaltung bei der Ausführung des Vuser-Skripts enden sollte.

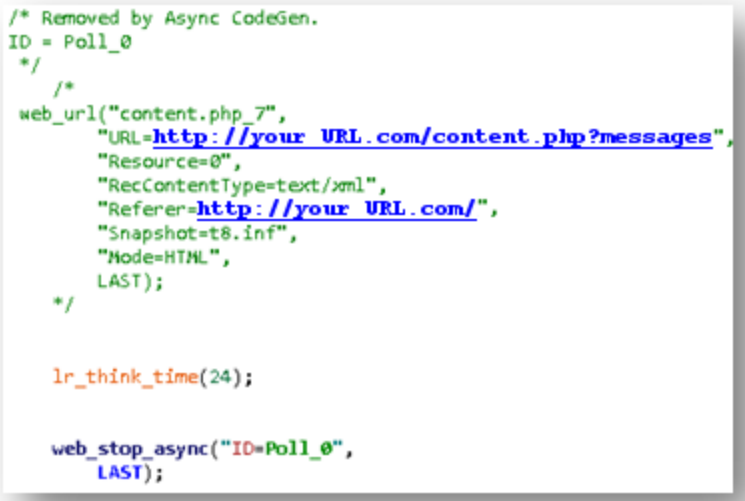
Um sicherzustellen, dass die asynchrone Unterhaltung bei der Skriptausführung korrekt beendet wird, kann es erforderlich sein, die Details des **web_stop_async**-Schritts im Vuser-Skript zu ändern. Weitere Informationen finden Sie unter "Feinabstimmung des Endes einer asynchronen Unterhaltung" auf Seite 491.

Hinweis: Alle asynchronen Konversationen werden automatisch am Ende jeder Iteration beendet, und zwar selbst dann, wenn die Laufzeitoption **Bei jeder Iteration neuen Benutzer simulieren** deaktiviert ist.

Details dazu, wie eine asynchrone Unterhaltung gestartet wird, finden Sie unter "Definieren des Anfangs einer asynchronen Unterhaltung" auf Seite 487.

Beispiel - web_stop_async:

Im folgenden Codebeispiel fügt VuGen einen **web_stop_async**-Schritt an das Ende einer *Poll*-Unterhaltung ein. In diesem Beispiel sind die ursprünglichen Abfrageschritte auskommentiert und die **lr_think_time**-Schritte, die sie getrennt haben, wurden in einem **lr_think_time**-Schritt zusammengeführt, um die Dauer der gesamten *Poll*-Unterhaltung zu emulieren.



Verwenden von Schwellenwerten für asynchrone Anforderungen

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Sie können einige VuGen-Verhaltensweisen beim Überprüfen eines Vuser-Skripts zum Suchen asynchroner Kommunikation optimieren. Sie verwenden Schwellenwerte für asynchrone Anforderungen von VuGen, um das VuGen-Verhalten zu optimieren. Jeder der Schwellenwerte ist nur mit einem der Typen asynchroner Unterhaltungen verbunden: *Push*, *Poll* oder *Long Poll*.

- **Schwellenwerte für asynchrone Anforderungen für Push-Unterhaltungen**

  Minimale Antwortgröße. Gibt die minimale Länge des Antwortinhalts in Byte zum Definieren asynchroner *Push*-Unterhaltungen an. Wenn der Server eine Meldung sendet, die kleiner als der angegebene Wert ist, ordnet VuGen die Unterhaltung nicht als asynchrone Push-Unterhaltung ein.

  Maximale Größe der untergeordneten Meldung. Gibt die maximale Größe der untergeordneten Meldung (in Byte) an, die vom Server zum Definieren asynchroner *Push*-Unterhaltungen gesendet wird. Wenn der Server eine untergeordnete Meldung sendet, die größer als der angegebene Wert ist, ordnet VuGen die Unterhaltung nicht als asynchrone Push-Unterhaltung ein.

  Minimale Anzahl von untergeordneten Meldungen. Gibt die minimale Anzahl untergeordneter Meldungen zum Definieren asynchroner *Push*-Unterhaltungen an. Eine Push-Unterhaltung, bei der weniger als die angegebene Anzahl untergeordneter Meldungen vom Server gesendet wurde, wird von VuGen nicht als asynchrone Push-Unterhaltung eingeordnet.

- **Schwellenwerte für asynchrone Anforderungen für Poll-Unterhaltungen**

  Toleranz bei Intervall. Gibt die Toleranz bei Intervallen (in Millisekunden) für die Klassifizierung von asynchronen *Poll*-Unterhaltungen an. Eine Unterhaltung, bei der die Intervalle um mehr als den angegebenen Wert voneinander abweichen, wird von VuGen nicht als asynchrone Poll-Unterhaltung eingestuft.

- **Schwellenwerte für asynchrone Anforderungen für Long Poll-Unterhaltungen**

  Maximales Intervall. Gibt das maximale Intervall (in Millisekunden) zwischen dem Ende einer Antwort und dem Beginn einer neuen Anforderung zum Klassifizieren asynchroner *Long Poll*-Unterhaltungen an. Eine Unterhaltung, bei der eine Anforderung nach dem angegebenen Wert nach dem Ende der vorherigen Antwort gestartet wird, ordnet VuGen nicht als asynchrone Long Poll-Unterhaltung ein.

Details zu den verfügbaren Schwellenwerten für asynchrone Anforderungen finden Sie unter "Dialogfeld "Asynchron-Optionen"" auf Seite 506.

Feinabstimmung des Endes einer asynchronen Unterhaltung

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Nachdem VuGen ein Vuser-Skript auf asynchrone Kommunikation geprüft hat, werden in der Registerkarte **Asynchron** von Design Studio die asynchronen Unterhaltungen aufgelistet, die im Skript gefunden wurden. In das Vuser-Skript wird am Ende jeder festgestellten asynchronen Unterhaltung ein **web_stop_async**-Schritt eingefügt. Damit jede asynchrone Unterhaltung bei der Skriptausführung korrekt beendet wird, kann es erforderlich sein, eine oder mehrere der folgenden Aufgaben durchzuführen:

- Entfernen des Schritts **web_stop_async**, sodass die asynchrone Unterhaltung am Ende der Iteration beendet wird.

- Verschieben des Schritts **web_stop_async** an eine Position hinter einem vorhandenen Aktionsschritt oder hinter einen vorhandenen **lr_think_time**-Schritt, damit die asynchrone Unterhaltung nach der Ausführung dieses Schritts beendet wird.

- Hinzufügen eines Schritts **lr_think_time** vor den Schritt **web_stop_async** oder Ändern des Zeitparameters in einem vorhandenen **lr_think_time**-Schritt. Vergewissern Sie sich, dass in den Laufzeiteinstellungen die Latenzzeit aktiviert ist. Weitere Informationen finden Sie unter "Allgemein > Latenzzeit (Knoten)" auf Seite 368.

- Hinzufügen eines **web_sync**-Schritts, um die asynchrone Unterhaltung zu beenden, nachdem ein angegebener Parameter einen Wert empfangen hat. Verwenden Sie die Callbacks für die

asynchrone Unterhaltung, um zu gewährleisten, dass der Parameter nur einen Wert empfängt, wenn Sie die Unterhaltung beenden möchten.

Korrelieren asynchroner Vuser-Skripts

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Nachdem Sie ein Vuser-Skript geändert haben, um mit ihm eine asynchrone Kommunikation zu emulieren, ist es möglicherweise erforderlich, an dem geänderten Skript Korrelationsaktivitäten durchzuführen. Aufgrund des asynchronen Verhaltens können dynamische Werte der asynchronen Kommunikation von Design Studio nicht verarbeitet werden und erfordern eine manuelle Korrelation.

Sie können mit der Funktion **lr_save_param_regexp** nach dynamischen Werten in Response-Callback-Funktionen suchen. Diese Funktion kann aus einem Callback aufgerufen werden, um den erforderlichen Wert aus der Serverantwort (**ResponseCB**) oder aus dem Antwortpuffer (**ResponseBodyBufferCB**) zu extrahieren und diesem Wert einem Parameter zuzuweisen. Dieser Parameter wird dann für Korrelationen verwendet.

Weitere Informationen zu der Funktion **lr_save_param_regexp** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Implementieren von Callbacks

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Nachdem VuGen ein Vuser-Skript auf asynchrone Kommunikation geprüft hat, werden auf der Registerkarte **Asynchron** von **Design Studio** die asynchronen Unterhaltungen aufgelistet, die im Skript gefunden wurden. Für jede während der Prüfung gefundene asynchrone Unterhaltung fügt VuGen die Callback-Funktionssignaturen hinzu, die denen entsprechen, die im Schritt **web_reg_async_attributes** angegeben wurden. Die Signaturen werden zur Zusatzdatei **AsyncCallbacks.c** hinzugefügt.

Folgende Callbacks stehen zur Verfügung:

1. RequestCB

   Dieser Callback wird aufgerufen, bevor eine Anforderung gesendet wird.

2. ResponseBodyBufferCB

   Dieser Callback wird aufgerufen, wenn im Puffer des Antworttexts Inhalt vorhanden ist, sowie am Ende des Antworttexts. Dieser Callback wird von VuGen automatisch für *Push*-Unterhaltungen generiert, steht aber bei *Poll*- und *Long-Poll*-Unterhaltungen ebenfalls zur Verfügung.

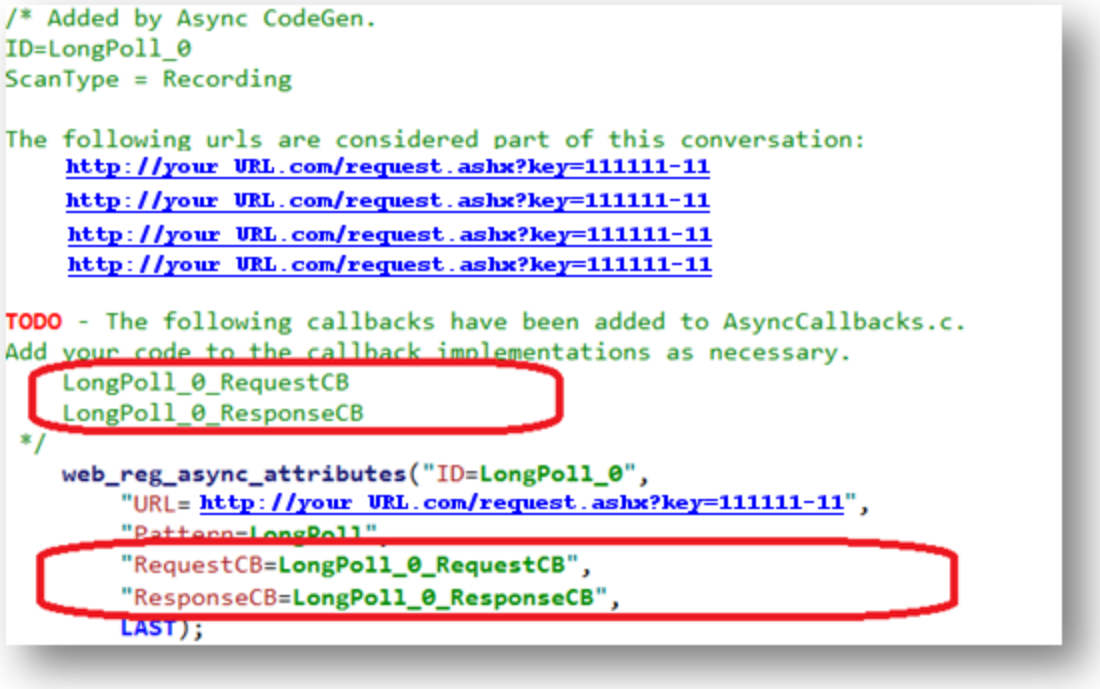
3. ResponseCB

   Dieser Callback wird nach jedem Antworteingang in der Unterhaltung aufgerufen.

- Die Namen der Callback-Funktionen beginnen mit der Unterhaltungs-ID der asynchronen Unterhaltung. So ist beispielsweise der RequestCB-Callback für eine asynchrone Unterhaltung mit ID "LongPoll_0" der Callback **LongPoll_0_RequestCB**.
- Die Namen der Callback-Funktionen sind im Schritt **web_reg_async_attributes** des Skripts deklariert.

Beispiel 1:

Im folgenden Beispielcode sind die drei Callback-Funktionen im Schritt **web_reg_async_attributes** deklariert.



Beispiel 2:

Im folgenden Beispielcode sind die zwei Callback-Funktionen in der zusätzlichen Datei **AsyncCallbacks.c** implementiert.

```
int LongPoll_0_RequestCB()
{
    //enter your implementation for RequestCB() here

    //call web_util_set_request_url() here to modify polling url
    //web_util_set_request_url("<request_url>");

    //call web_util_set_request_body() here to modify request body:
    //web_util_set_request_body("<request body>");

    return WEB_ASYNC_CB_RC_OK;
}

int LongPoll_0_ResponseCB(
    const char *    aResponseHeadersStr,
    int             aResponseHeadersLen,
    const char *    aResponseBodyStr,
    int             aResponseBodyLen,
    int             aHttpStatusCode)
{
    //enter your implementation for ResponseCB() here

    return WEB_ASYNC_CB_RC_OK;
}
```

Sie können die Callbacks ändern und das erforderliche Verhalten implementieren. Weitere Informationen finden Sie unter "Ändern von Callbacks" auf der nächsten Seite.

Beispiel 3:

Der folgende Beispielcode zeigt eine Implementierung der ResponseHeader-Callback-Funktion mit drei Argumenten: HTTP-Statuscode, Accumulated-Kopfzeilenzeichenfolge und Länge der Accumulated-Kopfzeilenzeichenfolge.

```
int Push_0_ResponseHeadersCB(
  int  aHttpStatusCode,
  const char * aAccumulatedHeadersStr,
  int aAccumulatedHeadersLen)
{
    //Enter your implementation for ResponseHeadersCB() here.
     lr_output_message("Response status code is :[%d]", aHttpStatusCode);
     lr_output_message("Response headers are :/n[%s]", aAccumulatedHeadersStr);
return WEB_ASYNC_CB_RC_OK;
}
```

Ein Beispiel der Ausgabe der oben dargestellten Callback-Funktion wird nachfolgend aufgeführt:

```
Response status code is :[200]
Response headers are :
[HTTP/1.1 200 OK
Connection: close
Date: Tue, 25 Jun 2013 09:03:33 GMT
Server: Microsoft-IIS/6,0
Content-Type: text/html
Cache-control: private]
```

Ändern von Callbacks

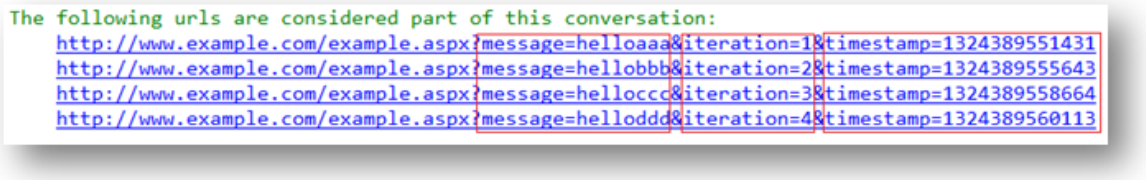
Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Nachdem VuGen ein Vuser-Skript auf asynchrone Kommunikation geprüft hat, werden auf der Registerkarte **Asynchron** von **Design Studio** die asynchronen Unterhaltungen aufgelistet, die im Skript gefunden wurden. VuGen fügt für jede während der Prüfung gefundene asynchrone Unterhaltung die erforderlichen Deklarationen der Callback-Funktion in die Datei **AsyncCallbacks.c** ein. Um das gewünschte Verhalten zu implementieren, können Sie die Callbacks ändern, die von VuGen hinzugefügt wurden. Das Ändern eines Callbacks beinhaltet:

Ändern des Anforderungs-URL im RequestCB-Callback

In Unterhaltungen des Typs *Poll* und *Long-Poll* ändern sich die angeforderten URLs häufig in jeder Abfrageiteration. Die Änderung wird in der Regel durch clientseitige Logik bestimmt, die normalerweise von JavaScript implementiert wird, das der Browser ausführt. Teile des URL können durch Korrelation eines bekannten Parameters, z. B. Sitzungs-ID, ermittelt werden. Details finden Sie unter "Parsen von URLs" auf Seite 499.

Die Anforderungs-URLs in einer asynchronen Unterhaltung werden geändert, bevor die Anforderung mit RequestCB und der Funktion **web_util_set_request_url** gesendet wird.



Ändern des Anforderungstexts im RequestCB-Callback

Der Anforderungstext in Anforderungen, die Teil einer asynchronen Unterhaltung sind, müssen

möglicherweise vor dem Senden der Anforderung geändert werden. Sie verwenden den RequestCB und die util-Funktion **web_util_set_request_body**, um den Anforderungstext zu ändern.

Das Ändern des Anforderungstexts ist hilfreich in Unterhaltungen des Typs *Poll* oder *Long-Poll*, bei denen jede neue Anforderung einen anderen Anforderungstext erfordert.

```
//an example of a parametrized request body sent in the RequestCB.
//the value of {request_body} may be set in the callback function,
//or elsewhere in the script.
web_util_set_request_body("{request_body}");
```

Jeder von VuGen generierte RequestCB enthält ein kommentiertes Snippet. Sie können das Snippet auskommentieren, um die util-Funktion **web_util_set_request_body** zu verwenden.

Wenn VuGen feststellt, dass unterschiedliche Anforderungen in der aufgezeichneten Unterhaltung über unterschiedliche Textwerte verfügen, enthält der generierte RequestCB einen Kommentar, mit dem Sie aufgefordert werden, die Aufzeichnung zu prüfen. Ziel ist, den Anforderungstext zu implementieren, der bei der Skriptausführung für jede Anforderung gesendet wurde.

```
//call web_util_set_request_body() here to modify request body:
//web_util_set_request_body("<request body>");
//TODO - use snapshot view to see examples of request bodies sent
```

Ändern der Antwort im ResponseCB-Callback

Sie können den Antwort-Callback in einer asynchronen Unterhaltung ändern, um die Gültigkeit der Antworten zu überprüfen oder auf ein bestimmtes Ereignis zu warten. Sie können z. B. die Antwortkopfzeilen für jede Antwort überprüfen, um festzustellen, ob ein bestimmter Wert empfangen wurde.

```
int Poll_0_ResponseCB(
    const char *    aResponseHeadersStr,
    int             aResponseHeadersLen,
    const char *    aResponseBodyStr,
    int             aResponseBodyLen,
    int             aHttpStatusCode)
```

Wenn der erwartete Wert empfangen wurde, können Sie in der Aktionsdatei einen **web_stop_async**-Schritt verwenden, um die asynchrone Unterhaltung zu beenden.

Im folgenden Codebeispiel ist eine solche Beendigung einer synchronisierten Unterhaltung dargestellt. In diesem Beispiel zählt das Skript in der Datei **AsyncCallback.c** 10 Iterationen der Poll-Unterhaltung ab und erstellt anschließend einen neuen Parameter, **stopAsync**.

```
int Poll_0_ResponseCB(
  ...{
       //increment iteration counter for every response received.
       static int iter = 0;
       iter++;

       //Once the desired number of responses has been reached,
       //create and save the parameter.
       if (iter > 10)  {
               lr_save_int(iter, "stopAsync");
                       }
       return WEB_ASYNC_CB_RC_OK;
     }
```

In der Aktionsdatei verwendet der **web_sync**-Schritt den generierten Parameter **stopAsync**, um die Unterhaltung zu beenden:

```
web_reg_async_attributes("ID=Poll_0","Pattern=Poll",
  "URL=http://pumpkin:2080/nioamfpoll;AMFSessionId=6F8D6108-E309-38B2-3D65-963B4
31D0A38",
  "PollIntervalMs=3000",
  "RequestCB=Poll_0_RequestCB",
  "ResponseCB=Poll_0_ResponseCB",
 LAST);

web_custom_request("nioamfpoll;AMFSessionId=6F8D6108-E309-38B2-3D65-963B431D0A3
8_2",
  "URL=http://pumpkin:2080/nioamfpoll;AMFSessionId=6F8D6108-E309-38B2-3D65-963B4
31D0A38",
   "Method=POST",
  "Resource=0",
  "RecContentType=application/x-amf",
  "Referer=http://pumpkin:8081/lcds-samples/traderdesktop/traderdesktop.swf/[[DY
NAMIC]]/3",
  "Snapshot=t11.inf",
  "Mode=HTML",
  "EncType=application/x-amf",
  "BodyBinary=\\x00\\x03\\x00\\x00\\x00\\x01\\x00\\x04null\\x00\\x02/6\\x00\\x00
\\x00Z\n\\x00\\x00\\x00\\x01\\x11\n\\x07\\x07DSC\\x8D\\x02\n\\x0B\\x01\\x01\\x06
\\x01\n\\x05\tDSId\\x06I6F8D611E-DC1C-9D0C-2BBA-36CC2AB8633B\\x01\\x0C!\\xC0\\xB
E\\xA6Z74\\xBE\\xC3\\xCF\\xC7\\xFA\\xE6\\xC3\t\\xE2\\x92\\x01\\x06\\x01\\x01\\x0
4\\x02",
 LAST);

lr_think_time(30);
```

```
//suspend the script until the desired number
//of iterations have been performed.
web_sync("ParamCreated=stopAsync", "RetryIntervalMs=1000", "RetryTimeoutMs=30000
0", LAST);

web_stop_async("ID=Poll_0",LAST);
```

Weitere Informationen zum Beenden einer asynchronen Unterhaltung finden Sie unter "Definieren des Endes einer asynchronen Unterhaltung" auf Seite 488.

Ändern von Callbacks in Flex-Vuser-Skripts

Bei asynchronen *Poll*- und *Long-Poll*-Unterhaltungen in Flex enthält der generierte RequestCB in der Datei **AsyncCallback.c** einen Aufruf für **web_util_set_formatted_request_body**. Damit wird ein in XML formatierter Anforderungstext für jede Anforderung festgelegt.

```
web_util_set_formatted_request_body("<AMFPacket AMF_version=\"3\">"
    "<AMFHeaders />"
    "<Messages>"
        "<Message method=\"null\" target=\"/{Target_Poll_0}\">"
              …
        "</Message>"
    "</Messages>"
    "</AMFPacket>");
```

Nach dem Auskommentieren des kommentierten Codes im TODO-Abschnitt und nach dem Hinzufügen Ihres Callback-Codes öffnen Sie das Design Studio für Skripts, um die Prüfung auf Korrelationen durchzuführen.

Beachten Sie, dass bei der Codegenerierung der Zielparameter automatisch im Anforderungstext parametriert wird. Außerdem wird vor jeder Abfrageiteration ein Code zur automatischen Inkrementierung dieses Parameters generiert.

Der generierte RequestCB enthält auch eine Erinnerung, um sicherzustellen, dass der Initialisierungsparameter des Zählers für **Target_Poll_0** in der Aktionsdatei mit dem Zielattribut im ersten Element *Meldung* in der ersten Abrufanforderung übereinstimmt.

```
lr_param_increment("Target_Poll_0", "{Target_Poll_0}");

web_util_set_formatted_request_body("<AMFPacket AMF_version=\"3\">"
   "<AMFHeaders />"
   "<Messages>"
     "<Message method=\"null\" target=\"/{Target_Poll_0}\">"
       …
```

Stellen Sie in der Aktionsdatei sicher, dass Sie denselben Abrufparameter verwenden, den Sie bereits in der Datei **AsyncCallbacks.c** verwendet haben. Im folgenden Segment aus der Aktionsdatei stimmt der Abrufparameter *Target_Poll_0* mit dem in der Datei **AsyncCallbacks.c** verwendeten Parameter überein.

```
/* Initialize target parameter before sending first request. */
/* Notice that parameter will be incremented once before first request. */
lr_save_int(5, "Target_Poll_0");
```

Weitere Details zur Verwendung von Callback-Funktionen finden Sie unter "Implementieren von Callbacks" auf Seite 492.

Parsen von URLs

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

In asynchroner Kommunikation verwendete URLs enthalten häufig Abfragezeichenfolgen, die auf unterschiedliche Art und Weise abgeleitet werden. Diese Zeichenfolgen können die folgenden Elemente enthalten:

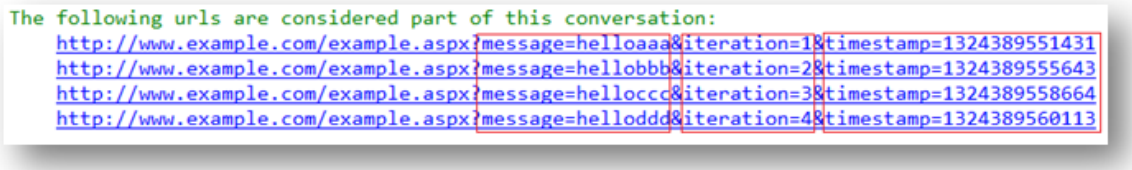
- Zeitstempel
- Indikatoren
- Komplexe Zeichenfolgen

Damit ein Vuser-Skript die asynchrone Kommunikation erfolgreich emulieren kann, muss VuGen in der Lage sein, die erforderlichen URLs neu zu erstellen.

- Wenn der URL einen Zeitstempel enthält, kann VuGen den erforderlichen URL in der Regel erstellen.
- Wenn der URL einen Zähler enthält, kann VuGen den Zähler in der Regel neu erstellen; allerdings kann es erforderlich sein, den Zähler im Skript manuell zu initialisieren.
- Wenn der URL mehrere komplexe Zeichenfolgen enthält, müssen die Algorithmen zum Generieren der URLs manuell in den Code des Vuser-Skripts eingefügt werden.

Beispiel:

Der folgende Beispielcode zeigt eine Gruppe von URLs, die Teil einer *Long-Poll*-Kommunikation sind. Die URLs befinden sich in einem Kommentar, der für einen **web_reg_async_attributes**-Schritt erstellt wurde:



Wenn keiner der in diesem Codebeispiel enthaltenen Parameter bei der Überprüfung des aufgezeichneten Vuser-Skripts gefunden wird, enthält die RequestCB-Implementierung möglicherweise ein unkommentiertes Codefragment, sodass der URL für jede Antwort mit benutzerdefiniertem Code festgelegt werden kann. Weitere Informationen finden Sie unter "Ändern von Callbacks" auf Seite 495.

```
//call web_util_set_request_url() here to modify request url:
//web_util_set_request_url("<request url>");
```

Wenn einige oder alle Parameter im oben gezeigten Beispielcode bei VuGens Überprüfung des aufgezeichneten Vuser-Skripts gefunden werden, enthält die RequestCB-Implementierung folgende Elemente:

- Einen Kommentar, der den Benutzer auffordert, **web_util_set_request_url** aufzurufen. Der Kommentar enthält eine parametrierte Version des URLs.
- Ein Codefragment zum Speichern des Zeitstempels in einem Parameter für jeden im URL gefundenen *Zeitstempel*-Parameter.
- Ein Codefragment zum Erhöhen des Zählers für jeden im URL gefundenen *Zähler*-Parameter. Ein ähnlicher Schritt für die Initialisierung des Zählerparameters wird zur Aktionsdatei hinzugefügt. Das Fragment enthält auch Beispiele des URL-Token, das den Zählerparameter enthält, wie er während der Aufzeichnung zu sehen ist.
- Ein Codefragment zum Speichern der Zeichenfolge in einem Parameter für jeden im URL gefundenen *komplexe Zeichenfolge*-Parameter. Es ist Aufgabe des Benutzers, die korrekte Zeichenfolge für den im Parameter zu speichernden URL zu erstellen. Das Fragment enthält auch Beispiele des URL-Token, das während der Aufzeichnung als unbekannter Parameter betrachtet wird.
- Ein Codefragment für die Übergabe der parametrierten Version eines URL an die Funktion **web_util_set_request_url**.

Beispiel: Codefragment mit der parametrierten Version eines URL.

```
//call web_util_set_request_url() here to modify polling url
//url is expected to be of the form:
//http://www.example.com/example.aspx?message={Unknown_LongPoll_0_0}
//&iteration={Counter_LongPoll_0_1}&timestamp={TimeStamp_LongPoll_0_2}
```

Beispiel: Codefragment, das den Benutzer auffordert, den Wert eines unbekannten Parameters einzugeben.

```
//TODO - implement parameter of type unknown: Unknown_LongPoll_0_0.
//Known examples for Unknown_LongPoll_0_0:
//message=[{"channel":"\/meta\/connect","connectionType":"long-polling","id":3,"clientId":"113fc44"}],
//message=[{"channel":"\/meta\/connect","connectionType":"long-polling","id":5,"clientId":"113fc44"}],
//message=[{"channel":"\/meta\/connect","connectionType":"long-polling","id":6,"clientId":"113fc44"}],
//message=[{"channel":"\/meta\/connect","connectionType":"long-polling","id":7,"clientId":"113fc44"}].
lr_save_string("[{\"channel\":\"\\/meta\\/connect\",\"connectionType\":\"long-polling\",\"id\":3,\"clientId\":\"113fc44\"}]",
"Unknown_LongPoll_0_0");
```

Beispiel: Codefragment zum Erhöhen eines Zählerparameters.

```
//TODO - check counter initialization for Counter_LongPoll_0_1 in Action file.
//Known examples for the token containing Counter_LongPoll_0_1:
//iteration=1, iteration=2, iteration=3, iteration=4 gPoll_0_1:
lr_param_increment("Counter_LongPoll_0_1", "{Counter_LongPoll_0_1}");
```

Beispiel: Codefragment zum Initialisieren eines Zählerparameters.

```
lr_save_int(0, "Counter_LongPoll_0_1");
```

Beispiel: Codefragment zum Speichern eines Zeitstempelparameters.

```
web_save_timestamp_param("TimeStamp_LongPoll_0_2",LAST);
```

Beispiel: Codefragment für die Übergabe der parametrierten Version eines URL an die Funktion **web_util_set_request_url**.

```
//once all parameters have been assigned, copy them to the updated url,
//and call web_util_set_request_url() with the updated url:
web_util_set_request_url("http://www.example.com/example.aspx?message={Unknown_LongPoll_0_0}
&iteration={Counter_LongPoll_0_1}&timestamp={TimeStamp_LongPoll_0_2}");
```

Regeln für die asynchrone Kommunikation - Übersicht

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

In einigen Fällen kann VuGen beim Überprüfen eines Vuser-Skripts auf asynchrone Kommunikation einige der im Vuser-Skript enthaltenen asynchronen Unterhaltungen möglicherweise nicht korrekt identifizieren. In anderen Fällen werden reguläre synchrone Schritte vielleicht fälschlicherweise als Teile von asynchronen Unterhaltungen klassifiziert. Um die Probleme in diesen beiden Fällen zu

vermeiden, können Sie Regeln für die asynchrone Kommunikation definieren, um festzulegen, wie Anforderungen an angegebene URLs bei einer asynchronen Prüfung klassifiziert werden.

Regeln für die asynchrone Kommunikation können positiv oder negativ sein.

- **Positiv:** Wenn VuGen in einem Vuser-Skript enthaltene asynchrone Unterhaltungen nicht identifizieren kann, implementieren Sie eine positive asynchrone Regel, damit VuGen das asynchrone Verhalten feststellen kann.

  Szenario: VuGen erkennt keine URLs unter **http://www.true-async.com/push_example.aspx** als asynchrone Push-Unterhaltungen, Sie wissen jedoch, dass sie Bestandteil asynchroner Push-Unterhaltungen sind. Fügen Sie eine positive Regel hinzu, damit VuGen die asynchronen Push-Unterhaltungen richtig erkennen kann. Wenn Sie das Skript erneut generieren, wendet die asynchrone Prüfung die hinzugefügte Regel an, und alle URLs, die mit **http://www.true-async.com/push_example.aspx** beginnen, werden als Bestandteile asynchroner Push-Unterhaltungen eingeschlossen.

- **Negativ:** Wenn VuGen fälschlicherweise reguläre synchrone Schritte als Teil einer asynchronen Unterhaltung klassifiziert, implementieren Sie eine negative asynchrone Regel, um zu verhindern, dass VuGen fälschlicherweise asynchrones Verhalten erkennt.

  Szenario: VuGen identifiziert alle URLs unter **http://www.not-async.com/** als asynchrone Poll-Unterhaltungen. Sie wissen, dass dies keine asynchronen Unterhaltungen sind. Implementieren Sie eine negative asynchrone Regel, um zu verhindern, dass VuGen fälschlicherweise asynchrones Verhalten erkennt. Wenn Sie das Skript erneut generieren, wendet die asynchrone Prüfung die hinzugefügte Regel an, sodass alle URLs, die mit **http://www.not-async.com/** beginnen, nicht als Bestandteile asynchroner Unterhaltungen klassifiziert werden.

Weitere Informationen zum Implementieren asynchroner Regeln finden Sie unter "Hinzufügen von Regeln für die asynchrone Kommunikation" unten.

Hinzufügen von Regeln für die asynchrone Kommunikation

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Wenn VuGen ein Vuser-Skript nach der Aufzeichnung oder erneuten Generierung des Skripts überprüft, werden asynchrone Unterhaltungen, die im Vuser-Skript enthalten sind, möglicherweise nicht erkannt. In anderen Fällen wird ein regulärer synchroner Schritt von VuGen vielleicht fälschlicherweise als asynchrone Unterhaltung klassifiziert.

Sie können Regeln für die asynchrone Kommunikation definieren, die festlegen, wie Anforderungen an die angegebenen URLs während einer asynchronen Prüfung klassifiziert werden. Regeln für die asynchrone Kommunikation können positiv oder negativ sein. In diesem Abschnitt wird beschrieben, wie Sie Regeln für die asynchrone Kommunikation implementieren. Eine Einführung zu den Regeln für die asynchrone Kommunikation finden Sie unter "Regeln für die asynchrone Kommunikation - Übersicht" auf der vorherigen Seite.

Hinzufügen einer positiven asynchronen Regel

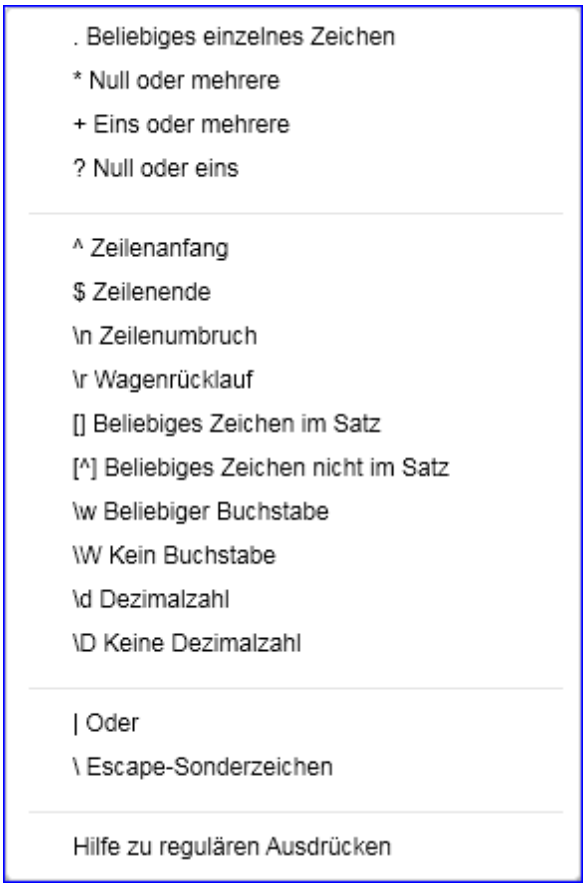
1. Wählen Sie **Aufzeichnen > Aufzeichnungsoptionen > Allgemein > Codegenerierung** und klicken Sie dann auf **Asynchron-Optionen**. Das Dialogfeld **Asynchron-Optionen** wird geöffnet.

2. Klicken Sie unter **Asynchrone reguläre Ausdrücke** auf **Asynchrone Regel hinzufügen** . Das Dialogfeld **Regel hinzufügen** wird geöffnet.

3. Wählen Sie aus der Liste **Typ** den gewünschten Typ aus: **Push**, **Poll** oder **Long Poll**.

4. Geben Sie unter **Regulärer URL-Ausdruck** einen regulären Ausdruck für URLs ein, der als Teil der asynchronen Unterhaltungen betrachtet werden soll.

Sonderzeichen, die in einem regulären Ausdruck enthalten sein können:



Hinweis: Um Zeichen wie "?" und "+" in den regulären Ausdruck einzuschließen, fügen Sie einen umgekehrten Schrägstrich "\" vor dem gewünschten Zeichen ein.

5. Klicken Sie auf **OK**. Die neue Regel wird in der Liste der asynchronen Regeln für das Vuser-Skript angezeigt.

Hinweis: Wenn Sie das Skript erneut generieren:
- Für jede Push-Unterhaltung, die einen URL enthält, der dem regulären Ausdruck entspricht, fügt VuGen asynchrone API-Funktionen in das Vuser-Skript ein, entfernt jedoch keinen aufgezeichneten Code aus dem Vuser-Skript.

- Für jede Poll- oder Long Poll-Unterhaltung, die einen URL enthält, der dem regulären Ausdruck entspricht, VuGen fügt asynchrone API-Funktionen in das Vuser-Skript ein. Außerdem werden möglicherweise Schritte oder Schrittparameter aus dem generierten Vuser-Skript entfernt. VuGen entfernt die Schritte oder Schrittparameter in Fällen, in denen die relevanten URLs durch die Ausführung der eingefügten asynchronen Funktionen angefordert werden und nicht durch die Ausführung der ursprünglichen Schritte, die entfernt wurden.

Details finden Sie unter "Änderung eines Vuser-Skripts durch VuGen für die asynchrone Kommunikation" auf Seite 484.

Hinzufügen einer negativen asynchronen Regel

1. Wählen Sie **Aufzeichnen > Aufzeichnungsoptionen > Allgemein > Codegenerierung** und klicken Sie dann auf **Asynchron-Optionen**. Das Dialogfeld **Asynchron-Optionen** wird geöffnet.

2. Klicken Sie unter **Asynchrone reguläre Ausdrücke** auf **Asynchrone Regel hinzufügen** . Das Dialogfeld **Asynchrone Regel hinzufügen** wird geöffnet.

3. Wählen Sie aus der Liste **Regeltyp** die Option **Nicht asynchron** aus.

4. Geben Sie unter **Regulärer URL-Ausdruck** einen regulären Ausdruck für URLs ein, der nicht als Teil der asynchronen Unterhaltungen betrachtet werden soll.

5. Klicken Sie auf **OK**. Die neue Regel wird in der Liste der asynchronen Regeln für das Vuser-Skript angezeigt.

   Wenn Sie das Skript erneut generieren, werden Schritte mit URLs, die dem regulären Ausdruck entsprechen, nicht in asynchrone Unterhaltungen eingeschlossen.

Registerkarte "Asynchron" [Design Studio]

Auf der Registerkarte **Asynchron** von Design Studio werden alle Vorkommen asynchroner Kommunikation aufgelistet, die VuGen im Vuser-Skript gefunden hat.

Zugriff	• Wählen Sie **Entwurf > Design Studio** aus und klicken Sie auf die Registerkarte **Asynchron**. • Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Design Studio** und anschließend auf die Registerkarte **Asynchron**.

Wichtige Informationen	• Die Schaltfläche **Design Studio** ist nur aktiviert, wenn im Lösungs-Explorer ein aufgezeichnetes Vuser-Skript angezeigt wird. • Mit der Registerkarte **Asynchron** können Sie die im Vuser-Skript enthaltene asynchrone Kommunikation nur anzeigen, Sie können jedoch keines der asynchronen Details von der Registerkarte **Asynchron** bearbeiten. Änderungen an den asynchronen Details müssen im Vuser-Skript vorgenommen werden.
Relevante Aufgaben	"Erstellen eines asynchronen Vuser-Skripts" auf Seite 480

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Typ	Gibt den Ursprung des asynchronen Codes im Vuser-Skript an: **Aufzeichnen.** Der asynchrone Code wurde bei der asynchronen Prüfung nach dem Aufzeichnen oder der Neuerstellung des Vuser-Skripts von VuGen hinzugefügt. **Regel.** Der asynchrone Code wurde aufgrund einer bestimmten Regel in der Regeldatei **Asynchron** von VuGen hinzugefügt. **Manuell.** Der asynchrone Code wurde manuell von einem Benutzer hinzugefügt.
Aktion	Der Abschnitt des Vuser-Skripts, in dem sich das asynchrone Verhalten befindet.
Auftreten	• Bei der Push-Kommunikation ist der Wert für **Auftreten** immer 1. • Bei der Poll- und der Long-Poll-Kommunikation wird mit **Auftreten** die Anzahl der Schritte oder zusätzlichen Ressourcenattribute angegeben, die von VuGen während der asynchronen Prüfung des Vuser-Skripts entfernt (auskommentiert) wurde.
Status	Weist immer den Wert **Angewendet** auf.
Asynchronisierungstyp	Der Typ des erkannten asynchronen Verhaltens: **Push**, **Poll** oder **Long Poll**.
URL	Der URL im Schritt **web_reg_async_attributes**, mit dem die asynchrone Kommunikation eingeleitet wird.
Filter	Wählen Sie aus, welche asynchrone Kommunikation in der Kommunikationsliste angezeigt werden soll.
Details	Erweitert das Dialogfeld, um Details zur ausgewählten asynchronen Kommunikation anzuzeigen.

Element der Oberfläche	Beschreibung
Name	Weist immer den Wert **web_reg_async_attributes** auf.
Zeile	Die Zeile im Vuser-Skript, die den Schritt **web_reg_async_ attributes** enthält.
Aktionsname	Der Abschnitt des Vuser-Skripts, in dem sich das asynchrone Verhalten befindet.
Beschreibung	Der Kommentar im Vuser-Skript, der dem Schritt **web_reg_async_ attributes** vorangestellt ist.
Vorkommen im Snapshot	• Bei der Push-Kommunikation wird der Antworttext angezeigt. • Bei der Poll- oder Long-Poll-Kommunikation werden die mit der asynchronen Kommunikation verknüpften HTTP-Attribute angezeigt.
Optionen	Öffnet das Dialogfeld **Schwellenwerte für asynchrone Anforderungen**.

Dialogfeld "Asynchron-Optionen"

In diesem Dialogfeld können Sie einige VuGen-Eigenschaften beim Scannen eines Vuser-Skripts zum Suchen asynchroner Kommunikation optimieren.

Zugriff	**VuGen > Aufzeichnen > Aufzeichnungsoptionen > Allgemein > Codegenerierung** und klicken Sie dann auf **Asynchron-Optionen**.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Minimale Antwortgröße	Gibt die minimale Größe (in Byte) an, die als Serverantwort zum Definieren asynchroner *Push*-Unterhaltungen gesendet werden kann. Wenn der Server eine Meldung sendet, die kleiner als der angegebene Wert ist, ordnet VuGen die Unterhaltung nicht als asynchrone Push-Unterhaltung ein.
Maximale Größe der untergeordneten Meldung	Gibt die maximale Größe der untergeordneten Meldung (in Byte) an, die vom Server zum Definieren asynchroner *Push*-Unterhaltungen gesendet wird. Wenn der Server eine untergeordnete Meldung sendet, die größer als der angegebene Wert ist, ordnet VuGen die Unterhaltung nicht als asynchrone Push-Unterhaltung ein.

Minimale Anzahl von untergeordneten Meldungen	Gibt die minimale Anzahl gültiger untergeordneter Meldungen zum Definieren asynchroner *Push*-Unterhaltungen an. Eine Push-Unterhaltung, bei der weniger als die angegebene Anzahl gültiger untergeordneter Meldungen vom Server gesendet wurde, wird von VuGen nicht als asynchrone Push-Unterhaltung eingeordnet.
Toleranz bei Intervall	Gibt die Toleranz bei Intervallen (in Millisekunden) für die Klassifizierung von asynchronen *Poll*-Unterhaltungen an. Eine Unterhaltung, bei der die Intervalle um mehr als den angegebenen Wert voneinander abweichen, wird von VuGen nicht als asynchrone Poll-Unterhaltung eingestuft.
Maximales Intervall	Gibt das maximale Intervall (in Millisekunden) zwischen dem Ende einer Antwort und dem Beginn einer neuen Anforderung zum Klassifizieren asynchroner *Long Poll*-Unterhaltungen an. Eine Unterhaltung, bei der eine Anforderung nach dem angegebenen Wert nach dem Ende der vorherigen Antwort gestartet wird, ordnet VuGen nicht als asynchrone Long Poll-Unterhaltung ein.
Tabelle mit asynchronen Regeln	
Regeltyp	**Push**, **Poll** und **Long Poll** sind positive asynchrone Regeln. **Nicht asynchron** ist eine negative Regel. Weitere Informationen zu den asynchronen Regeltypen finden Sie unter "Regeln für die asynchrone Kommunikation - Übersicht" auf Seite 501.
Regulärer Ausdruck	Ein regulärer Ausdruck für URLs, die als Teil von asynchronen Unterhaltungen betrachtet werden müssen. Eine Liste der Sonderzeichen, die Sie in einen regulären Ausdruck einschließen können, finden Sie unter "Hinzufügen von Regeln für die asynchrone Kommunikation" auf Seite 502.

Beispiel für asynchrone Kommunikation - Poll

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Das folgende Beispiel zeigt ein Vuser-Skript mit einer asynchronen Poll-Unterhaltung. Die vom Vuser emulierte Anwendung ist eine "Wiki"-Seite. Der Browser stellt die Wiki-Seite dar und sendet alle 5 Sekunden eine Anforderung auf Aktualisierung der Seite.

```
Action()
{

    web_url("wiki",
        "URL=http://example.com/",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=",
        "Snapshot=t1.inf",
        "Mode=HTML",
        LAST);

    web_url("content.php",
        "URL=http://example.com/content.php?messages",
        "Resource=0",
        "RecContentType=text/xml",
        "Referer=http://example.com/",
        "Snapshot=t2.inf",
        "Mode=HTML",
        LAST);

    lr_think_time(4);

    web_url("content.php_2",
        "URL=http://example.com/content.php?messages",
        "Resource=0",http://example.com/content.php?messages",
        "RecContentType=text/xml",
        "Referer=http://example.com/",
        "Snapshot=t3.inf",
        "Mode=HTML",
        LAST);

    lr_think_time(4);

    web_url("content.php_3",
        "URL=http://example.com/content.php?messages",
        "Resource=0",
        "RecContentType=text/xml",
        "Referer=http://example.com/",
        "Snapshot=t4.inf",
        "Mode=HTML",
        LAST);
```

Hinweis: Sie können die Schwellenwerte für asynchrone Anforderungen in VuGen ändern, um VuGen das Auffinden von Unterhaltungen des Typs Poll zu erleichtern. Weitere Informationen finden Sie unter "Verwenden von Schwellenwerten für asynchrone Anforderungen" auf Seite 490.

Das Skript wurde von VuGen nach dem Aufzeichnen der erforderlichen Geschäftsprozesse generiert. Es wurde keine Prüfung auf asynchrone Kommunikation durchgeführt. Beachten Sie, dass das Skript eine Reihe von **web_url**-Funktionen mit einem sich wiederholenden URL enthält: **http://example.com/content.php?messages**. Auf diese **web_url**-Funktionen folgen **lr_think_time**-Funktionen, mit denen festgelegt wird, dass die **web_url**-Funktionen in Intervallen von 4 Sekunden wiederholt werden.

Beim Ausführen des Vuser-Skripts sollen Anforderungen der Seite **http://example.com/content.php?messages** so lange wiederholt werden, bis die

Skriptausführung beendet ist. Zudem sollen die Anforderungen parallel (gleichzeitig) zu anderen im Vuser-Skript enthaltenen Aktionen ausgeführt werden.

Nachdem VuGen das Skript auf asynchrone Kommunikation geprüft hat, sieht es folgendermaßen aus:

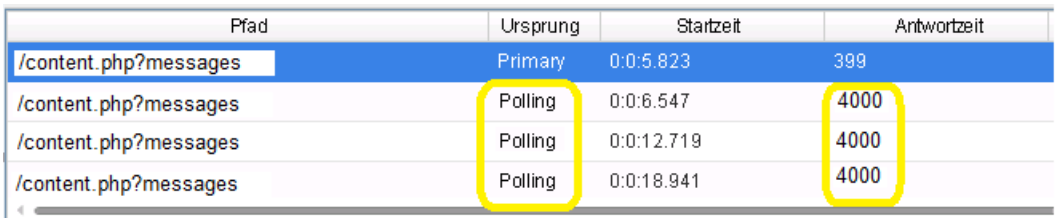
```
    /* Added by Async CodeGen.
ID=Poll_0
ScanType = Recording

The following urls are considered part of this conversation:
    http://example.com/content.php?messages

TODO - The following callbacks have been added to AsyncCallbacks.c.
Add your code to the callback implementations as necessary.
    Poll_0_RequestCB
    Poll_0_ResponseCB
 */
    web_reg_async_attributes("ID=Poll_0",
        "URL=http://example.com/content.php?messages",
        "Pattern=Poll",
        "PollIntervalMs=4000",
        "RequestCB=Poll_0_RequestCB",
        "ResponseCB=Poll_0_ResponseCB",
        LAST);


    web_url("content.php",
        "URL=http://example.com/content.php?messages",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://example.com",
        "Snapshot=t4.inf",
        "Mode=HTML",
        LAST);
```

Beachten Sie, dass die Funktion **web_reg_async_attributes** vor der ersten **web_url**-Funktion eingefügt wurde, mit der **http://example.com/content.php?messages** aufgerufen wird.

Abgesehen vom ersten Aufruf der Seite **http://example.com/content.php?messages** wurden alle anderen **web_url**-Funktionen, die denselben URL aufrufen, von VuGen auskommentiert.

Die **lr_think_time**-Funktionen wurden in einer **lr_think_time**-Funktion zusammengeführt.

Der Snapshot-Bereich für die verbleibende **web_url**-Funktion zeigt, dass die Snapshots für die entfernten **web_url**-Funktionen nun Polling als Ursprung angeben und dass sie in Intervallen von 5 Sekunden ausgeführt werden.



Pfad	Ursprung	Startzeit	Antwortzeit
/content.php?messages	Primary	0:0:5.823	399
/content.php?messages	Polling	0:0:6.547	4000
/content.php?messages	Polling	0:0:12.719	4000
/content.php?messages	Polling	0:0:18.941	4000

Da die Anforderungen mit einer Antwortzeit verbunden sind, wurde die Latenzzeit in den **lr_think_time**-Funktionen zwischen den Poll-Schritten im ursprünglichen Skript auf 4 Sekunden abgerundet.

Beispiel für asynchrone Kommunikation - Push

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Das folgende Beispiel beschreibt ein Vuser-Skript, das einen Browser emuliert, in dem eine Anwendung abläuft, die asynchrone Kommunikation des Typs **Push** nutzt Die Anwendung stellt eine Demoversion eines Börsentickers dar. Der Browser zeigt die Seite mit den Aktienkursen an und sendet eine Anforderung, auf die er eine Antwort mit den aktualisierten Kurswerten erhält. Die Anforderung bleibt offen, bis sie vom Benutzer geschlossen wird. Solange die Seite angezeigt wird, sendet der Server Unternachrichten als Teil der Antwort - wann immer aktualisierte Daten für die angezeigten Aktienkurse vorliegen. Wenn der Client eine solche Nachricht erhält, zeigt er die aktualisierten Aktienkurse an.

Hinweis: Sie können die Schwellenwerte für asynchrone Anforderungen in VuGen ändern, um VuGen das Auffinden von Unterhaltungen des Typs Push zu erleichtern. Weitere Informationen finden Sie unter "Verwenden von Schwellenwerten für asynchrone Anforderungen" auf Seite 490.

Name	Price	Time	Change	Bid Size	Bid	Ask	Ask Size	Min	Max	Ref.
Anduct	3.09	10:32:54	1.64	5000	3.09	3.1	35500	2.48	3.64	3.04
Ations Europe	17.92	10:33:05	11.37	86000	17.92	17.98	1000	12.8	19.33	16.09
Bagies Consulting	6.43	10:33:04	-10.57	4500	6.43	6.44	8500	5.74	8.64	7.19
BAY Corporation	3.2	10:32:03	-11.84	37500	3.2	3.21	63500	3.13	4.28	3.63
CON Consulting	6.8	10:33:05	-10.64	24000	6.8	6.83	51000	6.23	9.08	7.61
Corcor PLC	2.58	10:32:54	12.17	61000	2.58	2.59	42000	1.87	2.7	2.3
CVS Asia	13.48	10:33:06	-12.41	94500	13.45	13.48	11000	12.38	18.34	15.39
Datio PLC	5.81	10:33:00	9.41	42500	5.8	5.81	1500	4.4	6.34	5.31
Dentems	4.87	10:33:05	0.2	12000	4.86	4.87	35500	3.91	5.67	4.86
ELE Manufacturing	7.19	10:33:01	-5.51	13500	7.19	7.2	51000	6.25	9.09	7.61

Wenn Sie versuchen, ein Skript auszuführen, das einen Push-URL aufruft, ohne zuvor eine Prüfung auf asynchrone Kommunikation durchzuführen, wird die Ausführung angehalten und die Antwort auf die markierte Anforderung abgewartet. Nach zwei Minuten zeigt VuGen einen Fehler im Wiedergabeprotokoll an, der in etwa dem folgenden entspricht:

```
Action.c(140): Fehler -27782: Zeitüberschreitung (120 Sekunden) beim Warten auf
den Empfang von Daten für URL "http://push.example.com" [MsgId: MERR-27782]
```

Die Fehlermeldung weist darauf hin, dass die Antwort nie abgeschlossen wurde.

Die Neugenerierung des Skripts mit aktivierter asynchroner Prüfung erstellt ein Skript, das dem folgenden gleicht:

```
/* Added by Async CodeGen.
ID=Push_0
ScanType = Recording

The following urls are considered part of this conversation:
    http://push.example.com/STREAMING_IN_PROGRESS?LS_session=S343716d5eb050c64T2253451&LS_phase=4903&LS_domain=lights",

TODO - The following callbacks have been added to AsyncCallbacks.c.
Add your code to the callback implementations as necessary.
    Push_0_RequestCB
    Push_0_ResponseBodyBufferCB
    Push_0_ResponseCB
 */
    web_reg_async_attributes("ID=Push_0",
        "URL=http://push.example.com/STREAMING_IN_PROGRESS?LS_session={CorrelationParameter}&LS_phase=4903&LS_domain",
        "Pattern=Push",
        "RequestCB=Push_0_RequestCB",
        "ResponseBodyBufferCB=Push_0_ResponseBodyBufferCB",
        "ResponseCB=Push_0_ResponseCB",
        LAST);

    web_url("STREAMING_IN_PROGRESS",
        "URL=http://push.example.com/STREAMING_IN_PROGRESS?LS_session={CorrelationParameter}&LS_phase=4903&LS_domain=",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://www.app.example.com/GWT_StockListDemo_Basic/lightstreamer/lsengine.html",
        "Snapshot=t13.inf",
        "Mode=HTML",
        LAST);

    web_custom_request("control.js",
        "URL=http://push.example.com/lightstreamer/control.js", "",
        "Method=POST",
        "Resource=0",
        "RecContentType=text/plain",
        "Referer=http://push.example.com/ajax_frame.html?phase=594&domain=example.com&",
        "Snapshot=t14.inf",
        "Mode=HTML",
        "Body=LS_session={CorrelationParameter}&LS_table=1&LS_win_phase=19&LS_req_phase=311&LS_op=add&LS_mode=MERGE&LS_id=item1",
        LAST);
/* Added by Async CodeGen.
ID = Push_0
 */
    web_stop_async("ID=Push_0",
        LAST);


    return 0;
}
```

Beachten Sie, dass die Funktion **web_reg_async_attributes** vor der **web_url**-Funktion, die die *Push*-Unterhaltung startet, und die Funktion **web_stop_async** nach dem letzten Aktionsschritt im Skript hinzugefügt wurde. Das Skript wird nun erfolgreich ausgeführt. Die *Push*-Unterhaltung bleibt aktiv und läuft parallel zu anderen Skriptfunktionen bis zur Funktion **web_stop_async** oder bis das Ende des Skripts erreicht ist.

Beachten Sie, dass VuGen während der asynchronen Prüfung keinen Teil des generierten Codes im Vuser-Skript entfernt (auskommentiert) hat.

Beispiel für asynchrone Kommunikation - Long Poll

Hinweis: Dieses Thema bezieht sich auf Web (HTTP/HTML)-, Mobile Application - HTTP/HTML, Flex-, und Webdienst-Vuser-Skripts.

Das folgende Beispiel beschreibt ein Vuser-Skript, das eine Anwendung emuliert, die eine asynchrone Unterhaltung des Typs *long-poll* implementiert. Die Anwendung stellt eine Demo einer "Chat"-Seite dar. Ein Browser stellt die Chat-Seite dar und sendet eine Anfrage, die so lange offen

bleibt, bis ein anderer Benutzer eine neue Nachricht an den Chat sendet. Nachdem eine Nachricht gesendet wurde, geschieht Folgendes:

- Die Antwort wird fertiggestellt.
- Die neue Nachricht wird im Browser angezeigt.
- Der Browser sendet eine weitere Anfrage, um die nächste an den Chat gesendete Nachricht abzuhören.

Hinweis: Sie können die Schwellenwerte für asynchrone Anforderungen in VuGen ändern, um VuGen das Auffinden von Unterhaltungen des Typs Long Poll zu erleichtern. Weitere Informationen finden Sie unter "Verwenden von Schwellenwerten für asynchrone Anforderungen" auf Seite 490.

Das folgende Vuser-Skript hat VuGen nach dem Aufzeichnen der Anwendung und vor der Suche nach asynchroner Kommunikation erstellt. Das Skript enthält eine Reihe von **web_url**-Funktionen mit ähnlichen URLs. Da neue Anfragen gesendet werden, sobald die vorhergehende Antwort abgeschlossen ist, wurden keine **lr_think_time**-Funktionen zwischen den **web_url**-Funktionen eingefügt. Auch daran lässt sich erkennen, dass es sich um eine *Long-Poll*- und nicht um eine *Poll*-Unterhaltung handelt.

Beim Ausführen des Vuser-Skripts sollen immer dann wiederholt Anfragen an die Chat-Anwendung gesendet werden, wenn eine Antwort in der Chat-Anwendung abgeschlossen ist. Zudem sollen Anfragen parallel (gleichzeitig) zu anderen im Skript enthaltenen Aktionen ausgeführt werden.

Nachdem VuGen das Skript auf asynchrone Kommunikation geprüft hat, sieht das Skript folgendermaßen aus:

```
/* Added by Async CodeGen.
ID=LongPoll_0
ScanType = Recording

The following urls are considered part of this conversation:
    http://example.com/request.ashx?key=111111-1111-1111-1111-111111token=123488858&message=%5B%
    http://example.com/request.ashx?key=111111-1111-1111-1111-111111token=123488858&message=%5B%
    http://example.com/request.ashx?key=111111-1111-1111-1111-111111token=123488858&message=%5B%
    http://example.com/request.ashx?key=111111-1111-1111-1111-111111token=123488858&message=%5B%

TODO - The following callbacks have been added to AsyncCallbacks.c.
Add your code to the callback implementations as necessary.
    LongPoll_0_RequestCB
    LongPoll_0_ResponseCB
- */
    web_reg_async_attributes("ID=LongPoll_0",
        "URL=http://example.com/request.ashx?key=111111-1111-1111-1111-111111token=123488858&message=%5B%
        "Pattern=LongPoll",
        "RequestCB=LongPoll_0_RequestCB",
        "ResponseCB=LongPoll_0_ResponseCB",
        LAST);

    web_url("request.ashx_3",
        "URL=http://example.com/request.ashx?key=111111-1111-1111-1111-111111token=123488858&message=%5B%
        "Resource=0",
        "RecContentType=text/html",
        "Referer=",
        "Snapshot=t4.inf",
        "Mode=HTML",
        LAST);

/* Removed by Async CodeGen.
ID = LongPoll_0
- */
    /*  http://example.com/request.ashx?key=111111-1111-1111-1111-111111token=123488858&message=%5B%
 web_url("request.ashx_4",
        "URL=http://example.com/request.ashx?key=111111-1111-1111-1111-111111token=123488858&message=%5B%
        "Resource=0",
        "RecContentType=text/html",
        "Referer=",
        "Snapshot=t5.inf",
        "Mode=HTML",
        LAST);
-   */

/* Removed by Async CodeGen.
```

Beachten Sie, dass die Funktion **web_reg_async_attributes** vor der ersten **web_url**-Funktion, mit der die Chat-Anwendung aufgerufen wird, eingefügt wurde.

Abgesehen vom ersten Aufruf der Chat-Anwendung wurden alle **web_url**-Funktionen, die ähnliche URLs aufrufen, auskommentiert.

Ein Blick in den Snapshot-Bereich auf die restlichen **web_url**-Funktionen zeigt, dass die Snapshots der entfernten Schritte jetzt als Ursprung Polling angeben. Beachten Sie, dass die Antwortzeiten sehr unterschiedlich sind, da Antworten nur eintreffen, wenn ein anderer Benutzer eine Nachricht gesendet hat. Daran lässt sich erkennen, dass es sich um eine Long-Poll- und nicht um eine Poll-Unterhaltung handelt.

Anzeigen von Wiedergabeergebnissen

Im Abschnitt **Anzeigen von Wiedergabeergebnissen** wird erläutert, wie Sie die Ergebnisse einer Vuser-Skriptwiedergabe anzeigen und anpassen können.

Anzeigen von Wiedergabeergebnissen – Übersicht

Das Erstellen eines Vuser-Skripts umfasst die nachfolgend aufgeführten Schritte. Dieses Thema enthält eine Übersicht über den siebten Schritt (Anzeigen der Ergebnisse der Wiedergabe eines Vuser-Skripts).

Zur Unterstützung beim Debuggen eines Vuser-Skripts können Sie einen Bericht anzeigen, der die Ergebnisse Ihrer Skriptausführung zusammenfasst. VuGen generiert den Bericht während der Ausführung des Vuser-Skripts; Sie können ihn nach abgeschlossener Skriptausführung anzeigen.

Im Fenster **Testergebnisse** werden sämtliche Aspekte der Testausführung angezeigt, wie beispielsweise:

- Ein Ergebnisübersichtsbericht auf hoher Ebene (Status "Bestanden"/"Fehlgeschlagen")
- Die in allen Ausführungen verwendeten Daten
- Eine erweiterbare Struktur der Schritte mit exakter Angabe, wo Anwendungsfehler aufgetreten sind
- Die exakten Positionen im Skript, an denen die Fehler aufgetreten sind
- Eine Momentaufnahme des Status Ihrer Anwendung an einem bestimmten Schritt
- Ein Videoclip des Status Ihrer Anwendung an einem bestimmten Schritt oder des gesamten Tests
- Ausführliche Beschreibungen jedes Schritts und Informationen zu einem bestandenen oder nicht bestandenen Prüfpunkt in jeder Phase des Tests

Anpassen der Anzeige für die Testergebnisse

Jeder Ergebnissatz wird in einer **.xml**-Datei gespeichert (als **results.xml** bezeichnet). Diese **.xml**-Datei speichert Informationen zu jedem der Testergebnisknoten in der Anzeige. Die Informationen in diesem Knoten werden zur dynamischen Erstellung von **.htm**-Dateien verwendet, die im oberen rechten Bereich des Fensters **Testergebnisse** angezeigt werden.

Jeder Knoten in der Ausführungsergebnisstruktur ist ein Element in der **results.xml**-Datei. Darüber hinaus gibt es verschiedene Elemente, die verschiedene Typen von Informationen darstellen, die in den Testergebnissen angezeigt werden. Sie können Testergebnisinformationen aus der **.xml**-Datei verwenden und die erforderlichen Informationen mithilfe von XSL in einem angepassten Format anzeigen (entweder beim Drucken vom Fenster **Testergebnisse** aus, beim Anzeigen der Testergebnisse in Ihrer eigenen angepassten Umgebung zum Anzeigen von Ergebnissen oder beim Exportieren der Testergebnisse in eine HTML-Datei).

XSL stellt Ihnen die erforderlichen Tools bereit, um genau zu beschreiben, welche Testergebnisinformationen angezeigt werden sollen und wo sie genau angezeigt, gedruckt oder exportiert werden sollen. Mit einem XSL-Editor können Sie die **.css**- und **.xsl**-Dateien im Ergebnisordner bearbeiten, um die Darstellung des Berichts zu ändern, z. B. Schriftarten, Farben).

Beispielsweise enthält in der Datei **results.xml** ein Elementtag den Namen einer Aktion und ein anderes Elementtag enthält die Uhrzeit, zu der die Ausführung erfolgte. Mit XSL können Sie Ihrem angepassten Editor mitteilen, dass der Aktionsname an einer spezifischen Stelle auf der Seite und

in einer fetten grünen Schriftart angezeigt werden soll und dass die Uhrzeitinformationen nicht angezeigt werden sollen.

Verbinden zu Application Lifecycle Management über das Fenster "Testergebnisse"

Um Fehler vom Fenster **Testergebnisse** aus manuell an Application Lifecycle Management zu übermitteln, müssen Sie mit Application Lifecycle Management verbunden sein.

Der Verbindungsprozess umfasst zwei Stufen.

- Zunächst stellen Sie eine Verbindung zu einem lokalen oder Remote-Application Lifecycle Management-Server her. Dieser Server verarbeitet die Verbindungen zwischen den Testergebnissen und dem Application Lifecycle Management-Projekt.
- Anschließend melden Sie sich an und wählen das Projekt aus, auf das Sie zugreifen möchten. Das Projekt speichert Tests und Informationen zur Ausführungssitzung für die von Ihnen getestete Anwendung.

Hinweis: Application Lifecycle Management-Projekte sind durch ein Kennwort geschützt, daher müssen Sie einen Benutzernamen und ein Kennwort angeben.

Weitere Informationen zum Herstellen einer Verbindung zu einem ALM-Projekt finden Sie unter "Verwenden von Skripts in ALM-Projekten" auf Seite 150.

Senden von benutzerdefinierten Informationen an den Bericht

Zusätzlich zu den automatisch an den Bericht gesendeten Informationen können Sie für Webdienst-Vuser mit den Meldungsfunktionen **lr_output_message** oder **lr_error_message** Informationen an den Bericht senden.

Weitere Informationen über die Aufgabe finden Sie unter "Einfügen von Schritten in ein Skript" auf Seite 434.

Konfigurieren der Anzeige des Fensters "Testergebnisse"

Standardmäßig weist das Fenster **Testergebnisse** dieselbe Darstellung und Funktionsweise wie das Fenster auf, bei dem das Microsoft Office 2003-Design verwendet wird. Sie können die Darstellung und Funktionsweise des Fensters **Testergebnisse** wie gewünscht ändern.

Zum Ändern dieser Einstellungen wählen Sie **Ansicht > Fensterdesign** und wählen das gewünschte Design aus.

Hinweis: Sie können das Microsoft Windows XP-Design für das Fenster **Testergebnisse** nur anwenden, wenn Ihr Computer für die Verwendung eines Windows XP-Designs eingerichtet ist.

Öffnen des Fensters "Testergebnisse" einer bestimmten Ausführung

In dieser Aufgabe wird das Öffnen eines Testergebnisfensters für eine bestimmte Ausführung beschrieben.

1. Wählen Sie im Fenster **Testergebnisse** die Optionen **Datei > Öffnen** aus.
2. Wählen Sie eine Skriptdatei für die Anzeige des Testergebnisses und anschießend die gewünschte Testergebnisdatei aus. Standardmäßig sind Ergebnisdateien für Tests in **<Skript>\<Ergebnisname>.xml** gespeichert. Ist Ihr Skript in Application Lifecycle Management gespeichert, finden Sie weiter unten detaillierte Informationen.
3. Wählen Sie ein Ergebnis aus und klicken Sie auf **Öffnen**.

Hinweis: Ergebnisdateien für frühere Versionen wurden mit der Dateierweiterung **.qtp** gespeichert. Im Dialogfeld **Ergebnisdatei auswählen** werden nur Ergebnisdateien mit der Erweiterung **.xml** standardmäßig angezeigt. Zum Anzeigen von Dateien mit der Erweiterung **.qtp** im Dialogfeld **Ergebnisdatei auswählen** wählen Sie **Testergebnisse (*.qtp)** im Feld **Dateityp** aus.

Auswählen eines in Application Lifecycle Management gespeicherten Skripts

1. Klicken Sie auf die Schaltfläche **Application Lifecycle Management-Verbindung** und stellen Sie eine Verbindung zu Ihrem Application Lifecycle Management-Projekt her.
2. Geben Sie in das Dialogfeld **Testergebnisse öffnen** den Pfad des Ordners ein, der die Ergebnisdatei für Ihren Test enthält, oder klicken Sie auf die Schaltfläche **Durchsuchen**, um das Dialogfeld **Test öffnen aus dem HP ALM-Projekt** zu öffnen.
3. Wählen Sie **DB-Vuser** in der Testtypliste aus.
4. Suchen Sie das Skript, dessen Ergebnisse Sie anzeigen möchten, und klicken Sie auf **OK**.
5. Markieren Sie im Dialogfeld **Testergebnisse öffnen** den Testergebnissatz, den Sie anzeigen möchten, und klicken Sie auf **Öffnen**. Das Fenster **Testergebnisse** zeigt die ausgewählten Testergebnisse an.

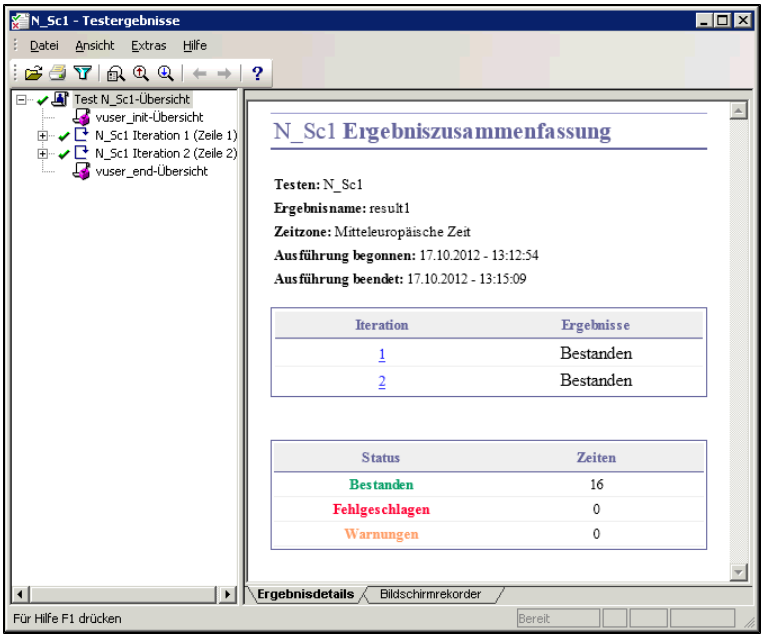
Auffinden von Schritten in den Testergebnissen

In dieser Aufgabe wird beschrieben, wie Sie die Testergebnisse für Schritte eines bestimmten Typs suchen.

1. Wählen Sie im Fenster **Testergebnisse** die Optionen **Extras > Suchen** aus.
2. Wählen Sie den Schritttyp aus, den Sie finden möchten. Es können mehrere Optionen ausgewählt werden.
3. Wählen Sie **Nach oben** oder **Nach unten** aus, um die Suchrichtung anzugeben.
4. Wählen Sie **Weiter suchen** aus, um das nächste Auftreten des ausgewählten Schritttyps zu finden.

Fenster "Testergebnisse"

In diesem Fenster wird ein Bericht angezeigt, in dem die Ergebnisse der Skriptausführung zusammengefasst werden.



Zugriff	• **Wird automatisch nach der Skriptausführung geöffnet.** • **VuGen > Wiedergeben > Testergebnisse.**

Forts.

Wichtige Informationen	• Zum Konfigurieren der Ergebniseinstellungen wählen Sie **Extras > Optionen > Skripterstellung > Wiedergeben** aus. • Das Fenster **Testergebnisse** kann Ergebnisse mit bis zu 300 Ebenen in der hierarchischen Struktur anzeigen. Wenn Ergebnisse mit über 300 verschachtelten Ebenen vorliegen, können Sie den gesamten Bericht anzeigen, indem Sie die Datei **results.xml** öffnen.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

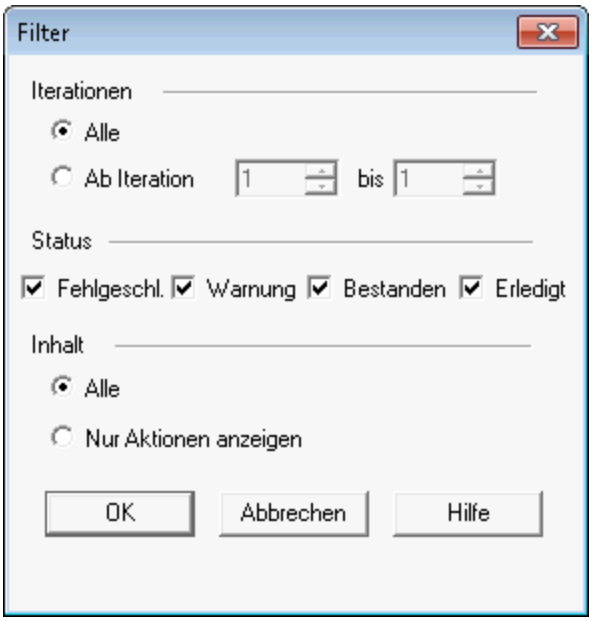
Element der Oberfläche	Beschreibung
Berichtstruktur	Eine grafische Darstellung der Testergebnisse, die in einem Ausschnitt links im Fenster angezeigt wird. Sie können eine Verzweigung in der Ausführungsergebnisstruktur ausblenden oder erweitern, um die von der Struktur angezeigte Detailebene zu ändern. Die Symbole neben den Schritten verweisen auf die folgenden Informationen: ✔ zeigt einen erfolgreichen Schritt an. ✘ zeigt einen fehlgeschlagenen Schritt an. ! zeigt eine Warnung an, die angibt, dass der Schritt nicht erfolgreich war, jedoch kein Fehlschlagen des Tests verursachte. ! ⊗ zeigt einen Schritt an, der unerwartet fehlgeschlagen ist. Die Optionen zum Ein- und Ausblenden der Knoten können Sie über das Menü **Ansicht** auswählen oder indem Sie auf * klicken.
Registerkarte "Ergebnisdetails" (Ergebnisübersicht)	Enthält die Einzelheiten der Testausführung, die sich je nach ausgewähltem Teil der Berichtsstruktur ändern können. Wenn Sie den oberen Knoten der Struktur auswählen, zeigt die Registerkarte **Ergebnisdetails** eine Zusammenfassung der Ergebnisse für Ihren Test an. Wenn Sie eine Verzweigung oder einen Schritt in der Struktur auswählen, finden Sie in der Registerkarte **Ergebnisdetails** die Details für diesen Schritt. Die Registerkarte **Ergebnisdetails** kann auch eine Momentaufnahme Ihrer Anwendung für den hervorgehobenen Schritt enthalten. • Iterationen, Aktionen und Schritte, die Prüfpunkte enthalten, sind im rechten Teil des Fensters **Testergebnisse** mit **Bestanden** oder **Fehlgeschlagen** markiert und durch Symbole im Strukturbereich gekennzeichnet. • Iterationen, Aktionen und Schritte, die erfolgreich ausgeführt wurden, aber keine Prüfpunkte enthalten, sind im rechten Teil des Fensters **Testergebnisse** mit **Fertig** markiert.

Forts.

Registerkarte "Bildschirmrekorder"	Enthält das aufgezeichnete Video zusammen mit den Testergebnissen. Wenn Ihren Testergebnissen kein Video zugeordnet ist, enthält die Registerkarte **Bildschirmrekorder** die folgende Meldung: `Den Ergebnissen ist kein Film zugewiesen.`
	Öffnet die Testergebnisse einer speziellen Ausführung. Weitere Informationen finden Sie unter "Öffnen des Fensters "Testergebnisse" einer bestimmten Ausführung" auf Seite 516.
	Öffnet das Dialogfeld **Drucken**, über das Sie die Testergebnisse drucken können. Weitere Informationen finden Sie unter "Dialogfeld "Drucken"" auf Seite 521.
	Öffnet das Dialogfeld **Filter**, über das Sie die Testergebnisse filtern können. Weitere Informationen finden Sie unter "Dialogfeld "Filter"" unten.
	Öffnet das Dialogfeld **Suchen**, über das Sie die Testergebnisse nach bestimmten Schritttypen durchsuchen können. Weitere Informationen finden Sie unter "Auffinden von Schritten in den Testergebnissen" auf Seite 517.
	Durchsucht die Testergebnisse nach dem vorherigen Schritt, der die Kriterien des Dialogfelds **Suchen** erfüllt.
	Durchsucht die Testergebnisse nach dem nächsten Schritt, der die Kriterien des Dialogfelds **Suchen** erfüllt.
	Wählt den vorherigen Knoten aus.
	Wählt den nächsten Knoten aus.
	Öffnet die Produktdokumentation.

Dialogfeld "Filter"

Über dieses Dialogfeld können Sie die Testergebnisse im Fenster **Testergebnisse** filtern.



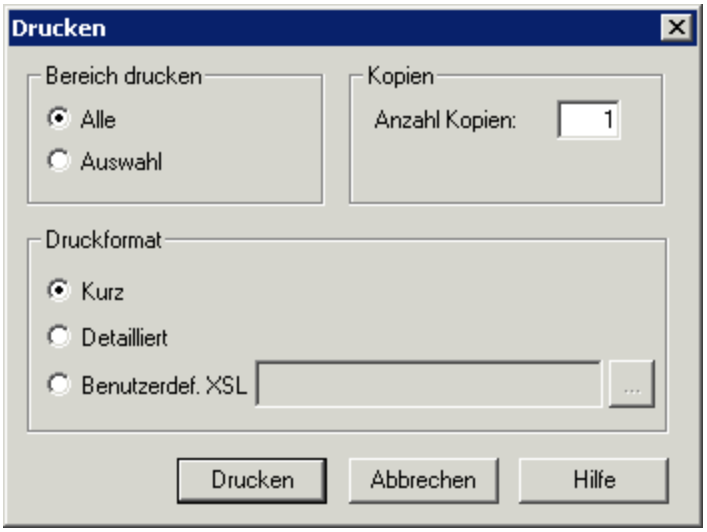
Zugriff	"Fenster "Testergebnisse"" auf Seite 517 **> Ansicht > Filter**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Iterationen	• **Alle.** Zeigt Testergebnisse von allen Iterationen an. • **Von Iteration x bis y.** Zeigt die Testergebnisse eines spezifischen Bereichs von Testiterationen an.
Status	• **Fehlgeschlagen.** Zeigt die Ergebnisse für die Schritte an, die fehlgeschlagen sind. • **Warnung.** Zeigt die Ergebnisse für die Schritte mit dem Status **Warnung** an (Schritte, die nicht den Status **Bestanden** aufweisen, die aber nicht zu einem Fehlschlagen des Skripts geführt haben). • **Bestanden.** Zeigt die Ergebnisse für die Schritte an, die bestanden wurden. • **Erledigt.** Zeigt die Ergebnisse für die Schritte mit dem Status **Erledigt** an (Schritte die erfolgreich durchgeführt wurden, die jedoch nicht auf den Status **Bestanden**, **Fehlgeschlagen** oder **Warnung** gesetzt wurden).
Inhalt	• **Alle.** Zeigt alle Schritte von allen Knoten im Test an. • **Nur Aktionen anzeigen.** Zeigt die Aktionsknoten im Test an (nicht die spezifischen Schritte in den Aktionsknoten).

Dialogfeld "Drucken"

Mithilfe dieses Dialogfelds können Sie die Testergebnisse drucken.



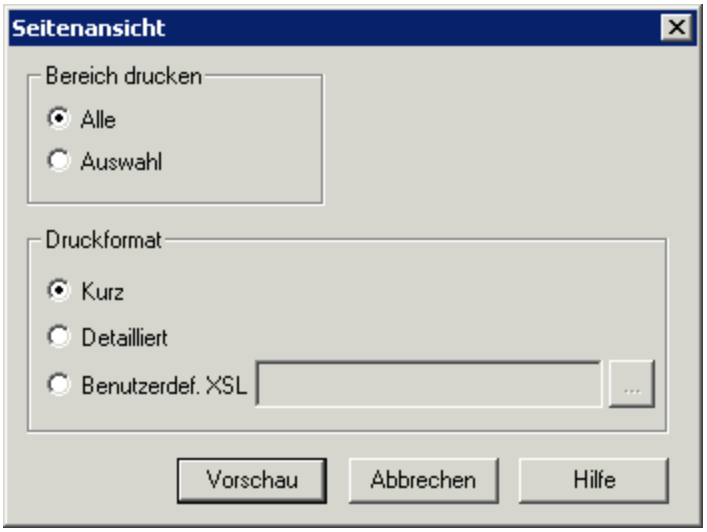
Zugriff	"Fenster "Testergebnisse"" auf Seite 517 **> Datei > Drucken**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Bereich drucken	• **Alle.** Druckt die Ergebnisse für das gesamte Skript. • **Auswahl.** Druckt die Ergebnisse für die ausgewählte Verzweigung in der Ausführungsergebnisstruktur.
Kopien	Die Anzahl der zu druckenden Kopien.
Druckformat	• **Kurz.** Druckt eine Zusammenfassungszeile (wenn verfügbar) für jedes Element in der Ausführungsergebnisstruktur. Diese Option steht nur zur Verfügung, wenn der Druckbereich auf **Alle** gesetzt ist. • **Detailliert.** Druckt alle verfügbaren Informationen für jedes Element in der Ausführungsergebnisstruktur oder für die ausgewählte Verzweigung. • **Benutzerdefinierte XSL.** Ermöglicht es Ihnen, zu einer angepassten **.xsl**-Datei zu navigieren und sie auszuwählen. Sie können eine angepasste **.xsl**-Datei erstellen, die angibt, welche Informationen in den gedruckten Bericht eingefügt und wie sie dargestellt werden sollen. Weitere Informationen finden Sie unter "Anpassen der Anzeige für die Testergebnisse" auf Seite 514.

Dialogfeld "Seitenansicht"

Mithilfe dieses Dialogfelds können Sie eine Seitenansicht der Testergebnisse anzeigen.



Zugriff	"Fenster "Testergebnisse"" auf Seite 517 **> Datei > Seitenansicht**
Wichtige Informationen	Wenn ein Teil der Informationen in der Vorschau abgeschnitten ist, beispielsweise bei Prüfpunktnamen, die zu lang sind, um in die Anzeige zu passen, können Sie auf die Schaltfläche **Vorschau** im Fenster **Seitenansicht** klicken und die Seitenausrichtung von **Hochformat** in **Querformat** ändern.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Bereich drucken	• **Alle.** Druckt die Ergebnisse für das gesamte Skript. • **Auswahl.** Druckt die Ergebnisse für die ausgewählte Verzweigung in der Ausführungsergebnisstruktur.

Forts.

Druckformat	• **Kurz.** Druckt eine Zusammenfassungszeile (wenn verfügbar) für jedes Element in der Ausführungsergebnisstruktur. Diese Option steht nur zur Verfügung, wenn der Druckbereich auf **Alle** gesetzt ist. • **Detailliert.** Druckt alle verfügbaren Informationen für jedes Element in der Ausführungsergebnisstruktur oder für die ausgewählte Verzweigung. • **Benutzerdefinierte XSL.** Ermöglicht es Ihnen, zu einer angepassten **.xsl**-Datei zu navigieren und sie auszuwählen. Sie können eine angepasste **.xsl**-Datei erstellen, die angibt, welche Informationen in den gedruckten Bericht eingefügt und wie sie dargestellt werden sollen. Weitere Informationen finden Sie unter "Anpassen der Anzeige für die Testergebnisse" auf Seite 514.

Dialogfeld "In HTML-Datei exportieren"

Mithilfe dieses Dialogfelds können Sie die Testberichte in eine HTML-Datei exportieren. Dadurch können Sie die Ergebnisse anzeigen, auch wenn die Umgebung zum Anzeigen der Testergebisse nicht verfügbar ist. Beispielsweise können Sie die Datei mit den Ergebnissen in einer E-Mail an eine dritte Partei senden. Sie können den Berichtstyp, den Sie exportieren möchten, auswählen. Außerdem können Sie einen angepassten Bericht erstellen und drucken.

Zugriff	"Fenster "Testergebnisse"" auf Seite 517 > Datei > In HTML-Datei exportieren
Wichtige Informationen	In Testergebnisberichten steht das Dienstprogramm **In HTML-Datei exportieren** nur für Skripts zur Verfügung, die in Version 9.50 oder höher wiedergegeben wurden. Um einen HTML-Bericht für Skripts zu erstellen, die mit früheren Versionen erstellt wurden, führen Sie das Skript erneut in Version 9.50 (oder höher) des Produkts aus.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Bereich exportieren	• **Alle.** Exportiert die Ergebnisse für das gesamte Skript. • **Auswahl.** Exportiert Ergebnisinformationen für die ausgewählte Verzweigung in der Ergebnisstruktur.
Exportformat	• **Kurz.** Druckt eine Zusammenfassungszeile (wenn verfügbar) für jedes Element in der Ausführungsergebnisstruktur. Diese Option steht nur zur Verfügung, wenn der Druckbereich auf **Alle** gesetzt ist. • **Detailliert.** Druckt alle verfügbaren Informationen für jedes Element in der Ausführungsergebnisstruktur oder für die ausgewählte Verzweigung. • **Benutzerdefinierte XSL.** Ermöglicht es Ihnen, zu einer angepassten **.xsl**-Datei zu navigieren und sie auszuwählen. Sie können eine angepasste **.xsl**-Datei erstellen, die angibt, welche Informationen in den gedruckten Bericht eingefügt und wie sie dargestellt werden sollen. Weitere Informationen finden Sie unter "Anpassen der Anzeige für die Testergebnisse" auf Seite 514.

Protokolle

Protokollunterstützung für asynchrone, IPv6- und 64-Bit-Aufzeichnung

In der folgenden Tabelle ist die Protokollunterstützung für asynchrone, IPv6- und 64-Bit-Aufzeichnung dargestellt:

Protokoll	Asynchron	IPv6	64-Bit-Aufzeichnung
DNS	Nein	Ja	Nein
Flex AMF	Ja	Ja	Nein
Flex RMTP	Nein	Nein	Nein
FTP	Nein	Ja	Nein
IMAP	Nein	Ja	Nein
Java over HTTP	Nein	Ja	Ja
LDAP	Nein	Nein	Ja
Microsoft .NET	Nein	Nein	Ja
Mobile Anwendungen (HTTP/HTML)	Nein	Ja	Nein
Windows Sockets (Mehrfachprotokoll)	Nein	Ja	Ja
Oracle 2-Tier	Nein	Nein	Ja
Oracle NCA	Nein	Ja	Ja
POP3	Nein	Ja	Nein
RDP	Nein	Ja	Ja
SAP – Web	Nein	Nein	Ja

Protokoll	Asynchron	IPv6	64-Bit-Aufzeichnung
Siebel - Web	Nein	Nein	Ja
Silverlight	Nein	Ja	Nein
SMTP	Nein	Ja	Nein
TruClient Ajax - FireFox	Nein	Ja	Nein
TruClient Ajax - IE	Nein	Ja	Nein
TruClient Ajax - Mobil	Nein	Ja	Nein
Click & Script-Familie (Ajax, SAP)	Nein	Ja	Nein
Web - HTTP/HTML	Ja	Ja	Ja
Webdienste	Ja	Ja	Ja

Hinweis: In allen Protokollen mit Unterstützung für asynchrone Sitzungen wird die Aufzeichnung nur auf web_*-Schritte angewendet.

Ajax - Click & Script-Protokoll

Ajax (Asynchronous JavaScript and XML) steht für eine Gruppe von Techniken zum Erstellen interaktiver Webanwendungen. Bei Ajax werden kleine Datenpakete mit dem Server ausgetauscht, anstatt Webseiten vollständig neu zu laden. Hierdurch kommt es zu geringeren Verzögerungen, wenn Daten vom Benutzer angefragt werden. Darüber hinaus wird die interaktive Funktionalität erweitert und die Nutzbarkeit dieser Anwendungen verbessert.

Mithilfe von Ajax lassen sich schnelle Webseiten entwickeln, die auf JavaScript und asynchronen Serveranfragen basieren. Die Anfragen sind entweder Benutzeraktionen, Zeitgeberereignisse bzw. andere vordefinierte Trigger.

Ajax-Komponenten, so genannte Ajax-Steuerelemente, sind benutzeroberflächenbasierte Steuerelemente, welche die Ajax-Technik nutzen: Beim Auslösen eines Triggers senden sie eine Anforderung an den Server.

Ein weit verbreitetes Ajax-Steuerelement ist beispielsweise **Reorder List**, mit dem Sie Elemente an eine gewünschte Position in einer Liste ziehen können. Die Ajax-Implementierung in VuGen basiert auf Microsofts ASP.NET Ajax Control-Toolkit (zuvor Atlas).

Ajax (Click and Script)-Protokoll - Übersicht

Ajax (Asynchronous JavaScript and XML) steht für eine Gruppe von Techniken zum Erstellen interaktiver Webanwendungen. Bei Ajax werden kleine Datenpakete mit dem Server ausgetauscht, anstatt Webseiten vollständig neu zu laden. Hierdurch kommt es zu geringeren Verzögerungen, wenn Daten vom Benutzer angefragt werden. Darüber hinaus wird die interaktive Funktionalität erweitert und die Nutzbarkeit dieser Anwendungen verbessert.

Mithilfe von Ajax lassen sich schnelle Webseiten entwickeln, die auf JavaScript und asynchronen Serveranfragen basieren. Die Anfragen sind entweder Benutzeraktionen, Zeitgeberereignisse bzw. andere vordefinierte Trigger.

Ajax-Komponenten, so genannte Ajax-Steuerelemente, sind benutzeroberflächenbasierte Steuerelemente, welche die Ajax-Technik nutzen: Beim Auslösen eines Triggers senden sie eine Anforderung an den Server.

Ein weit verbreitetes Ajax-Steuerelement ist beispielsweise **Reorder List**, mit dem Sie Elemente an eine gewünschte Position in einer Liste ziehen können. Die Ajax-Implementierung in VuGen basiert auf Microsofts ASP.NET Ajax Control-Toolkit (zuvor Atlas).

Einen Überblick über die Click and Script-Protokolle finden Sie unter "Click & Script-Protokolle – Überblick" auf Seite 656.

Unterstützte Frameworks für das Ajax (Click & Script)-Protokoll

Folgende Frameworks für AJAX Click and Script-Funktionen werden unterstützt:

- Atlas 1.0.10920.0/ASP.NET Ajax – Alle Steuerelemente
- Scriptaculous 1.8 – Autocomplete, Reorder List und Slider

VuGen unterstützt die folgenden Frameworks auf Modulebene. Dies bedeutet, dass in VuGen standardmäßige Click & Script-Schritte erstellt werden können, jedoch keine Ajax-spezifischen Funktionen:

- Prototype 1.6
- Google Web Toolkit (GWT) 1.4

Ajax (Click & Script)-Beispielskript

VuGen nutzt zur Simulation einer Operation mit einem GUI-Steuerelement die Steuerhandlerschicht. Wenn VuGen während der Aufzeichnung auf eines der unterstützten Ajax-Steuerelemente stößt, wird eine Funktion mit einem Präfix im Format **ajax_xxx** generiert.

Im folgenden Beispiel wählt der Benutzer Element Nr. 1 (index=1) in einem Accordion-Steuerelement aus. VuGen generiert eine **ajax_accordion**-Funktion.

```
web_browser("Accordion.aspx",
                DESCRIPTION,
                ACTION,
                "Navigate=http://labm1app08/AJAX/Accordion/.aspx",
                LAST);
lr_think_time(5);

ajax_accordion("Accordion",
                  DESCRIPTION,
                 "Framework=atlas",
                 "ID=ctl00_SampleContent_MyAccordion",
                    ACTION,
                 "UserAction=SelectIndex",
                 "Index=1",
                 LAST);
web_edit_field("free_text_2",
        "Snapshot=t18.inf",
        DESCRIPTION,
        "Type=text",
        "Name=free_text",
        ACTION,
        "SetValue=FILE_PATH",
        LAST);
```

Hinweis: Beim Aufzeichnen einer Ajax-Sitzung generiert VuGen standardmäßige Click & Script-Funktionen für nicht unterstützte Ajax-Steuerelemente. Im Beispiel oben wurde **FILE_PATH** in ein Bearbeitungsfeld eingegeben.

Ajax (Click & Script) - Aufzeichnungstipps

Dieser Abschnitt enthält Tipps für die Aufzeichnung von Click-and-Script-Vuser-Skripts.

Hinweis: Einige der unten aufgeführten Elemente gelten nur für bestimmte Click and Script-Protokolle.

Verwenden der Maus anstelle der Tastatur

Es empfiehlt sich, mit der Maus auf ein Objekt zu klicken, statt die Tastatur zu verwenden. Verwenden Sie während der Aufzeichnung nur GUI-Objekte, die sich innerhalb des Browserfensters befinden. Verwenden Sie keine Browsersymbole, keine Steuerelemente, keine Menüelemente wie **Ansicht > Aktualisieren** und verwenden Sie nicht die Schaltfläche **Anhalten**. Sie können jedoch die Schaltflächen **Aktualisieren**, **Startseite**, **Zurück Vorwärts** und die Adressleiste verwenden.

Zeichnen Sie nicht in einem vorhandenen Skript auf

Es empfiehlt sich, in einem neu erstellten Skript aufzuzeichnen und nicht in einem vorhandenen.

Vermeiden Sie Kontextmenüs

Vermeiden Sie die Verwendung von Kontextmenüs während der Aufzeichnung. Kontextmenüs sind Menüs, die angezeigt werden, wenn Sie mit der rechten Maustaste auf ein Element in der grafischen Benutzeroberfläche klicken.

Arbeiten Sie während der Aufzeichnung nicht in einem anderen Browser

Arbeiten Sie während der Aufzeichnung nur in den Browserfenstern, die von VuGen geöffnet werden, und nicht in anderen.

Warten Sie Downloads ab

Warten Sie, bis alle Downloads abgeschlossen sind, bevor Sie eine Aktion vornehmen, z. B. auf eine Schaltfläche klicken oder ein Textfeld ausfüllen.

Warten Sie, bis Seiten geladen sind

Während der Aufzeichnung empfiehlt es sich zu warten, bis eine Seite vollständig geladen ist, bevor Sie den nächsten Schritt ausführen. Wenn Sie nicht gewartet haben, bis alle Seiten geladen wurden, zeichnen Sie das Skript erneut auf.

Navigieren zur Startseite

Wenn die letzte Seite in einer Aktion nicht die Links und Schaltflächen enthält, die zu Beginn der Iteration zur Verfügung standen, schlägt die nächste Iteration fehl. Wenn die erste Seite beispielsweise den Textlink **Flugbuchung** enthält, müssen Sie sicherstellen, dass Sie am Ende der Aufzeichnung zu der entsprechenden Seite navigieren, sodass derselbe Link am Ende des Geschäftsprozesses angezeigt wird.

Verwenden Sie eine höhere Ereigniskonfigurationsebene

Zeichnen Sie den Geschäftsprozess erneut mit der Ereigniskonfigurationsebene **Hoch** auf. Weitere Informationen zum Ändern der Ereigniskonfigurationsebene finden Sie unter "Click & Script - Fehlerbeseitigung und Einschränkungen" auf Seite 666.

Deaktivieren Sie die Aufzeichnung auf Socketebene

In bestimmten Fällen unterbricht die Aufzeichnung der Meldungen auf Socketebene die Anwendung. Für die meisten Aufzeichnungen sind Daten der Socketebene nicht erforderlich. Um die Aufzeichnung dieser Daten zu verhindern, deaktivieren Sie die Option in den Aufzeichnungsoptionen. Weitere Informationen finden Sie unter "GUI-Eigenschaften > Erweitert (Knoten)" auf Seite 217.

Aktivieren Sie die Option "Renderingbezogene Eigenschaftswerte aufzeichnen"

Wenn die Skripts auf Clientseite der Anwendung zahlreiche Styling-Aktivitäten verwenden, aktivieren Sie die Option **Renderingbezogene Eigenschaftswerte aufzeichnen**, bevor Sie das Skript aufzeichnen. Aktivieren Sie beispielsweise diese Option, um zusätzliche DOM-Eigenschaften, wie z. B. **offsetTop**, zu erfassen. Beachten Sie, dass das Aktivieren dieser Option möglicherweise die Aufzeichnungsgeschwindigkeit verringert. Sie können die Option aktivieren, indem Sie **Aufzeichnungsoptionen > GUI-Eigenschaften > Erweitert** auswählen. Weitere Informationen finden Sie unter "GUI-Eigenschaften > Erweitert (Knoten)" auf Seite 217.

Ajax (Click & Script) - Wiedergabetipps

Dieser Abschnitt enthält Tipps für die Wiedergabe von Click-and-Script-Vuser-Skripts.

Hinweis: Einige der unten aufgeführten Elemente gelten nur für bestimmte Click and Script-Protokolle.

Beibehalten der Anweisungsreihenfolge in einem aufgezeichneten Skript

Ändern Sie die Reihenfolge von Anweisungen in einem aufgezeichneten Skript nicht. Es empfiehlt sich ferner nicht, Codesegmente aus einer Aktion in eine andere zu kopieren.

Konvertieren Sie Zeichen außerhalb des ASCII-Bereichs

Wenn Links Zeichen außerhalb des ASCII-Bereichs enthalten, sollten Sie VuGen anweisen, die Datei in das UTF-8-Format oder aus dem UTF-8-Format zu konvertieren.

Aktivieren der UTF-8-Konvertierung

1. Wählen Sie **Wiedergabe > Laufzeiteinstellungen** und dann den Knoten **Internetprotokoll > Einstellungen** aus.
2. Klicken Sie auf **Optionen**, um das Dialogfeld **Erweiterte Optionen** zu öffnen.
3. Wechseln Sie zu der Option **Aus/in UTF-8 konvertieren** und setzen Sie sie auf **Ja**.

Wahlweise können Sie die Liste der Optionen durchgehen, die eingeblendet wird, wenn ein Link nicht gefunden wird. Geben Sie den angezeigten Text unverändert ein, z. B. die hexadezimale Escapesequenz **\xA0** oder andere nicht standardmäßige Formate.

Zweimaliges Ausführen derselben Abfolge von Aktionen

In einigen Fällen können Sie einen bestimmten Prozess, z. B. einen Benutzer aus der Datenbank löschen, nur einmal durchführen. Die Wiedergabe schlägt nach der ersten Iteration fehl, da die Aktion nicht mehr gültig ist. Stellen Sie sicher, dass Ihr Geschäftsprozess mehrmals mit denselben Daten wiederholt werden kann.

Eindeutige Bild-Eigenschaften festlegen

Doppelklicken Sie im Schrittenavigator auf den vorherigen Bildschritt, um seine Eigenschaften zu öffnen. Wenn die Eigenschaften **Id**, **Name** und **Alt** leer sind, stellen Sie weitere Identifizierungsinformationen des Bilds bereit, z. B. seinen Dateinamen in der Eigenschaft **Src**.

Alternativ können Sie das Argument **Ordnungszahl** hinzufügen, um die Anzahl der Vorkommen des Bilds auf der jeweiligen Seite anzugeben. Das Argument **Ordnungszahl** identifiziert jedes Bild auf der Seite eindeutig; die anderen Identifikationsargumente hingegen sind nicht eindeutig. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Überprüfen der Schrittbeschreibung

Wenn Sie den Fehler **GUI-Objekt wurde nicht gefunden** erhalten, überprüfen Sie den Ausgabebereich auf eine Liste der Objekte in dem problematischen Schritt. In einigen Fällen ändert sich die Objektbeschreibung ein wenig von Lauf zu Lauf.

Es gibt mehrere Lösungen:

- Wenn der neue Wert stabil ist, öffnen Sie das Skript im Editor und ändern den Wert des Arguments BESCHREIBUNG manuell.
- Ändert sich die Beschreibung von Lauf zu Lauf, können Sie im DESCRIPTION-Argument einen regulären Ausdruck verwenden. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).
- Alternativ können Sie die problematische Objektbeschreibungseigenschaft wie Name mit der Eigenschaft **Ordnungszahl** ersetzen. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

ThreadingModel

Die Wiedergabe des COM-Skripts in VuGen schlägt fehl, wenn in der DLL-Registrierung die Zeichenfolge **ThreadingModel** unter dem Ordner **InprocServer32** der GUID fehlt.

Ajax (Click & Script) - Verschiedene Tipps

Die folgenden Tipps können sich bei Problemen mit Click-and-Script-Vuser-Skripts hilfreich erweisen.

Hinweis: Einige der unten aufgeführten Elemente gelten nur für bestimmte Click and Script-Protokolle.

Suchen Sie nach Warnungen

Suchen Sie im Ausgabebereich nach Warnungen oder Alarmen.

Überprüfen Sie die Antwort

Überprüfen Sie unter Verwendung von **web_reg_find**, ob die Antwort des vorherigen Schritts richtig ist. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Verwenden Sie die alternative Navigation

Verwenden Sie bei problematischen Schritten oder Schritten, die Java-Applets verwenden, die **alternative Navigation**, um den Web-Schritt durch einen Schritt auf HTTP-Ebene zu ersetzen. Beachten Sie, dass Schritte auf HTTP-Ebene möglicherweise manuelle Korrelationen erfordern. Um die alternative Navigation durchzuführen, wählen Sie einen Schritt im **Schrittenavigator** oder den Text in der Skriptansicht aus und wählen dann im Kontextmenü **Durch alternative Navigation ersetzen** aus.

Arbeiten mit dem Kerberos-Protokoll

Wenn Sie das Kerberos-Protokoll zur Authentifizierung verwenden, müssen Sie VuGen anpassen, um Autorisierungssitzungen ordnungsgemäß durchzuführen. Fortgeschrittene Benutzer können versuchen, diese Anpassung selbst vorzunehmen.

Damit das Kerberos-Protokoll ordnungsgemäß arbeitet, erstellen Sie die Datei krb5.ini und legen Sie sie in einem verfügbaren Ordner ab. Speichern Sie den vollständigen Pfadnamen der Datei **krb5.ini** in der Umgebungsvariablen KRB5_CONFIG.

Die Datei **krb5.ini** sollte detaillierte Informationen zu jeder Domäne (KDS- und AS-Adressen) und zu Vertrauensketten enthalten.

Wenn Sie weitere Informationen wünschen, setzen Sie sich bitte mit dem HP Software Support in Verbindung.

Click & Script - Fehlerbeseitigung und Einschränkungen

In diesem Abschnitt werden die Fehlerbeseitigung und die für Click and Script-Protokolle geltenden Einschränkungen beschrieben.

Hinweis: Einige der unten aufgeführten Elemente gelten nur für bestimmte Click and Script-Protokolle.

Aufzeichnungsprobleme und -beschränkungen

Kompatible Browser

- Für Web (Click and Script) wird nur Internet Explorer unterstützt. Um die Browseraktivität in Firefox aufzuzeichnen, verwenden Sie das Protokoll Web (HTTP/HTML).
- Nicht unterstützt für Internet Explorer 10.
- Bei Click & Script-Protokollen dauert das Öffnen des Dialogfelds "Aufzeichnungsoptionen" in VuGen unverhältnismäßig lange.

Sprachunterstützung

- Die Aufzeichnung einer Anwendung in einer bestimmten Sprache (z. B. Französisch, Japanisch) muss auf einem Computer durchgeführt werden, dessen Standardspracheinstellung (**Einstellungen > Systemsteuerung > Regions- und Sprachoptionen**) der Sprache der Aufzeichnung entspricht.
- Begrenzte Unterstützung von Rechts-nach-links-Sprachen (es treten z. B. Fehler bei der Verarbeitung von bidirektionalem oder umgedrehtem Text auf). Dies hängt von der standardmäßig verwendeten Umwandlungstabelle des Betriebssystems ab.
- Der Lastgenerator-Computer und der aufzeichnende Computer müssen auf dieselben Regionaleinstellungen eingestellt werden. Sie sollten nicht davon ausgehen, dass der Linux-Standardzeichensatz auf einem Computer mit englischem Betriebssystem dem Zeichensatz auf einem Windows-Computer mit englischem Betriebssystem entspricht. Überprüfen Sie dies vor der Aufzeichnung. Beispielsweise ist der Standardzeichensatz unter Linux UTF-8.

Die Anwendung verhält sich während der Aufzeichnung anders

Wenn die Anwendung sich während der Aufzeichnung anders verhält als ohne Aufzeichnung, sollten Sie feststellen, ob das Aufzeichnungsproblem nur bei Web auftritt. Die Auswirkungen des Problems sind z. B., dass eine Webseite nicht geladen wird, der Inhalt teilweise fehlt oder ein Popup-Fenster nicht geöffnet wird.
Problemumgehung: Erstellen Sie ein neues Web (HTTP/HTML)-Skript und wiederholen Sie die Aufzeichnung.

Tritt bei der Aufzeichnung in Web (HTTP/HTML) ein Fehler auf, wird empfohlen, die Aufzeichnung auf Socketebene zu deaktivieren (siehe "Click & Script - Aufzeichnungstipps" auf Seite 658).

Das Problem ist möglicherweise auf einen Ereignislistener zurückzuführen. Deaktivieren Sie die einzelnen Ereignislistener in den Aufzeichnungsoptionen unter **Webereigniskonfiguration** nach der Methode "Versuch und Irrtum" und zeichnen Sie Ihre Sitzung als Web-Benutzer (HTTP/HTML) erneut auf.

Bestimmte Click & Script-Schritte werden nicht ordnungsgemäß generiert

Wenn nach der Skriptaufzeichnung nicht alle Schritte ordnungsgemäß generiert werden, kann das Problem an der **Windows-Komponente > Verstärkte Sicherheitskonfiguration für den Internet Explorer** liegen.

Entfernen Sie die **Verstärkte Sicherheitskonfiguration für den Internet Explorer**, indem Sie **Systemsteuerung > Software > Windows-Komponenten hinzufügen/entfernen** auswählen und Ihr Skript neu aufzeichnen.

Deaktivieren eines Ereignislisteners

1. Klicken Sie auf **Aufzeichnen > Aufzeichnungsoptionen**, um das Dialogfeld **Aufzeichnungsoptionen** zu öffnen.

2. Wählen Sie den Knoten **GUI-Eigenschaften > Webereigniskonfiguration** aus.

3. Klicken Sie auf **Benutzerdefinierte Einstellungen** und erweitern Sie den Knoten **Web Objects**. Wählen Sie ein Objekt aus.

4. Wählen Sie **Deaktiviert** aus der Liste in der Spalte **Aufzeichnen** für das relevante Webobjekt aus. Wenn die Aufzeichnung immer noch nicht richtig durchgeführt wird, aktivieren Sie den Listener, den Sie zuvor deaktiviert haben, und deaktivieren Sie einen anderen. Wiederholen Sie diese Schritte, bis die Aufzeichnung erfolgreich durchgeführt wird.

Dynamische Menünavigation wurde nicht aufgezeichnet

Ein dynamisches Menü ist ein Menü, das sich dynamisch ändert, je nachdem, an welcher Stelle Sie es auswählen. Wenn die dynamische Menünavigation nicht erfasst wurde, zeichnen Sie erneut mit dem Ereigniskonfigurationsmodus **hoch** auf. Diese Einstellungen können in dem Knoten **Aufzeichnungsoptionen > GUI-Eigenschaften > Webereigniskonfiguration** gefunden werden.

Bestimmte Benutzeraktionen wurden nicht aufgezeichnet

Prüfen Sie, ob im Browser ein Java-Applet ausgeführt wird. Ist dies nicht der Fall, zeichnen Sie das Skript mit dem Protokoll Web (HTTP/HTML) auf.

Wiedergabeprobleme

GUI-Objekt wurde nicht gefunden

Tritt der Fehler zu Beginn der zweiten Iteration auf?

Tritt der Fehler zu Beginn des Abschnitts **Action** der zweiten Iteration auf, ist er möglicherweise auf eine Startseite zurückzuführen, die für die erste Iteration vorhanden war, für die zweite jedoch fehlte. Wenn die letzte Seite in einer Aktion nicht die Links und Schaltflächen enthält, die zu Beginn der Iteration zur Verfügung standen, schlägt die nächste Iteration fehl. Wenn die erste Seite beispielsweise den Textlink **Flugbuchung** enthält, müssen Sie sicherstellen, dass Sie zu der entsprechenden Seite navigieren, sodass derselbe Link am Ende des Geschäftsprozesses angezeigt wird.

Handelt es sich um einen Textlink, der Zeichen außerhalb des ASCII-Bereichs enthält?

Wenn das Problem bei Zeichen außerhalb des ASCII-Bereichs auftritt, sollten Sie VuGen anweisen, die Daten in einen geeigneten Zeichensatz zu konvertieren.

Aktivieren der Datenkonvertierung auf Windows-Computern

1. Wählen Sie **Wiedergeben > Laufzeiteinstellungen** und dann den Knoten **Internetprotokoll > Einstellungen** aus.
2. Klicken Sie auf **Optionen**, um das Dialogfeld **Erweiterte Optionen** zu öffnen.
3. Wechseln Sie unter **Web (Click and Script) > Allgemein** zu **Zeichensatzkonvertierungen durch HTTP** und setzen Sie die Option auf **Ja**.

Aktivieren der UTF-8-Konvertierung für Linux-Computer

1. Wählen Sie **Wiedergeben > Laufzeiteinstellungen** und dann den Knoten **Internetprotokoll > Einstellungen** aus.
2. Klicken Sie auf **Optionen**, um das Dialogfeld **Erweiterte Optionen** zu öffnen.
3. Wechseln Sie in den Optionen unter **Allgemein** zu **Aus/in UTF-8 konvertieren** und setzen Sie die Option auf **Ja**.

Wahlweise können Sie die Liste der Alternativen durchgehen, die eingeblendet wird, wenn ein Link nicht gefunden wird. Geben Sie den angezeigten Text unverändert ein, z. B. die hexadezimale Escapesequenz **\xA0** oder andere nicht standardmäßige Formate.

Können Sie dieselbe Abfolge von Aktionen in der Anwendung zweimal ausführen?

In einigen Fällen können Sie einen bestimmten Prozess, z. B. einen Benutzer aus der Datenbank löschen, nur einmal durchführen. Die Wiedergabe schlägt nach der ersten Iteration fehl, da die Aktion nicht mehr gültig ist. Stellen Sie sicher, dass Ihr Geschäftsprozess in der Anwendung mehrmals mit denselben Daten wiederholt werden kann, ohne dass eine neue Aufzeichnung erforderlich ist.

Waren die Bildeigenschaften **Id**, **Name** und **Alt** leer?

Doppelklicken Sie im **Schrittenavigator** auf den vorherigen Bildschritt, um seine Eigenschaften zu öffnen. Wenn die Eigenschaften **Id**, **Name** und **Alt** leer sind, stellen Sie weitere

Identifizierungsinformationen des Bilds bereit, z. B. seinen Dateinamen in der Eigenschaft **Src**.

Alternativ können Sie das Argument **Ordnungszahl** hinzufügen, um die Anzahl der Vorkommen des Bilds auf der jeweiligen Seite anzugeben. Das Argument **Ordnungszahl** identifiziert jedes Bild auf der Seite eindeutig; die anderen Identifikationsargumente hingegen sind nicht eindeutig. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Hat sich die Schrittbeschreibung geändert?

Prüfen Sie den Ausgabebereich auf eine Liste der Objekte in dem problematischen Schritt. In einigen Fällen ändert sich die Objektbeschreibung ein wenig von Lauf zu Lauf.

Es gibt mehrere Lösungen:

- Wenn der neue Wert stabil ist, öffnen Sie die Skriptansicht und ändern den Wert der DESCRIPTION-Argumente manuell.
- Ändert sich die Beschreibung von Lauf zu Lauf, können Sie in den DESCRIPTION-Argumenten einen regulären Ausdruck verwenden. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).
- Alternativ können Sie die problematische Objektbeschreibungseigenschaft, wie beispielsweise **Name**, mit der Eigenschaft **Ordnungszahl** ersetzen. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Wurde die Seite während der Aufzeichnung vollständig geladen?

Während der Aufzeichnung empfiehlt es sich zu warten, bis eine Seite vollständig geladen ist, bevor Sie den nächsten Schritt ausführen. Wenn Sie nicht gewartet haben, bis alle Seiten geladen wurden, zeichnen Sie das Skript erneut auf.

Wiedergabefehler

Kommt es bei der Wiedergabe an einem bestimmten Schritt zu einem Fehler, überprüfen Sie die Schrittbeschreibung. VuGen könnte ein einfaches Leerzeichen möglicherweise als doppeltes Leerzeichen interpretiert haben. Stellen Sie sicher, dass sich in der Zeichenfolge keine doppelten Leerzeichen befinden.

Wiedergabe-Snapshots

Wiedergabe-Snapshots können sich von der tatsächlichen Webseite unterscheiden.

Sonstige Probleme

Fehler "Speicherplatz nicht ausreichend" in JavaScript

Vergrößern Sie den JavaScript-Speicher in den Laufzeiteinstellungen.

Vergrößern des JavaScript-Speichers

1. Wählen Sie **Wiedergeben > Laufzeiteinstellungen** und dann den Knoten **Internetprotokoll > Einstellungen** aus.
2. Klicken Sie auf **Optionen**, um das Dialogfeld **Erweiterte Optionen** zu öffnen.

3. Wechseln Sie zu den Optionen **Speicherverwaltung Größe des JavaScript-Laufzeitspeichers (KB)** und **Speicherverwaltung Größe des JavaScript-Stackspeichers (KB)**.

4. Setzen Sie die Speichergrößen auf 512 KB oder höher.

VuGen zeigt JavaScript-Fehler an

Wenn VuGen im Ausgabebereich JavaScript-Fehler anzeigt, aktivieren Sie die Anzeige von IE-Skriptfehlern (Internet Explorer), um zu gewährleisten, dass das Java-Skript selbst keine Fehler enthält.

Anzeigen von Skriptfehlern

1. Öffnen Sie Internet Explorer.

2. Wählen Sie **Extras > Internetoptionen** aus und klicken Sie auf die Registerkarte **Erweitert**.

3. Aktivieren Sie im Abschnitt **Browsen** das Kontrollkästchen **Skriptfehler anzeigen**.

4. Führen Sie die Anwendung erneut in IE aus. Wenn IE Skriptfehler anzeigt, liegt ein Problem mit der JavaScript-Anwendung vor. Wenn das Problem in der Anwendung nicht behoben werden kann, können Sie die entsprechenden Wiedergabefehler ruhig ignorieren.

Probleme nach der Parametrierung

Wenn nach der Parametrierung von Werten Probleme auftreten, stellen Sie sicher, dass die Werte für Ihre Anwendung gültig sind. Führen Sie einen Geschäftsprozess mit dem Wert des Parameters durch und stellen Sie sicher, dass die Anwendung ihn akzeptiert.

Probleme mit Anwendungen, die Styling-Aktionen verwenden

Wenn die Skripts auf Clientseite der Anwendung zahlreiche Styling-Aktivitäten verwenden, sollten Sie das Skript erneut aufzeichnen, nachdem Sie die Option **Renderingbezogene Eigenschaftswerte aufzeichnen** aktiviert haben. Auf diese Weise wird die Aufzeichnung zusätzlicher DOM-Objekte ermöglicht.

Aktivieren Sie die Option "Renderingbezogene Eigenschaftswerte aufzeichnen"

1. Wählen Sie **Aufzeichnen > Aufzeichnungsoptionen** und dann den Knoten **GUI-Eigenschaften > Erweitert** aus.

2. Aktivieren Sie das Kontrollkästchen **Renderingbezogene Eigenschaftswerte aufzeichnen**.

Zeichnen Sie das Vuser-Skript erneut auf.

- ActiveX-Objekte und Java-Applets werden nur auf Windows-Plattformen unterstützt.
- Nicht unterstützt für Macromedia Flash oder VB Script.

- Für Click & Script-Protokolle werden keine Popup-Fenster unterstützt.

TruClient-Protokoll

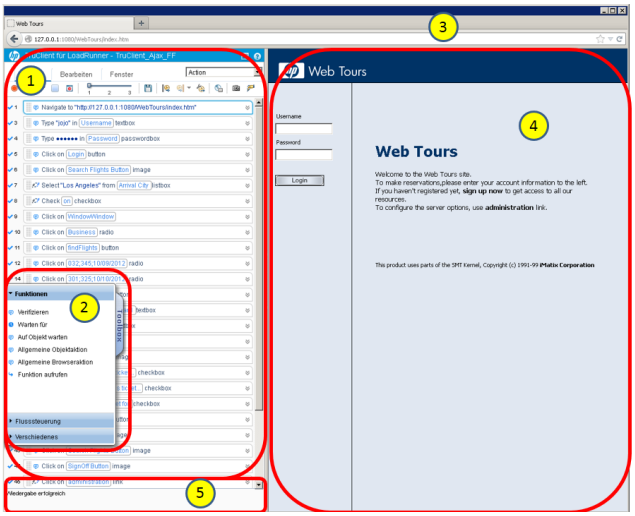
Was ist TrueClient?

TruClient ist ein Tool für die Aufzeichnung komplexer web-basierter Anwendungen. Das TruClient-Modul zeichnet Ihre Aktionen auf, während Sie den gewünschten Geschäftsprozess ausführen. Es wird ein Skript in Echtzeit erstellt, in dem Sie die Schritte während der Ausführung in einer Seitenleiste beobachten können. Derzeit können Sie Tests wahlweise mit den Browsern Mozilla Firefox oder Internet Explorer aufzeichnen.

Hinweis: In diesem Abschnitt werden die Funktionen dokumentiert, die sowohl in Firefox als auch in Internet Explorer verfügbar sind.

Funktionen, die nur von einem bestimmten Browser unterstützt werden, werden durch das Browsersymbol gekennzeichnet.

Die TruClient-Benutzeroberfläche



Die TruClient-Benutzeroberfläche setzt sich aus folgenden Abschnitten zusammen:

1. **TruClient-Seitenleiste.** Der Mittelpunkt der Oberfläche enthält alle Tools, die Sie zum Entwickeln Ihrer TruClient-Skripts benötigen.

2. **TruClient-Toolbox.** Die Toolbox enthält alle Schritttypen, die Sie zu einem TruClient-Skript hinzufügen können. Sie öffnen und schließen die Toolbox, indem Sie auf die Registerkarte klicken; zum Verschieben ziehen Sie sie nach oben oder unten.

3. **Navigationsleiste des Browsers.** Geben Sie den URL der Anwendung ein, für die Sie ein Skript entwickeln.

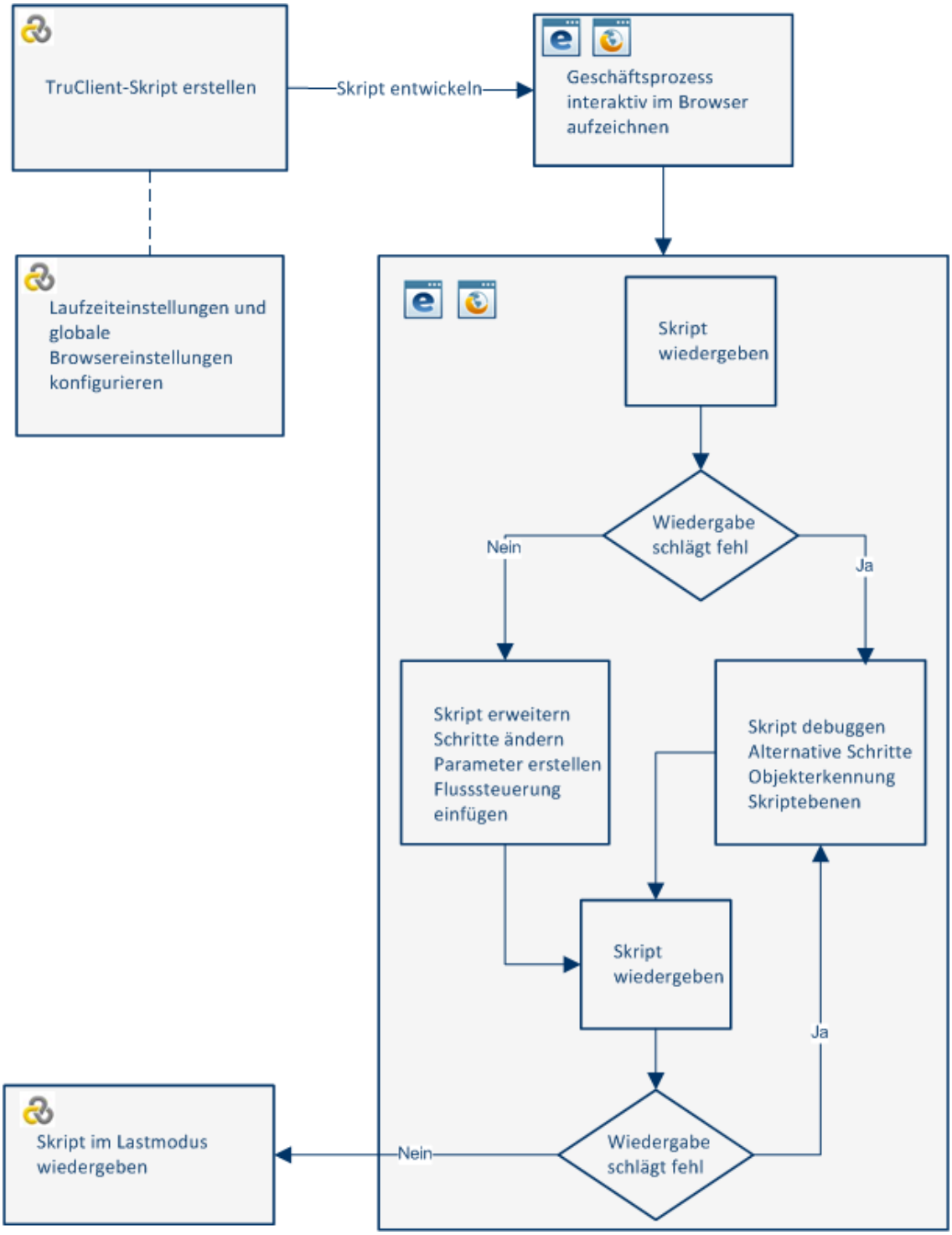
4. **Browserfenster der Anwendung.** Hier können Sie Ihr Skript interaktiv entwickeln und wiedergeben.

5. **Statusbereich auf der TruClient-Seitenleiste.** Zeigt Statusdetails über die aktive Aktion in der **TruClient-Seitenleiste** an.

Auf der VuGen-Startseite können Sie sich ein Demovideo zu LR ansehen. , um das Video wiederzugeben.

Der TruClient-Workflow für LoadRunner

Der Workflow beim TruClient-Protokoll unterscheidet sich vom Workflow bei anderen VuGen-Protokollen. Das folgende Flussdiagramm zeigt den grundlegenden TruClient-Workflow:



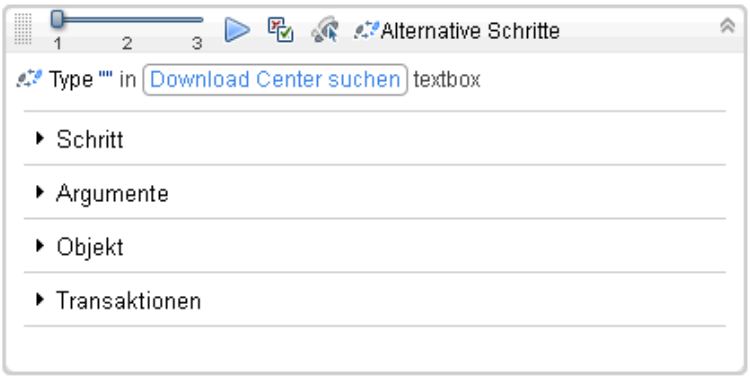
Im Folgenden sind einige der wesentlichen Unterschiede zwischen dem TruClient-Protokoll und anderen VuGen-Protokollen aufgelistet:

- Das Skript wird in der Skriptansicht von VuGen angezeigt, ist aber schreibgeschützt. Das Skript wird in der **TruClient-Seitenleiste** im Browser (FF oder IE) erstellt und geändert.
- TruClient-Skripts sind asynchron. Schritte müssen nicht warten, bis vorhergehende Schritte beendet sind. Jeder Schritt definiert ein **Endereignis**, das den Punkt bezeichnet, an dem die Ausführung der nachfolgenden Schritte beginnen kann.
- TruClient-Skripts werden auf Benutzerebene aufgezeichnet; insofern bestehen keine Korrelationen. Die Objekterkennung stellt die eigentliche Herausforderung dar.
- TruClient-Skripts werden auf Benutzerebene wiedergegeben, deshalb erfordert jeder Vuser eine Browserinstanz mit geladenem DOM und JavaScript. Die Ressourcenauslastung ist größer als beim (transportbasierten) Webprotokoll und abhängig von den DOM- und JavaScript-Funktionen der Anwendung. Weitere Informationen finden Sie unter "Berechnen der Anzahl an Lastgeneratoren, die für TruClient-Skripts erforderlich sind" auf Seite 553.
- Alle aufgezeichneten Ereignisse werden im Skript gespeichert. Als irrelevant erachtete Ereignisse werden verschiedenen Skriptebenen zugewiesen und nicht wiedergegeben, sofern die Ebene nicht vom Benutzer manuell geändert wird.
- TruClient-Transaktionen werden durch Schrittereignisse und nicht durch Schritte wie in anderen Protokollen definiert. Das Endereignis eines Schritts kann beispielsweise die Fortführung der Skriptausführung zulassen, während eine Transaktion, die bei diesem Schritt endet, weiter ausgeführt wird, bis das Schrittereignis, das die Transaktion definiert, erreicht ist.
- Die Ausführungslogik in TruClient-Skripts wird anders gesteuert. Es gibt nur eine Aktion.
- TruClient-Schrittargumente akzeptieren JavaScript-Code als Werte.

Die meisten Aufgaben, die beim Aufzeichnen, Wiedergeben und Ändern von Skripts anfallen, werden in der TruClient-Seitenleiste für Internet Explorer oder Mozilla Firefox durchgeführt.

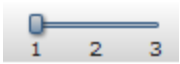
Struktur von TruClient-Schritten

TruClient-Schritte bestehen aus mehreren Abschnitten. Die Abschnitte und Elemente in jedem Abschnitt variieren abhängig vom Typ des Schritts.



Nachfolgend werden die Elemente der Oberfläche beschrieben:

Schrittstruktur

Element der Oberfläche	Beschreibung
	Schritt ziehen. Mit dieser Option können Sie die Reihenfolge Ihres Skripts ändern, indem Sie den Schritt an eine andere Position ziehen.
	Schritt erweitern. Zeigt die einzelnen Komponenten der Schritte an, die Schritt, Argument und Objekt enthalten.
	**Auswahl für Skriptebenen.** Ermöglicht Ihnen das Anzeigen und Ändern der Skriptebene des Schritts. Weitere Informationen finden Sie unter "TruClient-Skriptebenen" auf Seite 555.
	Wiedergabe. Gibt nur diesen Schritt wieder.
	Schritt deaktivieren/aktivieren. Deaktivierte Schritte werden nicht wiedergegeben. Diese Funktion ermöglicht Ihnen das vorübergehende Entfernen von Schritten aus dem Skript, ohne sie zu löschen.
	Optionaler Schritt. Wenn Sie einen Schritt als optional kennzeichnen und der Schritt sein Objekt nicht finden kann, wird das Skript fortgesetzt, ohne dass ein Fehler zurückgegeben wird.
	Alternative Schritte. Dieses Symbol weist auf einen Schritt hin, der auf andere Weise neu definiert werden kann. Um den Schritt neu zu definieren, klicken Sie auf das Symbol, wählen die gewünschte Schrittdefinition aus und klicken auf **Zurück**. Weitere Informationen finden Sie unter "Alternative Schritte für TruClient " auf Seite 555.

SchrittstrukturForts.

Schritt	• **Aktion.** Die Aktion, die den Schritt definiert. Die Liste der relevanten Aktionen wird durch die Objektrollen bestimmt. • **Objektzeitüberschreitung.** Wenn das Objekt nicht angezeigt wird, bevor diese Zeit in Sekunden abgelaufen ist, gibt der Schritt einen Fehler zurück. • **Schrittzeitüberschreitung.** Wenn das Endereignis nicht erreicht wird, bevor diese Zeit in Sekunden abgelaufen ist, gibt der Schritt einen Fehler zurück. Das Skriptverhalten bei einem solchen Fehler kann im Dialogfeld für die Laufzeiteinstellungen konfiguriert werden. • **Mindestzeit.** Die minimale Zeit in Sekunden, die zur Ausführung des Schritts erforderlich ist. Der Wert in diesem Feld kann entweder 0, "Wie aufgezeichnet" oder eine andere manuell festgelegte Zahl sein. Die Schrittausführung endet sofort nach dem Endereignis des Schrittes, wenn die Mindestzeit 0 ist. Ein Mindestzeitwert größer 0 zwingt TruClient, nach dem Endereignis des Schritts für eine zusätzliche Zeit zu warten (wenn diese noch nicht verstrichen ist), bevor mit dem nächsten Schritt fortgefahren wird. TruClient erfasst und speichert die Zeit, die zwischen aufgezeichneten Aktionen verstrichen ist. Des Weiteren haben Sie die Möglichkeit, die Mindestzeit auf "Wie aufgezeichnet" einzustellen. • **Endereignis.** TruClient definiert, wann ein Endereignis während der Wiedergabe auftritt. Ein **Endereignis** kann Folgendes sein: ▪ **Automatic: Not Set Yet.** Das automatische Endeereignis wurde noch nicht festgelegt. ▪ **Aktion abgeschlossen.** Schritt endet, wenn die Aktion abgeschlossen ist. Ein Beispiel für eine Aktion ist das Klicken auf eine Schaltfläche. ▪ **DOM content loaded.** Schritt endet, wenn das DOM (Document Object Model) bereit ist. Dies bedeutet, dass die API für die Interaktion mit Inhalt, Stil und Struktur einer Seite bereit ist, Anforderungen durch den clientseitigen Code Ihrer Anwendung zu empfangen. ▪ **Step synchronous network completed.** Schritt endet, wenn alle HTTP-Anforderungen abgeschlossen sind, außer Anforderungen in Bezug auf offene Verbindungen, die nicht relevant für den Schritt sind. Normalerweise werden diese Anforderungen mithilfe von XMLHttpRequest ausgelöst. ▪ **Laden des Dokuments.** Schritt endet, wenn das Laden eines

SchrittstrukturForts.

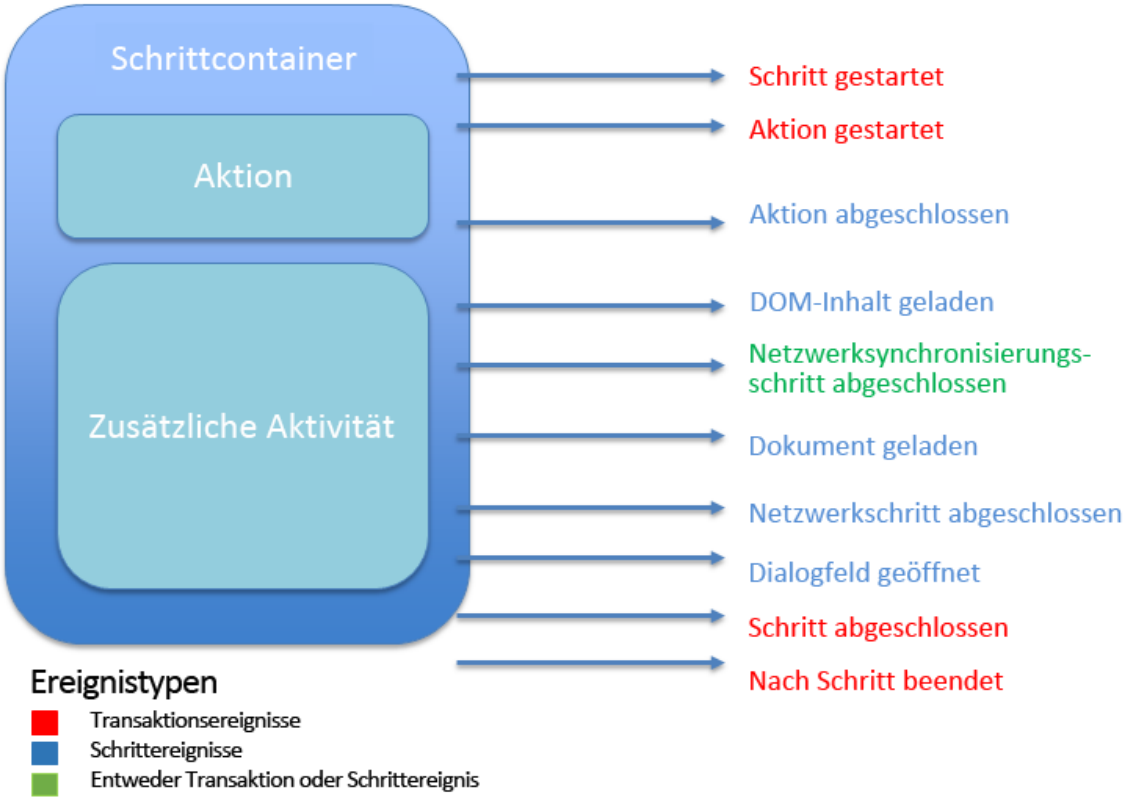
	Dokuments abgeschlossen ist. Dies bedeutet, dass alle Skripts und Stylesheets geladen und ausgeführt wurden und alle Bilder heruntergeladen und angezeigt wurden. ▪ **Step network completed.** Schritt endet, wenn alle HTTP-Anforderungen einschließlich von XMLHttpRequest initiierter Anforderungen abgeschlossen sind. ▪ **Dialog opened.** Schritt endet, wenn ein Dialogfeld geöffnet wird. Weitere Informationen finden Sie unter "Synchronisieren von TruClient-Skriptschritten" auf Seite 546.
Argumente	Enthält Schrittargumente. Diese Argumente sind für verschiedene Schrittaktionen und Rollen unterschiedlich. Eine Liste der Schrittargumente finden Sie unter "Argumente von TruClient-Schritten" auf Seite 630.
Objekt	• **Rollen.** Die Funktionen, die TruClient für ein Objekt versteht. Diese Informationen sind schreibgeschützt und werden abhängig von der Objektverwendung während der Aufzeichnung dynamisch aktualisiert. Die Liste der verfügbaren Schrittaktionen wird durch diese Rollen definiert. • **Name.** Ein logischer Name für das Objekt. Dieser wirkt sich nicht auf die Wiedergabe aus und kann geändert werden, um die Lesbarkeit zu verbessern. • **ID-Methode.** Die Methode zum Identifizieren des Objekts. ▪ **Automatisch.** Die Standardmethode zur Objektidentifizierung von TruClient. Wenn diese Methode das Objekt während der Wiedergabe nicht findet, klicken Sie auf die Schaltfläche **Objekterkennung verbessern**, wählen das richtige Objekt aus der Anwendung erneut aus und geben das Skript erneut wieder. ▪ **XPath.** Identifiziert das Objekt basierend auf dem zugehörigen XPath-Ausdruck, der das Objekt in der DOM-Struktur definiert. Wenn Sie diese Option auswählen, ist das nächste Bearbeitungsfeld mit **XPath** benannt und ermöglicht Ihnen das Auswählen eines XPath, um das Objekt zu definieren. Einzelheiten dazu finden Sie im Folgenden. ▪ **JavaScript.** JavaScript-Code, der ein Objekt zurückgibt. Wenn Sie diese Option auswählen, ist das nächste Bearbeitungsfeld mit JavaScript benannt und ermöglicht Ihnen das Definieren eines JavaScript, um das Objekt zu definieren. Einzelheiten dazu finden Sie im Folgenden. • **Zugehörige Objekte.** Tool, mit dem TruClient ein Zielobjekt in Bezug auf ein Ankerobjekt identifizieren kann. Weitere Informationen finden Sie unter "Lösen von Problemen bei der Objekterkennung" auf Seite 560.

SchrittstrukturForts.

Transaktionen	Ermöglicht Ihnen das Erstellen, Ändern und Anzeigen von Transaktionen. Weitere Informationen finden Sie unter "Erweitern von TruClient-Skripts" auf Seite 576.
XPath/JavaScript	
XPath	TruClient erzeugt je nach Objekt mehrere mögliche XPaths. Klicken Sie auf den Dropdownpfeil neben dem XPath-Bearbeitungsfeld, um einen vorgeschlagenen XPath für das Objekt auszuwählen. Sie können den vorgeschlagenen XPath manuell ändern. Um die Einstellung auf einen der ursprünglich von TruClient generierten Ausdrücke zurückzusetzen, wählen Sie in der Dropdownliste erneut eine der verfügbaren Optionen aus. Sie können auch auf die Schaltfläche **Ausdruck neu generieren** klicken und ein Objekt auswählen. TruClient basierend auf dem ausgewählten Objekt einen neuen Satz von Vorschlägen.
JavaScript	Wenn TruClient für das Objekt einen vorgeschlagenen XPath erzeugen kann, wird dieser XPath im JavaScript-Feld in einer **evalXPath**-Funktion als Argument eingegeben. Die Funktion **evalXPath** gibt ein Array der Objekte zurück, die durch den XPath in dem Argument definiert werden. Sie können den vorgeschlagenen XPath im Argument ändern, um eine andere Objektliste zurückzugeben. Sie können aber auch ein anderes JavaScript eingeben. Beispiel: document.getElementById("Suchschaltfläche") gibt ein Element mit dem DOM-ID-Attribut "Suchschaltfläche" zurück. TruClient enthält auch eine **Zufallsfunktion**, die ein zufälliges Element aus dem Array in Form des zugehörigen Arguments zurückgibt. Beispiel: `random (document.getElementsByTagName("a"))` **Hinweis:** Die evalXPath-Funktion und die Zufallsfunktion stehen nur als Methoden zur Objektidentifizierung zur Verfügung. Sie werden in einem Schritt zum Auswerten von JavaScript-Code nicht erkannt.

Grundlegendes zu Schrittereignissen

Ein TruClient-Schritt umfasst eine Aktion und für Schritte im Zusammenhang mit der Anwendung zusätzliche Aktivitäten. Das folgende Diagramm veranschaulicht die Abfolge von Ereignissen, die infolge einer Schrittausführung ausgelöst werden.



Was ist ein Endereignis?

TruClient-Skripts sind asynchron. D. h. Schritte müssen nicht warten, bis vorhergehende Schritte beendet sind. Jeder Schritt weist ein Ereignisablauf mit einem **Endereignis** auf, das den Punkt bezeichnet, an dem die Ausführung der nachfolgenden Schritte beginnen kann. Während der Skriptwiedergabe ermittelt TruClient das Endereignis für jeden der Schritte. Weitere Informationen finden Sie unter "Synchronisieren von TruClient-Skriptschritten".

Ein Schritt ist eine Art Container, in dem eine Aktion und zusätzliche Aktivitäten im Zusammenhang mit der Anwendung enthalten sind. Jede Schrittausführung ist unterschiedlich und durchläuft möglicherweise einen anderen Ereignisablauf. Das folgende Beispiel veranschaulicht, wie unterschiedlich Schrittausführungen sein können:

- **Beispiel 1: Abrufen von Parametern mit dem Schritt "EvaluateJS"**

  Der Schritt umfasst den Schritt "EvaluateJS" mit dem Code `TC.getParam("Name");`. Er umfasst keine Aktivitäten im Zusammenhang mit der Anwendung. Daher ist das Endereignis automatisch auf "Aktion abgeschlossen" festgelegt. Die anderen Ereignisse in der Kette sind nicht relevant.

- **Beispiel 2: Klicken auf ein einfaches Suchfeld (ohne automatische Vorschläge)**

  Dieser Schritt interagiert mit der Anwendung, aber er löst keine Netzwerk- oder DOM-Aktivitäten aus. Daher ist das Endereignis automatisch auf **Aktion abgeschlossen** festgelegt. Die anderen Ereignisse in der Kette sind nicht relevant.

- **Beispiel 3: Klicken auf eine Suchen-Schaltfläche**

  Dieser Schritt interagiert mit der Anwendung und löst Netzwerk- und DOM-Aktivitäten aus. Daher ist das Endereignis automatisch auf eines der Ereignisse festgelegt, die auf **Aktion abgeschlossen** folgen. Welches Ereignis ausgewählt wird, hängt vom Verhalten der jeweiligen Anwendung ab.

Wann Schritte und Ereignisse manuell zu ändern sind

- **Fall 1:**

  Der nächste Schritt kann noch nicht ausgeführt werden, weil das vorangegangene Endereignis nicht korrekt ist. Weitere Informationen finden Sie unter "Synchronisieren von TruClient-Skriptschritten".

- **Fall 2:**

  Sie haben eine Transaktion erstellt, die mehrere Schritte umfasst. Sie möchten einen bestimmten Aspekt der Transaktion genauer prüfen. Weitere Informationen finden Sie unter "Dialogfeld 'Transaktionseditor' (TruClient)" auf Seite 594.

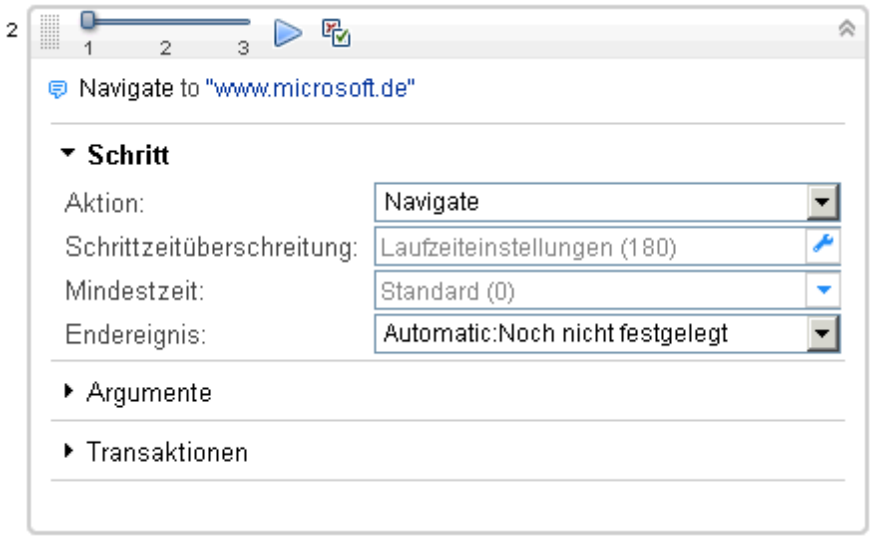
Synchronisieren von TruClient-Skriptschritten

In dieser Aufgabe wird der Prozess der Synchronisierung von Schritten in TruClient-Skripts beschrieben.

Hinweis: Diese Aufgabe ist Bestandteil einer übergeordneten Aufgabe. Weitere Informationen finden Sie unter "Entwickeln von TruClient-Skripts" auf Seite 549.

1. ## Interaktives Aufzeichnen des Geschäftsprozesses

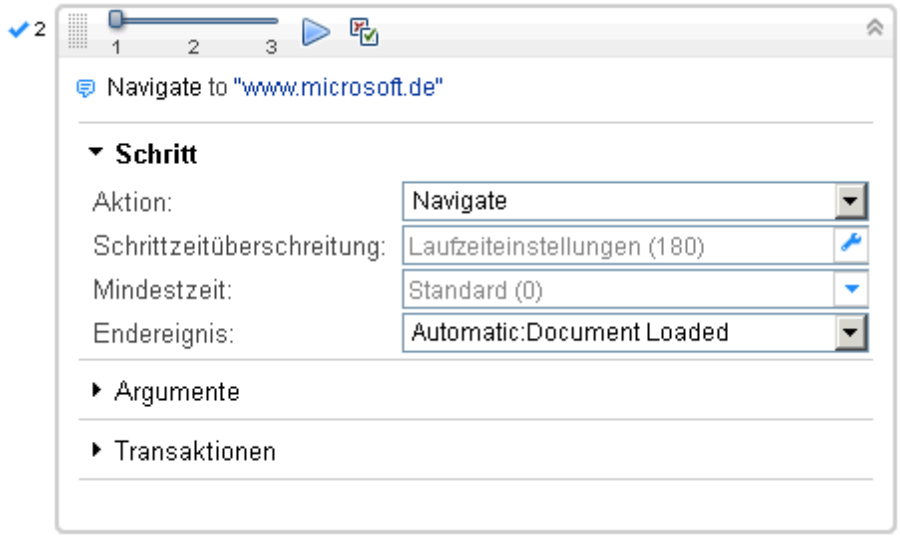
   TruClient-Skripts sind asynchron, d. h. Schritte müssen nicht auf den Abschluss vorhergehender Schritte warten. Jeder Schritt weist ein **Endereignis** auf, das den Punkt bezeichnet, an dem die Ausführung der nachfolgenden Schritte beginnen kann. Nach dem interaktiven Aufzeichnen, wird jedes **Endereignis** auf **Automatisch: Noch nicht festgelegt** gesetzt.



Weitere Informationen zum Entwickeln eines Skripts in TruClient finden Sie in Schritt 5 unter "Entwickeln von TruClient-Skripts" auf Seite 549

2. **Identifizieren des Endereignisses**

Bevor Sie ein TruClient-Skript erweitern, geben Sie das Skript zum Synchronisieren der Schritte wieder. Während der ersten Skriptwiedergabe versucht TruClient, das **Endereignis** für jeden Schritt automatisch zu erkennen.



Ein **Endereignis** kann Folgendes sein:

- **Automatic: Not Set Yet.** Das automatische Endeereignis wurde noch nicht festgelegt.
- **Aktion abgeschlossen.** Schritt endet, wenn die Aktion abgeschlossen ist. Ein Beispiel für eine Aktion ist das Klicken auf eine Schaltfläche.

- **DOM content loaded.** Schritt endet, wenn das DOM (Document Object Model) bereit ist. Dies bedeutet, dass die API für die Interaktion mit Inhalt, Stil und Struktur einer Seite bereit ist, Anforderungen durch den clientseitigen Code Ihrer Anwendung zu empfangen.
- **Step synchronous network completed.** Schritt endet, wenn alle HTTP-Anforderungen abgeschlossen sind, außer Anforderungen in Bezug auf offene Verbindungen, die nicht relevant für den Schritt sind. Normalerweise werden diese Anforderungen mithilfe von XMLHttpRequest ausgelöst.
- **Laden des Dokuments.** Schritt endet, wenn das Laden eines Dokuments abgeschlossen ist. Dies bedeutet, dass alle Skripts und Stylesheets geladen und ausgeführt wurden und alle Bilder heruntergeladen und angezeigt wurden.
- **Step network completed.** Schritt endet, wenn alle HTTP-Anforderungen einschließlich von XMLHttpRequest initiierter Anforderungen abgeschlossen sind.
- **Dialog opened.** Schritt endet, wenn ein Dialogfeld geöffnet wird.

Wenn während der ersten Wiedergabe ein Schritt den Assistenten für die Objekterkennung initiiert, um die Objekterkennung durchzuführen, wird das **Endereignis** des vorhergehenden Schrittes höchstwahrscheinlich nicht richtig erkannt, und TruClient setzt das Ereignis automatisch auf **Automatisch: Noch nicht festgelegt** zurück.

3. ## Bestätigen des Endereignisses

Während der zweiten Skriptwiedergabe bestätigt TruClient das automatische Endereignis und weist das Endereignis jeden Schrittes zu, das auf **Automatisch: Noch nicht festgelegt** zurückgesetzt worden war.

Wenn TruClient ein automatisches **Endereignis** nicht während der Wiedergabe zuweisen kann (in der Regel aufgrund einer Netzwerkzeitüberschreitung), erhalten Sie eine Warnmeldung, dass das **Endereignis** auf **Automatisch: Noch nicht festgelegt** zurückgesetzt wurde. Geben Sie das Skript wieder, um das **Endereignis** automatisch zuzuweisen, oder weisen Sie das **Endereignis** manuell zu.

Hinweis: Möglicherweise müssen Sie das Skript mehrmals wiedergeben, bis alle Schritte exakt synchronisiert sind.

Entwickeln von TruClient-Skripts

Entwickeln von TruClient-Skripts

In dieser Aufgabe werden die grundlegenden Schritte beim interaktiven Entwickeln eines TruClient-Skripts beschrieben.

1. **Erstellen Sie über die VuGen-Symbolleiste ein TruClient-Skript.**

   Weitere Informationen zum Erstellen eines VuGen-Skripts finden Sie unter "Erstellen oder Öffnen eines Vuser-Skripts" auf Seite 148.

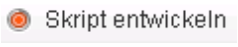
2. **Konfigurieren der allgemeinen Browsereinstellungen**

   In den Browserkonfigurationseinstellungen können Sie Einstellungen definieren, die für alle TruClient-Skripts gelten. Die Einstellungen werden in neue Skripts bei deren Erstellung importiert. Um diese Einstellungen zu bearbeiten, klicken Sie auf die Schaltfläche **TruClient - Allgemeine Einstellungen** in der Aufzeichnungssymbolleiste im VuGen-Hauptfenster und wählen dann die Registerkarte **Browsereinstellungen** aus. Weitere Informationen finden Sie unter "Dialogfeld "TruClient - Allgemeine Einstellungen"" auf Seite 609.

3. **Konfigurieren der Laufzeiteinstellungen.**

   Konfigurieren Sie in VuGen die Laufzeiteinstellungen, bevor Sie einen Lasttest aufzeichnen und durchführen. Um das Dialogfeld **Laufzeiteinstellungen** zu öffnen, drücken Sie **F4**, oder wählen Sie den Knoten **Laufzeiteinstellungen** aus dem **Lösungs-Explorer** von VuGen. Weitere Informationen finden Sie unter "Laufzeiteinstellungen" auf Seite 330.

4. **Starten der Skriptentwicklung**

   Klicken Sie auf 

   , um eine interaktive Aufzeichnungssitzung in der **TruClient-Seitenleiste** entweder für den Mozilla Firefox-Browser oder für den TruClient-Browser für IE zu öffnen.

5. **Interaktives Aufzeichnen**

   Navigieren Sie zu der Website, auf der Sie die Aufzeichnung starten möchten, und klicken Sie auf die Schaltfläche **Aufzeichnen**. Ihre sämtlichen Aktionen werden aufgezeichnet und in der **TruClient-Seitenleiste** auf der linken Seite angezeigt, während Sie Ihren Geschäftsprozess ausführen. Sie können die Aufzeichnung mit der Schaltfläche **Anhalten** anhalten. Sie können die Aufzeichnung von jeder beliebigen Stelle im Skript fortsetzen.

   Um die Aufzeichnung in einem anderen Abschnitt des Skripts fortzusetzen, klicken Sie mit der rechten Maustaste auf einen Schritt und wählen **Aufzeichnen > Nach Schritt** oder **Aufzeichnen > Vor Schritt**, um mit der Aufzeichnung der neuen Schritte an der gewünschten Stelle im Skript zu beginnen. Wenn Sie die Aufzeichnung in einen Gruppenschritt vornehmen,

wählen Sie **Aufzeichnen > Aufzeichnen in**. Weitere Informationen zu Gruppenschritten finden Sie unter "TruClient-Startregisterkarte" auf Seite 599.

6. ## Wiedergeben des Skripts

   Es wird dringend empfohlen, das Skript mindestens zweimal wiederzugeben und alle Fehler zu korrigieren, die während des Prozesses auftreten. Nachdem Sie das Skript zweimal hintereinander erfolgreich wiedergegeben haben, können Sie den nächsten Schritt in Angriff nehmen. Wenn weiterhin Fehler auftreten, schlagen Sie unter "Debuggen von TruClient-Skripts" auf Seite 558 nach.

   Bei der interaktiven Wiedergabe zeigt TruClient den Fortschritt bei jedem Schritt entsprechend dem echten Ausführungsfortschritt durch Auffüllen der Schrittleiste an. Der Fortschritt bei der Schrittausführung manifestiert sich beispielsweise durch folgende Ereignisse: Testobjekt gefunden, Aktion ausgeführt oder Schritt beim Endereignis synchronisiert. Weitere Informationen zum Synchronisieren von Skriptschritten finden Sie unter "Synchronisieren von TruClient-Skriptschritten" auf Seite 546.

   Darüber hinaus können Sie Details zu den Skriptaktionen im **Statusbereich auf der TruClient-Seitenleiste** anzeigen.

7. ## Erweitern des Skripts

   Sie können Ihr Skript in vielfältiger Weise erweitern, indem Sie z. B. Parameter, Transaktionen, Schleifen und Schritte zur Verifizierung einfügen. Weitere Informationen über die Aufgabe finden Sie unter "Erweitern von TruClient-Skripts" auf Seite 576.

8. ## Beenden der Entwicklung

   Klicken Sie auf die Schaltfläche **Speichern** , um das Skript zu speichern. Schließen Sie das Browserfenster. Stellen Sie das VuGen-Fenster wieder her.

9. ## Wiedergabe des Skripts im Lastmodus

   Bevor Sie das Skript in einem Controller-Szenario ausführen, führen Sie das Skript im Lastmodus von VuGen aus. Klicken Sie im VuGen-Hauptfenster auf die Schaltfläche **Wiedergabe** , um das Skript im Lastmodus wiederzugeben. Der Fortschritt kann im Wiedergabeprotokoll überwacht werden. Der Browser wird nicht geöffnet und es werden keine Snapshots angezeigt.

   Weitere Informationen zum Lastgenerator finden Sie unter "Berechnen der Anzahl an Lastgeneratoren, die für TruClient-Skripts erforderlich sind" auf Seite 553.

10. ## Allgemeine Tipps zur erfolgreichen interaktiven Wiedergabe

    Ändern Sie die Größe des Browsers zwischen Aufzeichnung und Wiedergabe bzw. während der Wiedergabe nicht. Dies kann bewirken, dass Objekte verschoben werden und TruClient diese nicht mehr findet.

Wechseln Sie während der interaktiven Wiedergabe nicht zwischen Anwendungen. Lassen Sie den Browser im Vordergrund.

Hinweis: Besonders wichtig ist dies, wenn die Funktion **Zugehörige Objekte** verwendet wird, da sich eine Größenänderung auf die relative Position der Objekte auswirken kann.

Hinweis: Anpassungen (wie Lesezeichen), die Sie in dieser Instanz von Firefox durchgeführt haben, werden nicht global gespeichert. Wie bereits erwähnt öffnet VuGen jedes Skript in einem eindeutigen Firefox-Profil. Wenn Sie Firefox noch für andere Dinge als die Erstellung dieses Skripts verwenden möchten (um z. B. im Internet zu surfen), sollten Sie ein weiteres Firefox-Fenster öffnen.

Aufzeichnen von TruClient-Skripts

In dieser Aufgabe wird beschrieben, wie Sie ein TruClient-Skript im FireFox-Browser mit dem TruClient-Seitenleiste-Add-in aufzeichnen.

1. ## Erstellen eines TruClient-Skripts

   Wählen Sie in der TruClient-Seitenleiste **Datei > Neu** aus.

2. ## Konfigurieren der allgemeinen Browsereinstellungen

   In den Browserkonfigurationseinstellungen können Sie Einstellungen definieren, die für alle TruClient-Skripts gelten. Die Einstellungen werden in neue Skripts bei deren Erstellung importiert. Um diese Einstellungen zu bearbeiten, klicken Sie auf der TruClient-Seitenleiste auf die Schaltfläche **TruClient - Allgemeine Einstellungen** , und wählen Sie die Registerkarte **Browsereinstellungen** aus. Weitere Informationen finden Sie unter "Dialogfeld "TruClient - Allgemeine Einstellungen"" auf Seite 609.

3. ## Interaktives Aufzeichnen

   Navigieren Sie zu der Website, auf der Sie die Aufzeichnung starten möchten, und klicken Sie auf die Schaltfläche **Aufzeichnen**. Ihre sämtlichen Aktionen werden aufgezeichnet und in der **TruClient-Seitenleiste** auf der linken Seite angezeigt, während Sie Ihren Geschäftsprozess ausführen. Sie können die Aufzeichnung mit der Schaltfläche **Anhalten** anhalten. Sie können die Aufzeichnung von jeder beliebigen Stelle im Skript fortsetzen.

   Um die Aufzeichnung in einem anderen Abschnitt des Skripts fortzusetzen, klicken Sie mit der rechten Maustaste auf einen Schritt und wählen **Aufzeichnen > Nach Schritt** oder **Aufzeichnen > Vor Schritt**, um mit der Aufzeichnung der neuen Schritte an der gewünschten Stelle im Skript zu beginnen. Wenn Sie die Aufzeichnung in einen Gruppenschritt vornehmen, wählen Sie **Aufzeichnen > Aufzeichnen in**. Weitere Informationen zu Gruppenschritten finden Sie unter "TruClient-Startregisterkarte" auf Seite 599.

4. ### Wiedergeben des Skripts

   Es wird dringend empfohlen, das Skript mindestens zweimal wiederzugeben und alle Fehler zu korrigieren, die während des Prozesses auftreten. Nachdem Sie das Skript zweimal hintereinander erfolgreich wiedergegeben haben, können Sie den nächsten Schritt in Angriff nehmen. Wenn weiterhin Fehler auftreten, schlagen Sie unter "Debuggen von TruClient-Skripts" auf Seite 558 nach.

   Bei der interaktiven Wiedergabe zeigt TruClient den Fortschritt bei jedem Schritt entsprechend dem echten Ausführungsfortschritt durch Auffüllen der Schrittleiste an. Der Fortschritt bei der Schrittausführung manifestiert sich beispielsweise durch folgende Ereignisse: Testobjekt gefunden, Aktion ausgeführt oder Schritt beim Endereignis synchronisiert. Weitere Informationen zum Synchronisieren von Skriptschritten finden Sie unter "Synchronisieren von TruClient-Skriptschritten" auf Seite 546.

   Darüber hinaus können Sie Details zu den Skriptaktionen im **Statusbereich auf der TruClient-Seitenleiste** anzeigen.

5. ### Erweitern des Skripts

   Sie können Ihr Skript in vielfältiger Weise erweitern, indem Sie z. B. Parameter, Transaktionen, Schleifen und Schritte zur Verifizierung einfügen. Weitere Informationen über die Aufgabe finden Sie unter "Erweitern von TruClient-Skripts" auf Seite 576.

6. ### Beenden der Entwicklung

   Klicken Sie auf die Schaltfläche **Speichern** , um das Skript zu speichern. Schließen Sie das Browserfenster.

7. ### Allgemeine Tipps zur erfolgreichen interaktiven Wiedergabe

   Ändern Sie die Größe des Browsers zwischen Aufzeichnung und Wiedergabe bzw. während der Wiedergabe nicht. Dies kann bewirken, dass Objekte verschoben werden und TruClient diese nicht mehr findet.

   Wechseln Sie während der interaktiven Wiedergabe nicht zwischen Anwendungen. Lassen Sie den Browser im Vordergrund.

   Hinweis: Besonders wichtig ist dies, wenn die Funktion **Zugehörige Objekte** verwendet wird, da sich eine Größenänderung auf die relative Position der Objekte auswirken kann.

Hinweis: Anpassungen (wie Lesezeichen), die Sie in dieser Instanz von Firefox durchgeführt haben, werden nicht global gespeichert. Wie bereits erwähnt öffnet TruClient jedes Skript in einem eindeutigen Firefox-Profil. Wenn Sie Firefox noch für andere Dinge als die Erstellung dieses Skripts verwenden möchten (um z. B. im Internet zu surfen), sollten Sie ein weiteres Firefox-Fenster öffnen.

Berechnen der Anzahl an Lastgeneratoren, die für TruClient-Skripts erforderlich sind

In dieser Aufgabe wird beschrieben, wie Sie die Anzahl der Lastgeneratoren berechnen, die zum Ausführen mehrerer TruClient-Vuser im Controller erforderlich sind.

Hinweis: Diese Aufgabe ist Bestandteil einer übergeordneten Aufgabe. Weitere Informationen finden Sie unter "Entwickeln von TruClient-Skripts" auf Seite 549.

Die TruClient-Technologie bietet Ihnen die Möglichkeit, komplexe Geschäftsprozesse schnell und einfach aufzuzeichnen. Da TruClient die Aufzeichnung jedoch auf der Benutzerebene durchführt und für die Wiedergabe einen Browser benötigt, bedeutet eine komplexere Clientlogik der Anwendung, dass zum Ausführen eines Vusers mehr CPU und Speicher erforderlich sind.

Hinweis: Die TruClient-Auslastung kann deutlich größer sein als die Auslastung anderer Vuser-Protokolle. Durch diese größere Auslastung ist mehr CPU und Speicherkapazität erforderlich, als zum Ausführen eines ähnlichen Geschäftsprozesses benötigt wird, der in einem anderen Protokoll aufgezeichnet wurde.

Verwenden Sie die folgende Methode, um die erforderlichen Anzahl Lastgeneratoren zu ermitteln:

1. Zeichnen Sie ein Skript mit TruClient auf. Details finden Sie in Schritt 5 unter "Entwickeln von TruClient-Skripts" auf Seite 549.

2. Geben Sie einen einzelnen Vuser im Controller wieder, und prüfen Sie die durchschnittliche CPU-Auslastung und den Spitzenspeicherverbrauch des Prozesses "mdrv.exe", indem Sie einen Zähler für **Prozessorzeit (%)** und **Private Bytes** hinzufügen. Weitere Informationen zum Arbeiten mit Windows-Ressourcenmonitoren finden Sie unter "Dialogfeld zum Hinzufügen von Messgrößen für Windows-Ressourcen".

3. Berechnen Sie anhand Ihrer Lastgenerator-Hardware sowie der CPU-Auslastung und des Speicherverbrauchs eines einzelnen Vusers die Anzahl der Vuser pro Computer.

   Beispiel:

   Angenommen, jeder Lastgenerator weist 8 Core-Prozessoren und 8 GB Arbeitsspeicher auf.

   Angenommen, ein einzelner Vuser verbraucht 80 MB maximalen Arbeitsspeicher und 10 % CPU als durchschnittlichen Wert für den angegebenen Geschäftsprozess.

   Wenn wir aus der CPU-Perspektive die CPU-Auslastung auf maximal 70 % Auslastung begrenzen, können wir 7 Vuser pro Core-Prozessor (70 %/10 %) verarbeiten. Wenn unser Lastgenerator insgesamt 8 Core-Prozessoren aufweist, entsprechen 8 * 7 Vuser pro Prozessor 56 Vusern pro Lastgenerator.

Aus der Speicherperspektive weist der Lastgeneratorcomputer 8 GB Arbeitsspeicher auf, von denen 7 GB für die Vuser zur Verfügung stehen. So sind ca. 87 Vuser pro Lastgeneratorcomputer (7 GB/80 MB) möglich.

Wenn also sowohl die Beschränkungen der CPU als auch der Speicherkapazitäten erfüllt werden sollen, verwenden wir die niedrigere Anzahl Vuser. Somit berechnen wir für diesen Geschäftsprozess, dass wir ca. 56 Vuser pro Lastgenerator ausführen.

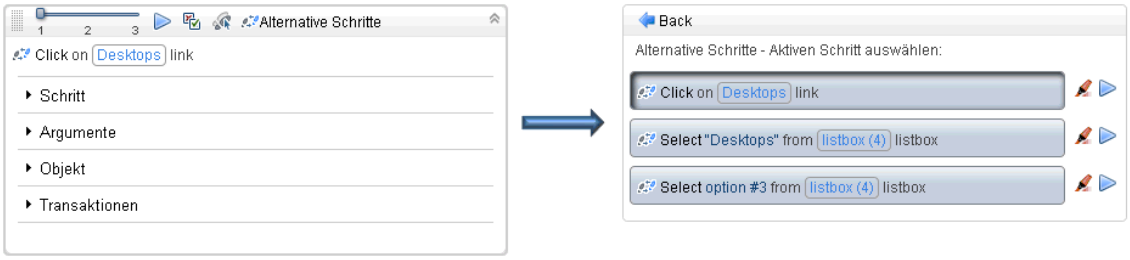
TruClient - Debuggen eines TruClient-Skripts

Alternative Schritte für TruClient

Alternative Schritte ermöglichen Ihnen, einen vorhandenen Schritt durch einen anderen Schritt zu ersetzen, der dieselbe Aktion ausführt. Dies kann beim Debuggen eines Skripts hilfreich sein. Nehmen wir beispielsweise an, der Schritt wählt ein Element auf Grundlage der Elementnummer aus einem Menü aus. TruClient gibt Ihnen die Möglichkeit, einen alternativen Schritt anzugeben, bei dem das Menülelement auf Grundlage seines Namens ausgewählt wird.

Schritte, für die alternative Optionen zur Verfügung stehen, sind mit einem Symbol für alternative Schritte gekennzeichnet. Klicken Sie auf dieses Symbol, um die alternativen Optionen für den Schritt anzuzeigen. Klicken Sie auf die gewünschte Alternative und wählen Sie **Zurück** aus.

Im Folgenden ist ein Snapshot eines Schrittes dargestellt, in dem das zweite Element in einem Listenfeld mit dem Namen "Desktop" ausgewählt wurde. Mit der Funktion für alternative Schritte können Sie den Schritt durch Klicken auf den Link "Desktop", durch Auswahl des Objekts "Desktop" im Listenfeld oder durch Auswahl des zweiten Elements im Listenfeld definieren.



TruClient-Skriptebenen

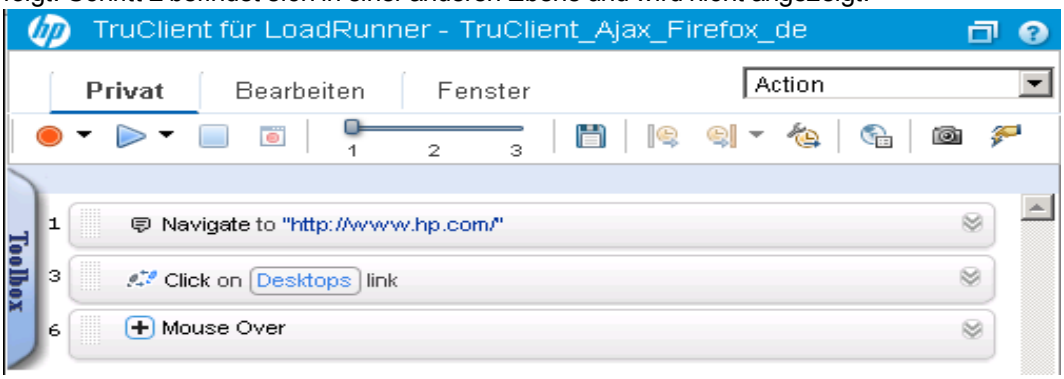
Beim Aufzeichnen eines Geschäftsprozesses werden einige Schritte vom Benutzer ausgeführt, die bei der Wiedergabe nicht notwendig sind. TruClient entfernt Schritte, die es für unnötig hält, und ordnet sie verschiedenen Skriptebenen zu. Ein Klickschritt, der in einem Anwendungsbereich stattfindet, der keinerlei Auswirkungen nach sich zieht, wird z. B. Ebene 2 zugeordnet. TruClient nimmt an, dass dieser Schritt nicht signifikant ist und dem Benutzer beim Emulieren eines Geschäftsprozesses in der Anwendung nicht weiterhilft. In der Standardansicht werden nur Schritte der ersten Ebene angezeigt. Um auch die Schritte der Ebenen 2 und 3 anzuzeigen, verwenden Sie den Schieberegler der Startregisterkarte. In der Wiedergabephase werden nur die sichtbaren Schritte ausgeführt.

In bestimmten Fällen müssen Sie die Entscheidungen von TruClient vielleicht revidieren und die Ebene eines Schrittes manuell ändern. Dies kann z. B. bei Mouse over-Schritten der Fall sein, die bei der Wiedergabe erforderlich sind. TruClient hält Mouse over-Schritte beim Wiedergeben generell für unnötig und weist sie der Ebene 3 zu. Weitere Informationen hierzu finden Sie unter "Debuggen von TruClient-Skripts" auf Seite 558.

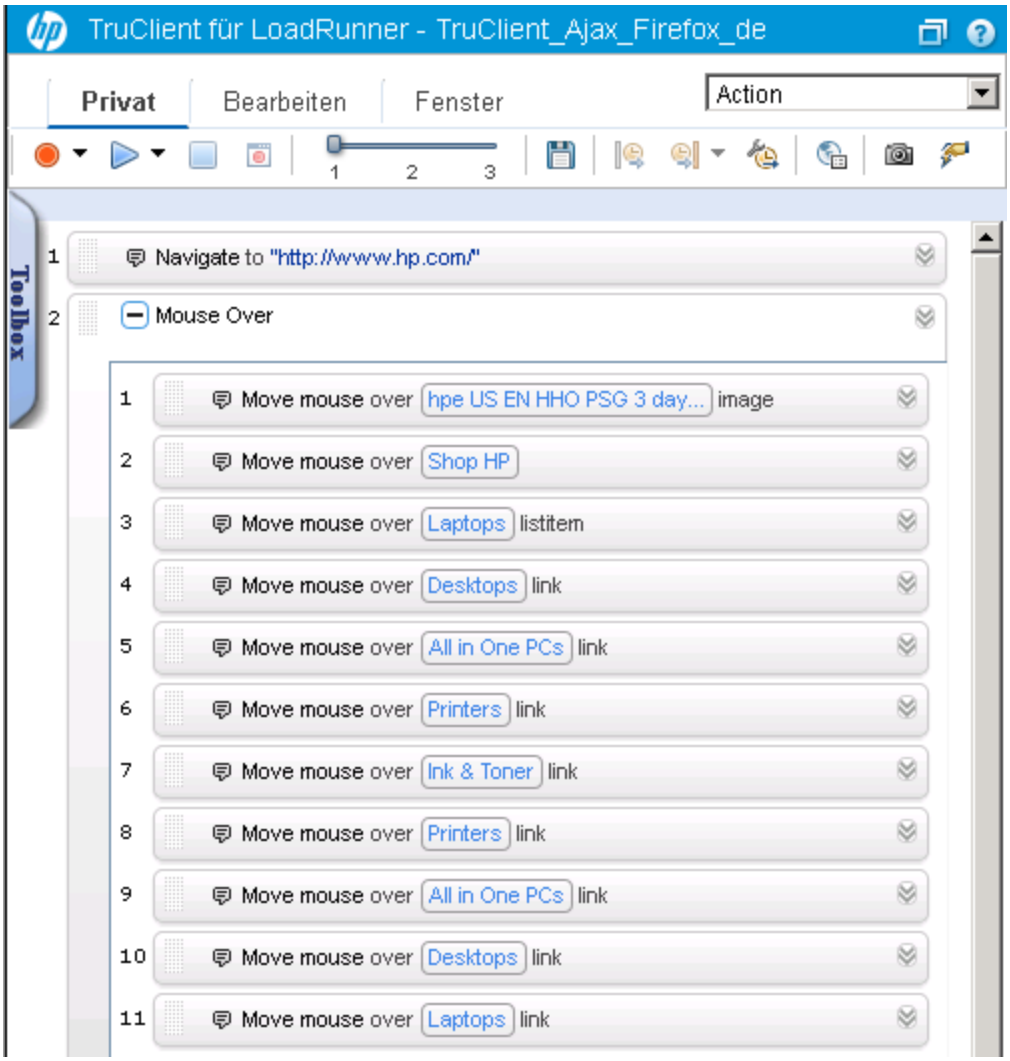
Hinweis: Automatische Ebenenzuweisung bei der Wiedergabe

Die Ebene eines Schrittes wird normalerweise beim Aufzeichnen anhand der Wichtigkeit der Ereignisse im Geschäftsprozess festgelegt. Es kann vorkommen, dass ein wichtiger Schritt für unbedeutend gehalten und einer niedrigeren Skriptebene zugeordnet wird. Infolgedessen kann bei der Wiedergabe ein Fehler des Typs "Objekt nicht gefunden" auftreten. Bei der Wiedergabe prüft TruClient, ob Schritte, die sich auf das Ergebnis des aktuellen Schrittes auswirken können, einer niedrigeren Ebene zugeordnet sind. Wird ein signifikanter Schritt gefunden, wird er einer höheren Skriptebene zugeordnet.

Der folgende Screenshot zeigt ein kleines Skript. Beachten Sie, dass auf Schritt 1 sofort Schritt 3 folgt. Schritt 2 befindet sich in einer anderen Ebene und wird nicht angezeigt.



Nachdem die Ansichtseinstellungen mit dem Schieberegler geändert wurden, werden nun alle Schritte angezeigt und bei der Wiedergabe im interaktiven Modus ausgeführt.



TruClient-Snapshots

Auf dem TruClient-Protokoll basierende Vuser-Skripts verwenden den Snapshot-Bereich von VuGen.

- Eine Einführung zum Snapshot-Bereich finden Sie unter "Snapshot-Bereich - Übersicht" auf Seite 62.
- Details zum Arbeiten mit dem Snapshot-Bereich finden Sie unter "Arbeiten mit Snapshots" auf Seite 65.
- Details zur Benutzeroberfläche des Snapshot-Bereichs finden Sie unter "Snapshot-Bereich" auf Seite 109.

TruClient generiert Snapshots während der Aufzeichnung gemäß den Einstellungen für die Snapshot-Generierung. Sie können diese Snapshots anzeigen, indem Sie den Mauszeiger auf das

Symbol der einzelnen Schritte bewegen. Die Snapshots werden vor der Implementierung der Aktion des Schrittes aufgezeichnet und als .png-Dateien gespeichert. Klicken Sie auf die einzelnen Snapshots, um diese in einer neuen Browser-Registerkarte anzuzeigen. Vergewissern Sie sich vor der Wiedergabe, dass die richtige Registerkarte aktiviert ist. Aufzeichnungs-Snapshots werden im Snapshot-Ordner gespeichert.

Sie können die Snapshot-Generierung während der Aufzeichnung und Wiedergabe konfigurieren, indem Sie auf die Schaltfläche **Allgemeine Einstellungen** klicken und die Registerkarte **Interaktive Optionen** auswählen. Weitere Informationen finden Sie unter "Dialogfeld "TruClient - Allgemeine Einstellungen"" auf Seite 609.

Sie können Snapshots im Snapshot-Viewer in VuGen anzeigen. Weitere Informationen finden Sie unter "Arbeiten mit Snapshots" auf Seite 65. Sie können Snapshots auch im TruClient-Snapshot-Viewer anzeigen. Weitere Informationen finden Sie unter "TruClient-Startregisterkarte" auf Seite 599.

TruClient kann entsprechend Ihren Spezifikationen in den Laufzeiteinstellungen auch Snapshots im Lastmodus erzeugen. Weitere Informationen finden Sie unter "Allgemein > Knoten "Wiedergeben"" auf Seite 360.

Wiedergabe-Snapshots werden im Ergebnisordner gespeichert und anhand des Wiedergabetyps (interaktiv oder Last), des Skriptabschnitts und der Iteration organisiert.

Debuggen von TruClient-Skripts

In dieser Aufgabe werden die unterschiedlichen Optionen zum Debuggen eines TruClient-Skripts beschrieben.

Anzeigen von Wiedergabefehlern in der TruClient-Seitenleiste

Sind während der Wiedergabe Schritte fehlgeschlagen, werden diese mit einem Fehlersymbol markiert. Bewegen Sie die Maus über diese Symbole, um Fehlerbeschreibungen zu erhalten.

Skript Schritt für Schritt ausführen

Sie können Ihr Skript Schritt für Schritt ausführen, um die Wiedergabe langsam und kontrolliert zu verfolgen. Wählen Sie dazu den Pfeil nach unten der Wiedergabe-Schaltfläche im Browser, und klicken Sie auf **Schritt für Schritt wiedergeben**. Wiederholen Sie diesen Vorgang nach jedem Schritt, um mit der schrittweisen Wiedergabe fortzufahren.

Anzeigen der Wiedergabeprotokolle

Im VuGen-Ausgabebereich können Sie Details zur Wiedergabe Ihres Skripts anzeigen. Wählen Sie die Protokolle **Ausgabebereich > Wiedergeben** oder **Ausgabebereich > Browserwiedergabe**. Details finden Sie im "Ausgabebereich" auf Seite 107.

Einfügen von Haltepunkten zum Umschalten

Haltepunkte weisen das Skript im interaktiven Modus an, die Ausführung während einer Wiedergabe anzuhalten. Damit können Sie das Skript einfacher debuggen. Zum Einfügen eines Haltepunkts zum Umschalten wählen Sie den gewünschten Schritt aus und klicken auf die Schaltfläche **Haltepunkte** .

Debuggen von Skripts mithilfe von Snapshots

Mit den während der Wiedergabe generierten Snapshots können Sie Skripts debuggen, indem Sie die Snapshots der fehlgeschlagenen Schritte anzeigen.

1. Wählen Sie die Schaltfläche in der **TruClient-Seitenleiste** und anschließend die Registerkarte **Interaktive Optionen** aus. Legen Sie die Einstellung **Generierung von Wiedergabe-Snapshots** auf **Bei Fehler** fest.

2. Geben Sie das Skript aus der **TruClient-Seitenleiste** wieder.

3. Durchsuchen Sie die Protokolle **Ausgabebereich > Wiedergeben** oder **Ausgabebereich > Browserwiedergabe** nach Fehlern. Notieren Sie sich die Nummern der Schritte, bei denen Fehler aufgetreten sind.

4. Um die Snapshots über die **TruClient-Seitenleiste** anzuzeigen, wählen Sie einen Schritt mit einem Fehler aus und klicken auf die Schaltfläche .

   oder

   Um die Snapshots über VuGen anzuzeigen, wählen Sie **Ansicht > Snapshot-Bereich**.

Sie erhalten eine Gruppe von Snapshots mit Fehlern im Skript.

Ändern und Anzeigen von Skriptebenen

Es kommt vor, dass aufgezeichnete Schritte, die für die Wiedergabe erforderlich sind, den Ebenen 2 und 3 zugeordnet werden. In dem Fall müssen Sie die Ebene manuell in Ebene 1 ändern.

- Zum Ändern der Wiedergabeebene eines Skripts ziehen Sie den Schieberegler auf der Symbolleiste zur gewünschten Ebene. Durch Ziehen des Reglers auf Ebene 3 werden die Schritte der Ebenen 1, 2 und 3 angezeigt und wiedergegeben.

- Wenn Sie einen Schritt auf eine andere Ebene verschieben möchten, öffnen Sie den Schritt und klicken Sie auf den Schrittabschnitt. Schieben Sie den Regler auf die gewünschte Ebene. Ist der Schritt Teil eines Gruppenschritts, müssen sowohl der Gruppenschritt als auch der einzelne Schritt geändert werden.

Weitere Informationen finden Sie unter "TruClient-Skriptebenen" auf Seite 555.

Einfügen von Warteschritten

Manchmal tritt bei der Wiedergabe eines Skripts ein Fehler auf, weil ein Objekt in einem Schritt nicht verfügbar ist, wenn der vorherige Schritt abgeschlossen ist. Sie können dieses Problem beheben, indem Sie **Warteschritte** in das Skript einfügen, mit denen die Skriptwiedergabe angehalten wird, bevor mit dem nächsten Schritt fortgefahren wird. Es gibt zwei verschiedene Arten von **Warteschritten**:

- Beim Schritt **Warten** wird das Skript für eine festgelegte Zeitspanne angehalten, bevor mit dem nächsten Schritt fortgefahren wird.

- Beim Schritt **Auf Objekt warten** wird das Skript angehalten, bis ein angegebenes Objekt geladen wird, bevor mit dem nächsten Schritt fortgefahren wird.

Warteschritte beginnen nach dem **Endereignis** des vorherigen Schritts. Der vorherige Schritt kann also weiter ausgeführt werden, nachdem der **Warteschritt** erreicht wurde.

Wählen Sie zum Einfügen eines **Warteschritts** die Option **Toolbox > Funktionen** aus, und ziehen Sie das Symbol **Warten** oder **Auf Objekt warten** an die gewünschte Position im Skript. Wenn Sie einen Schritt **Warten** hinzufügen, konfigurieren Sie das Intervall im Argumentbereich des Schrittes. Wenn Sie einen Schritt **Auf Objekt warten** hinzufügen, klicken Sie auf die Schaltfläche **Zum Auswählen eines Objekts klicken**, um das Zielobjekt in der Anwendung auszuwählen.

Hinweis: Warteschritte unterscheiden sich von **Latenzzeit**schritten in anderen Protokollen. Mit der **Latenzzeit** wird die Zeit gesteuert, die VuGen zwischen Aktionen wartet. Mit **Warteschritten** wird eine Skriptwiedergabe angehalten, bis entweder eine bestimmte Zeit verstrichen ist oder ein festgelegtes Objekt geladen wurde.

Weitere Tipps für das Debuggen von Skripts

Alarmfunktion

Da alle TruClient-Argumente JavaScript unterstützen, können Sie die **Alarm**funktion verwenden, um während der Skriptentwicklung Informationen anzuzeigen. Sie können ein DOM-Element auch mithilfe regulärer Funktionen, z. B. **location**, referenzieren.

Firebug für IE oder Dom Inspector für FF

Um die Debuggingfunktionen weiter zu verbessern, können Sie Plug-Ins wie DOM Inspector und Firebug installieren, die zusätzliche Informationen bezüglich der Anwendungsobjekteigenschaften bereitstellen.

Weitere Informationen zu Firebug Lite finden Sie unter " TruClient-Browser für IE" auf Seite 608.

Auflösen von Schrittzeitüberschreitungen

Schrittzeitüberschreitungen können aus unterschiedlichen Gründen auftreten:

- Anwendung reagiert langsam (z. B. unter Last). Dies ist ein wichtiges Testergebnis.
- Die Schrittzeitüberschreitung ist nicht korrekt und sollte im Bereich **Step** der Schritteigenschaften geändert werden.
- Das Endereignis des Schritts ist nicht korrekt und der Schritt wartet auf ein Ereignis, das nicht eintritt. Das Endereignis sollte im Bereich **Step** der Schritteigenschaften geändert werden.

Lösen von Problemen bei der Objekterkennung

Die Objekterkennung stellt eine der größten Herausforderungen beim Aufzeichnen und Wiedergeben von Web 2.0-Webanwendungen dar, weil aufgezeichnete Objekte verschoben oder

inhaltlich geändert werden können. Wenn sich aufgezeichnete Objekte während der Wiedergabe dynamisch ändern, kann TruClient die Fähigkeit zur automatischen Objekterkennung verlieren.

TruClient umfasst fortschrittliche Mechanismen, um dieses Problem zu lösen. Dazu zählen die Optionen Markieren, Objekterkennung verbessern, Objekt ersetzen und Zugehöriges Objekt sowie die Optionen des Assistenten für die Objekterkennung. In den folgenden Schritten sind Möglichkeiten zur Verwendung dieser Mechanismen beschrieben.

Hinweis: Wenn Objekte für Anwendungen erkannt werden sollen, die in mehreren Fenstern aufgezeichnet wurden, stellen Sie sicher, dass unter **TruClient-Seitenleiste > Registerkarte "Fenster" > Fenster "Wiedergabe"** das richtige Fenster ausgewählt ist.

Die Optionen Markieren, Objekterkennung verbessern, Objekt ersetzen und Zugehörige Objekte setzen voraus, dass der Benutzer in der Anwendung ein Objekt auswählt. Es gibt jedoch Fälle, in denen in der Anwendung mehrere Aktionen erforderlich sind, um das Objekt anzuzeigen, zum Beispiel Bewegen der Maustaste auf ein Objekt oder Mausklicks. Verwenden Sie in diesen Fällen STRG+ALT+F4, um den TruClient-Objektauswahlmodus anzuhalten, bis Sie das Objekt angezeigt haben. Drücken Sie dann erneut STRG+ALT+F4, um das Objekt auszuwählen.

Tipp: Nachdem Sie die Änderungen durchgeführt haben, geben Sie den betreffenden fehlgeschlagenen Schritt und anschließend das gesamte Skript erneut wieder. Auf diese Weise können Sie leichter feststellen, ob die Änderung das Problem gelöst hat.

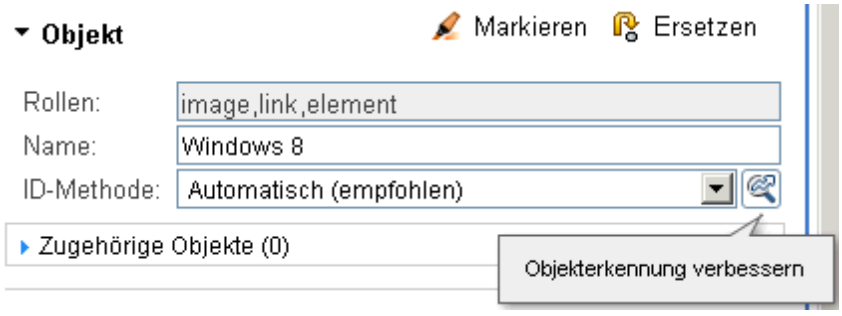
In den folgenden Schritten sind Möglichkeiten zur Lösung von Problemen bei der Objekterkennung beschrieben.

Markieren eines Objekts

Unabhängig von der verwendeten Methode der Objekterkennung können Sie die Markieren-Schaltfläche im Objektbereich des Schrittes verwenden, um zu überprüfen, ob ein Objekt in der Anwendung jederzeit angezeigt wird. Wird das Objekt nicht gefunden, kann ein Takt- und Zeitsteuerungsproblem vorliegen. Wird das Objekt nicht gefunden, wird eine Fehlermeldung angezeigt.

Verbessern der Objekterkennung

Wenn die Markieren-Funktion fehlschlägt, verwenden Sie die Funktion zur Verbesserung der Objekterkennung.



Diese Option befindet sich im Abschnitt "Objekt" des Schritts neben der Dropdownliste für die ID-Methode. Auf diese Weise lernt TruClient die Eigenschaften des Objekts neu und vergleicht diese

mit den bei der Aufzeichnung erlernten Eigenschaften. Basierend auf den Eigenschaften können die erforderlichen Anpassungen durchgeführt werden. Abhängig davon, wie dynamisch die Anwendung ist, müssen Sie die Funktion **Objekterkennung verbessern** möglicherweise mehr als einmal verwenden.

Nachdem Sie dies getan haben, versuchen Sie, den Schritt erneut wiederzugeben, um sicherzustellen, dass das Problem gelöst wurde.

Alternative Schritte

Die alternativen Schritte ermöglichen Ihnen das Anzeigen von Instanzen, in denen dieselbe Aktion in einem Schritt auf unterschiedliche Weise ausgeführt werden kann. Wenn die Funktion für die verbesserte Objekterkennung fehlschlägt, versuchen Sie, einen der alternativen Schritte zu verwenden.

Möglicherweise klicken Sie auf eine Option in einer Dropdownliste, in der sich der Text basierend auf einem Wert ändert.

Wenn Sie versuchen, basierend auf dem Text zu klicken, schlägt der Schritt möglicherweise fehl.

Wenn Sie einen alternativen Schritt verwenden, der ein Element in der Liste basierend auf der Ordnungszahl der Option innerhalb der Liste auswählt, verläuft der Klick unabhängig von dem Text erfolgreich.

Hinweis: Bevor Sie eine der Alternativen auswählen, versuchen Sie, das vom alternativen Schritt verwendete Objekt zu markieren und dieses wiederzugeben. Auf diese Weise stellen Sie sicher, dass der alternative Schritt die erforderliche Aktion ausführt.

Ändern der Methode der Objekterkennung

Sie können die Art und Weise der Objekterkennung durch TruClient ändern, indem Sie die Objekterkennungsmethode im Abschnitt **Objekt** der Schritteigenschaften ändern. Die folgenden Optionen sind verfügbar:

- **Automatisch.** Die Standardmethode zur Objektidentifizierung von TruClient. Die Methode "Automatisch" ermöglicht TruClient die Verwendung der internen, fortschrittlichen Algorithmen, um das Objekt zu finden. Wenn diese Methode das Objekt während der Wiedergabe nicht findet, klicken Sie auf die Schaltfläche **Objekterkennung verbessern** und geben das Skript erneut wieder.

- **XPath.** Wenn die automatische Erkennung auch nach Verwendung der Option zur Verbesserung der Objekterkennung oder der Option für zugehörige Objekte (im Folgenden beschrieben) fehlschlägt, versuchen Sie, die XPath-Erkennungsmethode zu verwenden. Diese Methode identifiziert das Objekt basierend auf einem XPath-Ausdruck, der das Objekt in der DOM-Struktur definiert. Klicken Sie auf den Dropdownpfeil neben dem Bearbeitungsfeld **XPath**, um einen vorgeschlagenen XPath für das Objekt auszuwählen. Sie können den vorgeschlagenen Pfad manuell ändern. Um die Einstellung auf einen der ursprünglich von TruClient generierten Ausdrücke zurückzusetzen, wählen Sie in der Dropdownliste erneut eine der verfügbaren Optionen aus.

Wenn Sie beispielsweise das erste Suchergebnis unabhängig von dem gesuchten Begriff auswählen müssen, kann die XPath-Erkennung helfen.

- **JavaScript.** JavaScript-Code, der ein Objekt zurückgibt. Beispiel: document.getElementById ("Suchschaltfläche") gibt ein Element mit dem DOM-ID-Attribut "Suchschaltfläche" zurück.

Mit der JavaScript-Erkennungsmethode können Sie JavaScript-Code schreiben, der das zurückgegebene Dokument referenziert, und CSS-Auswahlfunktionen und andere Standardfunktionen verwenden.

Nehmen wir an, die vom Server zurückgegebene Seite enthält mehrere Links mit demselben Attribut "Titel" (Suchergebnisse), und wir möchten, dass das Skript auf Zufallsbasis auf einen der verfügbaren Links klickt.

Die Objekterkennung kann in diesem Fall bei Verwendung der JavaScript-Erkennungsmethode wie folgt aussehen:

```
var my_results = document.querySelectorAll('a[title="SearchResult"]');

random(my_results);
```

Ändern der Zeitsteuerung des Skripts

Manchmal werden Objekte nicht gefunden, weil Zeitsteuerungs- und Synchronisierungsprobleme vorliegen. Beispielsweise kann das Skript nach einem in der Anwendung vorhandenen Objekt suchen, jedoch wurde das Skript zu schnell wiedergegeben und fährt bereits mit einer anderen Seite fort. Wenn Sie den Verdacht haben, dass das Objekt aufgrund von Zeitsteuerungs- oder Synchronisierungsproblemen nicht gefunden wurde, können Sie Warteschritte einfügen. Weitere Informationen finden Sie unter "Debuggen von TruClient-Skripts" auf Seite 558.

Herstellen von Beziehungen zwischen Objekten

Wenn das Problem mit der Funktion **Objekterkennung verbessern** und den alternativen Schritten nicht gelöst werden kann, verwenden Sie die Option **Zugehörige Objekte**.

Wenn ein Objekt an sich schwer zu erkennen ist, können Sie das Objekt basierend auf einem anderen, stabileren Objekt kennzeichnen. Sie können beispielsweise ein Objekt auswählen, das nicht dynamisch ist, und dieses mit dem Zielobjekt in Beziehung setzen. Beziehungen werden visuell definiert und verbinden Objekte basierend auf ihrem Pixel-Abstand zu anderen Objekten. Beziehungen werden pro ID-Methode und pro Objekt definiert. Wenn für eine ID-Methode eines vorhandenen Objekts mehr als eine Beziehung definiert ist, müssen beide Beziehungen dasselbe Objekt suchen, damit der Schritt erfolgreich ist. TruClient verwendet dieses Objekt, um das Finden des Zielobjekts zu unterstützen. Um diese Funktion zu verwenden, erweitern Sie den Schritt, wählen **Objekt > Zugehörige Objekte** aus und klicken auf die Schaltfläche ✚. Befolgen Sie die Anweisungen, um eine Beziehung herzustellen. Überprüfen Sie, ob dieser Vorgang erfolgreich war, indem Sie sowohl das Objekt als auch das verbundene Objekt markieren.

Tipps:

- Verwenden Sie diese Funktion nur, wenn andere Erkennungsmethoden fehlgeschlagen sind, da diese Funktion sehr ressourcenintensiv sein kann.

- Verwenden Sie den minimalen Suchbereich, um die Leistung zu verbessern.
- Verknüpfte Objekte reagieren empfindlich auf Fenstergrößen. Das Ändern der Fenstergröße kann sich auf die Objektpositionen und -beziehungen auswirken. Sie sollten dies berücksichtigen.
- Jede Erkennungsmethode (Automatisch, XPath und JavaScript) verfügt über einen eigenen Satz verknüpfter Objekte. Diese verknüpften Objekte werden von Erkennungsmethoden nicht gemeinsam verwendet.
- Wenn mehrere Beziehungen existieren, müssen alle dieser Beziehungen gefunden werden, damit die Erkennung erfolgreich verläuft.

Ersetzen eines Objekts

Wenn Sie bei der Aufzeichnung das falsche Objekt ausgewählt haben oder ein Objekt sich permanent geändert hat, können Sie dieses durch ein anderes Objekt ersetzen, ohne den Schritt zu ersetzen. Auf diese Weise wird der Schritt zurückgesetzt und die an dem ursprünglichen Schritt durchgeführten Änderungen werden gelöscht, wie zum Beispiel Beziehungen. Erweitern Sie den Schritt, wählen Sie **Objekt** aus und klicken Sie dann auf die Schaltfläche **Ersetzen** . Wählen Sie das neue Objekt aus und geben Sie das Skript erneut wieder.

Die Funktion **Objekt ersetzen** informiert TruClient darüber, dass das in dem Schritt aktuell referenzierte Objekt falsch ist. TruClient entfernt alle aktuellen Informationen zu dem Objekt und lernt, wie das ausgewählte Objekt gesucht werden soll.

Aus diesem Grund sollten Sie die Option **Objekt ersetzen** nur verwenden, wenn Sie bei der Aufzeichnung das falsche Objekt verwendet haben.

Assistent für Objekterkennung

Die Objekterkennung kann während der Wiedergabe aus verschiedenen Gründen fehlschlagen. Für jeden Grund startet TruClient den Assistenten für die Objekterkennung, um zu versuchen, das Problem mit der fehlgeschlagenen Erkennung zu lösen.

Objekt kann nicht bestätigt werden

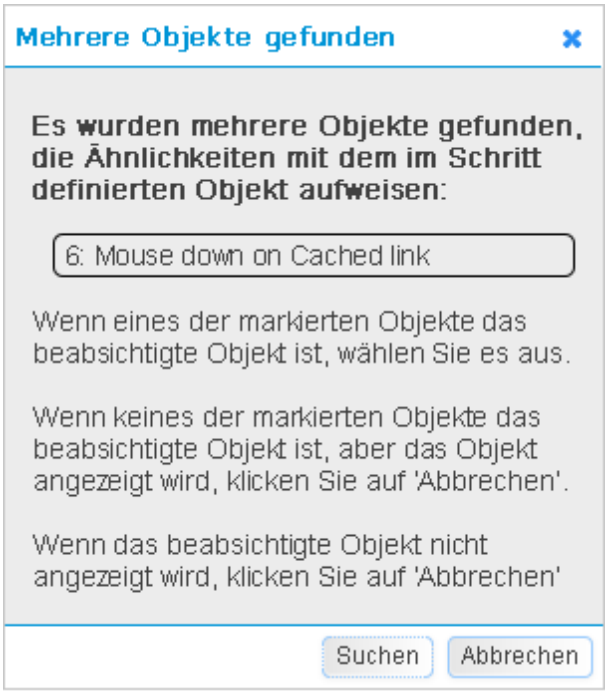
TruClient nimmt an, dass ein bestimmtes Objekt das gewünschte Objekt ist, jedoch kann dieses nicht positiv identifiziert werden. Das als gewünschtes Objekt angenommene Objekt wird auf dem Bildschirm markiert, und das folgende Dialogfeld des Assistenten wird angezeigt:

Optionen:

- **Ja**. Das angenommene (und markierte) Objekt ist das richtige Objekt.
- **Suchen**. Das angenommene Objekt ist nicht das richtige Objekt. Sie müssen das richtige Objekt in der Anwendung identifizieren.
- **Abbrechen**. Wiedergabe beenden.

Mehrere Objekte gefunden

TruClient hat mehrere Objekte gefunden, die der Identifikation des gewünschten Objekts entsprechen. Alle als gewünschte Objekte angenommenen Objekte werden auf dem Bildschirm markiert, und das folgende Dialogfeld des Assistenten wird angezeigt:

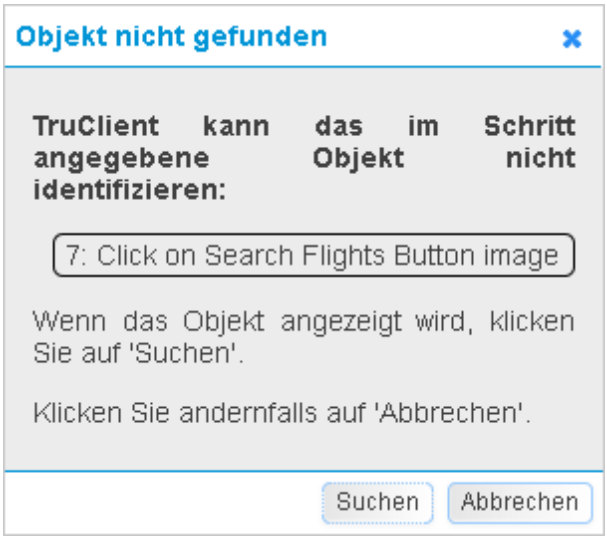


Optionen:

- Das korrekte Objekt ist eines der markierten Objekte. Klicken Sie in der Anwendung auf das Objekt, um das richtige Objekt anzugeben.
- **Suchen**. Das angenommene Objekt ist keines der markierten Objekte. In diesem Fall müssen Sie das richtige Objekt in der Anwendung markieren.
- **Abbrechen**. Wiedergabe beenden.

Objekt nicht gefunden

TruClient konnte das gewünschte Objekt nicht finden. Das folgende Dialogfeld des Assistenten wird angezeigt:



Optionen:

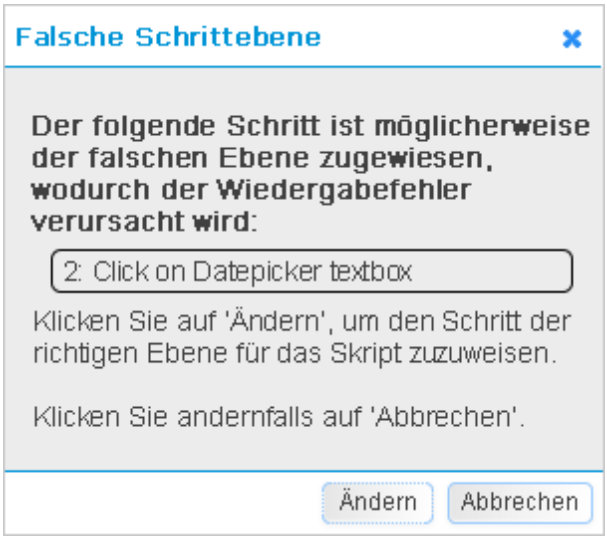
- **Suchen**. Suchen Sie das Objekt. Sie müssen das richtige Objekt in der Anwendung markieren.
- **Abbrechen**. Wiedergabe beenden.

Falsche Ebene für Schritt

TruClient hat festgestellt, dass ein Schritt auf einer höheren Ebene für die Erkennung des Objekts des aktuellen Schritts erforderlich ist. Dies tritt bei Mouse over-Schritten auf, die normalerweise in Ebene 3 aufgezeichnet werden, aber für Klick-Schritte erforderlich sind, die in Ebene 1 aufgezeichnet werden.

Wenn TruClient diese Abhängigkeit während der **Aufzeichnung** erkennt, wird der Mouse over-Schritt automatisch in Ebene 1 bewegt.

Wenn TruClient diese Abhängigkeit während der **Wiedergabe** erkennt, wird der folgende Assistent angezeigt:



Optionen:

- **Ja**. Den erforderlichen Schritt in Ebene 1 verschieben.
- **Abbrechen**. Wiedergabe beenden.

Fehlerbeseitigung bei der Objekterkennung

Wie verknüpfte Objekte die XPath- und JavaScript-Identifizierungsmethoden unterstützen können

Die XPath- und JavaScript-Identifizierungsmethoden geben abhängig vom verwendeten Ausdruck mehrere Elemente zurück.

Wenn der XPath-Wert beispielsweise //button lautet und die betreffende Webseite mehrere Schaltflächenelemente umfasst, werden mehrere Objekte zurückgegeben.

Um nur ein Objekt zurückzugeben, können Sie ein zugehöriges Objekt hinzufügen, um die Identifizierung einzugrenzen.

Fehler "Interaktive Wiedergabe mit Objekt nicht gefunden"

Wenn die Wiedergabe mit einem Fehler beendet wird, der angibt, dass das Objekt nicht gefunden wurde, versuchen Sie Folgendes:

- Wählen Sie den fehlgeschlagenen Schritt aus und klicken Sie auf **Markieren**. Wenn das richtige Objekt nicht markiert wird, verwenden Sie die Funktion zur Verbesserung der Objektidentifizierung.
- Wird das Objekt markiert, wurde der Schritt möglicherweise erreicht, bevor das Objekt auftrat. Fügen Sie einen Schritt "Warten" oder "Auf Objekt warten" vor dem problematischen Schritt ein.

In manchen Fällen müssen Sie einen verfügbaren alternativen Schritt auswählen, um das Problem zu lösen. Möglicherweise klicken Sie auf eine Option in einer Dropdownliste, in der sich der Text basierend auf einem Wert ändert. Wenn Sie versuchen, basierend auf dem Text zu klicken, schlägt der Schritt möglicherweise fehl. Wenn Sie einen alternativen Schritt verwenden, der das Element in der Liste basierend auf der Ordnungszahl dieser Option in der Liste auswählt, ist der Klick unabhängig von dem Text erfolgreich.

Die interaktive Wiedergabe schlägt aufgrund eines nicht gefundenen Objekts fehl, obwohl die Option "Markieren" das Objekt findet

Wenn die Wiedergabe fehlschlägt, obwohl die Option **Markieren** das richtige Objekt findet, kann ein Taktproblem vorliegen.

Das Laden des Objekts dauert etwas länger und der Schritt wird schneller ausgeführt. Daher wird das Objekt bei der Schrittausführung nicht gefunden, obwohl die Option **Markieren** beim Debuggen des fehlgeschlagenen Schritts das Objekt findet.

In diesem Fall wird empfohlen, das Skript zu verlangsamen, sodass ausreichend Zeit zum Laden des Objekts zur Verfügung steht. Verwenden Sie hierfür eine der folgenden Optionen:

- Ändern Sie die Objektzeitüberschreitung des fehlgeschlagenen Schritts. Diese Einstellung befindet sich im Abschnitt **Schritt** der Schritteigenschaften.
- Fügen Sie einen Schritt "Warten" oder "Auf Objekt warten" vor dem fehlgeschlagenen Schritt ein.

Die Wiedergabe kann ein Element in einer Liste nicht auswählen

Einer der häufigsten Gründe für diesen Fehler ist, dass die Namen der Elemente in der Liste dynamisch sind.

Beispielsweise kann die Liste eine Liste von Städten enthalten, die auf bereits eingegebenem Text basiert (automatisch vervollständigen).

Die Liste ändert sich basierend auf den Texttypen der Liste kontinuierlich.

Es gibt zwei Möglichkeiten, um dieses Problem zu lösen:

- Verwenden Sie einen alternativen Schritt, der ein Element in der Liste anhand der Ordnungs-ID und nicht anhand des Textes des tatsächlichen Elements auswählt.
- Wenn der Text nur teilweise dynamisch ist, können Sie einen regulären Ausdruck verwenden, um das erforderliche Element basierend auf dem teilweisen Textabgleich zu suchen.

Fehlerbehandlung von TruClient-Skripts

Prüfen auf bestimmten Text einschließlich Verzweigungen

Einen Verifizierungsschritt aus dem Funktionsbereich der Toolbox hinzuzufügen ist eine Option. In diesem Schritt können Sie verschiedene Validierungseinstellungen wie das Objekt, den gesuchten Text usw. auswählen.

Wenn Sie bestimmte Aktionen jeweils dann durchführen möchten, wenn eine Validierung erfolgreich war oder fehlgeschlagen ist, können Sie einen Fehlererfassungsschritt aus dem Flusssteuerungsbereich der Toolbox hinzufügen.

So können Sie sicherstellen, das der Schritt fortgesetzt wird, auch wenn die Validierung fehlschlägt, und in der Fehlererfassungsgruppe können Sie die Schritte definieren, die in diesem Fall ausgeführt werden sollen.

Sie können Validierungen auch programmatischer angehen. Mit JavaScript können Sie auf das DOM zugreifen und jede beliebige Eigenschaft validieren. Anschließend können Sie einen bedingten Unterbrechungs- oder Beendigungsschritt hinzufügen (über den Flusssteuerungsbereich der Toolbox), der auf dieser Verifizierung beruht.

Sie können den gesuchten Text auch direkt aus der IF-Anweisung überprüfen. Das Argument 'condition' der IF-Anweisung besteht einfach aus JavaScript-Code. Sie können JavaScript-Code verwenden, der auf das globale Fensterobjekt der getesteten Anwendung zugreift. Dies geschieht durch Verweis auf das Fenster.

Anschließend können Sie manuell prüfen, ob der Text auf der aktuellen Seite vorhanden ist. Bei einer Anwendung mit einem einzigen Frame könnten Sie z. B. die folgende Codezeile verwenden::

```
window.document.body.textContent.indexOf("Off") == -1
```

"Off" bezeichnet den gesuchten Text und -1 gibt an, dass der Text nicht gefunden wurde.

Der fragliche Code ist anwendungsabhängig.

Sie können den Code optimieren, wenn Sie mehr über die Anwendung wissen (indem Sie das spezifische Element abrufen).

Prüfen auf bestimmten Text ohne Berücksichtigung der Groß-/Kleinschreibung

In der Standardeinstellung wird beim Verifizierungsschritt die Groß-/Kleinschreibung berücksichtigt. Die Suche nach z. B. "Test" schlägt fehl, wenn "test" gefunden wird.

Wenn Sie den Verifizierungsschritt ohne Berücksichtigung der Groß-/Kleinschreibung ausführen möchten, gehen Sie folgendermaßen vor:

- Legen Sie für das Bedingungsargument im Verifizierungsschritt "Regulärer Ausdruck" fest.
- Um zu prüfen, ob die Zeichenfolge "test" unabhängig von der Groß-/Kleinschreibung im Text enthalten ist, können Sie folgenden Code verwenden:

```
RegExp("test", "i")
```

Auswählen einer zufälligen Option aus einer Liste

Setzen Sie das Ordnungszahlargument auf 0. TruClient wird daraufhin automatisch eine zufällige Option aus der Liste auswählen.

Nehmen wir z. B. eine AutoVervollständigen-Liste mit verschiedenen Städten, die von dem eingegebenen Text abhängen. Aktuell haben Sie die zweite Option ausgewählt und der Schritt lautet: Option Nr. 2 in der Städteliste auswählen.

Sie müssen nun lediglich den Argumentbereich der Schritteigenschaften öffnen und das Ordnungszahlargument in 0 ändern. Der Schritt lautet nun: Zufällige Option in der Städteliste auswählen.

Diese Option ist sehr wichtig, wenn der eingegebene Text ein Parameter ist und es keine einfache Methode gibt, im Voraus zu erfahren, wie viele und welche Werte in der Liste vorhanden sind.

Verwenden externer Funktionen im Skript

Fügen Sie Ihre JavaScript- und C-Funktionen zu den Dateien JS-Function.js und C-functions.c hinzu, die Teil des Skripts sind und in VuGen im linken Navigationsbereich angezeigt werden.

JavaScript-Funktionen können direkt aus dem TruClient-Skript aufgerufen werden, das alle Argumente und Parameter JavaScript unterstützen. Zu diesem Zweck können Sie auch einen JavaScript-Auswertungsschritt oder einen Schritt "LR.evalC" aus der Toolbox hinzufügen.

Um C-Funktionen aufzurufen, fügen Sie einen C-Auswertungsschritt aus der Toolbox hinzu.

Einige der Ereignisse und Aktionen sind im aufgezeichneten Skript nicht enthalten

Versuchen Sie das Problem mit einer der folgenden Methoden zu lösen:

- TruClient zeichnet alle Ereignisse in der Anwendung auf. Das gesuchte Ereignis kann einer anderen Skriptebene zugeordnet sein, die derzeit ausgeblendet ist.
  - Dass weitere Schritte in anderen Skriptebenen vorhanden sind, erkennen Sie daran, dass die Schritte der angezeigten Ebene nicht fortlaufend nummeriert sind.
  - Die aktuelle Skriptebene wird mit dem Schieberegler in der Symbolleiste festgelegt.

  Ändern Sie den Wert des Reglers und suchen Sie in anderen Ebenen nach dem fehlenden Ereignis oder Schritt.
  - Wenn Sie das gesuchte Ereignis gefunden haben, können Sie dessen Ebene ändern und es zu einem Teil der Ebene 1 machen.
  - Ändern Sie die Skriptebene in Ebene 1 und geben Sie das Skript erneut wieder.
- Sie können einen Schritt manuell in ein Skript einfügen:
  - Wählen Sie in der Toolbox die Option **Allgemeine Objektaktion** aus und passen Sie den Schritt so an, dass er die erforderliche Aktion ausführt.

Das Ziehen eines Schiebereglers oder einer Karte wird nicht korrekt wiedergegeben

Wenn das Ziehen nicht funktioniert (z. B. beim Einstellen eines Schiebereglers oder Ziehen einer Karte) und das Ergebnis das Steuerelement nicht an die gewünschte Position versetzt, versuchen Sie Folgendes:

- Verwenden Sie einen der alternativen Schritte.
- Legen Sie die Werte manuell fest (geben Sie z. B. die exakte Pixelzahl für jede Richtung an, in die das Objekt gezogen werden soll).
- Versuchen Sie die Funktion "Ziehen zu" (indem Sie die Aktion des Schritts "Ziehen" im Schrittbereich der Schritteigenschaften ändern). Auf diese Weise können Sie das Objekt an eine Position in Relation zu einem anderen Objekt ziehen.

Abfragen eines Objekts

Erstellen Sie eine Schleife, die folgende Schritte enthält:

- Einen Schritt, der eine Aktion für ein Objekt durchführt.
- Einen Fehlererfassungsbereich, der einen Fortsetzen-Schritt enthält.
- Einen Unterbrechungsschritt.

Der Fehlererfassungsbereich sorgt dafür, dass die Schleife so lange fortgesetzt wird, bis das Objekt gefunden wird und der Schritt erfolgreich war.

Erstellen einer WHILE-Schleife

Eine "For"-Schleife hat 3 Argumente: Initialisierung, Bedingung und Inkrement.

Eine "While"-Schleife ist im Prinzip eine "For"-Schleife, die nur das Bedingungsargument kennt.

Um eine While-Schleife zu erstellen, fügen Sie eine For-Schleife hinzu (über das Kontextmenü oder die Toolbox), löschen die Argumente Initialisierung und Inkrement und geben die Bedingung an.

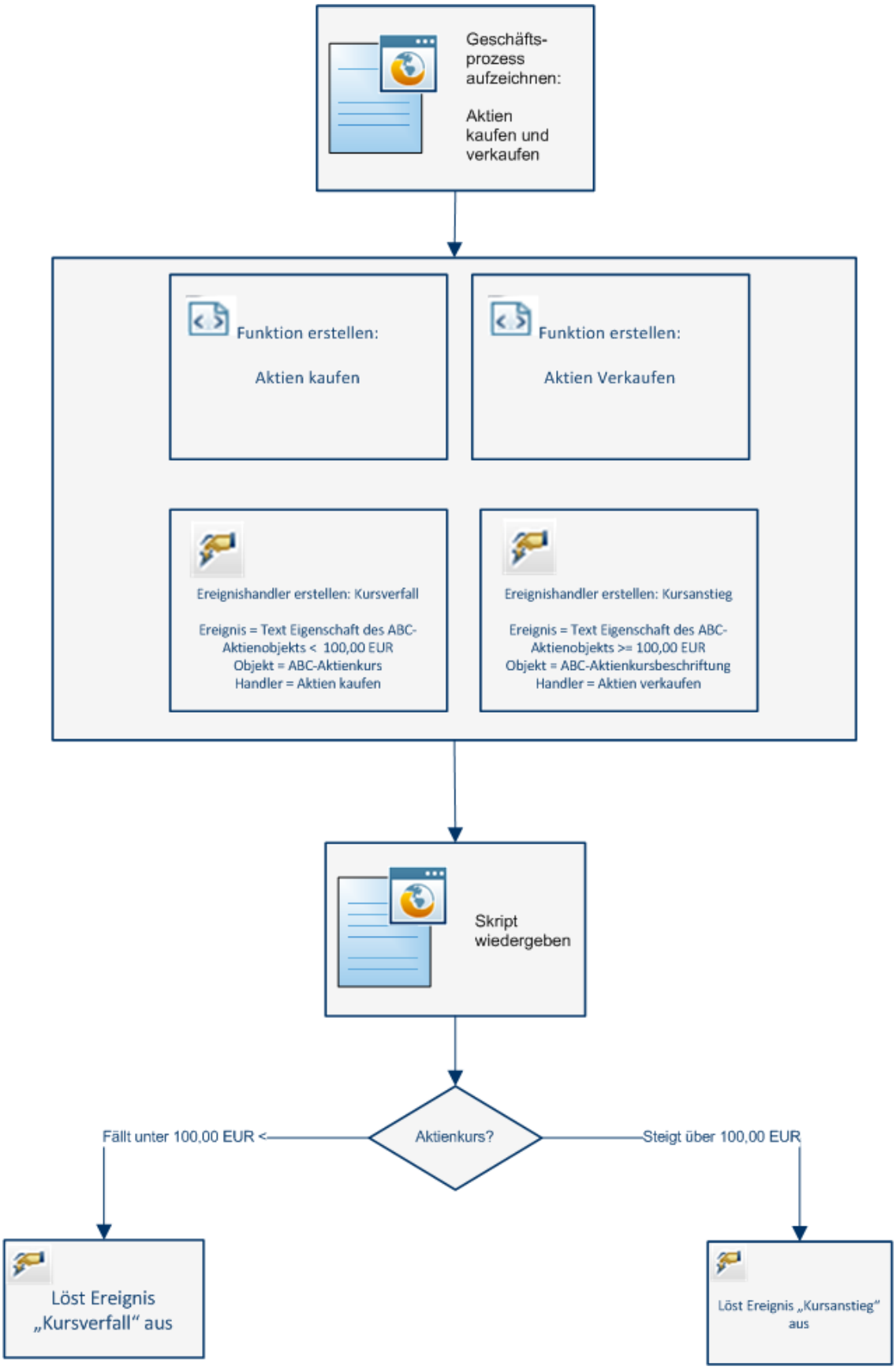
TruClient - Verbessern eines TruClient-Skripts

TruClient-Ereignishandler

Ereignishandler können Ereignisse verarbeiten, die zu einem beliebigen Zeitpunkt während der Ausführung des Skripts auftreten. Der Ereignistrigger wird auf ein Objekt gesetzt, das mit oder ohne eine zusätzlich angegebene Eigenschaft in der Anwendung angezeigt wird. Der Handler (Aktion) des Ereignisses ist eine Funktion, die aus den Funktionsbibliotheken des Skripts ausgewählt wurde.

Während der Wiedergabe des Skripts überprüft TruClient bei jedem Schritt, ob ein definiertes Ereignis vorliegt. Ist dies der Fall, führt TruClient die mit dem Ereignis verknüpfte Funktion aus.

Das folgende Flussdiagramm stellt den Workflow eines Ereignishandlers dar, wobei als Beispiel der Geschäftsprozess des Aktienkaufs und -verkaufs dient.


Geschäfts-
prozess
aufzeichnen:
Aktien
kaufen und
verkaufen
Funktion erstellen:
Aktien kaufen
Funktion erstellen:
Aktien Verkaufen
Ereignishandler erstellen: Kursverfall
Ereignis = Text Eigenschaft des ABC-Aktienobjekts < 100,00 EUR
Objekt = ABC-Aktienkurs
Handler = Aktien kaufen
Ereignishandler erstellen: Kursanstieg
Ereignis = Text Eigenschaft des ABC-Aktienobjekts >= 100,00 EUR
Objekt = ABC-Aktienkursbeschriftung
Handler = Aktien verkaufen
Skript
wiedergeben
Aktienkurs?
Fällt unter 100,00 EUR <
Steigt über 100,00 EUR
Löst Ereignis
„Kursverfall“ aus
Löst Ereignis „Kursanstieg“
aus

Weitere Informationen finden Sie unter:

- "Verwenden des Ereignishandlers" auf Seite 579
- "TruClient-Dialogfeld "Ereignishandler-Editor"" auf Seite 588

TruClient-Funktionen und -Funktionsbibliotheken

Eine TruClient-Funktion ist eine Gruppe von Schritten, beispielsweise eine Anmeldung, die Sie als Funktion definieren. Funktionen werden in Bibliotheken gespeichert, die Sie mehrmals in einem Skript wiederverwenden können. Jede Bibliothek kann mehrere Funktionen enthalten.

Jede Bibliothek kann entweder lokal oder global verwendet werden. Auf eine lokale Bibliothek können Sie mit dem Skript zugreifen, mit dem sie erstellt wurde. Auf eine globale Bibliothek können Sie mit allen TruClient-Skripts zugreifen. Zudem kann eine globale Bibliothek im Netzwerk gespeichert und von vielen Benutzern gemeinsame verwendet werden.

Weitere Informationen zum Erstellen und Verwenden von Funktionen und Funktionsbibliotheken finden Sie unter "Erstellen und Verwenden von Funktionen und Bibliotheken" auf Seite 586.

Arbeiten mit JavaScript in TruClient-Skripts

Der folgende Abschnitt enthält Hinweise zum Aufzeichnen von TruClient-Skripts.

JavaScript-Unterstützung

Die im Argumentbereich jedes Schritts aufgeführten Argumente basieren alle auf JavaScript und akzeptieren JavaScript-Ausdrücke, die während der Skriptwiedergabe ausgewertet werden. Beachten Sie, dass bei Angabe eines Zeichenfolgewerts Anführungszeichen verwendet werden müssen. Beispiel: City wird als Variable interpretiert, während "City" oder 'City' als Zeichenfolge ausgewertet werden.

Alle anderen Bereiche – Step, Object, Transactions usw. – basieren nicht auf JavaScript, werden nicht als JavaScript ausgewertet und unterstützen keine JavaScript-Ausdrücke. Einzige Ausnahme ist die Objekterkennung mittels JavaScript.

Objekterkennungsvariablen und Schrittvariablen verfügen jedoch nicht über einen gemeinsamen Kontext. Variablen, die in einem Kontext definiert wurden, werden im anderen Kontext nicht erkannt. Soll eine in einem Schritt definierte Variable in der Objekterkennung verwendet werden, fügen Sie dem Variablennamen das Präfix **ArgsContext** hinzu. Wurde die Variable **firstname** beispielsweise als Argument für einen Schritt definiert, kann sie in der Objekterkennung nicht verwendet werden. Um die Variable **firstname** in der Objekterkennung verwenden zu können, bezeichnen Sie die Variable als **ArgsContext.firstname**.

Weitere Informationen zu JavaScript

JavaScript ist die Skriptsprache des Web. JavaScript wird in Millionen von Webseiten verwendet, um zusätzliche Funktionen bereitzustellen, Formulare zu überprüfen, Browser zu erkennen usw. Mithilfe von Suchmaschinen finden Sie im Internet zahlreiche Ressourcen, um sich mit JavaScript vertraut zu machen.

Im Folgenden werden einige Beispiele für Tutorials und Referenzen aufgelistet:

http://www.javascriptkit.com/jsref/

http://www.w3schools.com

http://www.learn-javascript-tutorial.com/

Verwenden regulärer Ausdrücke

Bei der Verwendung regulärer Ausdrücke haben Sie zwei Möglichkeiten:

- Verwenden der Notation '/': Ersetzen Sie die Anführungszeichen einer Zeichenfolge durch einen Schrägstrich.

  Beispiel:

  /LoadRunner/ ist ein regulärer Ausdruck, der mit jeder Zeichenfolge übereinstimmt, die das Wort "LoadRunner" enthält.

- Wenn Sie einen regulären Ausdruck dynamisch erstellen möchten (z. B. mithilfe eines Parameters), können Sie den Konstruktor für reguläre Ausdrücke verwenden und die Zeichenfolge angeben. So würde das Äquivalent des obigen Beispiels **RegExp("LoadRunner")** lauten.

Eine vollständige Liste der unterstützten regulären Ausdrücke finden Sie unter:

https://developer.mozilla.org/en/JavaScript/Reference/Global_Objects/RegExp

Erweitern von TruClient-Skripts

Es gibt eine Reihe über den grundlegenden Workflow hinausgehende Erweiterungen, die Skripts optional hinzugefügt werden können. In dieser Aufgabe werden die Erweiterungen und deren Verwendung beschrieben.

Ändern von Schritten

Ändern Sie Schrittargumente und Objekte, indem Sie den betreffenden Schritt auswählen und die Optionen erweitern. Damit erweitern Sie den Schritt und können die Objekte und Eigenschaften ändern. Eine detaillierte Liste der Schrittstruktur finden Sie unter "TruClient-Toolbox" auf Seite 605.

- Einfügen von Schritten für die Flusssteuerung

  Mit Schleifen werden ausgewählte Teile des Skripts so oft wiederholt wie angegeben oder so lange, bis bestimmte Kriterien erfüllt sind. Um eine Schleife einzufügen, wählen Sie **Toolbox > Flusssteuerung > For-Schleife** aus. Weitere Informationen finden Sie unter " Einfügen und Ändern von Schleifen" auf Seite 581.

- Einfügen von If- oder If-else-Blöcken und Beendigungsschritten

  Um einen Teil des Skripts an Bedingungen zu knüpfen, können Sie If- oder If-else-Blöcke einfügen. Um einen If-Block einzufügen, wählen Sie **Toolbox > Flusssteuerung > If-Block** aus. Um eine Else-Bedingung einzufügen, klicken Sie auf den Link **Else-Aktion hinzufügen** neben dem Titel des If-Schritts. Weitere Informationen finden Sie unter "Argumente von TruClient-Schritten" auf Seite 630.

  Beendigungsschritte können die Iteration oder das gesamte Skript beenden. Sie können mit If-Anweisungen kombiniert werden, um die Beendigung des Skripts oder der Iteration von bestimmten Bedingungen abhängig zu machen. Um einen Beendigungsschritt einzufügen, wählen Sie **Toolbox > Flusssteuerung > Beenden** aus.

Einfügen von Kommentaren

Sie können Kommentare in Ihr Skript einfügen, indem Sie **Toolbox > Verschiedenes** auswählen und das Symbol **Kommentar** an die gewünschte Position ziehen.

Einfügen von Transaktionen

Sie können Transaktionen über den Transaktionseditor hinzufügen. Zum Öffnen des Transaktionseditors klicken Sie auf der Startregisterkarte auf die Schaltfläche für den Transaktionseditor oder verwenden Sie die Tastenkombination Strg + Alt + F7. TruClient-Transaktionen funktionieren aufgrund der asynchronen Eigenschaften der TruClient-Schritte anders als andere Protokolle. Transaktionen werden anhand von Start- und Endschritten sowie von Schrittereignissen definiert. Aufgrund dieser Definition kann das Ende einer Transaktion ausgelöst werden, bevor das wahre Ende eines Schrittes eintritt.

Wenn **Laufzeiteinstellungen > Wiedergeben > Profilerstellung aktivieren** ausgewählt ist, werden clientseitige Aufschlüsselungsinformationen im privaten Browser für jede Transaktion nach der erfolgreichen Wiedergabe des Skripts angezeigt. Weitere Informationen finden Sie unter " Clientseitige Aufschlüsselungsdaten für TruClient" auf Seite 591.

Erstellen von Parametern

Parameter für TruClient-Skripts können wie bei allen anderen Protokollen mit der Standardmethode erstellt werden.

Parameter können referenziert und innerhalb von Schrittargumenten oder JavaScript-Auswertungsschritten erstellt werden.

So parametrieren Sie einen Wert:

1. Wählen Sie ein Feld im Schrittargumentbereich aus.

2. Markieren Sie entweder die gesamte Zeichenfolge oder einen Teil davon.

3. Klicken Sie mit der rechten Maustaste und wählen Sie **Durch einen Parameter ersetzen** aus.

4. Geben Sie im Dialogfeld **Parameternamen eingeben** den Namen eines neuen oder vorhandenen Parameters ein.

Wenn der Parameter in der **Parameterliste** nicht aufgeführt ist, wird ein neuer Parameter mit dem Standardparametertyp **Datei** erstellt. Sie können zur **Parameterliste** wechseln, um dem angegebenen Parameter Werte hinzuzufügen oder vorhandene Werte zu löschen. Der ursprüngliche Wert wird standardmäßig in die Werteliste des neuen Parameters aufgenommen. Details finden Sie unter "Verwenden von bestehenden Parametern" auf Seite 446.

Das folgende Beispiel zeigt, wie Sie Parameter in Skripts verwenden. Diese Zeichenfolge kann als Argument in einem Schritt zur JavaScript-Auswertung verwendet werden.

Bei der Wiedergabe wird der aktuelle Wert des Parameters **paramname** zurückgegeben.

```
LR.getParam("paramname")
```

Bei der Wiedergabe wird dem Parameter **paramname** der Wert **value1** zugewiesen.

```
LR.setParam("paramname", "value1")
```

Nehmen wir z. B. eine AutoVervollständigen-Liste mit verschiedenen Wertegruppen, die von dem eingegebenen Text abhängen.

Der Versuch, eine auf dem Text der Option basierende Listenoption auszuwählen, ist zum Scheitern verurteilt, wenn der eingegebene Text durch einen Parameter definiert wird. Die Optionen ändern sich mit jeder Aktualisierung des Parameterwerts.

In diesen Fällen ist ein Schritt, der eine Ordnungszahl verwendet, besser geeignet.

Weitere Informationen zu Parametern finden Sie unter "Parameter" auf Seite 436.

Einfügen von Fehlererfassungsschritten

Fehlererfassungsschritte sind gruppierte Schritte, deren Inhalt ausgeführt wird, wenn der vorherige Schritt einen Fehler enthält. Zudem wird der Fehler "abgefangen" und nicht zurückgegeben. Sie können Fehlererfassungsschritte definieren, um alle Fehler oder Fehler eines bestimmten Typs abzufangen. Befinden sich zwei Fehlererfassungsschritte in einer Zeile, beziehen sich beide auf denselben Schritt. Um einen Fehlererfassungsschritt einzufügen, wählen Sie **Toolbox > Flusssteuerung > Fehlererfassung** aus.

Prüfen, ob ein Objekt existiert

Um sicherzustellen, dass eine Zeichenfolge oder ein Objekt in der Anwendung existiert, können Sie einen Verifizierungsschritt einfügen:

1. Wählen Sie **Toolbox > Funktionen** aus und ziehen Sie das Symbol **Verifizieren** an die gewünschte Position.

2. Klicken Sie auf das Objekt im Verifizierungsschritt.

3. Wählen Sie das Objekt aus, das Sie verifizieren möchten.

Einfügen allgemeiner Schritte

Sie können einen leeren Schritt einfügen und ihn manuell konfigurieren. Um einen allgemeinen Schritt einzufügen, wählen Sie **Toolbox > Funktionen > Allgemeine Objektaktion/Allgemeine Browseraktion** aus, und ziehen Sie den Schritt an die gewünschte Position. Erweitern Sie anschließend den Schritt und geben die gewünschten Schritteigenschaften ein. Allgemeine Objektaktionen führen eine nicht näher definierte Aktion für ein Objekt aus. Allgemeine Browseraktionen führen eine nicht näher definierte Aktion im Browser wie Vorherige Seite, Aktualisieren, Registerkarte wechseln usw. aus.

Verwenden des Ereignishandlers

In dieser Aufgabe wird beschrieben, wie Sie Ereignisse mit dem globalen Ereignishandler erstellen und verwenden. Eine Abbildung hierzu finden Sie unter "TruClient-Ereignishandler" auf Seite 573.

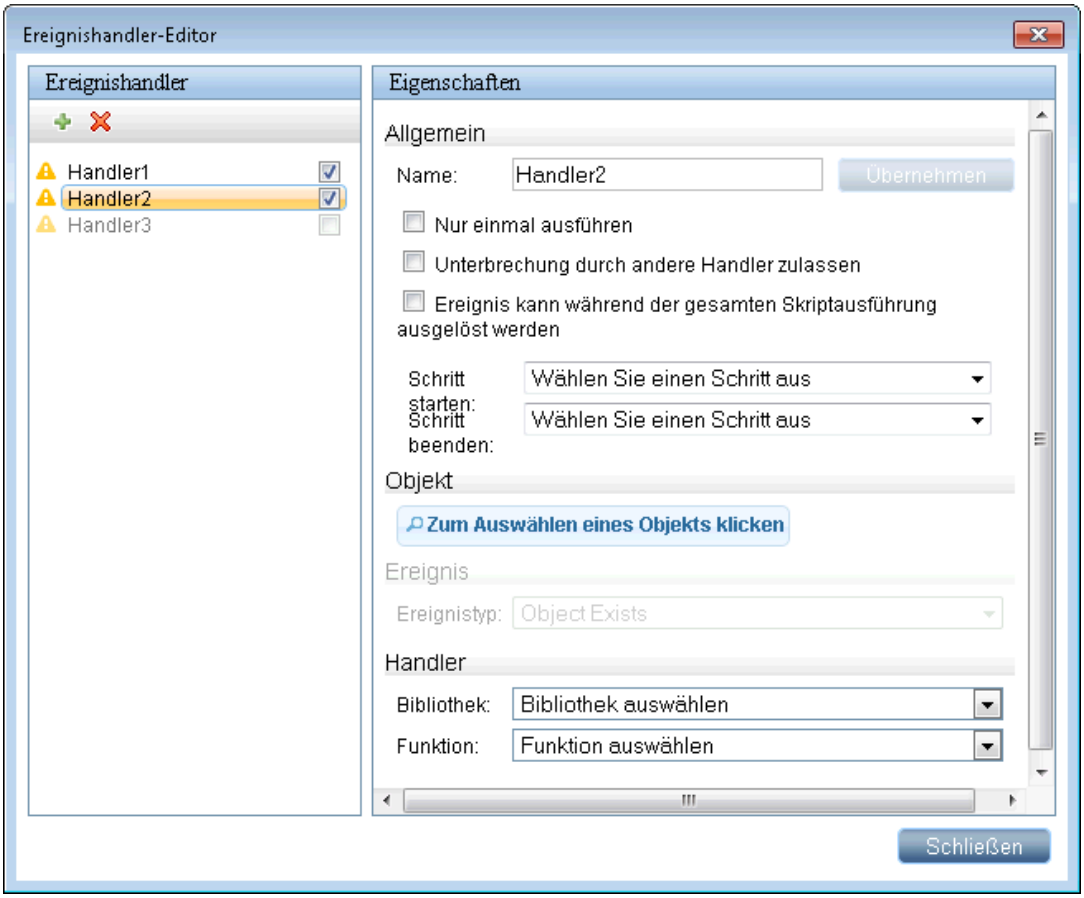
Voraussetzungen

Der globale Ereignishandler führt Funktionen aus, wenn ein bestimmtes Ereignis bei der Skriptwiedergabe auftritt. Bevor Sie Handler erstellen können, müssen Sie die zugehörigen Funktionen erstellen. Weitere Informationen finden Sie unter "Erstellen und Verwenden von Funktionen und Bibliotheken" auf Seite 586.

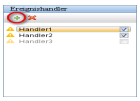
Erstellen eines Ereignishandlers

1. Klicken Sie auf die Schaltfläche in der TruClient-Symbolleiste. Damit öffnen Sie den Ereignishandler-Editor.

.



2. Klicken Sie auf die Schaltfläche im Ereignishandler-Bereich.



3. Definieren Sie die Eigenschaften des Handlers.

   Sie können angeben, ob der Handler zu jeder beliebigen Zeit oder nur zwischen bestimmten Schritten ausgeführt werden soll. Details zu dieser und anderen Eigenschaften finden Sie unter "TruClient-Dialogfeld "Ereignishandler-Editor"" auf Seite 588.

Definieren des mit dem Ereignis verknüpften Objekts

Wählen Sie das Objekt in der Anwendung mit der Schaltfläche aus, um das Ereignis auszulösen.

Wählen Sie z. B. das Textfeld für den ABC-Aktienkurs aus, wenn Sie die Funktion AktienVerkaufen auslösen möchten, sobald der Aktienkurs über 99,99 EUR steigt.

Nachdem Sie das Objekt ausgewählt haben, können Sie auf die Schaltfläche Markieren klicken, um zu bestätigen, dass Sie das richtige Objekt ausgewählt haben. Wenn Sie das falsche Objekt ausgewählt haben, klicken Sie auf die Schaltfläche Ersetzen, um das ausgewählte Objekt durch ein anderes zu ersetzen. Weitere Informationen finden Sie unter "Lösen von Problemen bei der Objekterkennung" auf Seite 560.

Definieren des Ereignisses

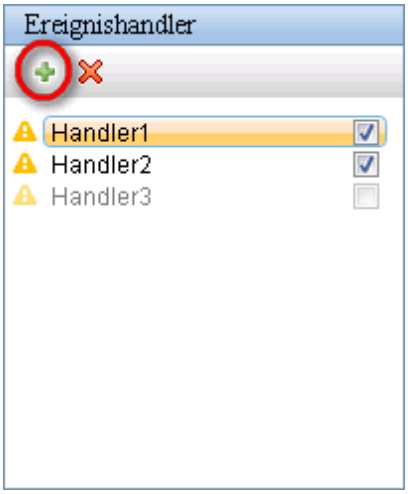
Sie können wählen, ob der Handler ausgeführt wird, wenn das Objekt existiert oder wenn die Eigenschaften des Objekts bestimmte Bedingungen erfüllen. Weitere Informationen finden Sie unter "TruClient-Dialogfeld "Ereignishandler-Editor"" auf Seite 588.

Zuweisen des Handlers zum Ereignis

Wählen Sie die Funktionsbibliothek und die Funktion aus und geben Sie die Argumentwerte an. Weitere Informationen finden Sie unter "TruClient-Dialogfeld "Ereignishandler-Editor"" auf Seite 588.

Aktivieren des Handlers

Um den Handler während der Skriptwiedergabe zu aktivieren, aktivieren Sie das Kontrollkästchen neben dem Handler im Ereignishandlerbereich.



Einfügen und Ändern von Schleifen

Mit Schleifen werden ausgewählte Teile des Skripts so lange wiederholt, bis bestimmte Kriterien erfüllt sind oder die angegebene Anzahl von Iterationen erreicht ist. Sie können Schleifen im Abschnitt **Funktionen** der **Toolbox** eingeben und ändern.

For-Schleifen

Schritte in **For-Schleifen** werden so lange ausgeführt, bis die Endbedingung erfüllt ist oder der Code auf eine Break-Anweisung trifft. Schleifenargumente richten sich nach der JavaScript-Syntax. Um eine For-Schleife einzufügen, wählen Sie **Toolbox > Flusssteuerung > Funktionen > For-Schleife** aus.

Break-Anweisungen

Break-Anweisungen geben an, dass die aktuelle Schleife sofort beendet werden soll. Wird beispielsweise eine Break-Anweisung in der zweiten von fünf Iterationen in einer For-Schleife angetroffen, werden die übrigen Iterationen nicht mehr ausgeführt, sondern die Schleife wird sofort beendet. Um eine Break-Anweisung einzufügen, wählen Sie **Toolbox > Funktionen > Unterbrechen** aus.

Continue-Anweisungen

Continue-Anweisungen geben an, dass die aktuelle Schleifeniteration sofort beendet werden soll. Anschließend wird die Schleifenbedingung überprüft, um festzustellen, ob die gesamte Schleife beendet werden soll. Wird beispielsweise eine Continue-Anweisung in der zweiten von fünf Iterationen in einer For-Schleife angetroffen, wird die zweite Iteration sofort beendet und die dritte Iteration gestartet. Um eine Continue-Anweisung einzufügen, wählen Sie **Toolbox > Funktionen > Flusssteuerung > Fortfahren** aus.

Einfügen von benutzerdefiniertem JavaScript- und C-Code in TruClient-Skripts

In dieser Aufgabe wird beschrieben, wie Sie Code in ein TruClient-Skript einfügen. Sie können den Code in einen schon vorhandenen Schritt als Teil des Schrittarguments einfügen oder Schritte einfügen, die komplett aus externem Code (C oder JavaScript) bestehen.

1. ### Einfügen von Code in einen vorhandenen Schritt

   Sie können JavaScript-Code in vorhandene Schritte in den Argumentfeldern der meisten Schritte einfügen. Auf diese Weise können Sie mehrere Änderungen durchführen.

2. ### Einfügen von Schritten, die vollständig aus Code bestehen

   Sie können Schritte, die gänzlich aus Code bestehen, in Ihr Skript einfügen. Wählen Sie dazu **Toolbox > Verschiedenes** aus und ziehen Sie das Symbol für **Javascript auswerten**, **C auswerten** oder **JS bei Objekt auswerten** an die gewünschte Position. Der Schritt **JS bei Objekt auswerten** führt den JavaScript-Code aus, nachdem das angegebene Objekt geladen wurde. Wir empfehlen, **C auswerten** zu vermeiden und stattdessen JavaScript zu verwenden, wo immer dies möglich ist. Sie können auf dieses Objekt als Variable "object" im JavaScript-Code innerhalb des Schritts verweisen.

   Beispiel:

   Der folgende Code erstellt eine Variable mit der Bezeichnung "amount", die eine Zufallszahl zwischen 1 und 5 generiert. Sie können diese Variable anschließend in den Argumentfeldern anderer Schritte verwenden.

   ```
   var amount=Math.floor(Math.random()*5)+1;
   ```

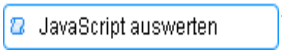
Verwenden von VTS in TruClient

In dieser Aufgabe wird ein Beispiel zur Verwendung von VTS in einem TruClient-Skript beschrieben.

Was ist VTS?

- HP LoadRunner Virtual Table Server, kurz VTS, ist eine webbasierte Anwendung, die mit LoadRunner-Vuser-Skripts ausgeführt werden kann. VTS bietet eine Alternative zur Standardparametrierung in LoadRunner.
- Wenn Sie die Standardparametrierung in LoadRunner verwenden, werden jedem Vuser Parameterwerte aus einer dedizierten Gruppe von Werten zugewiesen. Parameterwerte werden von Vusern nicht gemeinsam genutzt. Im Gegensatz dazu ermöglicht Ihnen VTS das Zuweisen von Parameterwerten aus einer Gruppe von Parameterwerten zu mehreren Vusern. So können Sie ggf. genauer eine Umgebung mit realen Benutzern emulieren.
- VTS enthält eine Tabelle, die Parameterwerte enthält, die von Ihren Vuser-Skripts verwendet werden können. Die VTS-Tabelle besteht aus Spalten und Zeilen. Jede Spalte stellt einen Satz von Werten dar, die von einem bestimmten Parameter in Ihren Vuser-Skripts verwendet werden können. Die Zellen innerhalb einer Spalte enthalten die tatsächlichen Werte des Parameters.

Aufbauen/Trennen der Verbindung zu einem VTS-Server

1. Wählen Sie **Toolbox > Verschiedenes** und fügen Sie einen 

-Schritt zu Ihrem Skript hinzu.
2. Um die Verbindung zu einem VTS herzustellen, geben Sie `TC.vtcConnect (“Server”,Port,"Alias")` im Abschnitt **Argumente > Code** des Schritts ein.

   Sie können den Server durch Angabe des Namens oder der IP-Adresse identifizieren.

   Beispiel: `TC.vtcConnect("myServer",8888,"myVTS");`
3. Um die Verbindung zu einem VTS zu trennen, geben Sie TC.vtcDisconnect("alias")> im Abschnitt **Argumente > Code** des Schritts ein.

Tipps:

- Spielen Sie vor der Wiedergabe des Skripts den Schritt `TC.vtcConnect` (aus dem Schritt heraus) ab, um sicherzustellen, dass eine Verbindung zum VTS-Server besteht.
- Nachdem Sie eine Verbindung mit dem VTS-Server hergestellt haben, können Sie die anderen VTS-API-Schritte wiedergeben, sie ändern und erneut wiedergeben.

- Durch Wiedergabe des gesamten Skripts wird die Verbindung zum VTS automatisch getrennt, auch wenn dieser Schritt nicht im Skript enthalten ist.

Beispiele für Aufbauen/Trennen der Serververbindung:

- Unter Verwendung von Servername und Alias

```
TC.vtcConnect(“MeinServer”,8888,”BenutzertabelleVts”);
TC.vtcConnect(”BenutzertabelleVts”);
```

Weitere Informationen zur Verwendung eines Alias finden Sie unter "Verwenden von VTS in TruClient" auf der vorherigen Seite.

- Unter Verwendung von Server-IP-Adresse und Alias

```
TC.vtcConnect(“192.168.1.133”,8888,”MitarbeitertabelleVts”);
TC.vtcConnect("MitarbeitertabelleVts”);
```

- Unter Verwendung von Servername ohne Alias

```
TC.vtcConnect(“MeinServer”,8888);
TC.vtcDiscconect();
```

- Unter Verwendung von Server-IP-Adresse ohne Alias

```
TC.vtcConnect(“192.168.1.133”,8888);
TC.vtcDiscconect();
```

Warum brauchen Sie einen Alias für den VTS-Server?

Wenn Sie mit mehreren Servern arbeiten, können Sie durch die Verwendung von Spitznamen deren Identifizierung innerhalb des Codes vereinfachen.

Beispiele:

In der folgenden Verbindungsanweisung ist "BenutzertabelleVts" der Alias für "MeinServer".

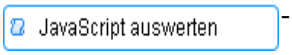
```
TC.vtcConnect(“MeinServer”,8888,”BenutzertabelleVts”);
```

Im folgenden Beispiele wird eine Verbindung zu zwei VTS-Servern hergestellt, wobei die Verbindung zum ersten Server über den zugehörigen Alias getrennt wird:

```
TC. vtcConnect("MeinServer", 8888, "BenutzertabelleVts");
TC.vtcConnect("MeinServer1",8888, "SkriptdatenVts");
TC.vtcDisconnect(”BenutzertabelleVts”);
```

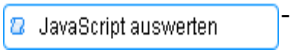
Verwenden eines JavaScript-Code-Auswertungsschritts zur Übergabe von VTS-Parametern an einen anderen Schritt.

Sie können den Java Script-Code-Auswertungsschritt zur Übergabe von VTS-Parametern an einen anderen Schritt in Ihrem Skript einsetzen. Das nachstehende Beispiel zeigt, wie Sie einen URL parametrieren.

1. Herstellen einer Verbindung zum VTS-Server
   a. Wählen Sie **Toolbox > Verschiedenes** und fügen Sie einen 

   -Schritt zu Ihrem Skript hinzu.

   b. Um die Verbindung zu einem VTS herzustellen, geben Sie `TC.vtcConnect (“Server”,Port,"Alias")` im Abschnitt **Argumente > Code** des Schritts ein.

2. Erstellen Sie einen Schritt, der eine Variable enthält.
   a. Wählen Sie **Toolbox > Verschiedenes** und fügen Sie einen 

   -Schritt zu Ihrem Skript hinzu.

   b. Definieren Sie Ihre Variable im Abschnitt **Argument > Code**.

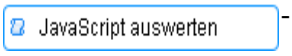
   Wenn Sie z. B. Daten vom Server abrufen möchten, geben Sie `var ws = TC.vtcGetCell ("Web Sites",1,"alias");` ein.

3. Wählen Sie **Toolbox > Funktionen** und fügen Sie eine **Allgemeine Objektaktion** in Ihr Skript ein.

4. Geben Sie die Variable ein, die Sie in Schritt 2 unter **Argument > Speicherort** definiert haben.

Verwenden einer VTS -API-Funktion unmittelbar in einem Schritt.

Sie können den Java Script-Code-Auswertungsschritt einsetzen, um eine VTS-API in Ihrem Skript zu implementieren. Das nachstehende Beispiel zeigt, wie Sie einen URL parametrieren.

1. Herstellen einer Verbindung zum VTS-Server
   a. Wählen Sie **Toolbox > Verschiedenes** und fügen Sie einen 

   -Schritt zu Ihrem Skript hinzu.

   b. Um die Verbindung zu einem VTS herzustellen, geben Sie `TC.vtcConnect (“Server”,Port,"Alias")` im Abschnitt **Argumente > Code** des Schritts ein.

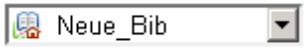
2. Wählen Sie **Toolbox > Funktionen** und fügen Sie eine **Allgemeine Objektaktion** in Ihr Skript ein.

3. Geben Sie `TC.vtcGetCell("Web Sites",1,"alias");` im Abschnitt **Argument > Speicherort** des Skripts ein.

Erstellen und Verwenden von Funktionen und Bibliotheken

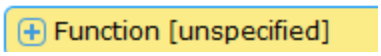
In dieser Aufgabe wird beschrieben, wie Sie Funktionsbibliotheken und Funktionen in TruClient-Skripts erstellen und nutzen.

Erstellen einer Funktionsbibliothek und einer Funktion aus der TruClient-Seitenleiste

1. Wählen Sie **Funktionen** aus der Dropdownliste auf der Registerkarte "TruClient".
2. Wählen Sie in der Dropdownliste 

eine vorhandene Bibliothek aus

   oder klicken Sie auf die Schaltfläche in der Funktionssymbolleiste, um eine neue Bibliothek zu erstellen.
3. Klicken Sie auf die Schaltfläche , um eine neue Funktion zu erstellen. Mit diesem Schritt fügen Sie eine neue, nicht definierte Funktion ein.

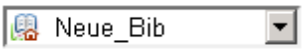
4. Klicken Sie auf die Schaltfläche , um die Funktion zu erweitern.
5. Definieren Sie den Schritt einschließlich Funktionsname und Endereignis.
6. Definieren Sie die Funktionsargumente mit dem **Argumenteditor**. Argumentnamen sollten so aussagefähig sein, dass Sie bei Verwendung der Funktion nicht raten müssen, welchen Wert das Argument erwartet.
7. Definieren Sie die Transaktion mit dem **Transaktionseditor**. Details finden Sie unter "Dialogfeld 'Transaktionseditor' (TruClient)" auf Seite 594.

Erstellen einer Funktion in einem Skript

1. Markieren Sie die Schritte im Skript, die Sie in die Funktion einbeziehen möchten. Zur Auswahl mehrerer Schritte halten Sie die Taste STRG gedrückt.
2. Klicken Sie mit der rechten Maustaste auf einen markierten Schritt und wählen Sie **Gruppieren in > Funktion** aus, um das Dialogfeld **Neue Funktion erstellen** zu öffnen. Weitere Informationen finden Sie unter " Dialogfeld "Neue Funktion erstellen"" auf Seite 593.
3. Um Schritte zuzuweisen, die Argumente enthalten, erweitern Sie den Argumentbereich und fügen die Funktion `FuncArgs.<Argumentname>` ein. .

   Tipp: Um JavaScript als Argumentwert einzufügen, klicken Sie auf die Schaltfläche , die den JavaScript-Editor öffnet.

Erstellen einer globalen Bibliothek

1. Wählen Sie **Funktionsbibliotheken** aus der Dropdownliste in der **TruClient-Seitenleiste**.
2. Wählen Sie eine vorhandene Bibliothek aus der Dropdownliste 

in der **TruClient-Seitenleiste**.
3. Klicken Sie auf die Schaltfläche , um die Bibliothek als XML-Datei in ein Dateiverzeichnis zu exportieren.

Verwenden von Kopieren/Einfügen in globalen Bibliotheken

Durch das Einfügen von Schritten, die Objekte aus einem lokalen Skript oder einer lokalen Bibliothek enthalten, in eine globale Bibliothek entstehen separate Testobjekte, die sich dasselbe Element der Benutzeroberfläche beziehen. Wenn Sie eines der Testobjekte in der globalen Bibliothek oder im Skript aktualisieren, müssen Sie das jeweils andere Testobjekt, das sich auf dasselbe Benutzeroberflächenelement bezieht, manuell anpassen.

Beispiele:

1. Die globale Bibliothek "GlobLib" enthält Schritt A mit einem Testobjekt "Foo", das sich auf die Anwendung bezieht.

   Skript enthält ebenfalls einen Schritt B mit dem Testobjekt "Foo", das sich auf die Anwendung bezieht.

   Wenn Sie Schritt B kopieren und in "GlobLib" einfügen, enthält die Bibliothek zwei unterschiedliche Testobjekte, die sich auf "Foo" beziehen. Wenn Sie also eines von ihnen ändern oder aktualisieren, sollten Sie das andere manuell ändern.

2. Die globale Bibliothek "GlobLib" enthält Schritt A mit einem Testobjekt "Foo", das sich auf die Anwendung bezieht.

   Das Skript enthält die Schritte B und C jeweils mit einem Testobjekt "Foo", das sich auf die Anwendung bezieht. Wenn Sie die Schritte B und C in die "GlobLib" kopieren, gibt es zwei verschiedene Testobjekte in "GlobLib", die sich auf "Foo" beziehen.

Hinweis:

- Beim Bearbeiten einer globalen Bibliothek können Sie Änderungen speichern, indem Sie auf die Schaltfläche klicken.
- Wenn Sie die Bibliothek in einem Netzwerkverzeichnis speichern, können andere Benutzer sie mit einem Klick auf die Schaltfläche importieren.
- Klicken Sie auf die Schaltfläche , um die Bibliothek vom globalen Modus zu trennen. Änderungen, die Sie im lokalen Modus durchführen, werden im Skript gespeichert.

Verwenden einer Funktion im Skript

1. Klicken Sie auf die TruClient-Registerkarte **Toolbox**.
2. Wählen Sie die Registerkarte **Funktionen** aus.
3. Ziehen Sie **Funktion aufrufen** an die richtige Position im Skript.
4. Klicken Sie auf den Link **Funktion aufrufen**, um die Funktion zu erweitern.
5. Geben Sie die Bibliothek und die Funktion im Schrittabschnitt an.
6. Geben Sie die Argumentwerte im Argumentabschnitt an. Weitere Informationen finden Sie unter "Erstellen einer Funktion in einem Skript" auf Seite 586. (Optional)
7. Geben Sie Transaktionen im Transaktionsabschnitt an. (Optional) Weitere Informationen finden Sie unter "Dialogfeld 'Transaktionseditor' (TruClient)" auf Seite 594.

Anzeigen clientseitiger Aufschlüsselungsdaten für TruClient-Skripts

In dieser Aufgabe wird beschrieben, wie Sie die clientseitigen Aufschlüsselungsdaten anzeigen können, die Transaktionen in einem TruClient-Skript zugeordnet sind.

1. **Profilerstellung aktivieren**

   Wählen Sie **Laufzeiteinstellungen > Wiedergeben > Profilerstellung aktivieren**.

2. **Erstellen und Aufzeichnen eines FireFox TruClient-Skripts**

   Weitere Informationen finden Sie unter "Entwickeln von TruClient-Skripts" auf Seite 549.

3. **Einfügen von Transaktionen**

   Sie können Transaktionen über den Transaktionseditor hinzufügen. Zum Öffnen des Transaktionseditors klicken Sie auf der Startregisterkarte auf die Schaltfläche für den Transaktionseditor oder verwenden Sie die Tastenkombination Strg + Alt + F7."Dialogfeld 'Transaktionseditor' (TruClient)" auf Seite 594

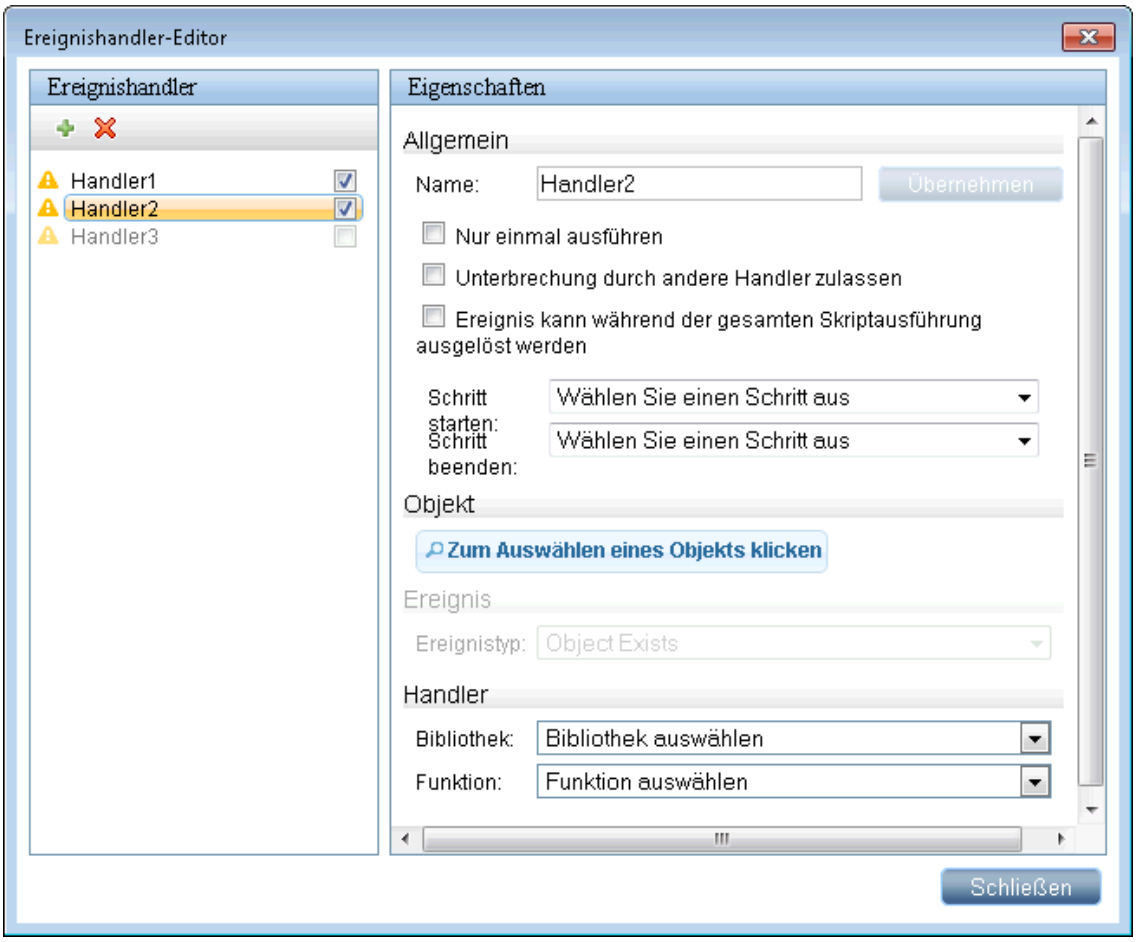
4. Wiedergeben des Skripts

Ergebnisse

Ein Browserfenster mit den clientseitigen Aufschlüsselungsdaten wird geöffnet. Weitere Informationen finden Sie unter " Clientseitige Aufschlüsselungsdaten für TruClient" auf Seite 591

TruClient-Dialogfeld "Ereignishandler-Editor"

In diesem Dialogfeld können Sie Ereignishandler und die zugehörigen Eigenschaften definieren.



Zugriff	Klicken Sie auf die Schaltfläche in der TruClient-Seitenleiste.
Wichtige Informationen	Ereignishandler können Ereignisse verarbeiten, die zu einem beliebigen Zeitpunkt während der Skriptwiedergabe auftreten.
Relevante Aufgaben	"Verwenden des Ereignishandlers" auf Seite 579 "Erstellen und Verwenden von Funktionen und Bibliotheken" auf Seite 586
Siehe auch	"TruClient-Ereignishandler" auf Seite 573

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Ereignishandlerbereich	**Ereignishandler hinzufügen**. Fügt einem Ereignishandler ein Ereignis hinzu. **Ereignishandler löschen**. Löscht ein Ereignis aus dem Ereignishandler.

Element der Oberfläche	Beschreibung
Eigenschaften	**Allgemein** • **Name.** Hier können Sie dem Ereignis einen Namen geben. • **Nur einmal ausführen.** Führt das Ereignis einmal aus. • **Unterbrechung durch andere Handler zulassen.** Aktivieren Sie das Kontrollkästchen, damit andere Ereignisse während der Ausführung dieses Ereignisses möglich sind. Sie können eine der folgenden Optionen auswählen: • **Ereignis kann während der gesamten Skriptausführung ausgelöst werden**. Ereignis kann während der gesamten Skriptausführung ausgelöst werden. oder • **Schritt starten.** Ermöglicht die Auswahl eines Startschrittes, bei dem der Ereignishandler ausgelöst werden kann. • **Schritt beenden.** Ermöglicht die Auswahl eines Endschrittes, nach dem der Ereignishandler ausgelöst werden kann. **Objekt** • Zum Auswählen eines Objekts klicken Ermöglicht die Auswahl eines Objekts aus der Anwendung, das den Ereignishandler auslöst. • **Rollen**. Zeigt die Rollen an, mit denen die möglichen Vorgänge für das Objekt bestimmt werden. Für die Rolle "Element" etwa sind Vorgänge wie "klicken" und "Mauszeiger darüber bewegen" möglich. Bei der Rolle "Textfeld" ist außerdem der Vorgang "eingeben" verfügbar. Andere Rollen: Element, Fokussierbarkeit, Textfeld, Kontrollkästchen, Dateifeld, Optionsfeld, Listenfeld, Mehrfach-Listenfeld, Schieberegler, Datumsauswahl, Video, Audio, Browser. • **Name.** Ermöglicht Ihnen, den Namen des Objekts anzugeben. • **ID-Methode** **Automatisch (empfohlen)**. Ermöglicht TruClient die Auflösung der Objektidentifikation.

Element der Oberfläche	Beschreibung
	XPath. Ermöglicht Ihnen das Einfügen einer XPath-Abfrage, um das Objekt zu identifizieren. **JavaScript**. Ermöglicht Ihnen das Einfügen von JavaScript-Code, um das Objekt zu identifizieren. • **Zugehörige Objekte.** Ermöglicht die Zuordnung eines Objekts zu einem anderen Objekt in der Anwendung, um die Objektidentifikation während der Wiedergabe zu vereinfachen. **Ereignis** • **Ereignistyp** Sie können einen von zwei Ereignistypen auswählen: **Objekt ist vorhanden**. Wenn das Objekt während der Wiedergabe vorhanden ist, wird der Ereignishandler ausgelöst. **Eigenschaft für Objekt ist vorhanden.** Wenn die Eigenschaft eines Objekts mit den definierten Kriterien übereinstimmt, wird der Ereignishandler ausgelöst. **Handler** • **Bibliothek**. Hiermit können Sie die Bibliothek mit der Funktion auswählen. • **Funktion**. Hiermit können Sie die Funktion auswählen. • **Argumente**. Hiermit können Sie die Argumentwerte angeben.

Clientseitige Aufschlüsselungsdaten für TruClient

Im Browserfenster werden clientseitige Aufschlüsselungsdaten im interaktiven Modus für jede Transaktion in Ihrem Skript angezeigt.

Wichtige Informationen	• Wählen Sie vor der Wiedergabe des Skripts **Laufzeiteinstellungen > Wiedergeben > Profilerstellung aktivieren**. • Wenn Sie ein Controller-Szenario mit Ihrem Skript erstellen, können Sie die clientseitigen Aufschlüsselungsdaten in Analysis anzeigen. Weitere Informationen finden Sie unter Client Side Breakdown (Over Time) Graph oder Client Side Java Script Breakdown (Over Time) Graph.
Relevante Aufgaben	" Anzeigen clientseitiger Aufschlüsselungsdaten für TruClient-Skripts " auf Seite 588

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Clientseitige Aufschlüsselung

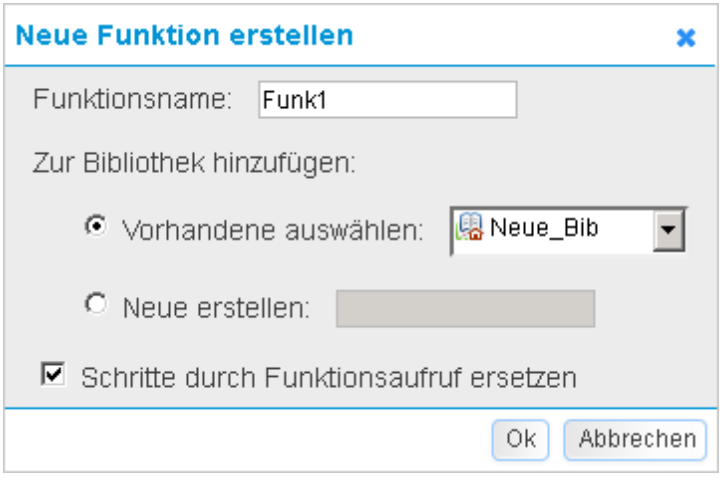
Element der Oberfläche	Beschreibung
JS-Zeit	**Gemessene JavaScript-Laufzeit** Die folgenden Datenelemente werden für jede Transaktion zurückgegeben: **Name.** Der Funktionsname. **Datei.** Der Name der JS-Datei, in der die Funktion vorhanden ist. **Aufrufe.** Protokoll des Ausführungsstapels und Drilldown zu den Funktionsaufrufen während der Transaktion. **Gesamtausführungszeit.** Verstrichene Zeit nach jedem Funktionsaufruf, einschließlich darin enthaltener Funktionsaufrufe. **Eigene Gesamtausführungszeit.** Verstrichene Zeit nach jedem Funktionsaufruf, ausschließlich darin enthaltener Funktionsaufrufe. **Max. Rekursionstiefe.** Die maximale Tiefe des Aufrufstapels der Funktion. **Min. Ausführungszeit.** Die minimale Laufzeit der Funktion, einschließlich darin enthaltener Funktionsaufrufe. **Max. Ausführungszeit.** Die maximale Laufzeit der Funktion, einschließlich darin enthaltener Funktionsaufrufe. **Min. eigene Ausführungszeit.** Die minimale Laufzeit der Funktion, ausschließlich darin enthaltener Funktionsaufrufe. **Max. eigene Ausführungszeit.** Die maximale Laufzeit der Funktion, ausschließlich darin enthaltener Funktionsaufrufe.

Clientseitige AufschlüsselungForts.

Netzwerkzeit	**Gemessene Netzwerkaktivität** Die folgenden Datenelemente werden zurückgegeben: **URL.** Der URL der Seitenressource. **Startzeit.** Startzeit der Transaktion im Netzwerk. **Endzeit.** Endzeit der Transaktion im Netzwerk. **Dauer.** Gesamtdauer der Transaktion im Netzwerk.
Wartezeit	**Wartezeit während der Transaktion** Die Wartezeit wird als Takt + minimale Zeit + Warteschritt definiert. Die folgenden Datenelemente werden für jede Transaktion zurückgegeben: **Start.** Startzeit der Wartezeit. **Type.** Entweder Takt, minimale Zeit oder Warteschritt. **Ende.** Endzeit der Wartezeit. **Dauer.** Gesamtdauer der Wartezeit in Millisekunden.
Gesamtzeit	Gesamttransaktionszeit (in Millisekunden) = JS + Netzwerk + Warten **Hinweis:** JS + Netzwerk oder Netzwerk + Warten können sich überlappen, daher entspricht die Gesamtzeit möglicherweise nicht der Summe aus JS + Netzwerk + Warten.

Dialogfeld "Neue Funktion erstellen"

In diesem Dialogfeld können Sie eine neue Funktion erstellen und die Funktion einer Bibliothek zuweisen.



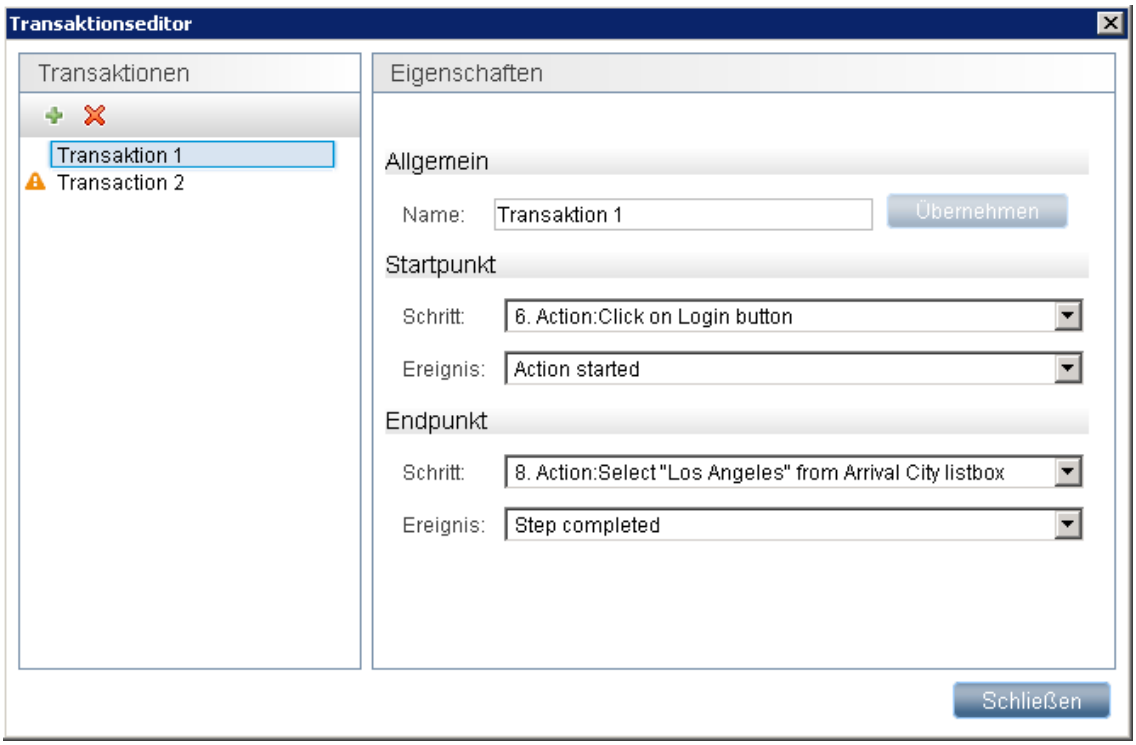
Zugriff	Klicken Sie mit der rechten Maustaste auf einen markierten Schritt im Skript und wählen Sie dann **Gruppieren in > Neue Funktion**.
Relevante Aufgaben	"Erstellen und Verwenden von Funktionen und Bibliotheken" auf Seite 586

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Funktionsname	Ermöglicht das Angeben eines Funktionsnamens.
Zur Bibliothek hinzufügen	• **Vorhandene auswählen.** Speichert eine neue Funktion in einer vorhandenen Bibliothek. • **Neue erstellen.** Speichert eine neue Funktion in einer neuen Bibliothek.
Schritte durch Funktionsaufruf ersetzen	Bei Auswahl dieser Funktion fügt TruClient automatisch eine Aufruffunktion anstelle von Schritten hinzu.

Dialogfeld 'Transaktionseditor' (TruClient)

In diesem Dialogfeld können Sie Transaktionen in TruClient-Vuser-Skripts verwalten.



Zugriff	Klicken Sie auf in der **TruClient-Seitenleiste**.

Wichtige Informationen	Wenn Sie einen Warteschritt innerhalb einer Transaktion in einem TruClient-Skript hinzufügen, werden im Protokoll Details zur Latenzzeit angegeben, die "Wartezeit + Takt + Max(Mindestzeit)" entspricht.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Fügt eine neue Transaktion hinzu oder löscht die ausgewählte Transaktion.
Allgemein	Ermöglicht Ihnen, den Namen der Transaktion zu ändern.

Startpunkt	Der Schritt und das Ereignis, das den Beginn der Transaktion markiert. Ein Ereignis kann Folgendes sein: • **Schritt gestartet.** Das erste Ereignis, das ausgelöst wird, wenn ein Schritt ausgeführt wird. Das Ereignis, das ausgelöst wird, bevor die eigentliche Aktion des Schritts ausgeführt wird und eine andere Aktivität stattfindet. • **Aktion gestartet.** Dieses Ereignis wird unmittelbar nach dem**Schritt gestartet**-Ereignis ausgelöst und bedeutet, dass die auszuführende Aktion gestartet wurde. Beispiel: Navigation zu einem URL oder das Klicken auf eine Schaltfläche. • **Aktion abgeschlossen.** Dieses Ereignis wird ausgelöst wird, wenn die auszuführende Aktion abgeschlossen ist. Der Schritt wurde möglicherweise jedoch noch nicht beendet. Für Schritte im Zusammenhang mit der Anwendung stehen möglicherweise weitere zusätzliche Netzwerk- oder DOM-Aktivitäten an. • **Step synchronous network completed.** Schritt endet, wenn alle HTTP-Anforderungen abgeschlossen sind, außer Anforderungen in Bezug auf offene Verbindungen, die nicht relevant für den Schritt sind. Normalerweise werden diese Anforderungen mithilfe von XMLHttpRequest ausgelöst. • **DOM content loaded.** Schritt endet, wenn das DOM (Document Object Model) bereit ist. Dies bedeutet, dass die API für die Interaktion mit Inhalt, Stil und Struktur einer Seite bereit ist, Anforderungen durch den clientseitigen Code Ihrer Anwendung zu empfangen. • **Laden des Dokuments.** Schritt endet, wenn das Laden eines Dokuments abgeschlossen ist. Dies bedeutet, dass alle Skripts und Stylesheets geladen und ausgeführt wurden und alle Bilder heruntergeladen und angezeigt wurden. • **Dialog opened.** Schritt endet, wenn ein Dialogfeld geöffnet wird. • **Schritt abgeschlossen.** Dieses Ereignis wird ausgelöst, sobald die Aktivität im Schritt-Container abgeschlossen ist. Es wird nur ausgelöst, wenn Schritt erfolgreich abgeschlossen wurde. • **After step ended.** Das letzte Ereignis in der Ereigniskette. Es folgt auf **Step completed**, wird jedoch immer ausgelöst, egal ob der Schritt erfolgreich abgeschlossen wurde oder nicht.

Endpunkt	Der Schritt und das Ereignis, die das Ende der Transaktion angeben. Ein Ereignis kann Folgendes sein: • **Schritt gestartet.** Das erste Ereignis, das ausgelöst wird, wenn ein Schritt ausgeführt wird. Das Ereignis, das ausgelöst wird, bevor die eigentliche Aktion des Schritts ausgeführt wird und eine andere Aktivität stattfindet. • **Aktion gestartet.** Dieses Ereignis wird unmittelbar nach dem**Schritt gestartet**-Ereignis ausgelöst und bedeutet, dass die auszuführende Aktion gestartet wurde. Beispiel: Navigation zu einem URL oder das Klicken auf eine Schaltfläche. • **Aktion abgeschlossen.** Dieses Ereignis wird ausgelöst wird, wenn die auszuführende Aktion abgeschlossen ist. Der Schritt wurde möglicherweise jedoch noch nicht beendet. Für Schritte im Zusammenhang mit der Anwendung stehen möglicherweise weitere zusätzliche Netzwerk- oder DOM-Aktivitäten an. • **Step synchronous network completed.** Schritt endet, wenn alle HTTP-Anforderungen abgeschlossen sind, außer Anforderungen in Bezug auf offene Verbindungen, die nicht relevant für den Schritt sind. Normalerweise werden diese Anforderungen mithilfe von XMLHttpRequest ausgelöst. • **DOM content loaded.** Schritt endet, wenn das DOM (Document Object Model) bereit ist. Dies bedeutet, dass die API für die Interaktion mit Inhalt, Stil und Struktur einer Seite bereit ist, Anforderungen durch den clientseitigen Code Ihrer Anwendung zu empfangen. • **Laden des Dokuments.** Schritt endet, wenn das Laden eines Dokuments abgeschlossen ist. Dies bedeutet, dass alle Skripts und Stylesheets geladen und ausgeführt wurden und alle Bilder heruntergeladen und angezeigt wurden. • **Dialog opened.** Schritt endet, wenn ein Dialogfeld geöffnet wird. • **Schritt abgeschlossen.** Dieses Ereignis wird ausgelöst, sobald die Aktivität im Schritt-Container abgeschlossen ist. Es wird nur ausgelöst, wenn Schritt erfolgreich abgeschlossen wurde. • **After step ended.** Das letzte Ereignis in der Ereigniskette. Es folgt auf **Step completed**, wird jedoch immer ausgelöst, egal ob der Schritt erfolgreich abgeschlossen wurde oder nicht.

TruClient-Seitenleiste

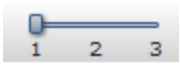
TruClient-Startregisterkarte

Mit dieser Registerkarte können Sie den Ablauf des Aufzeichnungsprozesses für TruClient-Skripts steuern.

Aktionen

Zugriff	Wählen Sie in der Dropdownliste der TruClient-Seitenleiste die Option **Action** aus.
Relevante Aufgaben	"Struktur von TruClient-Schritten" auf Seite 540

Nachfolgend werden die Elemente der Oberfläche beschrieben:

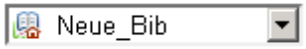
Element der Oberfläche	Beschreibung
	Aufzeichnung starten. Startet die Skriptaufzeichnung. Mit dem Pfeil können Sie angeben, ob vor oder nach dem ausgewählten Schritt aufgezeichnet werden soll.
	Wiedergeben. Gibt das Skript wieder. Sie können auch den Pfeil verwenden, um anzugeben, ob nur der ausgewählte Schritt oder das Skript Schritt für Schritt ausgeführt werden soll. Beim schrittweisen Ausführen wird die Wiedergabe nach jedem Schritt angehalten. Weitere Informationen finden Sie unter "Debuggen von TruClient-Skripts" auf Seite 558.
	Anhalten. Beendet das Aufzeichnen oder Wiedergeben des Skripts.
	Haltepunkt umschalten. Schaltet Haltepunkte für den ausgewählten Schritt ein oder aus.
	**Skriptebene.** Ändert die Skriptebenen, die im Skript angezeigt und abgespielt werden. Weitere Informationen finden Sie unter "TruClient-Skriptebenen" auf Seite 555.
	Transaktion starten/beenden. Fügt einen Start- oder Endpunkt für eine Transaktion ein.
	Transaktionseditor. Der Transaktionseditor wird geöffnet. Dort können Sie neue Transaktionen definieren und vorhandene ändern.

	Speichern. Speichert das Skript.
	Allgemeine Einstellungen. Öffnet das Dialogfeld **Allgemeine Einstellungen**. Weitere Informationen finden Sie unter "Dialogfeld "TruClient - Allgemeine Einstellungen"" auf Seite 609.
	Snapshotansicht. Zeigt die Snapshotansicht im rechten Bereich. Fügt die folgenden Dateien zum rechten Bereich hinzu: • **Snapshot-Typ.** Mit diesen Schaltflächen können Sie Snapshots anzeigen, die in verschiedenen Modi erstellt wurden: ▪ **Aufzeichnen.** Zeigt Snapshots an, die während der Aufzeichnung für einen bestimmten Schritt erstellt wurden. ▪ **Interaktive Wiedergabe.** Zeigt Snapshots an, die während der interaktiven Aufzeichnung für einen bestimmten Schritt erstellt wurden. ▪ **Lastmoduswiedergabe.** Zeigt Snapshots an, die während der Lastmoduswiedergabe für einen bestimmten Schritt erstellt wurden. ▪ **Diashow abspielen.** Zeigt Snapshots als Diashow an. ▪ **Diashow anhalten.** • **Ansicht.** Sie können aus folgenden Ansichten auswählen: ▪ **Einzeln.** Zeigt den Snapshot für einen einzelnen Schritt an. ▪ **Vergleichen.** Teilt den Bildschirm, damit Sie Snapshots aus verschiedenen Modi vergleichen können. Verwenden Sie die Schaltflächen für die Snapshotansicht in jedem Bereich, um die gewünschten Snapshots auszuwählen. Klicken Sie auf die Schaltfläche **Bildlauf synchronisieren**, um den Bildlauf zwischen den Bereichen zu synchronisieren. Durch das Fehlersymbol wird angegeben, dass der Snapshot für den Schritt nicht aktuell ist. ▪ **Miniaturansichten.** Zeigt die Snapshots als Miniaturansicht an. • **Zurück/Weiter.** Navigiert zum Screenshot für den vorherigen oder nächsten Schritt. Der entsprechende Schritt wird im Skript hervorgehoben.
	Öffnet das Feld des Ereignishandler-Editors. Weitere Informationen finden Sie unter "TruClient-Dialogfeld "Ereignishandler-Editor"" auf Seite 588.

Funktionen

Zugriff	Wählen Sie in der Dropdownliste der **TruClient-Seitenleiste** die Option **Funktionen** aus.
Relevante Aufgaben	"Erstellen und Verwenden von Funktionen und Bibliotheken" auf Seite 586

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Hiermit können Sie die aktive Bibliothek auswählen.
	Neue Bibliothek. Ermöglicht das Erstellen einer neuen Bibliothek.
	Bibliothek importieren. Importiert eine Funktionsbibliothekn aus einer XML-Datei.
	Bibliothek exportieren. Exportiert eine Funktionsbibliothek in eine XML-Datei.
	Bibliothek löschen. Löscht eine Funktionsbibliothek.
	Neue Funktion. Ermöglicht das Erstellen einer neuen Funktion. Weitere Informationen finden Sie unter "Erstellen und Verwenden von Funktionen und Bibliotheken" auf Seite 586.

Kontextmenü

Zugriff	Wählen Sie einen Schritt aus, und klicken Sie mit der rechten Maustaste, um das Kontextmenü anzuzeigen.
Relevante Aufgaben	"Struktur von TruClient-Schritten" auf Seite 540

Element der Oberfläche	Beschreibung
<Wiedergabeaktionen>	**Diesen Schritt wiedergeben.** Nur der ausgewählte Schritt wird wiedergegeben. **Ab diesem Schritt wiedergeben.** Das Skript wird ab dem ausgewählten Schritt wiedergegeben.

<Aufzeichnen>	• **Vor Schritt.** Fügt die nächste Reihe aufgezeichneter Schritte vor dem ausgewählten Schritt ein. • **In Schritt.** Fügt die nächste Reihe aufgezeichneter Schritte in die ausgewählte Gruppe ein. • **Nach Schritt.** Fügt die nächste Reihe aufgezeichneter Schritte nach dem ausgewählten Schritt ein.
Haltepunkt umschalten	Fügen Sie einen Haltepunkt zum Umschalten ein bzw. entfernen diesen.
<Transaktionsschritte>	• **Transaktion starten.** Fügt einen Schritt **Transaktion starten** in das Skript ein. • **Mit Transaktion umgeben.** Fügt die Schritte **Transaktion starten** und **Transaktion beenden** um einen einzelnen oder mehrere Schritte in die Skripts ein. • **Transaktion beenden.** Fügt einen Schritt **Transaktion beenden** in Ihr Skript ein.
<Aktionen gruppieren>	**Schritte gruppieren** Mehrere Schritte werden in einem einzelnen Schritt gruppiert. **Gruppieren in** Mehrere Schritte werden in folgende Gruppen gruppiert: • **If-Klausel.** Eine logische Struktur, mit der der Ablauf Ihres Skripts gesteuert wird. • **For-Schleife-Klausel**. Eine logische Struktur, bei der die in der Schleife enthaltenen Schritte so lange wiederholt werden, bis die angegebene Anzahl von Wiederholungen erreicht ist. • **Neue Funktion.** Eine Gruppe von Schritten, wie z. B. eine Anmeldung, die Sie als Funktion definieren.
<Aktion bearbeiten>	• **Schneidet aus.** Schneidet den ausgewählten Schritt aus dem Skript aus. • **Kopiert.** Kopiert den ausgewählten Schritt im Skript. • **Fügt ein.** Fügt den ausgewählten Schritt in das Skript ein.
Entf	Löscht einen Schritt aus dem Skript.
Deaktivieren/Aktivieren	Wechselt zwischen dem Deaktivieren oder Aktivieren eines Schritts während der Wiedergabe.

Schritt bearbeiten	Erweitert den Schritt, um Schritt-, Argument- und Transaktionseigenschaften anzuzeigen.
Anzeigen	**Schrittnummern.** Zeigt die Schrittnummern an. **Animationen.** Aktiviert/deaktiviert Schrittanimationen in der TruClient-Seitenleiste.

TruClient-Registerkarte "Bearbeiten"

Über diese Registerkarte können Sie Schritte und Daten in TruClient-Skripts ausschneiden, kopieren und einfügen.

Zugriff	Wählen Sie in der **TruClient-Seitenleiste** die Registerkarte **Bearbeiten** aus.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Schneidet die ausgewählten Daten bzw. den ausgewählten Schritt aus.
	Kopiert die ausgewählten Daten bzw. den ausgewählten Schritt.
	Fügt Daten vor dem ausgewählten Schritt ein.
	Fügt Daten nach dem ausgewählten Schritt ein.
	Fügt Daten in den ausgewählten Schritt ein.
	Löscht den ausgewählten Schritt.

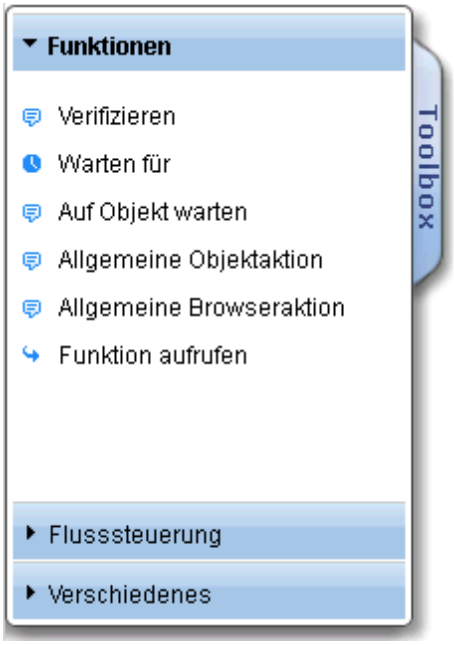
	Öffnet das Dialogfeld **Suchen**, in dem Sie das Skript nach Schritten mit bestimmten Namen oder Nummern durchsuchen können. **Hinweis:** Sie können einen Schritt erstellen, der eine Gruppe von untergeordneten Schritten darstellt. Wenn Sie nach einem Schritt suchen, der ein untergeordneter Schritt ist, müssen Sie sowohl die Gruppennummer als auch die Untergruppennummer angeben. Geben Sie beispielsweise 4.1 ein, wenn Sie nach dem ersten Unterschritt im vierten Schritt (der vierten Gruppe) suchen.
	Wechselt zu dem angegebenen Schritt.
	Die folgenden Aktionen können rückgängig gemacht werden: • Ausschneiden • Kopieren • Einfügen • Löschen • Gruppe • Ziehen und Ablegen (Schritt) • Schritt aus Toolbox hinzufügen
	Die folgenden Aktionen können wiederholt werden: • Ausschneiden • Kopieren • Einfügen • Löschen • Gruppe • Ziehen und Ablegen (Schritt) • Schritt aus Toolbox hinzufügen

Registerkarte "Fenster"

Mit dieser Registerkarte können Sie mehrere Browserfenster für dasselbe Skript steuern. Wählen Sie das Anwendungsfenster aus, das den Fokus erhalten soll. Dies ist für die Debuggingphase der Skriptentwicklung notwendig, um z. B. ein Schrittobjekt markieren zu können.

TruClient-Toolbox

Die Toolbox enthält alle Schritttypen, die Sie zu einem TruClient-Skript hinzufügen können. Sie öffnen und schließen die Toolbox, indem Sie auf die Registerkarte klicken; zum Verschieben ziehen Sie sie nach oben oder unten.



Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Funktionen	• **Verifizieren**. Verifiziert, dass ein Objekt in der Anwendung vorhanden ist. • **Warten für.** Wartet die angegebene Anzahl von Sekunden, bevor der nächste Schritt gestartet wird. • **Auf Objekt warten.** Wartet, bis ein Objekt geladen ist, bevor der nächste Schritt erfolgt. • **Allgemeine Objektaktion/Allgemeine Browseraktion**. Leere Schritte, die eingefügt und manuell konfiguriert werden können. • **Funktion aufrufen.** Fügt eine Funktion in das Skript ein. Weitere Informationen finden Sie unter "Erstellen und Verwenden von Funktionen und Bibliotheken" auf Seite 586.

Flusssteuerung	• **For-Schleife**. Eine logische Struktur, bei der die in der Schleife enthaltenen Schritte so lange wiederholt werden, bis die angegebene Anzahl von Wiederholungen erreicht ist. • **If-Block.** Eine logische Struktur, bei der die im Block enthaltenen Schritte ausgeführt werden, wenn die Bedingung erfüllt ist. • **Else-Aktion hinzufügen.** Klicken Sie auf den Link **Else-Aktion hinzufügen**, um einen **Else**-Abschnitt zu Ihrem **If**-Block hinzuzufügen. Wenn die Bedingung nicht erfüllt ist, werden die im **Else**-Abschnitt enthaltenen Schritte ausgeführt. • **Else-Aktion entfernen.** Entfernt den **Else**-Abschnitt aus dem **If**-Block. **Hinweis:** Wenn der **Else**-Abschnitt Schritte enthält und Sie auf **Else-Aktion entfernen** klicken, werden die Schritte gelöscht. Kopieren Sie sie und fügen Sie sie im Hauptteil Ihres Skripts ein, um sie zu speichern. • **If Exists.** Eine Kombination aus "If-Block" und "Verifizieren"; eine logische Struktur, bei der die im Block enthaltenen Schritte ausgeführt werden, wenn die Bedingung für eine Eigenschaft des ausgewählten Objekts erfüllt ist. • **If Verify.** Eine logische Struktur, bei der die im Block enthaltenen Schritte ausgeführt werden, wenn das ausgewählte Objekt in der Anwendung vorhanden ist. • **Unterbrechen.** Bewirkt, dass die Schleife sofort beendet wird und die aktuelle Iteration oder die restlichen Iterationen nicht mehr ausgeführt werden. • **Fortfahren**. Bewirkt, dass die aktuelle Schleifeniteration sofort beendet wird. Die Skriptausführung wird mit der nächsten Iteration fortgesetzt. • **Fehlererfassung.** Fängt einen Fehler im unmittelbar vorhergehenden Schritt ab und führt den Inhalt des Fehlererfassungsschritts aus. Weitere Informationen finden Sie unter "Erweitern von TruClient-Skripts" auf Seite 576. • **Beenden.** Beendet die Iteration oder das gesamte Skript je nach definierter Einstellung.

Verschiedenes	• **JavaScript auswerten**. Führt den im Schritt enthaltenen JavaScript-Code aus. • **JS bei Objekt auswerten.** Führt den im Schritt enthaltenen JavaScript-Code aus, nachdem das angegebene Objekt in die Anwendung geladen wurde. • **C auswerten**. Führt den im Schritt enthaltenen C-Code aus. • **Kommentar.** Ein leerer Schritt, der es Ihnen ermöglicht, Kommentare in Ihre Skripts zu schreiben. • **Rendezvous.** Synchronisiert alle Vuser in einem Controller-Szenario, so dass diese alle gleichzeitig einen bestimmten Schritt ausführen. So könnten Sie beispielsweise testen, was passiert, wenn sich mehrere Benutzer gleichzeitig bei Ihrer Anwendung anmelden. **Hinweis:** Um einen Rendezvous-Schritt einzufügen, ziehen Sie diesen aus der Toolbox und platzieren Sie ihn oberhalb des zu synchronisierenden Schritts.

Browser in TruClient

Dieser Abschnitt umfasst die folgenden Themen:

Allgemein Browsereinstellungen (TruClient)

Jedes TruClient-Skript in einem Browser mit einem anderen Profil geöffnet. Im Browserprofil werden Benutzerdaten wie Cookies, Clientzertifikate, Chronik, Cache usw. gespeichert. Führen Sie die Änderungen im Browser aus, während Sie das Skript interaktiv mithilfe von TruClient entwickeln. Diese Änderungen beziehen sich nur auf das aktuelle Skript.

Browseroptionen, die sich auf den interaktiven Modus auswirken, können im Dialogfeld **TrueClient – Allgemeine Einstellungen** geändert werden. Alle Einstellungen in diesem Dialogfeld werden in jedes neu erstellte Skript importiert, wirken sich aber nur auf den interaktiven Modus aus. Wird das Skript gespeichert, werden diese Einstellungen auch in die Laufzeiteinstellungen kopiert.

Im Lastmodus ausgeführte Skripts verwenden die Einstellungen, die im Knoten **Last > Browsereinstellungen** des Dialogfelds **Laufzeiteinstellungen** definiert sind. Weitere Informationen finden Sie unter "Last > Knoten "Browsereinstellungen" (Ajax TruClient)" auf Seite 355.

Die erweiterten Firefox-Einstellungen werden jedes Mal, wenn ein Skript in Firefox geöffnet wird, in das Skript importiert. Weitere Informationen finden Sie unter "Dialogfeld "TruClient - Allgemeine Einstellungen"" auf Seite 609.

Einige standardmäßigen Browseroptionen sind deaktiviert. Dazu gehören unter anderem einige Optionen der Menüs **Datei**, **Lesezeichen**, **Extras** und **Hilfe**. Um Lesezeichen zu speichern, die Sie in Skripts verwenden möchten, verwenden Sie "Dialogfeld "TruClient - Allgemeine Einstellungen"" auf der nächsten Seite.

Privater Modus (TruClient)

Der private Browsermodus ermöglicht es Ihnen, im Web zu surfen, ohne dass Sitzungsinformationen gespeichert werden. Einige Beispiele für nicht gespeicherte Elemente sind Kennwörter, Cookies und Verlaufsdaten.

Um reale Benutzer präziser zu emulieren, gibt TruClient Skripts im privaten Modus wieder. Damit wird sichergestellt, dass der Browser keine gespeicherten Sitzungsinformationen verwendet, wenn ein Skript mehrmals ausgeführt wird.

TruClient-Browser für IE

Der TruClient-Browser für IE simuliert die Funktionen eines IE-Browsers. Damit können Sie Skripts aufzeichnen und wiedergeben, als ob Sie in einer IE-Umgebung arbeiten würden.

Hinweis: Das TruClient für IE-Protokoll wurde für Anwendungen entwickelt, die nur im Standardmodus ausgeführt werden.

Die TruClient-Engine hängt von bestimmten Browserfunktionen ab, die erst ab Internet Explorer Version 9 unterstützt werden. Diese Funktionen werden auch in IE9 nicht unterstützt, wenn Mechanismen für die Abwärtskompatibilität verwendet werden.

Weitere Informationen zum Dokumente- und Browsermodus finden Sie unter Browsermodus.

Hinweis: Der Tru Client-Browser für IE erfordert ebenfalls Lesezugriff auf den folgenden Registrierungsschlüssel: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer`

Firebug Lite

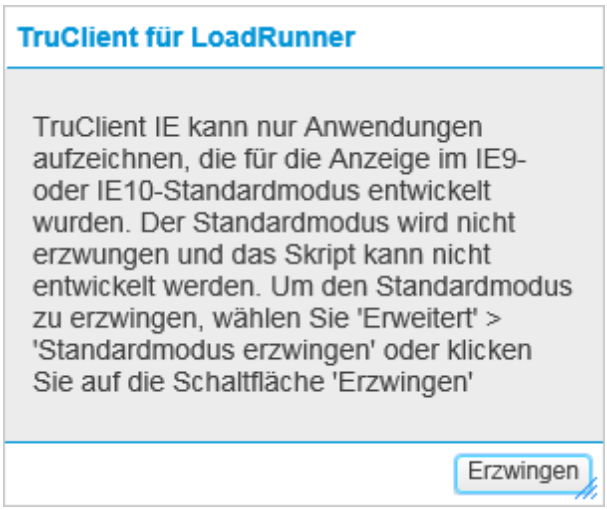
Firebug Lite ist ein Dienstprogramm eines anderen Herstellers, das viele wertvolle Entwicklungstools bereitstellt. Sie können damit CSS, HTML und JavaScript in Echtzeit auf einer beliebigen Webseite bearbeiten, debuggen und überwachen. Wählen Sie **Erweitert > FireBug Lite** aus oder drücken Sie im IE-Browser von TruClient auf **F12**.

Weitere Informationen zu den Funktionen von FireBug Lite finden Sie unter What is Firebug?

Standardmodus erzwingen

Mit TruClient für IE können Sie eine Anwendung nur im Standardmodus aufzeichnen und wiedergeben. TruClient erzwingt diesen Modus auf allen Seiten, zu denen Sie bei der Aufzeichnung und Wiedergabe navigieren.

Wenn Sie untersuchen möchten, wie Seiten in einem nicht standardmäßigen IE-Modus gerendert werden, können Sie den erzwungenen Standardmodus vorübergehend deaktivieren, indem Sie die Auswahl unter **Erweitert > Standardmodus erzwingen** aufheben. Wenn die Option deaktiviert wird, wird in der Seitenleiste folgende Meldung angezeigt:



In diesem Modus können Sie keine Skripts im TruClient IE-Browser entwickeln oder bearbeiten. Klicken Sie im Meldungsfeld auf die Schaltfläche **Erzwingen** oder aktivieren Sie **Erweitert > Standardmodus erzwingen**, um die Aufzeichnung und Wiedergabe im TruClient für IE-Browser fortzusetzen.

Dialogfeld "TruClient - Allgemeine Einstellungen"

Mithilfe dieses Dialogfelds können Sie mehrere Optionen festlegen, die sich auf den Browser auswirken, während Sie Skripts im interaktiven Modus entwickeln.

Allgemeine Einstellungen

Zugriff	Schaltfläche **TruClient - Allgemeine Einstellungen**
Wichtige Informationen	Der Zugriff auf das Dialogfeld **TruClient - Allgemeine Einstellungen** erfolgt über das VuGen-Fenster oder die TruClient-Seitenleiste im interaktiven Modus.
Relevante Aufgaben	"Entwickeln von TruClient-Skripts" auf Seite 549

Registerkarte "Sucheinstellungen"

Auf dieser Registerkarte können Sie die Einstellungen des TruClient-Browsers für Skripts konfigurieren, die im interaktiven Modus ausgeführt werden.

Die Einstellungen auf der Registerkarte **Sucheinstellungen** stimmen mit den Einstellungen im Knoten **Laufzeiteinstellungen > Last > Browsereinstellungen** von VuGen überein. Eine Beschreibung der verfügbaren Optionen finden Sie unter "Last > Knoten "Browsereinstellungen" (Ajax TruClient)" auf Seite 355.

Einstellungen, die Sie auf der Registerkarte **Sucheinstellungen** ändern, wirken sich nur auf den interaktiven Modus aus und gelten für alle TruClient-Skripts. Wenn Sie Ihr Skript im interaktiven Modus speichern, werden alle Einstellungen, die Sie auf der Registerkarte **Sucheinstellungen** geändert haben, auf die Lastmodus-Browsereinstellungen angewendet.

Firefox-Optionen

Tab	

Registerkarte "Interaktive Optionen"

Element der Oberfläche	Beschreibung
Aktion bei Fehler	**Skript abbrechen**. Bricht das Skript bei einem Fehler ab. **Mit der nächsten Iteration fortfahren.** Stoppt die Iteration bei einem Fehler und fährt mit der nächsten Iteration fort.
Snapshot-Generierung	**Generierung von Aufzeichnungs-Snapshots** • **Nie.** Aufzeichnungs-Snapshots werden nie generiert. • **Immer.** Aufzeichnungs-Snapshots werden immer generiert. ▪ **Maximale Ebene.** Ermöglicht die Auswahl der Skriptebene für die Generierung von Snapshots (1-3). **Generierung von Wiedergabe-Snapshots** • **Nie.** Snapshots werden während der Wiedergabe nie generiert. • **Bei Fehler.** Snapshots werden während der Wiedergabe generiert, wenn ein Fehler auftritt. • **Immer.** Snapshots werden während der Wiedergabe immer generiert.
Assistent für Objekterkennung	**Aktivieren.** Bei Auswahl dieser Option wird der Assistent für die Objekterkennung aktiviert. Der Assistent für die Objekterkennung unterstützt die Behebung von Erkennungsfehlern während der Wiedergabe. Weitere Informationen finden Sie unter "Lösen von Problemen bei der Objekterkennung" auf Seite 560.

	Serverparameter	**Server durch Parameter ersetzen.** Bei Auswahl dieser Option wird der Serverdomänenname in den Navigationsschritten automatisch durch einen Parameter ersetzt.
	Konsoleneinstellungen	**Protokollmeldungen anzeigen.** Protokollmeldungen der Konsole werden während der Wiedergabe angezeigt.
	Debugeinstellungen	**Springen zu 'Funktionen' deaktivieren.** Deaktiviert die Anzeigeschritte für Funktionen während interaktiver Wiedergabe.
Registerkarte "Verschlüsselung"	Weitere Details zur Konfiguration finden Sie unter Mozilla Support for Encryption.	
Registerkarte "Erweiterung"	Weitere Details zur Konfiguration finden Sie unter Mozilla Support for Extensions.	

Internet Explorer-Optionen

Registerkarte "Interaktive Optionen"

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Aktion bei Fehler	**Skript abbrechen** Bricht das Skript bei einem Fehler ab. **Mit der nächsten Iteration fortfahren**. Stoppt die Iteration bei einem Fehler und fährt mit der nächsten Iteration fort

Snapshot-Generierung	**Generierung von Aufzeichnungs-Snapshots** • **Nie.** Aufzeichnungs-Snapshots werden nie generiert. • **Immer.** Aufzeichnungs-Snapshots werden immer generiert. ▪ **Maximale Ebene.** Ermöglicht die Auswahl der Skriptebene für die Generierung von Snapshots (1-3). **Generierung von Wiedergabe-Snapshots** • **Nie.** Snapshots werden während der Wiedergabe nie generiert. • **Bei Fehler.** Snapshots werden während der Wiedergabe generiert, wenn ein Fehler auftritt. • **Immer.** Snapshots werden während der Wiedergabe immer generiert.
Assistent für Objekterkennung	**Aktivieren.** Bei Auswahl dieser Option wird der Assistent für die Objekterkennung aktiviert. Der Assistent für die Objekterkennung unterstützt die Behebung von Erkennungsfehlern während der Wiedergabe. Weitere Informationen finden Sie unter "Lösen von Problemen bei der Objekterkennung" auf Seite 560.
Serverparameter	**Server durch Parameter ersetzen.** Bei Auswahl dieser Option wird der Serverdomänenname in den Navigationsschritten automatisch durch einen Parameter ersetzt.

Registerkarte "Clientzertifikat"

Ermöglicht das manuelle Exportieren des Clientzertifikats in den Skriptordner.

TrueClient – Fehlerbehebung und Einschränkungen (Allgemein)

In diesem Abschnitt werden die Fehlerbeseitigung und die Einschränkungen für TrueClient sowie andere interessante Dinge vorgestellt.

Fehlerbehebung bei der Kommunikation zwischen VuGen und TruClient

Fehlermeldung: VuGen kann den TruClient-Browser nicht öffnen, da der Portbereich belegt oder ungültig ist.

Empfehlung: Zum Öffnen eines Servers ist ein freier Port erforderlich. Der Server ist offen unter http://127.0.0.1:<free_Port>/ url.

Der freie Port kann in den folgenden XML-Dateien konfiguriert werden.

```
%APPDATA%\Hewlett-Packard\LoadRunner\Data\Settings\VuGenProperties.xml
```

```
<VuGenProperties>
```

```
<BrowserCommunicationServerEndPort value="8090" />
```

```
<BrowserCommunicationServerStartPort value="8080" />
```

Der Standardwert für den Portbereich liegt zwischen 8080 und 8090. Wenn ein Problem mit den Ports vorliegt (alle Ports sind belegt oder der Anfangsport ist größer als der Endport), können Sie den Bereich in der Konfigurationsdatei ändern.

Wenn Sie das Problem nicht beheben können, wenden Sie sich an die HP Software-Unterstützung.

Fehlermeldung: TruClient-Browser kann nicht geöffnet werden. Dieser Fehler tritt auf, weil Sie nicht die erforderlichen Berechtigungen besitzen, um eine Verbindung zwischen VuGen und dem TruClient-Browser herzustellen.

Empfehlung: Bitten Sie Ihren Administrator oder einen Hauptbenutzer, diesen Befehl auszuführen.

```
netsh http add urlacl url=http://127.0.0.1:<In_der_Fehlermeldung_angegebener_
Port>/ user=<DOMÄNE>\<BENUTZERNAME>
```

Beispiel: `netsh http add urlacl url=http://127.0.0.1:8080/ user=DOMAIN\john`

Wortverifizierungsfunktion im Geschäftsprozess

Viele Websites verwenden besondere Bestätigungsfelder (auch Captcha genannt), bei denen der angezeigte Text eingegeben werden muss, um sicherzustellen, dass ein Formular tatsächlich von einem Benutzer ausgefüllt wird.

Diese Maßnahme soll verhindern, dass Webcrawler, Spider usw. die Website nutzen und wertvolle Systemressourcen belegen.

Da diese Felder entwickelt wurden, um automatische Tools zu blockieren, ist auch LoadRunner davon betroffen.

Damit Ihr Geschäftsprozess vollständig ausgeführt wird, müssen Sie diese Funktion auf der Website, für die Sie den Lasttest ausführen, deaktivieren.

Protokollbeschränkungen

Die Antivirensoftware von McAfee hält die Ausführung von Vusern im Lastmodus an

Problem: Neuere Versionen der Antivirensoftware von McAfee verursachen ggf. Netwerkhookkollisionen mit dem TruClient für IE-Protokoll. Dies führt u. U. zu einem unerwarteten Verhalten während eines Lasttests. Beispielsweise werden Vuser ggf. angehalten. Dies hat zur Folge, dass im Bereich **Controller > Szenariostatus > Fehler** eine Fehlermeldung und der Vuser-Status als fehlerhaft angezeigt wird.

Schritte für die Behebung: Deaktivieren Sie McAfee.

Flash-, Silverlight-, ActiveX-Einschränkung für TruClient IE

Benutzeraktionen auf Elemente der Benutzeroberfläche, die auf Flash basieren, Silverlight, Active X im Allgemeinen und Java-Applets-Technologien werden derzeit nicht aufgezeichnet und wiedergegeben.

Einschränkungen in Bezug auf Java-Applets und Silverlight für TruClient FF

Benutzeraktionen auf Elemente der Benutzeroberfläche, die auf Java-Applets- und Silverlight-Technologien basieren, können derzeit nicht aufgezeichnet und wiedergegeben werden.

- Wenn Sie für TruClient-Vuser Diagramme zur Aufschlüsselung von Webseitenkomponenten anzeigen, sind Messungen in TruClient möglicherweise nicht verfügbar oder ungenauer als die anderer Protokolle. Beispielsweise stehen möglicherweise das Diagramm **Zeit bis zur ersten Puffer-Aufschlüsselung** und Messwerte wie **SSL Handshake** und **Client/Fehler-Zeit** nicht zur Verfügung.

- TruClient ist ein auf der Benutzeroberfläche basierendes Protokoll. Daher kann es beim Ausführen eines TruClient-Skripts im Controller zu einer Ausnahme wie der folgenden kommen: `Fehler beim Initialisieren. Ursache Zeitüberschreitung`. Es wird empfohlen, die anfängliche Zeitüberschreitung zu erhöhen (**Controller > Extras > Optionen**), damit die Vuser ausreichend Zeit haben, die Initialisierungsphase abzuschließen.

- Nachdem Sie TruClient durch Klicken auf **Skript entwickeln** geöffnet und anschließend geschlossen haben, wird in VuGen möglicherweise weiterhin das gelbe Banner des Entwicklungsmodus angezeigt. **Problemumgehungen**: Wenn Sie mit einem Proxy arbeiten, müssen Sie angeben, dass lokale Verbindungen umgangen werden. Wenn Sie mit einer PAC-Datei arbeiten, muss PAC eine direkte Verbindung für 127.0.0.1 angeben.

- Wenn Sie ein TruClient-Skript mit einer späteren Version von LoadRunner bearbeiten als mit der, mit der Sie es aufgenommen haben, schlägt die Wiedergabe möglicherweise in einigen Fällen fehl.
  Erstellen Sie vor der Bearbeitung eine Sicherungskopie des Skripts. **Problemumgehung:** Sollte die Wiedergabe fehlschlagen, zeichnen Sie das Skript mit der aktuellen Version des TruClient-Protokolls auf.

- TruClient zeichnet keine "Mauszeiger darüber bewegen"-Vorgänge auf, wenn für das Element die CSS-Regel "Mouse Hover" gilt.
  Problemumgehung: Statt die Maus über ein Element zu bewegen, klicken Sie auf die einzelnen Elemente, um das Dropdownmenü zu öffnen (die Klicks werden aufgezeichnet).

- Schritttypen und Argumente des TruClient-Protokolls sind nicht lokalisiert.

- Allgemeine Zeichenfolgen für TruClient sind in der japanischen Version nicht lokalisiert.

TruClient - IE-Protokoll

- Das TruClient für IE-Protokoll unterstützt keine Diagnose für J2EE/.NET.
- Beim Aufzeichnen eines Skripts im TruClient für IE-Protokoll wird ein Tastaturereignis mit der STRG-Taste nicht unterstützt und somit nicht aufgezeichnet.
- Das TruClient für IE-Protokoll erfordert die Installation von Internet Explorer 9 oder 10 auf dem Lastgenerator.
- Aufgrund des Verhaltens von Internet Explorer 9 bei der Verwendung des TruClient-IE-Protokolls für Internet Explorer 9 und bei Verwendung eines JRE-Plug-Ins vor Version 7 stürzt der Browser möglicherweise ab, wenn Sie Webseiten aufrufen, die gerendert werden und Applets enthalten, die das Plug-In instanziieren müssen.

  Problemumgehung: Installieren Sie das JRE 7-Plug-In oder eine höhere Version.
- TruClient unterstützt nur den IE9- oder IE10-Standardmodus.

TruClient für Firefox-Protokoll

- Während der Skriptentwicklung werden in TruClient keine Dialogfelder mit Sicherheitswarnungen angezeigt.
- Wenn beim Ausführen von TruClient für Firefox-Vusern im Lastmodus Probleme auftreten, ändern Sie die Einstellungen für **Takt** (**Wiedergeben > Laufzeiteinstellungen > Allgemein > Takt**) wie folgt: Wählen Sie **Nach Beendigung der vorherigen Iteration** oder **Zu festgelegten Intervallen** bzw. **In zufälligen Intervallen** aus.
- Das TruClient für Firefox-Protokoll kann nur mit Anwendungen verwendet werden, die den Browser Mozilla Firefox unterstützen.
- TruClient für Firefox unterstützt die Kerberos-Authentifizierung nicht. Wenn der Server auch NTLM unterstützt, verlagert TruClient die Authentifizierung automatisch auf NTLM, und der Vorgang wird fortgesetzt.

Konvertieren eines TruClient-Skripts in ein HTTP/HTML-Skript

Profitieren Sie gleichzeitig von einer Skriptentwicklung und der Ausführung einer Vielzahl von Vusern, indem Sie ein TruClient-Skript in ein HTTP/HTML umwandeln.

Hinweis:

Sie müssen über Kenntnisse des Web-HTTP/HTML-Protokolls verfügen, um das Skript erfolgreich wiedergeben zu können.

Während der Konvertierung werden Kommentare und APIs zum Web-HTTP/HTML-Protokollskript hinzugefügt, um den Konvertierungsvorgang zu dokumentieren.

1. Erstellen Sie ein TruClient-Skript. Weitere Informationen finden Sie unter "Entwickeln von TruClient-Skripts" auf Seite 549.
2. Speichern Sie das Skript und schließen Sie die TruClient-Seitenleiste.
3. Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche

, um das Skript zu konvertieren.
4. Nachdem das Skript generiert wurde, überprüfen Sie es und denken Sie dabei daran, dass das Web-HTTP/HTML-Protokoll auf der Transportebene aufgezeichnet wird. So könnte es beispielsweise sein, dass Sie Probleme mit der Korrelation oder mit Parametern in dem konvertierten Skript beheben müssen. Weitere Informationen finden Sie unter "Webprotokoll (HTTP/HTML) – Übersicht" auf Seite 881.

Einschränkungen

Bei der Konvertierung von TruClient-Skripts in Web-HTTP/HTML-Skripts werden die Konvertierungsschritte, die die Adresse 127.0.0.1 (localhost) aufrufen, nicht unterstützt.

LRFunktionsreferenz

Dieser Abschnitt umfasst die folgenden Themen:

TruClient-Funktionen

Die folgenden Funktionen können als Werte in TruClient-Schrittelemente eingegeben werden.

Methode	Beschreibung	Argumente
Die Argumente in dieser Zeile können in allen Methoden verwendet werden.		• **Objekt.** Das Objekt des Schritts entsprechend der Definition in der Anwendung. • **Fenster.** Zeigt auf das globale Fensterobjekt der Anwendung. • **Dokument.** Das globale Dokumentobjekt der Anwendung.
IO.createDir(Pfad)	Erstellt den angegebenen Ordner. Erstellt bei Bedarf alle Ordner, die für den vollständigen Pfad erforderlich sind.	**Pfad.** Der absolute Pfad des Ordners.
IO.delete(Pfad)	Löscht den angegebenen Ordner bzw. die Datei. Bei Angabe eines Ordners werden alle Dateien in dem Ordner einschließlich Unterverzeichnisse gelöscht.	**Pfad.** Der absolute Pfad des Ordners oder der Datei.

Methode	Beschreibung	Argumente
IO.isExist(Pfad)	Gibt True zurück, wenn der angegebene Ordner bzw. die Datei existiert, und False, wenn dies nicht der Fall ist. Gibt False für ein nicht verbundenes oder nicht authentifiziertes, zugeordnetes Laufwerk zurück.	**Pfad.** Der absolute Pfad des Ordners oder der Datei.
IO.read(Pfad, Zeichensatz)	Gibt alle Daten der angegebenen Datei zurück. Konvertiert die Daten aus dem angegebenen Zeichensatz in Unicode.	**Pfad.** Der absolute Pfad zu der Datei. **Zeichensatz.** Der Zeichensatz der Datei, wenn es sich nicht um UTF 8 handelt.
IO.write(Pfad, Zeichenfolge, Anfügen, Zeichensatz)	Schreibt die Zeichenfolge in die angegebene Datei. Wenn die Datei nicht existiert, wird sie erstellt.	**Absoluter Pfad.** Der absolute Pfad zu der Datei. **Zeichenfolge.** Die in die Datei zu schreibende Zeichenfolge. **Anfügen.** Boolescher Wert. **True.** (Standard) Die Zeichenfolge an das Dateiende anhängen. **False.** Die Datei mit der Zeichenfolge überschreiben. **Zeichensatz.** Der Zeichensatz der Datei, wenn es sich nicht um UTF 8 handelt.
LR.advanceParam (Parameter)	Legt für den angegebenen Parameter den nächsten Wert in der Datei fest.	**Parameter.** Der Name des weiterzusetzenden Parameters. Dies muss ein Parameter des Typs Datei oder eindeutige Zahl sein.
LR.getUserIP	Gibt die IP-Adresse zurück, wenn IP-Spoofing aktiviert ist und das Skript im Lastmodus im Controller ausgeführt wird.	
LR.setParam(Name, Wert)	Speichert eine Zeichenfolge in einem Parameter und erstellt den Parameter, wenn dieser nicht existiert.	**Name.** Der Name des Parameters, in dem der Wert gespeichert werden soll. **Wert.** Der Wert.
LR.crossTransactionStart (Name,Wert)	Startet eine Transaktion in einem Vuser-Skript, die in einem anderen Vuser-Skript beendet wird.	**Name.** Der Name der Transaktion. **Wert.** Ein Zeiger auf eine Parameterliste.
LR.crossTransactionEnd (Name,Wert,Status)	Beendet eine Transaktion in einem Vuser-Skript, die in einem anderen Vuser-Skript gestartet wurde.	**Name.** Der Name der Transaktion. **Wert.** Ein Zeiger auf eine Parameterliste. **Status.** Einer der folgenden Statusangaben: "Bestanden" oder "Fehler".
LR.transactionDuration (Name)	Gibt die Dauer einer bestimmten Transaktion zurück.	**Name.** Der Name der Transaktion.
LR.getParam(Name)	Gibt den Wert des angegebenen Parameters zurück.	**Name.** Der Parametername.
LR.getLRAttr(Name)	Gibt den Wert des angegebenen mdrv-Befehlsparameters zurück.	**Name.** Der Name des Befehlszeilenparameters.

Methode	Beschreibung	Argumente
LR.evalC (Funktionsname)	Führt die angegebene Funktion aus, deren Definition sich in der angegebenen Datei befindet.	**Funktionsname.** Der Funktionsname. Verwenden Sie den Befehl `#include` zum Einschließen der Datei, in der die Funktion definiert ist.
LR.log(Text, Ebene)	Protokolliert eine Meldung.	**Text.** Die Meldung. **Ebene.** Eine der Folgenden: ● "Fehler", ● "Warnung", ● "Standard", ● "Erweitert", ● "Status". LR.log(Ebene)= Status sendet eine Zeichenfolge an den Statusbereich von Controller. Die Zeichenfolge wird zudem an das Vuser.Protokoll gesendet. Bei Ausführung von VuGen aus wird die Meldung an "output.txt" gesendet. **Beispiel:** LR.log("Text", "Fehler");
LR.decrypt(Text)	Gibt den Text nach der Entschlüsselung zurück.	**Text.** Der verschlüsselte Text.
LR.userDataPoint(Name, Wert)	Zeichnet einen benutzerdefinierten Datenpunkt für die Analyse auf.	**Name.** Der Name des Datenpunkts. Ein Datenpunktname darf nicht mit einer dieser Zeichenfolgen beginnen: HTTP, NON_HTTP, RETRY, mic_, stream_, mms_ **Wert.** Der numerische Wert.
evalXPath(XPath)	Gibt ein Array der Objekte zurück, die durch den XPath in dem Argument definiert werden.	**XPath.** Die XPath-Abfrage.
LR.startTransaction (Name)	Startet eine LoadRunner-Transaktion.	**Name.** Der Name der zu startenden Transaktion.
LR.endTransaction (Name, Status)	Beendet eine LoadRunner-Transaktion.	● **Name.** Der Name der zu beendenden Transaktion. ● **Status.** Einer der Folgenden: "Pass", "Fail", "Auto"

Methode	Beschreibung	Argumente
LR.vuserStatusMessage (Zeichenfolge)	Gibt an, welcher Vuser ein bestimmtes Skript verarbeitet.	**Zeichenfolge.** Beliebige Zeichenfolge. **Beispiel:** LR.vuserStatusMessage ("FlightStatus")
Utils.clearCookies()	Entfernt alle gegenwärtig vom Vuser gespeicherten Cookies.	
Utils.clearCache()	Löscht den Inhalt des Cachesimulators.	
Utils.getEnv(Name)	Gibt den Wert der angegebenen Umgebungsvariablen zurück. Gibt eine leere Zeichenfolge zurück, wenn die angegebene Variable nicht existiert.	**Name.** Der Name der Umgebungsvariable.
Utils.import(Pfad)	Wertet die angegebene JavaScript-Datei im Argumentkontext aus.	**Pfad.** Der absolute Pfad zu der JavaScript-Datei.
Utils.isEnvExists(Name)	Gibt **True** zurück, wenn die angegebene Umgebungsvariable existiert, und **False**, wenn dies nicht der Fall ist.	**Name.** Der Name der Umgebungsvariable.
Utils.setEnv(Name, Wert)	Legt für die angegebene Umgebungsvariable den angegebenen Wert fest. Wenn die Variable bereits einen Wert besitzt, wird dieser überschrieben.	• **Name.** Der Name der Umgebungsvariable. • **Wert.** Der Wert, auf den die Umgebungsvariable eingestellt wird.
Utils.addAutoFilter(Filter, isIncluded)	Hinzufügen eines Filters zu einer der Black Lists oder White Lists der URLs. Der URL jeder HTTP-Anforderung wird gegen diese Listen geprüft. Zuerst die Black List und anschließend, wenn diese nicht leer ist, die White List. HTTP-Anforderungen, deren Prüfung nicht erfolgreich verläuft, werden blockiert.	• **Filter.** Zeichenfolge, die den URL repräsentiert. • **isInclude.** True gibt die White List an, ansonsten False.
Utils.removeAutoFilter (Filter, isIncluded)	Entfernen eines Filters aus einer Black List oder White List der URLs.	• **Filter.** Zeichenfolge, die den URL repräsentiert. • **isInclude.** True gibt die White List an, ansonsten False.
Utils.cleanupAutoFilters ()	Entfernen aller Filter aus der Black List und der White List der URLs.	

Methode	Beschreibung	Argumente	
Utils.addAutoHeader (Kopfzeile, Wert, Zusammenführen)	Hinzufügen einer HTTP-Kopfzeile zu jeder nachfolgenden HTTP-Anforderung, die diesem Funktionsaufruf folgt.	• **Kopfzeile.** Der Name der hinzuzufügenden Kopfzeile. • **Wert.** Der Wert der hinzuzufügenden Kopfzeile. • **Zusammenführen. True** gibt an, dass der Wert mit dem Wert einer vorhandenen Kopfzeile mit demselben Namen zusammengeführt wird. **False** gibt an, dass die Kopfzeile überschrieben wird.	
Utils.removeAutoHeader (Header)	Beendet das Hinzufügen einer HTTP-Kopfzeile zu jeder nachfolgenden HTTP-Anforderung, die diesem Funktionsaufruf folgt.	**Kopfzeile.** Der Name der Kopfzeile, die nicht mehr hinzugefügt wird.	
Utils.cleanupAutoHeaders ()	Entfernt alle HTTP-Kopfzeilen und beendet das Hinzufügen von HTTP-Kopfzeilen zu allen nachfolgenden HTTP-Anforderungen, die auf diesen Funktionsaufruf folgen.		

TruClient-VTS-Funktionen

Die folgenden Funktionen können als Werte in TruClient-Schrittelemente eingegeben werden.

Methode	Beschreibung	Argumente	Hinweise zum Funktions-verhalten
LR.vtcAddCell(colName, value, VtsName)	Legt das letzte Feld einer Spalte auf einen Wert fest.	**colName.** Der Name der Spalte. **value.** Der Wert als Zeichenfolge. **VtsName.** Der Alias des VTS-Servers.	Wenn die im Argument angegebene Spalte nicht vorhanden ist, wird sie erstellt, und der Inhalt der Zelle wird auf den Argumentwert festgelegt.

Methode	Beschreibung	Argumente	Hinweise zum Funktions-verhalten
LR.vtcAddCells (colNames, values,option, VtsName)	Legt die Daten in mehreren Spalten fest. Wenn die ausgewählte Option 2 ist, wird ein Wert von 0 zurückgegeben, sofern ein Wert eingefügt wurde. Gibt einen Wert von 1 zurück, wenn der Wert bereits vorhanden ist.	**ColNames.** Die Spaltennamen, voneinander getrennt durch ein Semikolon. **values.** Die Werte als Zeichenfolge, voneinander getrennt durch ein Semikolon. **option.** Definiert das Hinzufügen der Werte. **0.** Zum Hinzufügen als identische Zeile in allen Spalten basierend auf der Spalte mit den meisten Zeilen. Die erstellte Zeile ist für alle Spalten n+1. **1.** Zum Hinzufügen als Stapel - letzte Zeile in jeder Spalte. **2.** Zum Hinzufügen als einzelner Stapel - letzte Zeile in jeder Spalte nur, wenn der Wert in der Spalte nur einmal vorkommt. VtsName. Der Alias des VTS-Servers.	Wenn die im Argument angegebenen Spalten nicht vorhanden sind, werden sie erstellt, und die Inhalte der Zellen werden auf die Argumentwerte festgelegt.

Methode	Beschreibung	Argumente	Hinweise zum Funktions-verhalten
LR.vtcAddUniqueCell (colName, value, VtsName)	Legt das letzte Feld einer Spalte auf einen Wert fest, wenn der Wert in der Spalte nicht vorhanden ist. Gibt einen Wert von 0 zurück, wenn ein Wert eingefügt wurde. Gibt einen Wert von 1 zurück, wenn der Wert bereits vorhanden ist.	**ColName.** Der Name der Spalte. **value.** Der Wert. **VtsName.** Der Alias des VTS-Servers.	Wenn die im Argument angegebene Spalte nicht vorhanden ist, wird sie erstellt, und der Inhalt der Zelle wird auf den Argumentwert festgelegt.
LR.vtcClearColumn (colName,VtsName)	Löscht alle Daten in einer Spalte.	**colName**. Der Name der Spalte. **VtsName.** Der Alias des VTS-Servers.	Wenn die im Argument angegebene Spalte nicht vorhanden ist, wird der Schritt ausgeführt, ohne dass ein Fehler zurückgegeben wird, und in VTS werden keine Daten geändert.
LR.vtcClearCell (colName,rowIndex,VtsName)	Löscht die Daten in einem Feld.	**colName.** Der Name der Spalte. **rowIndex.** Die Indexnummer des Felds als Ganzzahl. 1 ist das erste Feld in der Spalte. **VtsName.** Der Alias des VTS-Servers.	Wenn die im Argument angegebene Spalte bzw. der dort angegebene Zeilenindex nicht vorhanden ist, wird der Schritt ausgeführt, ohne dass ein Fehler zurückgegeben wird, und in VTS werden keine Daten geändert.

Methode	Beschreibung	Argumente	Hinweise zum Funktions-verhalten
LR.vtcColumnSize (colName, VtsName)	Gibt die Anzahl der Felder in einer Spalte zurück, die Daten enthalten.	**colName.** Der Name der Spalte. **VtsName.** Der Alias des VTS-Servers.	Wenn die im Argument angegebene Spalte nicht vorhanden ist, wird der Schritt ausgeführt, ohne dass ein Fehler zurückgegeben wird, und der Rückgabewert ist 0.
LR.vtcConnect (serverName, port, VtsName)	Stellt eine Verbindung mit dem Server her.	**serverName.** Entweder der IP-Name oder der Servername. **port.** Die Portnummer **VtsName.** Der Alias des VTS-Servers.	
LR.vtcDisconnect (VtsName)	Trennt die Verbindung mit dem Server.	**VtsName.** Der Alias des VTS-Servers.	
LR.vtcDropIndex (colName,VtsName)	Löscht den Index einer Spalte.	**colName.** Der Spaltenname. **VtsName.** Der Alias des VTS-Servers.	Wenn die im Argument angegebene Spalte nicht vorhanden ist, wird der Schritt ausgeführt, ohne dass ein Fehler zurückgegeben wird. Auch wenn der Index bereits in der Spalte gelöscht wurde, wird der Schritt ausgeführt, ohne dass ein Fehler zurückgegeben wird.

Methode	Beschreibung	Argumente	Hinweise zum Funktions-verhalten
LR.vtcEnsureIndex (colName,VtsName)	Erstellt einen Index für eine Spalte.	**colName.** Der Name der Spalte. **VtsName.** Der Alias des VTS-Servers.	Wenn die im Argument angegebene Spalte nicht vorhanden ist, wird der Schritt ausgeführt, ohne dass ein Fehler zurückgegeben wird. Auch wenn der Index bereits für die Spalte erstellt wurde, wird der Schritt ausgeführt, ohne dass ein Fehler zurückgegeben wird.

Methode	Beschreibung	Argumente	Hinweise zum Funktions-verhalten
LR.vtcIncrement (colName,rowIndex,value ,VtsName)	Erhöht einen in einem Feld gespeicherten Zähler.	**colName.** Der Name der Spalte. **rowIndex.** Die Indexnummer des Felds als Ganzzahl. 1 ist das erste Feld in der Spalte. **value.** Der Wert als Zeichenfolge. **VtsName.** Der Alias des VTS-Servers.	Wenn der im Argument angegebene Spaltenname nicht vorhanden ist, wird die Spalte erstellt, und die Zelle, auf die der Index verweist, wird auf den Argumentwert festgelegt. Wenn der im Argument angegebene Index die Spaltengröße überschreitet, wird die Zelle vom angegebenen Index erstellt, und die Inhalte der Zelle werden auf den Argumentwert festgelegt. Wenn die Spalte, auf die der Index verweist, eine Zeichenfolge enthält, werden die Inhalte der Zelle durch das Wertargument ersetzt.

Methode	Beschreibung	Argumente	Hinweise zum Funktions-verhalten
LR.vtcGeLRell (colName,rowIndex,VtsName)	Gibt die Daten in einem Feld zurück.	**colName.** Der Name der Spalte. **rowIndex.** Die Indexnummer des Felds als Ganzzahl. 1 ist das erste Feld in der Spalte. **VtsName.** Der Alias des VTS-Servers.	Wenn der im Argument angegebene Spaltenname oder Indexname nicht vorhanden ist, wird ein NULL-Wert zurückgegeben.
LR.vtcGetRowCells (rowIndex,VtsName)	Gibt die Daten in einer Zeile als JavaScript-Objekt zurück. Die Eigenschaften des Objekts werden auf die Spaltennamen festgelegt.	**rowIndex.** Die Indexnummer des Felds. 1 ist das erste Feld in der Spalte. **VtsName.** Der Alias des VTS-Servers.	Wenn die im Argument angegebene Spalte nicht vorhanden ist, wird für jede Spalte ein NULL-Wert zurückgegeben.
LR.vtcPopCell (colName,VtsName)	Blendet das erste Feld einer Spalte ein.	**colName.** Der Name der Spalte. **VtsName.** Der Alias des VTS-Servers.	Wenn der im Argument angegebene Spaltenname nicht vorhanden ist, wird ein NULL-Wert zurückgegeben.
LR.vtcPopMultipleCells (colNames,VtsName)	Blendet die ersten Felder der angegebenen Spalten ein. Wird als JavaScript-Objekt zurückgegeben. Die Eigenschaften des Objekts werden auf die Spaltennamen festgelegt.	**colNames.** Die Namen der Spalten. **VtsName.** Der Alias des VTS-Servers.	Wenn die im Argument angegebenen Spaltennamen nicht vorhanden sind, wird ein NULL-Wert zurückgegeben.

Methode	Beschreibung	Argumente	Hinweise zum Funktions-verhalten
LR.vtcUpdateCell (colName,rowIndex,value,VtsName)	Ersetzt die Daten in einem Feld.	**colName.** Der Name der Spalte. **rowIndex.** Die Indexnummer des Felds als Ganzzahl. 1 ist das erste Feld in der Spalte. **value.** Der Wert als Zeichenfolge. **VtsName.** Der Alias des VTS-Servers.	Wenn der Spaltenname oder Index nicht vorhanden ist, wird die Spalte und/oder die Zelle, auf die der Index verweist, erstellt, und die Zelleninhalte werden auf den Argumentwert festgelegt.
LR.vtcUpdateEqualsCell (colName,rowIndex,value,comparedValue,VtsName)	Ersetzt die Daten in einem Feld, wenn die aktuellen Daten einem bestimmten Wert entsprechen. Gibt einen Wert von 0 zurück, wenn der Wert aktualisiert wurde. Gibt einen Wert von 1 zurück, wenn der Wert nicht mit dem Vergleichswert übereinstimmt.	**colName.** Der Name der Spalte. **rowIndex.** Die Indexnummer des Felds als Ganzzahl. 1 ist das erste Feld in der Spalte. **value.**Der Wert als Zeichenfolge. **comparedValue.** **VtsName.** Der Alias des VTS-Servers.	Wenn der Spaltenname oder Index nicht vorhanden ist, wird die Spalte und/oder die Zelle, auf die der Zeilenindex zeigt, erstellt, und die Zelleninhalte werden auf den Argumentwert festgelegt.

Methode	Beschreibung	Argumente	Hinweise zum Funktions-verhalten
LR.vtcUpdateRowCells (colNames,rowIndex,values,VtsName)	Ersetzt die Daten in einer Zeile.	**colNames.** Die Namen der Spalten. **rowIndex.** Die Indexnummer des Felds als Ganzzahl. 1 ist das erste Feld in der Spalte. **values.** Die Werte als Zeichenfolge, voneinander getrennt durch ein Semikolon. **VtsName.** Der Alias des VTS-Servers.	Wenn die Spaltennamen oder der Index nicht vorhanden sind, werden die Spalten und/oder die Zellen, auf die der Index verweist, erstellt, und die Zelleninhalte werden auf die Argumentwerte festgelegt.

TruClient-Eigenschaften

Diese Eigenschaften können als Argumente in TruClient-Funktionen verwendet werden.

Eigenschaft	Beschreibung
LR.userId	Die Benutzer-ID, wie sie in der MDRV-Befehlszeile erscheint. MDRV ist der Hauptprozess zum Ausführen aller Protokolle.
LR.groupName	Der Gruppenname, wie er in der MDRV-Befehlszeile erscheint. MDRV ist der Hauptprozess zum Ausführen aller Protokolle. Wurde der Prozess von VuGen gestartet, ist der Wert 0.
LR.scenarioId	Die Szenario-ID, wie sie in der MDRV-Befehlszeile erscheint. MDRV ist der Hauptprozess zum Ausführen aller Protokolle. Die Szenario-ID ist nur vorhanden, wenn MDRV vom Controller gestartet wurde. Beim Start durch VuGen ist der Wert 0.
LR.outputDir	Der Ausgabeordner mit der gesamten Ausgabe für das Skript. Bei VuGen sind Ausgabeordner und Skriptordner identisch. Beim Controller unterscheiden sich die Ordner. Der zurückgegebene Pfad enthält das Trennzeichen des letzten Ordners.
LR.scriptDir	Der Skriptordner des Benutzers. Sie können externe Dateien im Skriptordner speichern. Sollen diese im Skript enthalten sein, können Sie den Namen Ihrer externen Datei an "LR.scriptDir" anhängen.

Argumente von TruClient-Schritten

In der folgenden Tabelle sind die Schrittargumente kategorisiert nach Rolle aufgelistet. Erforderliche Argumente sind in der Benutzeroberfläche links neben dem Argumentnamen mit einem roten Stern gekennzeichnet. Alle Argumente können JavaScript-Code und LoadRunner-Funktionen als Werte akzeptieren. Eine Liste der LoadRunner-Funktionen finden Sie unter TruClient-Funktionen.

Rolle	Aktion	Argumente
Element	JavaScript auswerten	Code: JavaScript-Code
Element	Mausaktionen: Maus nach unten, Maus nach oben, Mauszeiger auf ein Objekt, Mausklick, Doppelklick	• **Taste.** Die geklickte Maustaste. • **X-Koordinate.** Die Offset-Position der Aktion in Bezug zur oberen linken Ecke des Objekts. Dieser Wert muss positiv sein. Wenn kein Wert angegeben wird, ist der Standard die Mitte des Objekts. • **Y-Koordinate.** Die Offset-Position der Aktion in Bezug zur oberen linken Ecke des Objekts. Wenn kein Wert angegeben wird, ist der Standard die Mitte des Objekts. • **Strg-Taste.** Ob diese Taste während der Aktion gedrückt wird oder nicht. • **Alt-Taste.** Ob diese Taste während der Aktion gedrückt wird oder nicht. • **Umschalttaste.** Ob diese Taste während der Aktion gedrückt wird oder nicht.
Element	Ziehen	• **Schaltfläche.** Die geklickte Maustaste. • **X-Offset.** Die Zahl der Pixel beim Ziehen des Objekts auf der X-Achse. Ein positiver Wert gibt ein Ziehen nach rechts an. • **Y-Offset.** Die Zahl der Pixel beim Ziehen des Objekts auf der Y-Achse. Ein positiver Wert gibt ein Ziehen nach unten an. • **Pfad.** Liste der Koordinaten, die den Ziehen-Pfad eines Benutzers repräsentieren. Ändern Sie dieses Argument nicht.

Rolle	Aktion	Argumente
Element	Ziehen nach	• **Zielobjekt.** Das Schrittobjekt wird auf dieses Zielobjekt gezogen. • **X-Offset.** Der Offset von der oberen linken Position des Zielobjekts auf der X-Achse. Dieser Wert muss positiv sein. • **Y-Offset.** Der Offset von der oberen linken Position des Zielobjekts auf der Y-Achse. Dieser Wert muss positiv sein.
Element	Eigenschaft abrufen	• **Eigenschaft.** Die Eigenschaft, deren Wert in der angegebenen Variable gespeichert wird. Die Liste der verfügbaren Eigenschaften hängt von der Rolle des Objekts ab. Die folgenden Standardeigenschaften stehen für alle Objekte zur Verfügung: ▪ **Sichtbarer Text.** Der sichtbare Text des Elements entsprechend der DOM textContent-Eigenschaft. ▪ **Gesamter Text.** Der gesamte Text des Elements entsprechend der DOM textContent-Eigenschaft. ▪ **Inner HTML.** Der Inner HTML-Wert des Elements entsprechend der DOM innerHTML-Eigenschaft. ▪ **Variable.** Der Name der Variablen, in der der angegebene Eigenschaftswert gespeichert werden soll.

Rolle	Aktion	Argumente
Element	Verifizieren	• **Wert.** Die zu überprüfende Zeichenfolge bzw. der Wert. • **Eigenschaft.** Die Objekteigenschaft, deren Wert überprüft wird. Die Liste der verfügbaren zu prüfenden Eigenschaften hängt von der Rolle des Objekts ab. Die folgenden Standardeigenschaften stehen zur Überprüfung aller Objekte zur Verfügung: ▪ **Sichtbarer Text.** Objekte, die in der Anwendung angezeigt werden. ▪ **Gesamter Text.** Objekte, die in der Anwendung vorhanden sind, aber nicht notwendigerweise angezeigt werden. Objekte in dieser Kategorie sind in der DOM-Eigenschaft **textContent** enthalten. ▪ **Inner HTML.** In der DOM-Eigenschaft **innerHTML** enthaltene Objekte. ▪ **Bedingung.** Die Beziehung zwischen dem Wert und den Eigenschaftsargumenten.

Rolle	Aktion	Argumente
Element	Warten auf Eigenschaft	• **Wert.** Der Wert der angegebenen Eigenschaft, auf die der Schritt wartet, bevor der Schritt erfolgreich ausgeführt wird. • **Eigenschaft.** Die Objekteigenschaft, auf deren Wert das Skript wartet. Die Liste der verfügbaren Eigenschaften, auf die gewartet werden kann, hängt von der Rolle des Objekts ab. Die folgenden Standardeigenschaften stehen für alle Objekte zur Verfügung: ▪ **Sichtbarer Text.** Objekte, die in der Anwendung angezeigt werden. ▪ **Gesamter Text.** Objekte, die in der Anwendung vorhanden sind, aber nicht notwendigerweise angezeigt werden. Objekte in dieser Kategorie sind in der DOM-Eigenschaft **textContent** enthalten. ▪ **Inner HTML.** In der DOM-Eigenschaft **innerHTML** enthaltene Objekte. ▪ **Bedingung.** Die Beziehung zwischen dem Wert und den Eigenschaftsargumenten.
Fokussierbar	Tastendruck	• **Tastenname.** Eingabe- oder Leertaste.
Textfeld	Typ	• **Wert.** Was eingegeben wird. • **Löschen.** Das Textfeld vor der Eingabe löschen. Die Standardeinstellung lautet **True**. • **Eingabeintervall.** Die durchschnittliche Zeit zwischen Tastenanschlägen in Millisekunden.
Kontrollkästchen	Einstellen	• **Aktiviert.** Einstellen des Kontrollkästchens auf aktiviert (T) oder deaktiviert (F).

Rolle	Aktion	Argumente
Listenfeld	Auswählen	• **Text.** Die ausgewählte Zeichenfolge. • **Ordnungszahl.** Die Reihenfolge des ausgewählten Elements in der Liste. Wenn das Textargument ebenfalls angegeben wird, bezieht sich dieses Argument auf die Instanz des angegebenen Textwerts in dem Listenfeld. Eine Ordnungszahl 0 generiert einen Zufallswert.
Optionsfeldgruppe	Auswählen	• **Text.** Die ausgewählte Zeichenfolge. • **Ordnungszahl.** Die Reihenfolge des ausgewählten Elements in der Liste. Wenn das Textargument ebenfalls angegeben wird, bezieht sich dieses Argument auf die Instanz des angegebenen Textwerts in dem Listenfeld. Eine Ordnungszahl 0 generiert einen Zufallswert.
Dateifeld	Einstellen	• **Pfad.** Der ausgewählte Pfad.
Schieberegler	Einstellen	• **Wert.** Der Wert, auf den der Schieberegler eingestellt ist.
Datumsauswahl	Tag festlegen	• **Tag.** Eine Ganzzahl zwischen 1 und 31, die den Tag des Monats repräsentiert.
Browser	Aktivieren	• **Ordnungszahl.** Definiert als Ganzzahl. Verschiebt das angegebene Browserfenster in den Vordergrund.
Browser	Registerkarte aktivieren	• **Ordnungszahl.** Die zu aktivierende Registerkarte (Ganzzahl).
Browser	Registerkarte schließen	• **Ordnungszahl.** Die zu schließende Registerkarte (Ganzzahl).
Browser	Registerkarte hinzufügen	• **Speicherort.** Der URL, zu dem in der neu geöffneten Registerkarte navigiert werden soll.
Browser	Navigieren	• **Speicherort.** Der URL, zu dem navigiert werden soll.
Browser	Zurück	• **Anzahl.** Die Anzahl der zurückzublätternden Seiten.

Rolle	Aktion	Argumente
Browser	Vorwärts	• **Anzahl.** Die Anzahl der vorzublätternden Seiten.
Browser	Größe ändern	• **Breite.** Die neue Breite. Wird kein Wert eingegeben, bleibt die Breite unverändert. • **Höhe.** Die neue Höhe. Wird kein Wert eingegeben, bleibt die Höhe unverändert.
Browser	Bildlauf	• **X-Koordinate.** Die neue X-Koordinate. Wird kein Wert eingegeben, bleibt die Position auf der X-Achse unverändert. • **Y-Koordinate.** Die neue Y-Koordinate. Wird kein Wert eingegeben, bleibt die Position auf der Y-Achse unverändert.
Browser	Dialogfeld - Bestätigen	• **Schaltfläche. Ok** oder **Abbrechen**.
Browser	Dialogfeldaufforderung	• **Wert.** Die einzugebende Zeichenfolge. • **Schaltfläche. Ok** oder **Abbrechen**.
Browser	Dialogfeld - Authentifizierung	• **Benutzername.** Der einzugebende Benutzername. • **Kennwort.** Das einzugebende Kennwort. • **Domäne.** Die einzugebende Domäne. • **Schaltfläche. Ok** oder **Abbrechen**.
Browser	Verifizieren	• **Wert.** Der zu verifizierende Wert der Eigenschaft. • **Eigenschaft.** Die zu verifizierende Eigenschaft. Sie können die folgenden Eigenschaften eines Browserobjekts verifizieren: • **Titel.** Der Titel des Browserfensters. • **Speicherort.** Der Speicherort des Browserfensters. • **Bedingung.** Die Beziehung zwischen dem Wert und den Eigenschaftsargumenten.

Citrix-Protokoll

Citrix-Protokoll - Übersicht

Citrix-Vuser-Skripts emulieren die Kommunikation zwischen einem Citrix-Client und einem Server über das Citrix ICA-Protokoll. VuGen zeichnet während der Kommunikation alle Aktivitäten auf und erstellt ein Vuser-Skript.

Wenn Sie Aktionen auf dem Remoteserver ausführen, generiert VuGen Funktionen zur Beschreibung dieser Aktionen. Alle diese Funktionen beginnen mit dem Präfix **ctrx**. Durch diese Funktionen werden analoge Mausbewegungen und Tastenanschläge emuliert. Darüber hinaus können Sie über die **ctrx**-Funktionen die Wiedergabe der Aktionen synchronisieren, indem auf das Öffnen bestimmter Fenster gewartet wird.

VuGen ermöglicht außerdem die Aufzeichnung einer Citrix NFuse-Sitzung. Mit Citrix NFUSE wird zwar der Client installiert, allerdings mit einem Browser anstelle einer Clientoberfläche. Für die Aufzeichnung von NFuse-Sitzungen müssen Sie eine Mehrfachprotokollaufzeichnung für Citrix und Web-Vusers durchführen. VuGen generiert bei der Aufzeichnung im Mehrfachprotokollmodus Funktionen aus Citrix- und Webprotokollen.

Im folgenden Beispiel wird mit der Funktion **ctrx_mouse_click** ein Klick mit der linken Maustaste simuliert.

```
ctrx_mouse_click(44, 318, LEFT_BUTTON, 0, CTRX_LAST);
```

Weitere Informationen zur Syntax und zu den Parametern finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Citrix-Aufzeichnungstipps

Gehen Sie beim Aufzeichnen eines Citrix Vuser-Skripts wie folgt vor, um ein möglichst wirksames Skript zu erhalten.

Einzel- und Mehrfachprotokoll-Skripts

Wenn Sie ein neues Skript erstellen, können Sie zwischen einem Einzel- und einem Mehrfachprotokoll wählen. Planen Sie die Aufzeichnung einer einfachen Citrix ICA-Sitzung, reicht ein Einzelprotokoll-Skript aus. Wenn Sie jedoch eine NFUSE Web Access-Sitzung aufzeichnen möchten, müssen Sie ein Mehrfachprotokollskript für Citrix ICA und Web (HTML/HTTP) erstellen, damit beide Protokolle berücksichtigt werden.

Aufzeichnung in den richtigen Abschnitten

Zeichnen Sie den Verbindungsprozess im Abschnitt **vuser_init** und den Trennungsprozess im Abschnitt **vuser_end** auf. Dadurch werden Iterationen beim Herstellen und Trennen der Verbindung verhindert. Weitere Informationen zum Aufzeichnen in Abschnitten finden Sie unter "Abschnitte des Vuser-Skripts" auf Seite 260.

Eine leere Sitzung ausführen

Vergewissern Sie sich beim Aufzeichnen einer Sitzung, dass Sie den gesamten Geschäftsprozess von Herstellung der Verbindung bis zur Bereinigung ausführen. Beenden Sie Ihre Sitzung immer an einem Punkt, an dem Sie den gesamten Prozess wieder von Beginn an starten könnten. Lassen Sie keine Client- oder Anwendungsfenster geöffnet.

Explizites Klicken

Wenn Sie auf die Optionen eines erweiterten Menüs zugreifen möchten, klicken Sie explizit auf jede einzelne Option und verlassen sich nicht auf das erweiterte Menü. Vergewissern Sie sich beispielsweise bei der Auswahl von **Start > Programme > Microsoft Word**, dass Sie explizit auf **Programme** klicken.

Beibehaltung der Fenstergröße

VuGen unterstützt zwar die Größenanpassung von Fenstern während der Sitzung, trotzdem empfiehlt es sich, Fenster während der Aufzeichnung weder zu verschieben noch in der Größe zu ändern. Wenn Sie die Größe oder Position eines Fensters ändern möchten, doppelklicken Sie im Schrittenavigator auf den entsprechenden Schritt **Synchronisierung bei Fenster** und bearbeiten die Fensterkoordinaten.

Konsistente Auflösungseinstellungen

Für eine erfolgreiche Bitmapsynchronisierung müssen die Auflösungseinstellungen übereinstimmen. Überprüfen Sie auf dem Aufzeichnungscomputer die Einstellungen des ICA-Clients, die Aufzeichnungsoptionen sowie die Laufzeiteinstellungen. Überprüfen Sie auf Lastgeneratoren die Einstellungen des ICA-Clients und vergewissern Sie sich, dass sie mit denen der Aufzeichnungscomputer übereinstimmen. Falls es zu einer Inkonsistenz zwischen den Auflösungen kommt, nimmt der Serverdatenverkehr zu, sodass Sie die erforderlichen Anpassungen vornehmen können.

Manuelle Synchronisierungspunkte hinzufügen

Wenn Sie während der Aufzeichnung auf ein Ereignis warten, z. B. darauf, dass eine Anwendung geöffnet wird, empfehlen wir das Hinzufügen von manuellen Synchronisierungspunkten wie **Synchronisierung bei Bitmap** oder **Synchronisieren bei Text**. Weitere Informationen finden Sie unter "Citrix - Automatische Synchronisierung" auf Seite 640.

Clientaktualisierungen deaktivieren

Deaktivieren Sie Clientaktualisierungen, wenn Sie vom Citrix-Client eine entsprechende Aufforderung erhalten. Dadurch werden Probleme im Zusammenhang mit der Aufwärtskompatibilität zwischen VuGen und neueren, bislang ungetesteten Citrix-Clients verhindert.

Windows-Stil

In **Synchronisierung bei Bitmap**-Schritten werden die Aufzeichnungsfenster im "klassischen" Windows-Stil und nicht im XP-Stil dargestellt.

Festlegen des "klassischen" Windows-Stils

1. Klicken Sie auf den Desktop.
2. Wählen Sie im Kontextmenü die Option **Personalisieren** aus.
3. Klicken Sie auf die Registerkarte **Design**.
4. Wählen Sie in der Liste **Design Windows - klassisch** aus.

5. Klicken Sie auf **OK**.

Citrix-Wiedergabetipps

Platzhalter

Für die Definition von Fensternamen können Sie Platzhalter (*) verwenden. Dies ist besonders nützlich, wenn sich das Präfix oder Suffix des Fensternamens möglicherweise bei der Wiedergabe ändert.

Im folgenden Beispiel wird der Titel des **Microsoft Internet Explorer**-Fensters mittels eines Platzhalters geändert.

```
ctrx_mouse_click(573, 61, LEFT_BUTTON, 0,
"Welcome to MSN.com - Microsoft Internet Explorer");ctrx_mouse_click(573, 61, LE
FT_BUTTON, 0,
"* - Microsoft Internet Explorer");
```

Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Festlegen der Höchstquote für die Initialisierung

Um bei der Verbindung mehrerer Vuser eine Überlastung zu verhindern, können Sie eine Höchstquote für die Initialisierung von 4 bis 10 Vusern (je nach Serverkapazität) festlegen oder mithilfe der Planung die Initialisierung allmählich aufstocken.

Aktivierung der Latenzzeit

Beste Ergebnisse erzielen Sie, wenn Sie Latenzzeit in den Laufzeiteinstellungen aktivieren. Latenzzeit ist von besonderer Bedeutung für die Funktionen **ctrx_sync_on_window** und **ctrx_sync_on_bitmap**; diese benötigen Zeit für die Stabilisierung.

Skript neu generieren

Während der Aufzeichnung speichert VuGen alle Agenteninformationen zusammen mit dem Skript. Standardmäßig werden bis auf die **Sync On Text**-Schritte auch alle Informationen innerhalb des Skripts gespeichert. Bei Problemen mit der Textsynchronisierung können Sie das Skript unter Einbeziehung der Textsynchronisierungsschritte erneut generieren.

Weiterhin können Sie, falls Sie die Erstellung von Agenteninformationen in den Aufzeichnungsoptionen zuvor deaktiviert hatten, das Skript erneut generieren, damit die Agenteninformationen berücksichtigt werden.

Die erneute Generierung von Skripts empfiehlt sich auch für manuell geänderte Skripts. Wenn Sie das Skript erneut generieren, verwirft VuGen alle manuellen Änderungen und stellt die ursprünglich aufgezeichnete Version wieder her.

Führen Sie zum erneuten Generieren des Skripts den Befehl **Aufzeichnen > Skript neu generieren** aus und wählen Sie die gewünschten Optionen aus. Weitere Informationen zum erneuten Generieren von Skripts finden Sie unter "Erneutes Generieren eines Vuser-Skripts" auf Seite 271.

Konsistenz zwischen Computern

Wenn Sie beabsichtigen, das Skript auf einem anderen Computer auszuführen, vergewissern Sie sich, dass auf dem aufzeichnenden und dem wiedergebenden Computer folgende Elemente übereinstimmen: Window Size (resolution), Window Colors, System Font und weitere Standardoptionen für den Citrix-Client. Diese Einstellungen wirken sich auf den Hashwert von Bitmaps aus, wobei eine fehlende Übereinstimmung zu Fehlern bei der Wiedergabe führen kann. Wählen Sie zum Anzeigen der Einstellungen des Citrix-Clients ein Element aus der Citrix-Programmgruppe aus und klicken Sie im Kontextmenü auf **Application Set Settings** oder **Custom Connection Settings**. Klicken Sie auf die Registerkarte **Default Options**.

Erhöhen der Anzahl von Vusern pro Lastgeneratorcomputer

Auf Lastgeneratorcomputern, auf denen Citrix-Vuser ausgeführt werden, kann die Anzahl der ausführbaren Vuser aufgrund der verfügbaren Grafikressourcen, der so genannten Graphics Device Interface (GDI), beschränkt sein. Zur Erhöhung der Vuser-Anzahl pro Computer können Sie eine Terminalserversitzung öffnen, die dann als zusätzlicher Lastgenerator fungiert.

Die GDI-Anzahl ist abhängig vom Betriebssystem. Die tatsächliche GDI-Anzahl (Graphics Device Interface) für ein System unter Verwendung von LoadRunner mit hoher Auslastung beträgt ca. 7.500. Die maximal verfügbare GDI-Anzahl auf Systemen unter Windows 2000 beträgt 16.384.

Weitere Informationen zum Erstellen einer Terminalserversitzung finden Sie unter Übersicht über Terminaldienste.

Hinweis: Standardmäßig wird in Terminalserversitzungen eine Palette mit 256 Farben verwendet. Wenn Sie für Lasttests eine Terminalsitzung verwenden möchten, vergewissern Sie sich, dass die Aufzeichnung auf Computern mit einer Palette mit 256 Farben erfolgt.

Citrix-Synchronisierung

Synchronisierung bedeutet in diesem Zusammenhang, auf Fenster und Objekte zu warten, bis diese zur Verfügung stehen, bevor eine Aktion ausgeführt wird. Dies ist zur Aufzeichnung von Citrix-Skripts erforderlich. Wenn beispielsweise während eines Skriptschritts ein Fenster geöffnet wird und im nächsten Schritt eine Aktion in diesem Fenster ausgeführt werden soll, dann kann der zweite Schritt erst dann stattfinden, wenn das Fenster geöffnet wurde. Damit bei der Skriptwiedergabe keine Fehler entstehen, generiert VuGen automatisch Funktionen, die das Skript synchronisieren, indem so lange gewartet wird, bis Fenster oder Objekte zur Verfügung stehen. Außerdem können Sie ein Skript auch manuell synchronisieren.

Informationen zur automatischen Synchronisierung finden Sie unter "Citrix - Automatische Synchronisierung" auf der nächsten Seite.

Informationen zum manuellen Setzen von Synchronisierungspunkten finden Sie unter "Manuelles Synchronisieren von Citrix-Skripts" auf Seite 647.

Citrix - Automatische Synchronisierung

Während der Aufzeichnung erstellt VuGen automatisch Schritte zur Synchronisierung der Skriptwiedergabe:

Sync on Window

Der **Sync On Window**-Schritt weist den Vuser an, vor der Fortsetzung der Wiedergabe auf ein bestimmtes Ereignis zu warten. Folgende Ereignisse sind verfügbar: **Create** und **Active**. Bei Auswahl des Create-Ereignisses muss der Vuser die Erstellung des Fensters abwarten. Bei Auswahl des Active-Ereignisses muss der Vuser die Erstellung und Aktivierung (Fokussierung) des Fensters abwarten. In der Regel generiert VuGen eine Funktion mit einem create-Ereignis. Wenn die nächste Anweisung allerdings ein Tastaturereignis darstellt, generiert das Programm eine Funktion mit einem active-Ereignis.

Im Editor lautet der entsprechende Funktionsaufruf für den Schritt **Synchronisierung bei Fenster** **ctrx_sync_on_window**.

Sync on Obj Info

Der **Sync On Obj Info**-Schritt weist den Vuser an, vor der Fortsetzung der Wiedergabe auf ein bestimmtes Objekt zu warten. Die verfügbaren Attribute lauten: **Enabled**, **Visible**, **Focused**, **Text**, **Checked**, **Lines** oder **Item**. Bei den Attributen **Enabled**, **Visible**, **Focused** und **Checked** handelt es sich um boolesche Werte, die den Wert **true** oder **false** annehmen können. Die übrigen Attribute erfordern einen Text- oder Zahlenwert.

Das hauptrangige Ziel dieses Schrittes besteht darin, dass auf die Fokussierung eines Objekts gewartet wird, bevor eine Aktion mit dem Objekt ausgeführt werden kann.

VuGen generiert automatisch **Sync On Obj Info**-Schritte, wenn der Citrix-Agent installiert und in den Aufzeichnungsoptionen die Option **Eingaben des Citrix-Agenten bei Codegenerierung verwenden** aktiviert ist. Diese Aufzeichnungsoption ist standardmäßig aktiviert. Weitere Informationen finden Sie unter "Citrix > Codegenerierung" auf Seite 165.

```
ctrx_sync_on_obj_info("Run=snapshot9", 120, 144, TEXT, "OK",
                                                CTRX_LAST);
```

Sync on Text

Der Textsynchronisierungsschritt **Sync On Text** weist den Vuser an, vor dem Fortfahren auf die Anzeige einer Textzeichenfolge an der angegebenen Position zu warten. Bei der Wiedergabe von **Synchronisieren bei Text** durchsuchen Vuser das Rechteck, dessen veränderbare Koordinaten in den Schritteigenschaften festgelegt sind, nach dem Text.

Achtung: Die maximal zulässige Länge einer Textzeichenfolge ist auf 255 Zeichen begrenzt.

Wenn Sie den Agenten installiert haben (siehe "Agent für Citrix Presentation Server – Übersicht" auf Seite 642), können Sie VuGen anweisen, vor jedem Klicken oder Doppelklicken automatisch einen Textsynchronisierungsschritt zu generieren. Standardmäßig ist die automatische Textsynchronisierung deaktiviert. Weitere Informationen finden Sie unter "Citrix > Codegenerierung" auf Seite 165.

Anmerkung: Wenn Sie einen Schritt mit deaktivierter Option ausführen, die Option aktivieren und das Skript anschließend erneut genrieren, fügt VuGen Textsynchronisierungsaufrufe in das gesamte Skript ein.

Im Editor lautet der entsprechende Funktionsaufruf für den Schritt **Synchronisieren bei Text ctrx_sync_on_text_ex**.

Das folgende Segment zeigt eine **ctrx_sync_on_text_ex**-Funktion, die während einer Citrix-Aufzeichnung mit installiertem HP Citrix-Agenten und aktivierter Textsynchronisierung aufgezeichnet wurde.

```
ctrx_sync_on_window ("ICA Seamless Host Agent", ACTIVATE, 0, 0,391,224, "snapsho
t1", CTRX_LAST);
ctrx_sync_on_text_ex (196, 198, 44, 14, "OK", "ICA Seamless Host Agent=snapshot
2", CTRX_LAST);
ctrx_obj_mouse_click ("<class=Button text=OK>", 196, 198, LEFT_BUTTON, 0, "ICA S
eamless Host Agent=snapshot2", CTRX_LAST);
```

Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Weitere Informationen finden Sie unter "Citrix - Alternative Synchronisierung" unten.

Citrix - Alternative Synchronisierung

Neben der automatischen Synchronisierung in Citrix Vuser-Skripts können Sie verschiedene andere Schritte hinzufügen, die sich indirekt auf die Synchronisierung auswirken:

Festlegen der Wartezeit

Mit dem Schritt **Wartezeit festlegen** stellen Sie die Wartezeit für die anderen Citrix-Synchronisierungsfunktionen ein. Diese Einstellung gilt dann jeweils für alle folgenden Funktionen im Skript. Wenn es beispielsweise bei Ihren **Synchronisierung bei Fenster**-Schritten zu einer Zeitüberschreitung kommt, können Sie die Standardzeitüberschreitung von 60 auf 180 Sekunden erhöhen.

Wählen Sie zum Einfügen dieses Schrittes die Option **Einfügen > Schritt hinzufügen > Wartezeit festlegen** aus.

Überprüfen, ob ein Fenster vorhanden oder geschlossen ist

Mit dem Schritt **ctrx_win_exist** können Sie herausfinden, ob ein Fenster im Citrix-Client sichtbar ist. Wenn Sie Ablaufanweisungen für Steuerelemente hinzufügen, können Sie diese Funktion dazu verwenden, nach Fenstern zu suchen, die sich nicht immer öffnen, z. B. Dialogfelder mit Warnungen. Im folgenden Beispiel wird mit der **ctrx_win_exist**-Funktion überprüft, ob ein Browserfenster gestartet wurde. Hierbei wird durch das zweite Argument angegeben, wie lange auf das Öffnen des Fensters gewartet werden soll. Wenn es sich nicht innerhalb der angegebenen Zeit öffnet, wird ein Doppelklick auf das Symbol des Fensters ausgeführt.

```
if (!ctrx_win_exist("Welcome",6, CTRX_LAST))
   ctrx_mouse_double_click(34, 325, LEFT_BUTTON, 0, CTRX_LAST)
```

Wählen Sie zum Einfügen dieses Schrittes die Option **Einfügen > Schritt hinzufügen > Fenster vorhanden** aus.

Des Weiteren lässt sich mit diesem Schritt herausfinden, ob ein Fenster geschlossen wurde. Wenn Sie darauf warten müssen, dass ein Fenster geschlossen wird, verwenden Sie einen Synchronisierungsschritt wie **UnSet Window** oder **ctrx_unset_window**.

Ausführliche Informationen zu diesen Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Warten auf eine Bitmapänderung

Manchmal wissen Sie nicht genau, welche Daten oder Bilder in einem bestimmten Bereich dargestellt werden, möchten aber eine Änderung daran vornehmen. Um dies zu emulieren, können Sie den **Sync on Bitmap Change**-Schritt bzw. dessen entsprechende Funktion **ctrx_sync_on_bitmap_change** verwenden. Klicken Sie mit der rechten Maustaste in den Snapshot und wählen Sie im Kontextmenü die Option **Einfügen: Synchronisierung bei Bitmap** aus. VuGen fügt den Schritt oder die Funktion an der Cursorposition ein.

Die Funktionen weisen folgende Syntax auf:

```
ctrx_sync_on_bitmap (x_start, y_start, width, height, hash, CTRX_LAST);
ctrx_sync_on_bitmap_change (x_start, y_start, width, height,
                                        [initial_wait_time,] [timeout,]
[initial_bitmap_value,] CTRX_LAST);
```

Für **ctrx_sync_on_bitmap_change** stehen folgende optionale Argumente zur Verfügung:

- Anfangswert für Wartezeit – Startzeitpunkt der Überprüfung auf eine Änderung.
- Zeitüberschreitung – Wartezeit (in Sekunden) auf eine Änderung, bevor ein Fehler ausgegeben wird.
- Anfangswert für Bitmap – Anfangshashwert der Bitmapgrafik. Vuser warten, bis sich der Hashwert vom angegebenen Anfangswert für die Bitmapgrafik unterscheidet.

Im folgenden Beispiel wurde die aufgezeichnete Funktion geändert und ihr wurde ein Anfangswert für die Wartezeit von 300 Sekunden sowie eine Zeitüberschreitung von 400 Sekunden zugewiesen.

```
ctrx_sync_on_bitmap_change(93, 227, 78, 52,
                          300,400, "66de3122a58baade89e63698d1c0d5dfa",CTRX_LAS
T);
```

Hinweis: Vergewissern Sie sich bei der Verwendung von **Sync on Bitmap**, dass die Konfigurationseinstellungen von Controller, Lastgeneratorcomputer und Bildschirm übereinstimmen. Andernfalls kann es passieren, dass VuGen die richtigen Bitmaps bei der Wiedergabe nicht findet. Weitere Informationen zur Konfiguration der Clienteinstellungen finden Sie unter "Aufzeichnungsoptionen" auf Seite 164.

Agent für Citrix Presentation Server – Übersicht

Der Agent für Citrix Presentation Server, auch Citrix Agent, ist ein optionales Dienstprogramm, das auf dem Citrix-Server installiert werden kann. Dieses optimiert die reguläre Citrix-Funktionalität. In

den folgenden Abschnitten werden die Verbesserungen einzeln vorgestellt.

Sie finden das Dienstprogramm auf dem Installationsdatenträger des Produkts; es kann auf jedem beliebigen Citrix-Server installiert werden. Weitere Informationen zur Installation von Citrix Agent finden Sie unter "Installieren bzw. Deinstallieren des Citrix-Agenten" auf Seite 648.

Detaillierte Aufzeichnung von Objekten

Wenn der Agent für Citrix Presentation Server installiert ist, zeichnet VuGen anstelle von allgemeinen Information zur Aktion spezifische Informationen zum aktiven Objekt auf. Beispielsweise generiert VuGen anstelle der Schritte **Mouse Click** und **Mouse Double Click** bei installiertem Agenten die Schritte **Obj Mouse Click** und **Obj Mouse Double Click**.

Das folgende Beispiel demonstriert die Aufzeichnung einer Mausklickaktion mit und ohne den Agenten. Hinweis: Mit dem Agenten generiert VuGen für alle Mausaktionen ctrx_obj_xxx-Funktionen, z. B. Doppelklicken und Loslassen der Maustaste.

```
/* WIthout Agent Installation */
ctrx_mouse_click(573, 61, LEFT_BUTTON, 0, test3.txt - Notepad);
/* WIth Agent Installation */
ctrx_obj_mouse_click("<text=test3.txt - Notepad class=Notepad>" 573,
         61, LEFT_BUTTON, 0, test3.txt - Notepad=snapshot21, CTRX_LAST);
```

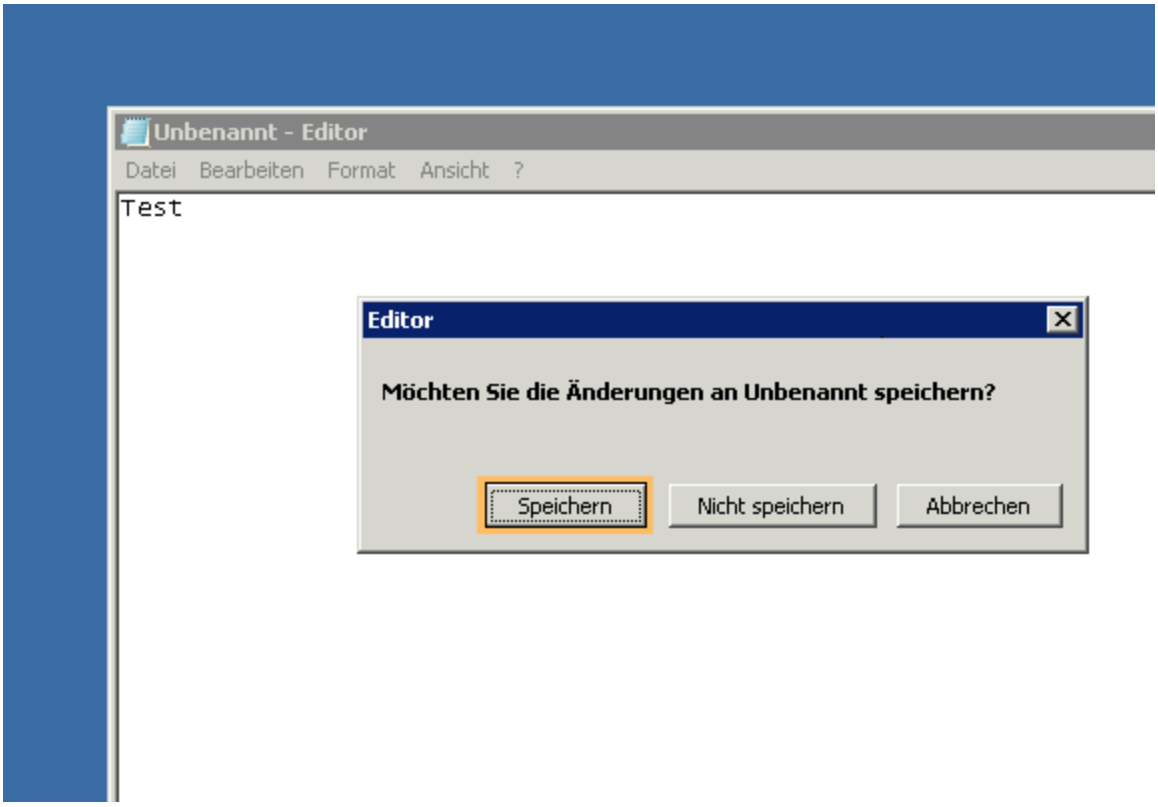
Im obigen Beispiel enthält das erste Argument der Funktion **ctrx_obj_mouse_click** den Text des Fenstertitels sowie die Klasse, Notepad. Hinweis: Obwohl der Agent zusätzliche Information zu den einzelnen Objekten bereitstellt, können Vuser nur über den Fensternamen und die Koordinaten auf das Objekt zugreifen.

Aktive Objekterkennung

In der Agenteninstallation können Sie sehen, welche Objekte im Clientfenster von VuGen erkannt werden. Hierzu gehören alle einfachen Windows-Objekte wie Bearbeitungsfelder, Schaltflächen und Listen im aktuellen Fenster.

Bewegen Sie die Maus im Snapshot, um zu sehen, welche Objekte im einzelnen erkannt wurden. VuGen hebt die Ränder der erkannten Objekte hervor, wenn Sie mit der Maus darüber fahren.

Im folgenden Beispiel gehört die Schaltfläche **Yes** zu den erkannten Objekten.



Erweitertes Kontextmenü

Wenn Sie in einen Snapshot klicken, können Sie über das Kontextmenü verschiedene Funktionen in das Skript einfügen. Ohne den Agenten sind lediglich die Optionen **Insert Mouse Click**, **Insert Mouse Double Click**, **Insert Sync on Bitmap** und **Insert Get Bitmap Value** verfügbar. Bei Verwendung einer Palette mit 256 Farben fehlen dem Kontextmenü die Schritte **Insert Sync on Bitmap** und **Get Bitmap Value**.

Bei installiertem Agenten für Citrix Presentation Server finden Sie im Kontextmenü des fokussierten Fensters zusätzlich folgende Optionen vor:

- **Obj Get Info** und **Sync on Obj Info**. Stellt Information zum Objektstatus bereit: ENABLED, FOCUSED, VISIBLE, TEXT, CHECKED und LINES.
- **Insert Sync on Obj Info**. Weist VuGen an, vor dem Fortfahren einen bestimmten Status abzuwarten. Diese Aktion wird als **ctrx_sync_on_obj_info**-Funktion generiert.
- **Insert Obj Get Info**. Ruft den aktuellen Status einer beliebigen Objekteigenschaft ab. Diese Aktion wird als **ctrx_get_obj_info**-Funktion generiert.
- **Insert Sync on Text** und **Get Text**. Siehe Abschnitt "Agent for Citrix Presentation Server - Overview".

Diese Befehle sind interaktiv, d. h. wenn Sie sie in das Skript einfügen, markieren Sie gleichzeitig das entsprechende Objekt oder den Textbereich im Snapshot.

Im folgenden Beispiel bewirkt die Funktion **ctrx_sync_on_obj_info** eine Synchronisierung, da sie veranlasst, dass auf die Fokussierung des Dialogfelds **Schriftart** gewartet wird.

```
ctrx_sync_on_obj_info("Font", 31, 59, FOCUSED, "TRUE", CTRX_LAST);
```

Mit der Objekterkennungsfunktion von VuGen können Sie aus dem Snapshot heraus Aktionen interaktiv für bestimmte Objekte ausführen.

Interaktives Einfügen einer Funktion mit dem Agenten

1. Klicken Sie im Schrittenavigator an die Stelle, an der Sie den neuen Schritt einfügen möchten. Vergewissern Sie sich, dass der Snapshot sichtbar ist.

2. Klicken Sie in den Snapshot.

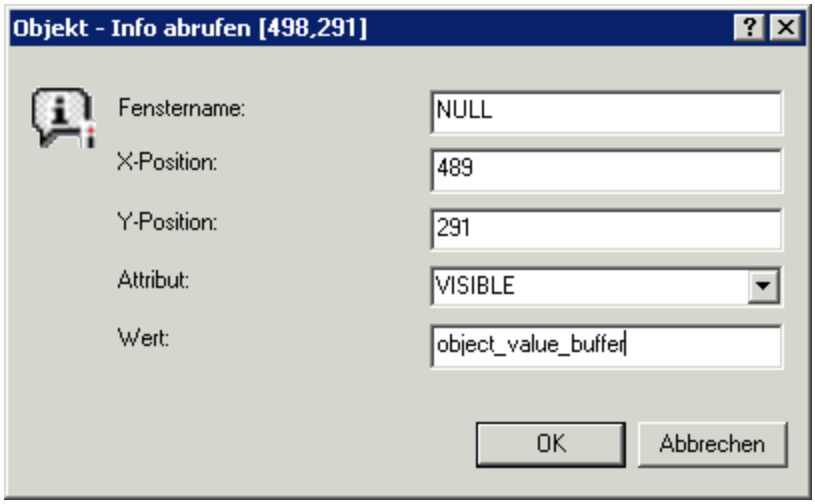
3. Klicken Sie mit der rechten Maustaste auf die Bitmapgrafik, um diese zu markieren, und wählen Sie **Insert Sync on Bitmap** aus.

   VuGen informiert Sie mit einer Meldung darüber, dass Sie den gewünschten Bereich durch Ziehen mit dem Cursor markieren müssen. Klicken Sie auf **OK** und ziehen Sie den Cursor diagonal über die Bitmapgrafik, die Sie auswählen möchten.

   Wenn Sie die Maustaste loslassen, fügt VuGen den Schritt hinter dem aktuell ausgewählten Schritt in das Skript ein.

4. Bewegen Sie für alle anderen Schritte die Maus über die Snapshotobjekte, um herauszufinden, welche Elemente aktiv sind. VuGen hebt die Ränder der aktiven Objekte hervor, wenn Sie mit der Maus darüber fahren.

   Klicken Sie mit der rechten Maustaste und wählen Sie einen der **Insert**-Befehle aus. Ein Dialogfeld mit den Eigenschaften des Schritts wird geöffnet.



   Legen Sie die gewünschten Eigenschaften fest und klicken Sie auf **OK**. VuGen fügt den Schritt in Ihr Skript ein.

Abruf von Text

Bei installiertem Agenten können Sie in VuGen Standardtext in einem Puffer speichern. Hinweis: VuGen kann lediglich reinen Text speichern und keine grafische Darstellung von Text in Form von Bildern.

Speichern Sie den Text mithilfe des **Sync On Text**-Schrittes entweder während oder nach der Aufzeichnung.

Weitere Informationen über die Aufgabe finden Sie unter "Manuelles Synchronisieren von Citrix-Skripts" auf der nächsten Seite.

Fehlerbehebung bei XenApp 5.0

Um ein Citrix- und Web-Skript mit mehreren Protokollen mithilfe von XenApp 5.0 aufzuzeichnen, müssen manuelle Änderungen vorgenommen werden, um eine einwandfreie Aufzeichnung zu gewährleisten.

Aufzeichnen mit XenApp 5.0

1. **Ändern der XenApp-Einstellungen:**

   a. Greifen Sie über einen Webbrowser auf den XenApp-Server zu.

   b. Wählen Sie **Preferences > Session Settings** aus, und stellen Sie **Window Size** auf **No Preference** ein.

2. **Ändern der Korrelationsregeln**

3. **Ändern der Aufzeichnungsoptionen**

   a. Erweitern Sie im Dialogfeld **Aufzeichnungsoptionen** den Knoten **Allgemein > Aufzeichnung**.

   b. Wählen Sie **Erweitertes HTML** aus, um das Dialogfeld **Erweitertes HTML** zu öffnen.

   c. Wählen Sie **Ein Skript, das nur explizite URLs enthält** aus.

4. **Manuelles Ändern des Skripts**

Konfigurieren von Citrix-Client und -Server

Vergewissern Sie sich vor der Skripterstellung, dass auf Ihrem Computer ein unterstützter Citrix-Client installiert und der Server ordnungsgemäß konfiguriert ist.

Konfigurieren des Citrix-Client

Damit Sie Ihr Skript ausführen können, müssen Sie auf jedem Lastgeneratorcomputer einen Citrix-Client installieren. Wenn Sie noch keinen Client installiert haben, können Sie ihn von der Citrix-Website unter der Adresse **www.citrix.com** im Abschnitt **Download** herunterladen.

VuGen unterstützt Citrix-Webclients und alle Citrix-Clients, ausgenommen Version 8.00 sowie Version 6.30.1060 und älter.

Installieren des MetaFrame-Servers

Vergewissern Sie sich, dass MetaFrame Server (Version 3, 4, oder 4.5) installiert ist. Wählen Sie zum Überprüfen der Serverversion in der Symbolleiste der Serverkonsole die Option **Citrix Connection Configuration** aus und klicken Sie auf **Help > About**.

Konfigurieren des MetaFrame-Servers

Konfigurieren Sie den Citrix-Server so, dass eine Sitzung vollständig geschlossen wird. Wenn ein Citrix-Client die Verbindung beendet, ist der Server standardmäßig so konfiguriert, dass die Sitzung bis zum nächsten Öffnen einer neuen Verbindung durch den Client gespeichert wird. Wenn über denselben Client eine neue Verbindung hergestellt wird, wird folglich derselbe Arbeitsbereich angezeigt, über den die Verbindung zuvor beendet wurde. Es empfiehlt sich jedoch, jeden Test in einem leeren Arbeitsbereich auszuführen.

Damit jeder Test in einem leeren Arbeitsbereich ausgeführt wird, müssen Sie den Citrix-Server so konfigurieren, dass die vorherige Sitzung nicht gespeichert wird. Die Verbindung soll statt dessen bei jeder Zeitüberschreitung und jedem Verbindungsabbruch des Clients zurückgesetzt werden.

Konfigurieren des Servers

1. Öffnen Sie das Dialogfeld **Citrix Connection Configuration**. Wählen Sie dazu die Option **Programme > Citrix > Administration Tools > Citrix Connection Configuration Tool** aus.

2. Wählen Sie den Namen einer ICA-TCP-Verbindung aus und wählen Sie anschließend die Option **Connection > Edit** aus. Alternativ können Sie auch auf die Verbindung doppelklicken. Das Dialogfeld **Edit Connection** wird geöffnet.

3. Klicken Sie auf **Erweitert**. Das Dialogfeld **Advanced Connection** wird geöffnet.

4. Deaktivieren Sie im unteren Bereich des Dialogfelds neben dem Listenfeld **On a broken or timed-out connection** das Kontrollkästchen **inherit user config**. Ändern Sie den Listeneintrag in **reset**.

Manuelles Synchronisieren von Citrix-Skripts

Zusätzlich zur automatischen Synchronisierung können Sie die Synchronisierung sowohl vor als auch nach der Aufzeichnung manuell hinzufügen. Diese Funktion wird häufig dann verwendet, wenn nicht das Fenster an sich, sondern nur ein Objekt innerhalb des Fensters geändert wurde. Da das Fenster unverändert ist, hat VuGen keinen **Sync on Window**-Schritt erkannt oder aufgezeichnet.

Wenn Sie beispielsweise erreichen möchten, dass mit der Wiedergabe gewartet wird, bis ein Browserfenster eine bestimmte Grafik enthält, fügen Sie eine manuelle Synchronisierung ein. Auch wenn Sie ein großes Fenster mit mehreren Registerkarten aufzeichnen, können Sie mit einem Synchronisierungsschritt bewirken, dass die Darstellung des Inhalts der nächsten Registerkarte abgewartet wird.

Manuelles Synchronisieren während der Aufzeichnung

Verwenden Sie zum Hinzufügen von Synchronisierungsfunktionen während der Aufzeichnung die unverankerte Symbolleiste. Mit den Funktionen **Synchronisierung bei Bitmap** und **Synchronisieren bei Text** können Sie einen Bereich oder Text im Clientfenster markieren, der fokussiert sein muss, bevor die Wiedergabe fortgesetzt wird.

- Um einen Schritt vom Typ **Synchronisierung bei Bitmap** einzufügen, klicken Sie auf die Schaltfläche **EinfügenSynchronisierung bei Bitmap** in der Symbolleiste und ziehen einen rechteckigen Rahmen um den gewünschten Bereich.
- Um einen Schritt vom Typ **Synchronisieren bei Text** einzufügen, (Citrix-Agent erforderlich), klicken Sie auf die Schaltfläche **Synchronisierung bei Text-Schritt einfügen** in der Symbolleiste, und ziehen Sie um den gewünschten Text einen rechteckigen Rahmen auf.

Manuelles Synchronisieren nach der Aufzeichnung

Sie können auch nach der Aufzeichnung Synchronisierungsfunktionen hinzufügen. Klicken Sie zum Hinzufügen eines Synchronisierungsschrittes mit der rechten Maustaste in das Snapshotfenster und wählen Sie eine Synchronisierungsoption aus:

- **Sync on Bitmap.** Wartet auf die Anzeige einer Bitmapdatei.
- **Sync on Obj Info** Wartet, bis die Attribute eines Objekts die angegebenen Werte aufweisen (gilt nur bei Installation des Agenten).
- **Sync on Text.** Wartet, bis der angegebene Text angezeigt wird (gilt nur bei Installation des Agenten)

Installieren bzw. Deinstallieren des Citrix-Agenten

Die Installationsdatei für den Citrix Presentation Server-Agenten befindet sich auf dem LoadRunner-Installationsdatenträger im Ordner **Additional Components\Agent for Citrix Presentation Server**.

Hinweis: Der Agent darf nur auf Citrix-Servern, nicht auf Lastgenerator-Computern installiert werden.

Installieren von Agent für Citrix Presentation Server:

1. Wenn Sie ein Upgrade des Agenten durchführen, vergewissern Sie sich vor der Installation, ob die Vorversion entfernt wurde.
2. Wenn auf Ihrem Server für die Installation von Software Administratorberechtigungen erforderlich sind, melden Sie sich als Administrator an.
3. Suchen Sie die Installationsdatei **SetupCitrixAgent<arch>.exe** auf dem

Installationsdatenträger zum Produkt im Ordner **Additional Components\ Agent for Citrix Presentation Server\Win32 or Win64**.

4. Folgen Sie den Anweisungen des Installations-Assistenten.

Hinweis: Nach der Installation ist der Agent nur für Citrix-Sitzungen aktiv, die von LoadRunner aufgerufen wurden. Es können keine Citrix-Sitzungen ohne LoadRunner gestartet werden.

Deinstallieren von Agent für Citrix Presentation Server:

1. Wenn auf Ihrem Server für die Entfernung von Software Administratorberechtigungen erforderlich sind, melden Sie sich als Administrator an.

2. Öffnen Sie auf dem Servercomputer in der Systemsteuerung die Option **Software**. Wählen Sie **HP Software Agent for Citrix Presentation Server 32 bzw. 64** und klicken Sie auf **Ändern/Entfernen**.

Citrix-Funktionen

Bei einer Citrix-Aufzeichnungssitzung generiert VuGen Funktionen zur Emulation der Kommunikation zwischen einem Client und einem Remoteserver. Die generierten Funktionen haben das Präfix **ctrx**. Durch die Funktion **ctrx_obj_mouse_click** wird beispielsweise ein Mausklick für ein bestimmtes Objekt emuliert. Sie können alle Funktionen nach der Aufzeichnung auch manuell in Ihrem Vuser-Skript bearbeiten oder weitere Funktionen hinzufügen.

Weitere Informationen zu den **ctrx**-Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Hinweis: Für Funktionen, mit denen ein Fenstername angegeben wird, können Sie als Platzhaltersymbol einen Stern (*) verwenden. Sie können den Platzhalter am Anfang, in der Mitte oder am Ende der Zeichenfolge positionieren.

Citrix - Informationen über ICA-Dateien

Bei Citrix ICA-Clientdateien handelt es sich um Textdateien mit Konfigurationsinformationen für die Anwendungen, auf die der Citrix-Client zugreift. Diese Dateien erhalten die Erweiterung **.ica** und weisen das folgende Format auf:

```
[WFClient]
Version=
TcpBrowserAddress=

[ApplicationServers]
AppName1=
```

```
[AppName1]
Address=
InitialProgram=#
ClientAudio=
AudioBandwidthLimit=
Compress=
DesiredHRES=
DesiredVRES=
DesiredColor=
TransportDriver=
WinStationDriver=
Username=
Domain=
ClearPassword=
```

Hinweis: Wenn Sie mithilfe der Aufzeichnungsoptionen eine ICA-Datei laden, speichert VuGen die Datei zusammen mit Ihrem Skript. Dadurch ersparen Sie sich das Kopieren der ICA-Datei auf jeden einzelnen Lastgeneratorcomputer.

Im Folgenden sehen Sie eine ICA-Beispieldatei für die Verwendung von Microsoft Word auf einem Remotecomputer über den Citrix-Client:

```
[WFClient]
Version=2
TcpBrowserAddress=235.119.93.56

[ApplicationServers]
Word=

[Word]
Address=Word
InitialProgram=#Word
ClientAudio=On
AudioBandwidthLimit=2
Compress=On
DesiredHRES=800
DesiredVRES=600
DesiredColor=2
TransportDriver=TCP/IP
WinStationDriver=ICA 3.0
Username=test
Domain=user_lab
ClearPassword=test
```

Weitere Informationen finden Sie auf der Citrix-Website www.citrix.com..

Dialogfeld "Fehler bei der Bitmapsynchronisierung"

In diesem Dialogfeld können Sie festlegen, wie bei einer fehlgeschlagenen Bitmapsynchronisierung vorzugehen ist.

Zugriff	Wenn bei einem Bitmapsynchronisierungsschritt eine Abweichung zwischen dem Aufzeichnungs- und Wiedergabe-Snapshot vorliegt, wird dieses Dialogfeld automatisch während der Wiedergabe geöffnet.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Weiter	Abweichung annehmen und ursprünglichen und neuen Snapshot als Vergleichsbasis für Bildschirme bei der künftigen Wiedergabe verwenden. Wenn bei der Wiedergabe eine der beiden Bitmaps zurückgegeben wird, wird kein Vuser-Fehler ausgegeben.
Aufzeichnungs-Snapshot	Ansicht des Aufzeichnungs-Snapshots.
Wiedergabe-Snapshot	Ansicht des Wiedergabe-Snapshots.
Anhalten	Die Abweichung wird als Fehler behandelt. Dieser Fehler wird wie alle anderen Fehler behandelt, d. h. die Ausführung wird angehalten. Alternativ dazu können Sie für eine bestimmte Funktion die Option **Bei Fehler fortfahren** festlegen, wie in "Citrix - Fehlerbehebung und Einschränkungen" unten beschrieben.

Citrix - Fehlerbehebung und Einschränkungen

In diesem Abschnitt werden die Fehlerbehebung und die für Citrix-Vuser geltenden Einschränkungen beschrieben.

Allgemeine Einschränkungen

- Der Registrierungspatch muss installiert werden, damit Citrix-Clients höher als Version 10.x unterstützt werden können. Darüber hinaus muss **Enable_Citrix_API.reg** aus dem Ordner **LoadRunner\dat** auf Vugen- oder Load Generator-Computern installiert werden, wenn ein Citrix-Client nach der Installation von LoadRunner installiert wird.

- Das Ausführen von Citrix Vusern auf virtuellen Computern kann sich aufgrund der gemeinsamen Nutzung der physischen Ressourcen ungünstig auf die Leistung auswirken.

- Citrix XenApp Desktop kann nicht mit Citrix Web Access (vormals Citrix NFuse) aufgezeichnet werden, wenn Desktop View (Desktop-Symbolleiste) aktiviert ist.
- Die Texterkennung funktioniert bei überlappenden Anwendungen auf Windows 2012-Servern möglicherweise nicht korrekt.
- Stellen Sie für die Aufzeichnung von Citrix Web Access-Anwendungen (vormals Citrix NFuse) die Fenstergröße auf der Konfigurationsseite des Citrix XenApp-Webportals so ein, dass keine Voreinstellung ausgewählt wird.
- Beim Aufzeichnen der Fenstergröße mithilfe des XenApp-Plug-Ins für Hosted Applications 11. Die Option zum Aufzeichnen der Fenstergröße funktioniert beim XenApp-Plug-In für Hosted Applications 11 nicht ordnungsgemäß. Die Größe des Clientfensters wird festgelegt; die Auflösung des Serverbildschirms jedoch nicht. Hierbei handelt es sich um einen Fehler des Citrix-Clients, der in zukünftigen Citrix-Client-Versionen behoben wird.
  Problemumgehung: Weisen Sie der Fenstergröße während der Aufzeichnung den Wert der lokalen Bildschirmauflösung zu. Weisen Sie bei der Wiedergabe bzw. bei Lasttests der VuGen- oder Lastgenerator-Bildschirmauflösung den Wert zu, der bei Aufzeichnung des Skripts verwendet wurde. Die bei der Aufzeichnung verwendete Auflösung finden Sie in der Window-Eigenschaft in der Datei **<Skriptordner>\default.cfg**.
- Das Citrix Connection Center verhindert möglicherweise die Aufzeichnung und Wiedergabe von Citrix ICA-Skripts, wenn es in einer anderen Benutzersitzung auf demselben Computer ausgeführt wird.

  Lösung: Schließen Sie alle Instanzen des Prozesses *concenter.exe* für alle Benutzer. Damit das Citrix Connection Center nicht automatisch gestartet wird, legen Sie den Registrierungsschlüssel *ConnectionCenter* auf einen leeren Wert fest"". Diesen Schlüssel finden Sie unter:
  32-Bit-Systeme: HKEY_LOCAL_
  MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
  64-Bit-Systeme: HKEY_LOCAL_
  MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\Curr

Auswirkungen und Speicheranforderungen des Citrix-Agenten

Wenn Sie Citrix-Vuser mit installiertem Agenten ausführen, wird für jeden Vuser ein eigener **ctrxagent.exe**-Prozess ausgeführt. Das führt dazu, dass die Anzahl der auf dem Server ausführbaren Vuser sich um ca. 7 % verringert.

Der Speicherbedarf pro Citrix ICA-Vuser liegt bei etwa 4,35 MB, wenn für jeden Vuser ein eigener **ctrxagent.exe**-Prozess ausgeführt wird. Für die Ausführung von 25 Vusern sind folglich 110 MB Arbeitsspeicher erforderlich.

Data Execution Prevention (DEP) und Leistung des Citrix-Agenten

Bei DEP handelt es sich um eine Sicherheitsfunktion, die in Windows-Versionen ab Windows XP Service Pack 2 enthalten ist. DEP kann zu Beeinträchtigungen bei einigen Funktionen von Agent für Citrix Presentation Server führen, u. a. auch dazu, dass VuGen die Aufzeichnung abbricht.

Wenn ein ungewöhnliches Verhalten während der Aufzeichnung in diesen Umgebungen auftritt, müssen Sie die Einstellungen der Datenausführungsverhinderung ändern.

So ändern Sie die Einstellungen der Windows-Datenausführungsverhinderung

1. Wählen Sie **Start > Systemsteuerung > System**.
2. Klicken Sie auf der Registerkarte **Erweitert** unter **Systemleistung** auf **Einstellungen**.
3. Wählen Sie auf der Registerkarte **Datenausführungsverhinderung** die erste Option **Datenausführungsverhinderung nur für erforderliche Windows-Programme und -Dienste aktivieren**.

   Wenn Sie diese Option nicht ändern können, klicken Sie auf **Hinzufügen**. Suchen Sie das Client-Programm, zum Beispiel IEXPLORE.EXE.
4. Wenn keine dieser beiden Optionen möglich ist, versuchen Sie die Datenausführungsverhinderung vollständig zu deaktivieren.
   a. Öffnen Sie eine Befehlszeile.
   b. Führen Sie den folgenden Befehl aus: **bcdedit.exe /set {current} nx AlwaysOff**
   c. Starten Sie den Computer neu.
   d. Stellen Sie sicher, dass die Einstellungen wirksam wurden, indem Sie folgenden Befehl an der Befehlszeile ausführen: **BCDEdit /enum**
   e. Stellen Sie sicher, dass in der letzten Zeile **nx AlwaysOff** angezeigt wird.

Debuggingtipps

Im folgenden Abschnitt finden Sie Hinweise zur Fehlerbehebung in Citrix-Skripts.

Installation mit einem einzelnen Client

Wenn Sie in Ihrer Citrix-Sitzung keine Aktionen aufzeichnen können, vergewissern Sie sich, dass auf Ihrem Computer nur ein Citrix-Client installiert ist. Öffnen Sie hierzu in der Systemsteuerung das Dialogfeld **Software** und überprüfen Sie, ob für den Citrix ICA-Client ein Eintrag vorhanden ist.

Hinzufügen von Haltepunkten

Durch Hinzufügen von Breakpoints zu Ihrem Skript in VuGen können Sie problematische Codezeilen bestimmen.

Synchronisieren des Skripts

Bei einem Wiedergabefehler müssen Sie möglicherweise Synchronisierungsfunktionen in Ihr Skript einfügen, um dem gewünschten Fenster mehr Zeit für die Fokussierung zu gewähren. Sie können zwar mithilfe der Funktion **lr_think_time** manuell eine Verzögerung einfügen, trotzdem empfiehlt es sich, eine der unter "Citrix - Automatische Synchronisierung" auf Seite 640 beschriebenen Synchronisierungsfunktionen zu verwenden.

Fortsetzen bei Fehler

Sie können Vuser auch nach dem Auftreten von Fehlern weiterhin ausführen lassen, z. B. wenn das passende Fenster nicht gefunden wird. Sie können die Option **Bei Fehler fortfahren** pro Schritt einzeln festlegen.

Dies ist besonders hilfreich, wenn Sie nicht wissen, welches von zwei Fenstern geöffnet wird. Beide Fenster sind zulässig, aber nur eines wird tatsächlich geöffnet.

Geben Sie an, ob bei einem Fehler fortgefahren werden soll:

Klicken Sie im **Schrittenavigator** mit der rechten Maustaste auf den Schritt und wählen Sie **Argumente anzeigen** aus. Wählen Sie im Feld **Continue on Error** die Option **CONTINUE_ON_ERROR** aus.

Suchen Sie in der **Skriptansicht** die Funktion und fügen Sie als letztes Argument vor **CTRX_LAST** das Argument **CONTINUE_ON_ERROR** hinzu.

Diese Option ist in den folgenden Funktionen nicht verfügbar: **ctrx_key**, **ctrx_key_down**, **ctrx_key_up**, **ctrx_type**, **ctrx_set_waiting_time**, **ctrx_save_bitmap**, **ctrx_execute_on_window** und **ctrx_set_exception**.

Erweitertes Protokoll

Es werden weitere Informationen angezeigt, z. B. folgende:

- die Gesamtzahl der verwendeten GDI-Handles
- eine Liste der ausgeführten Citrix-Prozesse ("concentr.exe", "receiver.exe", "wfcica32.exe", "wfcrun32.exe") mit ihren PIDs und Benutzernamen (wenn der Prozess nicht unter dem LoadRunner-Benutzer ausgeführt wird)
- der Name des Citrix-Clients
- Inkompatibilitätswarnungen

Aktivieren Sie zum Anzeigen dieser Informationen die Option "Erweiterte Protokollierung" (Taste F4) auf der Registerkarte **Protokoll** in den Laufzeiteinstellungen. Sie können diese Informationen auf der Registerkarte **Wiedergabeprotokoll** oder in der Datei **output.txt** anzeigen, die sich im Skriptordner befindet.

Snapshotbitmap

Bei Auftreten eines Fehlers speichert VuGen einen Snapshot des Bildschirms im Ausgabeordner des Skripts. Sie können die Bitmapdatei anzeigen und die Fehlerursache bestimmen.

Während der Aufzeichnung werden die für die Funktion **ctrx_sync_on_bitmap** generierten Bitmaps im Datenordner des Skripts gespeichert. Der Name der Bitmapdatei weist folgendes Format auf: **hash_value.bmp**. Tritt während der Wiedergabe ein Synchronisierungsfehler auf, wird die generierte Bitmapdatei in den Ausgabeordner des Skripts bzw. in den entsprechenden Ordner für Ausgabedateien geschrieben, falls Sie das Skript in einem Szenario ausführen. Anhand der neuen Bitmapdatei können Sie die Fehlerursache ermitteln.

Anzeigen von Vusern

Geben Sie Folgendes an der Vuser-Befehlszeile ein, um Vuser während eines Szenarios anzuzeigen: **-lr_citrix_vuser_view**. Öffnen Sie im Controller das Dialogfeld **Vuser Details** und klicken Sie zur Erweiterung des Dialogfelds auf **Mehr**. Hinweis: Da sich dieser Vorgang negativ auf die Skalierbarkeit des Tests auswirkt, sollte er nur ausgeführt werden, um problematisches Vuser-Verhalten zu untersuchen.

Um die Auswirkungen auf die Skalierbarkeit des Skripts zu minimieren, können Sie die Details für einen einzelnen Vuser anzeigen, indem Sie die Befehlszeileneingabe mit der Vuser-ID beenden: `-lr_citrix_vuser_view <VuserID>`.

Um mehrere Vuser zu öffnen, beenden Sie die Befehlszeileneingabe mit den gewünschten, durch Kommata getrennten IDs. Es sind nur Kommata und Bindestriche zulässig, keine Leerzeichen. Beispiel: Bei der Eingabe 1,3-5,7 würden die Vuser mit der ID 1,3,4,5 und 7 angezeigt, nicht aber die Vuser mit der ID 2 oder 6 bzw. Vuser mit einer ID höher als 7.

Um ein Citrix- und Web-Skript mit mehreren Protokollen mithilfe von XenApp 5.0 aufzuzeichnen, müssen manuelle Änderungen vorgenommen werden, um eine einwandfreie Aufzeichnung zu gewährleisten.

Aufzeichnen mit XenApp 5.0

1. **Ändern der XenApp-Einstellungen**

   a. Stellen Sie in Internet Explorer eine Verbindung zum XenApp-Server her.

   b. Melden Sie sich dabei mit demselben Konto an, das auch für die Aufzeichnung verwendet wird.

   c. Wählen Sie **Preferences > Session Settings** aus, und stellen Sie **Window Size** auf **No Preference** ein.

2. **Ändern der Korrelationsregeln**

   a. Öffnen Sie im Dialogfeld **Aufzeichnungsoptionen** den Knoten **HTTP-Korrelation**.

   b. Vergewissern Sie sich, dass die Regel **Citrix_XenApp** ausgewählt ist.

3. **Ändern der Aufzeichnungsoptionen**

   a. Erweitern Sie im Dialogfeld **Aufzeichnungsoptionen** den Knoten **Allgemein > Aufzeichnung**.

   b. Wählen Sie **Erweitertes HTML** aus, um das Dialogfeld **Erweitertes HTML** zu öffnen.

   c. Wählen Sie **Ein Skript, das nur explizite URLs enthält** aus.

Lastgenerator und Citrix XenApp 11.2-Client

- Ein Lastgenerator kann nicht als Dienst für das Citrix-Protokoll verwendet werden, wenn der Citrix XenApp-Client 11.2 verwendet wird.
  Problemumgehung: Verwenden Sie eine andere Version des Citrix-Clients.

- Das Aufzeichnen eines Citrix NFuse-Skripts auf IE9 wird ab Citrix-Client, Version 12.1.0.44, unterstützt.
- Die Citrix-Clients ab Version 11.2 bieten keine Unterstützung für den TCP/IP-Modus im Dialogfeld **Aufzeichnungsoptionen** (Knoten **Citrix > Anmelden**).

Citrix Access Gateway-Unterstützung

LoadRunner unterstützt Citrix Access Gateway (CAG) für den Citrix-Client Version 10.200 (oder niedriger) und den Citrix-Client Version 13.x.

Click and Script-Protokolle

Hinweis: Ab LoadRunner 12.00 wird Web (Click & Script) nur für die Wiedergabe, nicht für die Aufzeichnung unterstützt.

Click & Script-Protokolle – Überblick

Die Web Click & Script-Protokolle dienen zum Aufzeichnen von Websitzungen auf der Ebene von Benutzeraktionen, die in der grafischen Benutzeroberfläche ausgeführt werden. VuGen erstellt ein GUI-Skript, das für die intuitive Darstellung von Aktionen auf der Weboberfläche sorgt. So generiert das Skript beispielsweise eine Funktion des Typs **web_button**, wenn Sie zum Übermitteln von Informationen auf eine Schaltfläche klicken, und eine Funktion des Typs **web_edit_field**, wenn Sie in ein Bearbeitungsfeld Text eingeben.

Click & Script-Vuser unterstützen Nicht-HTML-Code, wie z. B. JavaScript auf der Clientseite. VuGen erstellt ein intuitives Skript, das Ihre Aktionen auf der Webseite emuliert, und führt den erforderlichen JavaScript-Code aus.

Click & Script-Vuser bewältigen die meisten Korrelationen automatisch und verkürzen so die für die Skripterstellung benötigte Zeit. In den meisten Fällen müssen Sie nach der Aufzeichnung weder Regeln für die Korrelationen definieren noch Korrelationen manuell durchführen.

Ferner können Sie mit Click & Script-Vusern detaillierte Geschäftsprozessberichte generieren, die das Skript zusammenfassen. Wenn Sie beispielsweise auf eine Schaltfläche klicken, um Daten zu übermitteln, generiert VuGen die Funktion **web_button**. Handelt es sich bei der Schaltfläche um ein Bild, generiert VuGen die Funktion **web_image_submit**. Im folgenden Beispiel klickt ein Benutzer auf die Schaltfläche zum Anmelden.

```
...
web_image_submit("Login",
        "Snapshot=t4.inf",
        DESCRIPTION,
        "Alt=Login",
        "Name=login",
        "FrameName=navbar",
        ACTION,
```

```
        "ClickCoordinates=31,6",
        LAST);}
```

Im nächsten Abschnitt wird beschrieben, wie ein Benutzer in der Verzweigung **Manage Assets** bis zum Prozess **Asset ExpressAdd** navigiert. Zum Navigieren klickt der Benutzer auf die Textlinks der gewünschten Verzweigungen und generiert so Funktionen des Typs **web_text_link**.

```
web_text_link("Manage Assets_2",
        DESCRIPTION,
        "Text=Manage Assets",
        "Ordinal=2",
        "FrameName=main",
        ACTION,
        "UserAction=Click",
        LAST);
web_text_link("Use",
        DESCRIPTION,
        "Text=Use",
        "FrameName=main",
        ACTION,
        "UserAction=Click",
        LAST);
web_text_link("Asset ExpressAdd",
        DESCRIPTION,
        "Text=Asset ExpressAdd",
        "FrameName=main",
        ACTION,
        "UserAction=Click",
        LAST);
```

Im folgenden Beispiel emuliert die Funktion **web_list** die Auswahl eines Listenelements.

```
...
web_list("Year",
        DESCRIPTION,
        "Name=Year",
        "FrameName=CalFrame",
        ACTION,
        "Select=2000",
        LAST);
```

Wenn Sie auf ein Bild klicken, das einer Image-Map zugeordnet ist, generiert VuGen eine Funktion des Typs **web_map_area**.

```
web_map_area("map2_2",
        DESCRIPTION,
        "MapName=map2",
        "Ordinal=20",
        "FrameName=CalFrame",
        ACTION,
        "UserAction=Click",
```

```
LAST);
```

Hinweis: Web Click & Script-Vuser unterstützen weder Applets noch VB Script. Wenn die zu testende Website eines dieser Elemente enthält, müssen Sie das Web (HTTP/HTML)-Protokoll verwenden.

Click & Script - Aufzeichnungstipps

Dieser Abschnitt enthält Tipps für die Aufzeichnung von Click-and-Script-Vuser-Skripts.

Hinweis: Einige der unten aufgeführten Elemente gelten nur für bestimmte Click and Script-Protokolle.

Verwenden der Maus anstelle der Tastatur

Es empfiehlt sich, mit der Maus auf ein Objekt zu klicken, statt die Tastatur zu verwenden. Verwenden Sie während der Aufzeichnung nur GUI-Objekte, die sich innerhalb des Browserfensters befinden. Verwenden Sie keine Browsersymbole, keine Steuerelemente, keine Menüelemente wie **Ansicht > Aktualisieren** und verwenden Sie nicht die Schaltfläche **Anhalten**. Sie können jedoch die Schaltflächen **Aktualisieren**, **Startseite**, **Zurück Vorwärts** und die Adressleiste verwenden.

Zeichnen Sie nicht in einem vorhandenen Skript auf

Es empfiehlt sich, in einem neu erstellten Skript aufzuzeichnen und nicht in einem vorhandenen.

Vermeiden Sie Kontextmenüs

Vermeiden Sie die Verwendung von Kontextmenüs während der Aufzeichnung. Kontextmenüs sind Menüs, die angezeigt werden, wenn Sie mit der rechten Maustaste auf ein Element in der grafischen Benutzeroberfläche klicken.

Arbeiten Sie während der Aufzeichnung nicht in einem anderen Browser

Arbeiten Sie während der Aufzeichnung nur in den Browserfenstern, die von VuGen geöffnet werden, und nicht in anderen.

Warten Sie Downloads ab

Warten Sie, bis alle Downloads abgeschlossen sind, bevor Sie eine Aktion vornehmen, z. B. auf eine Schaltfläche klicken oder ein Textfeld ausfüllen.

Warten Sie, bis Seiten geladen sind

Während der Aufzeichnung empfiehlt es sich zu warten, bis eine Seite vollständig geladen ist, bevor Sie den nächsten Schritt ausführen. Wenn Sie nicht gewartet haben, bis alle Seiten geladen wurden, zeichnen Sie das Skript erneut auf.

Navigieren zur Startseite

Wenn die letzte Seite in einer Aktion nicht die Links und Schaltflächen enthält, die zu Beginn der Iteration zur Verfügung standen, schlägt die nächste Iteration fehl. Wenn die erste Seite

beispielsweise den Textlink **Flugbuchung** enthält, müssen Sie sicherstellen, dass Sie am Ende der Aufzeichnung zu der entsprechenden Seite navigieren, sodass derselbe Link am Ende des Geschäftsprozesses angezeigt wird.

Verwenden Sie eine höhere Ereigniskonfigurationsebene

Zeichnen Sie den Geschäftsprozess erneut mit der Ereigniskonfigurationsebene **Hoch** auf. Weitere Informationen zum Ändern der Ereigniskonfigurationsebene finden Sie unter "Click & Script - Fehlerbeseitigung und Einschränkungen" auf Seite 666.

Deaktivieren Sie die Aufzeichnung auf Socketebene

In bestimmten Fällen unterbricht die Aufzeichnung der Meldungen auf Socketebene die Anwendung. Für die meisten Aufzeichnungen sind Daten der Socketebene nicht erforderlich. Um die Aufzeichnung dieser Daten zu verhindern, deaktivieren Sie die Option in den Aufzeichnungsoptionen. Weitere Informationen finden Sie unter "GUI-Eigenschaften > Erweitert (Knoten)" auf Seite 217.

Aktivieren Sie die Option "Renderingbezogene Eigenschaftswerte aufzeichnen"

Wenn die Skripts auf Clientseite der Anwendung zahlreiche Styling-Aktivitäten verwenden, aktivieren Sie die Option **Renderingbezogene Eigenschaftswerte aufzeichnen**, bevor Sie das Skript aufzeichnen. Aktivieren Sie beispielsweise diese Option, um zusätzliche DOM-Eigenschaften, wie z. B. **offsetTop**, zu erfassen. Beachten Sie, dass das Aktivieren dieser Option möglicherweise die Aufzeichnungsgeschwindigkeit verringert. Sie können die Option aktivieren, indem Sie **Aufzeichnungsoptionen > GUI-Eigenschaften > Erweitert** auswählen. Weitere Informationen finden Sie unter "GUI-Eigenschaften > Erweitert (Knoten)" auf Seite 217.

Click and Script - Wiedergabetipps

Dieser Abschnitt enthält Tipps für die Wiedergabe von Click-and-Script-Vuser-Skripts.

Hinweis: Einige der unten aufgeführten Elemente gelten nur für bestimmte Click and Script-Protokolle.

Beibehalten der Anweisungsreihenfolge in einem aufgezeichneten Skript

Ändern Sie die Reihenfolge von Anweisungen in einem aufgezeichneten Skript nicht. Es empfiehlt sich ferner nicht, Codesegmente aus einer Aktion in eine andere zu kopieren.

Konvertieren Sie Zeichen außerhalb des ASCII-Bereichs

Wenn Links Zeichen außerhalb des ASCII-Bereichs enthalten, sollten Sie VuGen anweisen, die Datei in das UTF-8-Format oder aus dem UTF-8-Format zu konvertieren.

Aktivieren der UTF-8-Konvertierung

1. Wählen Sie **Wiedergabe > Laufzeiteinstellungen** und dann den Knoten **Internetprotokoll >**

Einstellungen aus.

2. Klicken Sie auf **Optionen**, um das Dialogfeld **Erweiterte Optionen** zu öffnen.

3. Wechseln Sie zu der Option **Aus/in UTF-8 konvertieren** und setzen Sie sie auf **Ja**.

Wahlweise können Sie die Liste der Optionen durchgehen, die eingeblendet wird, wenn ein Link nicht gefunden wird. Geben Sie den angezeigten Text unverändert ein, z. B. die hexadezimale Escapesequenz **\xA0** oder andere nicht standardmäßige Formate.

Zweimaliges Ausführen derselben Abfolge von Aktionen

In einigen Fällen können Sie einen bestimmten Prozess, z. B. einen Benutzer aus der Datenbank löschen, nur einmal durchführen. Die Wiedergabe schlägt nach der ersten Iteration fehl, da die Aktion nicht mehr gültig ist. Stellen Sie sicher, dass Ihr Geschäftsprozess mehrmals mit denselben Daten wiederholt werden kann.

Eindeutige Bild-Eigenschaften festlegen

Doppelklicken Sie im Schrittenavigator auf den vorherigen Bildschritt, um seine Eigenschaften zu öffnen. Wenn die Eigenschaften **Id**, **Name** und **Alt** leer sind, stellen Sie weitere Identifizierungsinformationen des Bilds bereit, z. B. seinen Dateinamen in der Eigenschaft **Src**.

Alternativ können Sie das Argument **Ordnungszahl** hinzufügen, um die Anzahl der Vorkommen des Bilds auf der jeweiligen Seite anzugeben. Das Argument **Ordnungszahl** identifiziert jedes Bild auf der Seite eindeutig; die anderen Identifikationsargumente hingegen sind nicht eindeutig. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Überprüfen der Schrittbeschreibung

Wenn Sie den Fehler **GUI-Objekt wurde nicht gefunden** erhalten, überprüfen Sie den Ausgabebereich auf eine Liste der Objekte in dem problematischen Schritt. In einigen Fällen ändert sich die Objektbeschreibung ein wenig von Lauf zu Lauf.

Es gibt mehrere Lösungen:

- Wenn der neue Wert stabil ist, öffnen Sie das Skript im Editor und ändern den Wert des Arguments BESCHREIBUNG manuell.

- Ändert sich die Beschreibung von Lauf zu Lauf, können Sie im DESCRIPTION-Argument einen regulären Ausdruck verwenden. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

- Alternativ können Sie die problematische Objektbeschreibungseigenschaft wie Name mit der Eigenschaft **Ordnungszahl** ersetzen. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

ThreadingModel

Die Wiedergabe des COM-Skripts in VuGen schlägt fehl, wenn in der DLL-Registrierung die Zeichenfolge **ThreadingModel** unter dem Ordner **InprocServer32** der GUID fehlt.

Click & Script - Sonstige Tipps

Die folgenden Tipps können sich bei Problemen mit Click-and-Script-Vuser-Skripts hilfreich erweisen.

Hinweis: Einige der unten aufgeführten Elemente gelten nur für bestimmte Click and Script-Protokolle.

Suchen Sie nach Warnungen

Suchen Sie im Ausgabebereich nach Warnungen oder Alarmen.

Überprüfen Sie die Antwort

Überprüfen Sie unter Verwendung von **web_reg_find**, ob die Antwort des vorherigen Schritts richtig ist. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Verwenden Sie die alternative Navigation

Verwenden Sie bei problematischen Schritten oder Schritten, die Java-Applets verwenden, die **alternative Navigation**, um den Web-Schritt durch einen Schritt auf HTTP-Ebene zu ersetzen. Beachten Sie, dass Schritte auf HTTP-Ebene möglicherweise manuelle Korrelationen erfordern. Um die alternative Navigation durchzuführen, wählen Sie einen Schritt im **Schrittenavigator** oder den Text in der Skriptansicht aus und wählen dann im Kontextmenü **Durch alternative Navigation ersetzen** aus.

Arbeiten mit dem Kerberos-Protokoll

Wenn Sie das Kerberos-Protokoll zur Authentifizierung verwenden, müssen Sie VuGen anpassen, um Autorisierungssitzungen ordnungsgemäß durchzuführen. Fortgeschrittene Benutzer können versuchen, diese Anpassung selbst vorzunehmen.

Damit das Kerberos-Protokoll ordnungsgemäß arbeitet, erstellen Sie die Datei krb5.ini und legen Sie sie in einem verfügbaren Ordner ab. Speichern Sie den vollständigen Pfadnamen der Datei **krb5.ini** in der Umgebungsvariablen KRB5_CONFIG.

Die Datei **krb5.ini** sollte detaillierte Informationen zu jeder Domäne (KDS- und AS-Adressen) und zu Vertrauensketten enthalten.

Wenn Sie weitere Informationen wünschen, setzen Sie sich bitte mit dem HP Software Support in Verbindung.

Click & Script - Erweiterungen

Im folgenden Abschnitt werden mehrere Erweiterungen beschrieben, die bei der Erstellung eines Skripts nützlich sein können.

Der Großteil der unten beschriebenen Funktionen sind Erweiterungen der API-Funktionen. Weitere Informationen zu den Funktionen und ihren Argumenten finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**). Sie können auch eine Funktion auswählen und dann F1 drücken.

Hinzufügen von bedingten Schritten

Die Web (Click and Script)-Funktionen **web_xxxx** ermöglichen Ihnen die Angabe bedingter Aktionen bei der Wiedergabe. Bedingungen sind hilfreich, wenn Sie z. B. auf ein Element prüfen und eine Aktion nur durchführen, wenn das Element gefunden wird.

Angenommen, Sie führen eine Internetsuche durch und möchten zu allen Ergebnisseiten navigieren, indem Sie auf **Next** klicken. Da Sie nicht wissen, wie viele Ergebnisseiten es geben wird, müssen Sie prüfen, ob eine Schaltfläche **Next** vorhanden ist, die eine andere Seite angibt, ohne dass der Schritt fehlschlägt. Der folgende Code fügt einen Verifizierungsschritt mit einer Benachrichtigung hinzu; wird die Schaltfläche **Next** gefunden, wird darauf geklickt.

```
While (web_text_link("Next",
DESCRIPTION,
    "Text=Next",
    VERIFICATION,
    "NotFound=Notify",
    ACTION,     "UserAction=Click",
    LAST) == LR_PASS);
```

Weitere Informationen zur Syntax und zur Verwendung des Abschnitts VERIFICATION finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Prüfen eines Seitentitels

In **web_browser**-Schritten können Sie die Aufzeichnungsoption zur Titelverifizierung verwenden, um sicherzustellen, dass die richtige Seite heruntergeladen wird. Sie können den Vuser anweisen, diese Prüfung automatisch für jeden Schritt oder jede Navigation zu einem neuen Fenster auf oberster Ebene durchzuführen.

Zusätzlich können Sie Titelverifikationen an den gewünschten Stellen manuell zu Ihrem Skript hinzufügen und dabei sowohl genaue Übereinstimmungen als auch Übereinstimmungen regulärer Ausdrücke verwenden.

```
web_browser("test_step",
DESCRIPTION,
...
VERIFICATION,
   "BrowserTitle=Title",
   ACTION,]
,
LAST);
```

Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Die Optionen der Titelverifikation können Sie direkt in den Aufzeichnungsoptionen festlegen. Weitere Informationen finden Sie im Abschnitt zum Aufzeichnen mit Click and Script.

Verifikation von Textprüfungen

Unter Verwendung von Textprüfpunkten können Sie sicherstellen, dass eine Textzeichenfolge auf einer Webseite oder in einer Anwendung an der richtigen Stelle angezeigt wird und dann basierend auf den Ergebnissen eine Aktion durchführen. Anhand von genauen Übereinstimmungen oder von Übereinstimmungen regulärer Ausdrücke können Sie prüfen, ob eine Textzeichenfolge vorhanden ist (**ContainsText**) oder nicht (**DoesNotContainText**).

Angenommen, auf einer Webseite wird der Satz **Flug von New York nach San Francisco** angezeigt. Sie können einen Textprüfpunkt erstellen, der prüft, ob die Wörter **New York** zwischen **Flug von** und **nach San Francisco** angezeigt werden. (In diesem Beispiel müssten Sie die Kriterien eines regulären Ausdrucks verwenden.)

Um diese Prüfpunkte zu implementieren, fügen Sie dem Abschnitt VERIFICATION des Schritts auf die Textprüfung bezogene Argumente hinzu. Bei der Wiedergabe suchen Vuser den innerText des HTML-Dokuments im Browser sowie untergeordnete Frames. Das Argument **NotFound** gibt die Aktion an, die bei einem Verifzierungsfehler vorzunehmen ist, der entweder auftritt weil das Objekt nicht gefunden wurde oder weil die Textverifizierung fehlgeschlagen ist: **Error**, **Warning** oder **Notify**.

Sie können Ihrem Skript Textverifizierungen für vorhandene Schritte manuell hinzufügen. Platzieren Sie eine Textverifizierung nach dem Schritt, der das Element generiert hat.

Die Textvalidierungsargumente sind für die folgenden Aktionsfunktionen gültig: **web_browser**, **web_element**, **web_list**, **web_text_link**, **web_table** und **web_text_area**.

Hinweis: Sie können nicht mehrere gleiche Textverifizierungstypen pro Schritt verwenden (z. B. zweimal **ContainsText**). Wenn Sie auf mehrere Texte prüfen möchten, trennen Sie sie in einzelne Schritte. Sie können jedoch unterschiedliche Verifikationen in demselben Schritt verwenden (z. B. **ContainsText** =; **DoesNotContainText**). In diesem Fall müssen alle Bedingungen erfüllt sein, damit der Schritt bestanden wird.

Im folgenden Beispiel prüfen die Verifikationsargumente, ob keine Umleitung von der Website **www.acme.com** auf die französische Website **acme.com/fr** erfolgt ist.

```
web_browser("www.acme.com",
        ACTION,
        "Navigate=http://www.acme.com/",
    LAST);
web_browser("Verify",
        VERIFICATION,
        "ContainsText=Go to Acme France",
        "DoesNotContainText=acme.com in English",
    LAST);
```

Speichern eines Java-Skriptwerts in einem Parameter

Das Argument **EvalJavaScript** ermöglicht die Auswertung eines Java-Skripts auf der Webseite.

Angenommen, Sie möchten auf einen Link klicken, dessen Name mit dem Seitentitel übereinstimmt. Im folgenden Beispiel wird der Dokumenttitel ausgewertet und in der nächsten web_text_link-Funktion verwendet.

```
web_browser("GetTitle",
    ACTION,
    "EvalJavaScript=document.title;",
    "EvalJavaScriptResultParam=title",
    LAST);
web_text_link("Link",
    DESCRIPTION,
```

```
    "Text={title}",
    LAST);
```

Arbeiten mit benutzerdefinierten Beschreibungen

Angenommen, Sie klicken zufällig auf einen Link, der zu einer Gruppe gehört. Beispiel: Sie wählen auf der Webseite **hp.com** zufällig ein Land aus. In der normalen Beschreibungszuordnung ist dieser Vorgangstyp nicht zulässig. Wenn Sie jedoch das Argument einer benutzerdefinierten Beschreibung verwenden, können Sie die Gruppe mit einem Attribut identifizieren, das allen Links in der Gruppe gemein ist.

Unter Verwendung des Arguments einer benutzerdefinierten Beschreibung geben Sie ein beliebiges Attribut des Elements an, einschließlich solcher Attribute, die für das betreffende Element nicht vordefiniert sind. bei der Wiedergabe sucht der Vuser nach diesen Attributen, die im Abschnitt DESCRIPTION angegeben wurden. Die Wiedergabe schlägt aufgrund eines unbekannten Arguments im Abschnitt DESCRIPTION nicht fehl.

Um beispielsweise den Hyperlink

Yahoo zu finden, verwenden Sie Folgendes:

```
web_text_link("yahoo",
    DESCRIPTION,
    "Text=yahoo",
    "my_attribute=bar",
    LAST);
```

Im folgenden Beispiel können Sie ein zufälliges Klicken unter Verwendung des folgenden Codes ausführen, da alle relevanten Links denselben Klassennamen **newmerc-left-ct** aufweisen:

```
web_text_link("Click",
                        DESCRIPTION,
                        "Class=newmerc-left-ct",
                        "Ordinal=random",
                        LAST);
```

Die folgenden Funktionen unterstützen keine Argumente benutzerdefinierter Beschreibungen: **web_browser**, **web_map_area**, **web_radio_group** und **web_reg_dialog**.

Kopieren von Text in die Zwischenablage

VuGen bietet Ihnen die Möglichkeit, Text aus einem Browser in die Zwischenablage zu kopieren. Diese Funktionalität steht sowohl in der Seitenansicht als auch in der Seitenquellenansicht zur Verfügung. Details zum Kopieren von Text in die Zwischenablage finden Sie unter "Arbeiten mit Snapshots" auf Seite 65.

Hinweise zur Click & Script-API

Dieser Abschnitt enthält allgemeine Hinweise zu Webfunktionen. Beachten Sie, dass Sie für die meisten Objektbeschreibungen einen regulären Ausdruck angeben können, indem Sie dem Text nach dem Gleichzeichen ein "/RE" voranstellen. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**). Beispiel:

```
web_text_link("Manage Assets",
        DESCRIPTION,
        "Text/RE=(Manage Assets)|(Configure Assets)",
        ACTION,
        "UserAction=Click",
        LAST);
```

Ordnungszahlen

Das Attribut **Ordinal** ist ein bei der Zahl 1 beginnender Index zur Unterscheidung mehrerer Vorkommen von Objekten mit einer identischen Beschreibung. Im folgenden Beispiel weisen zwei aufgezeichnete Funktionen des Typs **web_text_link** bis auf die Ordnungszahl identische Argumente auf. Die Ordnungszahl verweist auf das zweite Vorkommen.

```
web_text_link("Manage Assets",
        DESCRIPTION,
        "Text=Manage Assets",
        "FrameName=main",
        ACTION,
        "UserAction=Click",
        LAST);
web_text_link("Manage Assets_2",
        DESCRIPTION,
        "Text=Manage Assets",
        "Ordinal=2",
        "FrameName=main",
        ACTION,
        "UserAction=Click",
        LAST);
```

Leere Zeichenfolgen

Es besteht ein Unterschied darin, ob der Benutzer ein Argument überhaupt nicht oder als leere Zeichenfolge angibt. Wenn Sie kein Argument angeben, verwendet VuGen den Standardwert oder ignoriert ihn. Wenn Sie ein Argument auflisten, diesem jedoch eine leere Zeichenfolge als Wert zuweisen, versucht VuGen eine Übereinstimmung mit einer leeren Zeichenfolge zu finden oder sucht überhaupt nicht nach einer Zeichenfolge. Wenn Sie beispielsweise keine Argument-ID angeben, wird VuGen angewiesen, die Eigenschafts-ID des HTML-Elements zu ignorieren. Bei der Angabe von "ID=" sucht VuGen nach HTML-Elementen ohne Eigenschafts-ID oder mit einer leeren ID.

```
web_text_link("Manage Assets_2",
        DESCRIPTION,
        "Text=Manage Assets",
        "Id=",
        "FrameName=main",
        ACTION,
        "UserAction=Click",
        LAST);
```

Click & Script - Fehlerbeseitigung und Einschränkungen

In diesem Abschnitt werden die Fehlerbeseitigung und die für Click and Script-Protokolle geltenden Einschränkungen beschrieben.

Hinweis: Einige der unten aufgeführten Elemente gelten nur für bestimmte Click and Script-Protokolle.

Aufzeichnungsprobleme und -beschränkungen

Kompatible Browser

- Für Web (Click & Script) wird nur Internet Explorer unterstützt. Um die Browseraktivität in Firefox aufzuzeichnen, verwenden Sie das Protokoll Web (HTTP/HTML).
- Citrix Agent kann keinen Text aus Java-basierten Anwendungen und Internet Explorer Version 9 und höher erfassen.

Sprachunterstützung

- Die Aufzeichnung einer Anwendung in einer bestimmten Sprache (z. B. Französisch, Japanisch) muss auf einem Computer durchgeführt werden, dessen Standardspracheinstellung (**Einstellungen > Systemsteuerung > Regions- und Sprachoptionen**) der Sprache der Aufzeichnung entspricht.
- Begrenzte Unterstützung von Rechts-nach-links-Sprachen (es treten z. B. Fehler bei der Verarbeitung von bidirektionalem oder umgedrehtem Text auf). Dies hängt von der standardmäßig verwendeten Umwandlungstabelle des Betriebssystems ab.
- Der Lastgenerator-Computer und der aufzeichnende Computer müssen auf dieselben Regionaleinstellungen eingestellt werden. Sie sollten nicht davon ausgehen, dass der Linux-Standardzeichensatz auf einem Computer mit englischem Betriebssystem dem Zeichensatz auf einem Windows-Computer mit englischem Betriebssystem entspricht. Überprüfen Sie dies vor der Aufzeichnung. Beispielsweise ist der Standardzeichensatz unter Linux UTF-8.

Die Anwendung verhält sich während der Aufzeichnung anders

Wenn die Anwendung sich während der Aufzeichnung anders verhält als ohne Aufzeichnung, sollten Sie feststellen, ob das Aufzeichnungsproblem nur bei Web auftritt. Die Auswirkungen des Problems sind z. B., dass eine Webseite nicht geladen wird, der Inhalt teilweise fehlt oder ein Popup-Fenster nicht geöffnet wird.

Erstellen Sie ein neues Web (HTTP/HTML)-Skript und wiederholen Sie die Aufzeichnung.

Tritt bei der Aufzeichnung in Web (HTTP/HTML) ein Fehler auf, wird empfohlen, die Aufzeichnung auf Socketebene zu deaktivieren (siehe "Click & Script - Aufzeichnungstipps" auf Seite 658).

Das Problem ist möglicherweise auf einen Ereignislistener zurückzuführen. Deaktivieren Sie die einzelnen Ereignislistener in den Aufzeichnungsoptionen unter **Webereigniskonfiguration** nach der Methode "Versuch und Irrtum" und zeichnen Sie Ihre Sitzung als Web-Benutzer erneut auf.

Bestimmte Click & Script-Schritte werden nicht ordnungsgemäß generiert

Wenn nach der Skriptaufzeichnung nicht alle Schritte ordnungsgemäß generiert werden, kann das Problem an der **Windows-Komponente > Verstärkte Sicherheitskonfiguration für den Internet Explorer** liegen.

Entfernen Sie die **Verstärkte Sicherheitskonfiguration für den Internet Explorer**, indem Sie **Systemsteuerung > Software > Windows-Komponenten hinzufügen/entfernen** auswählen und Ihr Skript neu aufzeichnen.

Deaktivieren eines Ereignislisteners

1. Klicken Sie auf **Aufzeichnen > Aufzeichnungsoptionen**, um das Dialogfeld **Aufzeichnungsoptionen** zu öffnen.
2. Wählen Sie den Knoten **GUI-Eigenschaften > Webereigniskonfiguration** aus.
3. Klicken Sie auf **Benutzerdefinierte Einstellungen** und erweitern Sie den Knoten **Web Objects**. Wählen Sie ein Objekt aus.
4. Wählen Sie **Deaktiviert** aus der Liste in der Spalte **Aufzeichnen** für das relevante Webobjekt aus. Wenn die Aufzeichnung immer noch nicht richtig durchgeführt wird, aktivieren Sie den Listener, den Sie zuvor deaktiviert haben, und deaktivieren Sie einen anderen. Wiederholen Sie diese Schritte, bis die Aufzeichnung erfolgreich durchgeführt wird.

Dynamische Menünavigation wurde nicht aufgezeichnet

Ein dynamisches Menü ist ein Menü, das sich dynamisch ändert, je nachdem, an welcher Stelle Sie es auswählen. Wenn die dynamische Menünavigation nicht erfasst wurde, zeichnen Sie erneut mit dem Ereigniskonfigurationsmodus **hoch** auf. Diese Einstellungen können in dem Knoten **Aufzeichnungsoptionen > GUI-Eigenschaften > Webereigniskonfiguration** gefunden werden.

Bestimmte Benutzeraktionen wurden nicht aufgezeichnet

Prüfen Sie, ob im Browser ein Java-Applet ausgeführt wird. Ist dies nicht der Fall, zeichnen Sie das Skript mit dem Protokoll Web (HTTP/HTML) auf.

Wiedergabeprobleme

GUI-Objekt wurde nicht gefunden

Tritt der Fehler zu Beginn der zweiten Iteration auf?

Tritt der Fehler zu Beginn des Abschnitts **Action** der zweiten Iteration auf, ist er möglicherweise auf eine Startseite zurückzuführen, die für die erste Iteration vorhanden war, für die zweite jedoch fehlte. Wenn die letzte Seite in einer Aktion nicht die Links und Schaltflächen enthält, die zu Beginn der Iteration zur Verfügung standen, schlägt die nächste Iteration fehl. Wenn die erste Seite beispielsweise den Textlink **Flugbuchung** enthält, müssen Sie sicherstellen, dass Sie zu der entsprechenden Seite navigieren, sodass derselbe Link am Ende des Geschäftsprozesses angezeigt wird.

Handelt es sich um einen Textlink, der Zeichen außerhalb des ASCII-Bereichs enthält?

Wenn das Problem bei Zeichen außerhalb des ASCII-Bereichs auftritt, sollten Sie VuGen anweisen, die Daten in einen geeigneten Zeichensatz zu konvertieren.

Aktivieren der Datenkonvertierung auf Windows-Computern

1. Wählen Sie **Wiedergeben > Laufzeiteinstellungen** und dann den Knoten **Internetprotokoll > Einstellungen** aus.

2. Klicken Sie auf **Optionen**, um das Dialogfeld **Erweiterte Optionen** zu öffnen.

3. Wechseln Sie unter **Web (Click and Script) > Allgemein** zu **Zeichensatzkonvertierungen durch HTTP** und setzen Sie die Option auf **Ja**.

Aktivieren der UTF-8-Konvertierung für Linux-Computer

1. Wählen Sie **Wiedergeben > Laufzeiteinstellungen** und dann den Knoten **Internetprotokoll > Einstellungen** aus.

2. Klicken Sie auf **Optionen**, um das Dialogfeld **Erweiterte Optionen** zu öffnen.

3. Wechseln Sie in den Optionen unter **Allgemein** zu **Aus/in UTF-8 konvertieren** und setzen Sie die Option auf **Ja**.

Wahlweise können Sie die Liste der Alternativen durchgehen, die eingeblendet wird, wenn ein Link nicht gefunden wird. Geben Sie den angezeigten Text unverändert ein, z. B. die hexadezimale Escapesequenz **\xA0** oder andere nicht standardmäßige Formate.

Können Sie dieselbe Abfolge von Aktionen in der Anwendung zweimal ausführen?

In einigen Fällen können Sie einen bestimmten Prozess, z. B. einen Benutzer aus der Datenbank löschen, nur einmal durchführen. Die Wiedergabe schlägt nach der ersten Iteration fehl, da die Aktion nicht mehr gültig ist. Stellen Sie sicher, dass Ihr Geschäftsprozess in der Anwendung mehrmals mit denselben Daten wiederholt werden kann, ohne dass eine neue Aufzeichnung erforderlich ist.

Waren die Bildeigenschaften **Id**, **Name** und **Alt** leer?

Doppelklicken Sie im **Schrittenavigator** auf den vorherigen Bildschritt, um seine Eigenschaften zu öffnen. Wenn die Eigenschaften **Id**, **Name** und **Alt** leer sind, stellen Sie weitere Identifizierungsinformationen des Bilds bereit, z. B. seinen Dateinamen in der Eigenschaft **Src**.

Alternativ können Sie das Argument **Ordnungszahl** hinzufügen, um die Anzahl der Vorkommen des Bilds auf der jeweiligen Seite anzugeben. Das Argument **Ordnungszahl** identifiziert jedes Bild auf der Seite eindeutig; die anderen Identifikationsargumente hingegen sind nicht eindeutig. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Hat sich die Schrittbeschreibung geändert?

Prüfen Sie den Ausgabebereich auf eine Liste der Objekte in dem problematischen Schritt. In einigen Fällen ändert sich die Objektbeschreibung ein wenig von Lauf zu Lauf.

Es gibt mehrere Lösungen:

- Wenn der neue Wert stabil ist, öffnen Sie die Skriptansicht und ändern den Wert der DESCRIPTION-Argumente manuell.

- Ändert sich die Beschreibung von Lauf zu Lauf, können Sie in den DESCRIPTION-Argumenten einen regulären Ausdruck verwenden. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

- Alternativ können Sie die problematische Objektbeschreibungseigenschaft, wie beispielsweise **Name**, mit der Eigenschaft **Ordnungszahl** ersetzen. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Wurde die Seite während der Aufzeichnung vollständig geladen?

Während der Aufzeichnung empfiehlt es sich zu warten, bis eine Seite vollständig geladen ist, bevor Sie den nächsten Schritt ausführen. Wenn Sie nicht gewartet haben, bis alle Seiten geladen wurden, zeichnen Sie das Skript erneut auf.

Wiedergabefehler

Kommt es bei der Wiedergabe an einem bestimmten Schritt zu einem Fehler, überprüfen Sie die Schrittbeschreibung. VuGen liest ein einfaches Leerzeichen möglicherweise als doppeltes Leerzeichen. Stellen Sie sicher, dass sich in der Zeichenfolge keine doppelten Leerzeichen befinden.

Sonstige Probleme

Fehler "Speicherplatz nicht ausreichend" in JavaScript

Vergrößern Sie den JavaScript-Speicher in den Laufzeiteinstellungen.

Vergrößern des JavaScript-Speichers

1. Wählen Sie **Wiedergeben > Laufzeiteinstellungen** und dann den Knoten **Internetprotokoll > Einstellungen** aus.

2. Klicken Sie auf **Optionen**, um das Dialogfeld **Erweiterte Optionen** zu öffnen.

3. Wechseln Sie zu den Optionen **Speicherverwaltung Größe des JavaScript-Laufzeitspeichers (KB)** und **Speicherverwaltung Größe des JavaScript-Stackspeichers (KB)**.

4. Setzen Sie die Speichergrößen auf 512 KB oder höher.

   VuGen zeigt JavaScript-Fehler an

   Wenn VuGen im Ausgabebereich JavaScript-Fehler anzeigt, aktivieren Sie die Anzeige von IE-Skriptfehlern (Internet Explorer), um zu gewährleisten, dass das Java-Skript selbst keine Fehler enthält.

Anzeigen von Skriptfehlern

1. Öffnen Sie Internet Explorer.

2. Wählen Sie **Extras > Internetoptionen** aus und klicken Sie auf die Registerkarte **Erweitert**.

3. Aktivieren Sie im Abschnitt **Browsen** das Kontrollkästchen **Skriptfehler anzeigen**.

4. Führen Sie die Anwendung erneut in IE aus. Wenn IE Skriptfehler anzeigt, liegt ein Problem mit der JavaScript-Anwendung vor. Wenn das Problem in der Anwendung nicht behoben werden kann, können Sie die entsprechenden Wiedergabefehler ruhig ignorieren.

Probleme nach der Parametrierung

Wenn nach der Parametrierung von Werten Probleme auftreten, stellen Sie sicher, dass die Werte für Ihre Anwendung gültig sind. Führen Sie einen Geschäftsprozess mit dem Wert des Parameters durch und stellen Sie sicher, dass die Anwendung ihn akzeptiert.

Probleme mit Anwendungen, die Styling-Aktionen verwenden

Wenn die Skripts auf Clientseite der Anwendung zahlreiche Styling-Aktivitäten verwenden, sollten Sie das Skript erneut aufzeichnen, nachdem Sie die Option **Renderingbezogene Eigenschaftswerte aufzeichnen** aktiviert haben. Auf diese Weise wird die Aufzeichnung zusätzlicher DOM-Objekte ermöglicht.

Aktivieren Sie die Option "Renderingbezogene Eigenschaftswerte aufzeichnen"

1. Wählen Sie **Aufzeichnen > Aufzeichnungsoptionen** und dann den Knoten **GUI-Eigenschaften > Erweitert** aus.

2. Aktivieren Sie das Kontrollkästchen **Renderingbezogene Eigenschaftswerte aufzeichnen**.

Zeichnen Sie das Vuser-Skript erneut auf.

COM-Protokoll

Übersicht über das COM-Protokoll

Beim Aufzeichnen von COM-Clientanwendungen erzeugt VuGen Funktionen, welche die COM-Client/Server-Aktivität beschreiben. Das aufgezeichnete Skript enthält Schnittstellendeklarationen, API-Aufrufe und Instanzenaufrufe für Methoden. Alle COM-Funktionen beginnen mit dem Präfix **lrs**. Sie können die Programmiersprache festlegen, die bei der Erstellung eines Vuser-Skripts verwendet werden soll, entweder C oder Visual Basic.

Für jedes COM-Vuser-Skript generiert VuGen Folgendes:

- Schnittstellenzeiger und andere Variablendeklarationen in der Datei interfaces.h.
- Funktionsaufrufe, die Sie in den Abschnitten vuser_init, actions oder vuser_end der Vuser-Datei aufzeichnen können.
- Die Datei user.h, in der das Vuser-Skript in Aufrufe auf niedriger Ebene übersetzt wird.

Übersicht über die COM-Technologie

Dieser Abschnitt bietet einen Überblick über die COM-Technologie. Die hier zur Verfügung gestellten Informationen sollten ausreichen, damit Sie mit der Arbeit mit COM-Vuser-Skripts beginnen können. Weitere Details erhalten Sie im Microsoft Developers Network (MSDN) und anderen Dokumentationen.

COM (Component Object Model) ist eine Technologie zur Entwicklung von wieder verwendbaren Softwarekomponenten (Plug-Ins). DCOM (Distributed COM) ermöglicht die Verwendung von COM-Komponenten auf Remotecomputern. Microsoft Transaction Server (MTS), Visual Basic und Explorer nutzen alle die COM/DCOM-Technologie. Möglicherweise nutzt also auch die von Ihnen getestete Anwendung die COM-Technologie, obwohl Sie sich dessen vielleicht nicht bewusst sind. Sie müssen wahrscheinlich einige, aber sicherlich nicht alle COM-Aufrufe Ihrer Anwendung in das Vuser-Skript aufnehmen.

Objekte, Schnittstellen und Typbibliotheken

COM-Objekte sind Binärcodemodule. Jedes COM-Objekt implementiert eine oder mehrere Schnittstellen, über die die Kommunikation mit Clientprogrammen stattfindet. Sie müssen diese Schnittstellen kennen, damit Sie die COM-Aufrufe in den Vuser-Skripts nachverfolgen können. Typbibliotheken, die als Referenz für den Zugriff auf COM-Schnittstellen und -Parameter verwendet werden, enthalten Beschreibungen von COM-Objekten und -Schnittstellen. Jede COM-Klasse, -Schnittstelle und -Typbibliothek wird durch eine GUID (Global Unique Identifier) eindeutig identifiziert.

COM-Schnittstellen

Eine COM-Schnittstelle stellt eine Sammlung zusammengehöriger Methoden dar. Beispielsweise kann ein Objekt vom Typ **Clock** die Schnittstellen **Clock**, **Alarm** und **Timer** haben. Jede Schnittstelle verfügt wiederum über mindestens eine Methode. Die Schnittstelle **Alarm** stellt beispielsweise die Methoden **AlarmOn** und **AlarmOff** zur Verfügung.

Schnittstellen haben darüber hinaus auch mindestens eine Eigenschaft. Manchmal kann dieselbe Funktion entweder durch Aufruf einer Methode oder durch Festlegen und Abrufen eines Eigenschaftenwerts ausgeführt werden. Sie können beispielsweise entweder die Eigenschaft **Alarm Status** auf **On** setzen oder die Methode **AlarmOn** aufrufen.

Ein COM-Objekt kann eine Vielzahl von Schnittstellen unterstützen. Die Schnittstelle **IUnknown** wird von allen Komponenten implementiert. Mit ihr können Informationen über andere Schnittstellen gesammelt werden. Viele Komponenten implementieren darüber hinaus die Schnittstelle **IDispatch**, die alle anderen Schnittstellen und Methoden des Objekts zur Verfügung stellt und so die Implementierung der COM-Automatisierung in Skriptsprachen ermöglicht.

COM-Klassenkontext und Ortstransparenz

COM-Objekte können entweder auf demselben Computer wie die Clientanwendung oder auf einem Remoteserver ausgeführt werden. Die von einer Anwendung erzeugten COM-Objekte können sich in einer lokalen Bibliothek, einem lokalen Prozess oder auf einem Remotecomputer ("Remoteobjektproxy") befinden. Der Speicherort des COM-Objekts, der so genannte "Kontext", kann für die Anwendung transparent sein. Die meisten Benutzer überprüfen mithilfe von Vusern die Last auf Remoteservern. Aus diesem Grund haben die Objekte, auf die über den Remoteobjektproxy zugegriffen wird, in der Regel für Tests die größte Relevanz.

COM-Datentypen

COM stellt darüber hinaus eine Reihe spezieller Datentypen zur Verfügung, darunter sichere Arrays, BSTR-Zeichenfolgen und Varianten. Sie benötigen diese Datentypen für das Debugging, die Parametrierung und ähnliche Aufgaben.

Struktur der COM-Vuser-Skripts

VuGen-COM-Skripts sind strukturell an die Erfordernisse von COM-Schnittstellen angepasst.

Schnittstellenmethoden

Für Schnittstellenmethoden gelten die folgenden Benennungs- und Syntaxkonventionen:

```
lrc_<Schnittstellenname>_<Methodenname>(Instanz,...);
```

Beachten Sie, dass die Instanz immer der erste Parameter ist, der übergeben wird.

Die Anbieter der einzelnen COM-Komponenten stellen in der Regel eine Dokumentation zu ihren Schnittstellenfunktionen bereit.

Schnittstellenzeiger

In der Schnittstellenkopfzeilendatei werden die Schnittstellenzeiger sowie weitere Variablen definiert, die im Skript verwendet werden können. Jede Schnittstelle besitzt zur eindeutigen Kennzeichnung eine eigene Schnittstellen-ID (IID).

Schnittstelledefinitionen weisen das folgende Format auf:

```
<Schnittstellentyp>*<Schnittstellenname> = 0; //”{<IID des Schnittstellentyps>}”
```

Im folgenden Beispiel lautet der Schnittstellentyp **IDispatch**, der Name der Schnittstelleninstanz **IDispatch_0** und die IID des Typs **IDispatch** ist die lange Zahlenfolge:

```
IDispatch* IDispatch_0= 0;//"{00020400-0000--C000-000000000046}"
```

Vuser-Skriptanweisungen

COM-Vuser-Skripts bestehen aus Code, der Objektinstanzen erstellt, Schnittstellenzeiger abruft und Schnittstellenmethoden aufruft. Bei jeder Benutzeraktion wird mindestens ein COM-Aufruf generiert. Jeder der COM-Aufrufe wird von VuGen als eine Gruppe von Anweisungen kodiert. Jede dieser Gruppen ist in einem separaten Bereich enthalten, der durch Klammern begrenzt wird. Der Hauptaufruf wird in mehreren unterschiedlichen Anweisungen durch das Zuweisen von Werten und durch Typkonvertierungen vorbereitet. Die zum Erstellen eines Objekts erforderlichen vorbereitenden Aufrufe könnten z. B. folgendermaßen aussehen:

```
{
GUID pClsid = lrc_GUID("student..1");
IUnknown * pUnkOuter = (IUnknown*)NULL;
unsigned long dwClsContext = lrc_ulong("7");
GUID riid = IID_IUnknown;
lrc_CoCreateInstance(=;pClsid, pUnkOuter, dwClsContext, =;riid, (void**)=;IUnkno
wn_0, CHECK_HRES);
}
```

Fehlerüberprüfung

Bei jedem Aufruf einer COM-Methode oder eines APIs wird ein Fehlerwert zurückgegeben. VuGen setzt ein Kennzeichnen, das angibt, ob Fehler bei der Wiedergabe überprüft werden sollen oder nicht. Dies hängt davon ab, ob der Aufruf während der Aufzeichnung erfolgreich war oder nicht. Das Kennzeichen wird am Ende des Funktionsaufrufs als letztes Argument gesetzt. Es hat die folgenden Werte:

CHECK_HRES	Dieser Wert wird eingefügt, wenn die Funktion während der Aufzeichnung erfolgreich ausgeführt wurde und bei der Wiedergabe auf Fehler geprüft werden soll.
DONT_CHECK_HRES	Dieser Wert wird eingefügt, wenn die Funktion während der Aufzeichnung fehlgeschlagen ist und bei der Wiedergabe nicht auf Fehler geprüft werden soll.

COM-Beispiel-Vuser-Skripts

In diesem Abschnitt wird anhand von Beispielen veranschaulicht, wie VuGen eine COM-Clientanwendung emuliert. Es wird in die grundlegenden COM-Skriptoperationen unterteilt. Jede dieser Operationsarten hat ihren eigenen Ausführungsbereich.

Instantiierung des Objekts

Um ein COM-Objekt verwenden zu können, muss die Anwendung es zunächst instantiieren und dann einen Zeiger auf eine der Schnittstellen des Objekts abrufen.

VuGen geht folgendermaßen vor, um ein Objekt zu instanziieren:

1. VuGen ruft die Funktion **lrc_GUID** auf, um die eindeutige **ProgID** des Objekts abzurufen, die dann in **pClsid** gespeichert wird:

   ```
   GUID pClsid = lrc_GUID("student..1");
   ```

   pClsid ist die eindeutige, globale CLSID des Objekts, die aus der ProgID **student.student.1** umgewandelt wurde.

2. Wenn der unbekannte Schnittstellenzeiger auf ein aggregiertes Objekt zeigt, ruft VuGen den Zeiger für dieses Objekt ab oder gibt andernfalls den Wert NULL zurück:

   ```
   IUnknown * pUnkOuter = (IUnknown*)NULL;
   ```

3. VuGen legt den Kontext für das zu erstellende Objekt fest:

   ```
   unsigned long dwClsContext = lrc_ulong("7");
   ```

   dwClsContext enthält den Objektkontext (**in Bearbeitung**, **lokal**, **remote** oder Kombinationen daraus.)

4. VuGen setzt eine Variable, in der die angefragte Schnittstellen-ID (in diesem Fall **IUnknown**) gespeichert wird:

```
GUID riid = IID_IUnknown;
```

riid enthält die Schnittstellen-ID der Schnittstelle **IUnknown**.

5. Nach Vorbereitung der Eingabeparameter wird das Objekt mithilfe eines Aufrufs der Funktion **lrc_CoCreateInstance** unter Verwendung der in den vorangegangenen Anweisungen definierten Parameter erstellt. Der Zeiger für die Schnittstelle **IUnknown** wird dem Ausgabeparameter **IUnknown_0** zugewiesen. Dieser Zeiger wird für die nachfolgenden Aufrufe benötigt:

```
lrc_CoCreateInstance(=;pClsid, pUnkOuter, dwClsContext, =;riid, (void**)=;IUn
known_0, CHECK_HRES);
```

Die Eingabeparameter wurden in den vorangegangenen Schritten vorbereitet und erklärt. Da der Aufruf erfolgreich war, aktiviert VuGen durch Einfügen des Wertes **CHECK_HRES** die Fehlerprüfung während der Benutzersimulation. Der Aufruf gibt den Zeiger auf die Schnittstelle **IUnknown** an den Parameter **IUnknown_0** zurück, welcher für nachfolgende Aufrufe verwendet werden kann.

Abrufen einer Schnittstelle

Nach dem Erstellen eines Objekts kann VuGen lediglich auf die Schnittstelle **IUnknown** zugreifen. VuGen nutzt die Schnittstelle **IUnknown** für die Kommunikation mit dem Objekt. Hierzu wird die von der Standardschnittstelle **IUnknown** bereitgestellte Methode **QueryInterface** verwendet. Der erste Parameter in einem VuGen-Methodenaufruf ist die Schnittstelleninstanz. In diesem Fall ist dies der Zeiger **IUnknown_0**, der zuvor durch **CoCreateInstance** gesetzt wurde. Für den Aufruf der Funktion **QueryInterface** muss die ID der abzurufenden Schnittstelle eingegeben werden. Der Aufruf gibt dann einen Zeiger zurück, der auf die durch die ID gekennzeichnete Schnittstelle verweist.

Abrufen der Schnittstelle

1. Zunächst setzt VuGen den Parameter **riid** auf die ID der Schnittstelle **Istudent**:

```
GUID riid = IID_Istudent;
```

2. Beim Aufruf von **QueryInterface** wird der Zeiger auf die Schnittstelle **Istudent** dem Ausgabeparameter **Istudent_0** zugewiesen, falls das Objekt **Istudent** über eine solche Schnittstelle verfügt:

```
lrc_IUnknown_QueryInterface(IUnknown_0, =;riid, (void**)=;Istudent_0, CHECK_H
RES);
```

Festlegen von Daten mithilfe einer Schnittstelle

Das folgende Beispiel zeigt, wie Sie die Methoden einer Schnittstelle zum Festlegen von Daten verwenden können. Nehmen wir an, der Benutzer wird in der Anwendung aufgefordert, einen Namen einzugeben. Dabei wird eine Methode zum Festlegen von Namen aufgerufen. VuGen zeichnet diesen Schritt mithilfe zweier Anweisungen auf. Die erste Anweisung dient dem Einrichten der Zeichenfolge für den Namen, die zweite legt die Namenseigenschaft fest.

Einrichten des kompletten Funktionsaufrufs

1. Zunächst setzt VuGen eine Variable (Prop Value), die der Zeichenfolge entspricht. Der Parameter hat den Typ BSTR, ein Zeichenfolgentyp der in COM-Dateien verwendet wird:

   ```
   BSTR PropValue = lrc_BSTR("John Smith");
   ```

   Danach werden Sie diesen Aufruf wahrscheinlich parametrieren, indem Sie **John Smith** durch einen Parameter ersetzen, so dass bei jeder Ausführung des Vuser-Skripts ein anderer Name verwendet wird.

2. In einem nächsten Schritt ruft VuGen die Methode **Put_Name** der Schnittstelle **Istudent** zur Eingabe des Namens auf:

   ```
   lrc_Istudent_put_name(Istudent_0, PropValue, CHECK_HRES);
   ```

Abrufen von Daten mithilfe einer Schnittstelle

Das Abrufen von Daten von einer Anwendung unterscheidet sich von der Eingabe von Daten, da diese Daten voraussichtlich gespeichert und zur Parametrierung als Eingabe in nachfolgenden Aufrufen genutzt werden.

Es folgt ein Beispiel dafür, wie VuGen vorgeht, wenn die Anwendung Daten abruft:

1. Es wird eine geeignete Variable (in diesem Fall vom Typ BSTR) erstellt, in welcher der Eigenschaftenwert gespeichert wird.

   ```
   BSTR pVal;
   ```

2. Es wird der Eigenschaftenwert (in diesem Fall ein Name) abgerufen und in der oben erstellten Variable **pVal** gespeichert. Hierzu wird in diesem Beispiel die von der Schnittstelle **Istudent** zur Verfügung gestellte Methode **get_name** verwendet.

   ```
   lrc_Istudent_get_name(Istudent_0, =;pVal, CHECK_HRES);
   ```

3. VuGen generiert nun eine Anweisung zum Speichern der Werte.

   ```
   //lrc_save_BSTR("param-name",pVal);
   ```

   Diese Anweisung ist auskommentiert. Sie können die Kommentarzeichen entfernen und <param-name> durch eine Variable mit einem aussagekräftigen Namen ersetzen, in welcher der Wert gespeichert wird. VuGen verwendet die Variable, um den Wert von **pVal**, der im vorherigen Aufruf zurückgegeben wurde, zu speichern. Danach können Sie die Variable als parametrierte Eingabe für nachfolgende Aufrufe anderer Methoden verwenden.

Die Schnittstelle "IDispatch"

Die meisten COM-Objekte verfügen über eigene spezielle Schnittstellen. Viele implementieren auch eine allgemeine Schnittstelle namens **IDispatch**, die VuGen auf spezielle Weise umsetzt. **IDispatch** ist eine Art "Superschnittstelle", die alle anderen Schnittstellen und Methoden eines COM-Objekts zur Verfügung stellt. Aufrufe der Methode **IDispatch:Invoke** in VuGen-Skripts werden mithilfe von **lrc_Disp**-Funktionen implementiert. Der Aufbau dieser Aufrufe unterscheidet

sich geringfügig von dem anderer Schnittstellen.

Die von der Schnittstelle **IDispatch** bereitgestellte Methode **Invoke** kann entweder eine Methode ausführen, einen Eigenschaftenwert abrufen oder einen Wert bzw. einen Referenzwert für eine Eigenschaft festlegen. Bei der Standardmethode **IDispatch:Invoke** werden diese Möglichkeiten durch den Parameter **wflags** angegeben. In der VuGen-Implementierung werden sie durch unterschiedliche Prozeduraufrufe realisiert, die entweder eine Methode aufrufen oder eine Eigenschaft festlegen bzw. abrufen.

Ein Aufruf von **IDispatch** zur Aktivierung der Methode **GetAgentsArray** könnte beispielsweise folgendermaßen aussehen:

```
retValue = lrc_DispMethod1((IDispatch*)IDispatch_0, "GetAgentsArray", /*locale*/
1033, LAST_ARG, CHECK_HRES);
```

In dem Aufruf oben werden die folgenden Parameter verwendet:

IDispatch_0	Der Zeiger auf die Schnittstelle **IDispatch**, der durch einen vorangegangenen Aufruf der Methode **IUnknown:Queryinterface** zurückgegeben wurde.
GetAgentsArray	Der Name der aufzurufenden Methode. Im Hintergrund ruft VuGen die ID der Methode aus dem Namen ab.
1033	Das Gebietsschema der Sprache.
LAST_ARG	Das Kennzeichen, das der Schnittstelle **IDispatch** signalisiert, dass keine weiteren Argumente folgen.
CHECK_HRES	Durch dieses Kennzeichen wird die Überprüfung von HRES aktiviert, da der Aufruf bei der Aufzeichnung erfolgreich war.

Darüber hinaus kann noch ein weiterer Parameter, **OPTIONAL_ARGS**, vorhanden sein. Dieser signalisiert, dass VuGen zusätzlich zu den Standardparametern weitere optionale Argumente übergibt. Jedes optionale Argument enthält ein Wertepaar bestehend aus ID bzw. Name und Wert des Arguments. Im folgenden Aufruf von **lrc_DispMethod** werden beispielsweise die optionalen Argumente "#3" und "var3" übergeben:

```
{
        GUID riid = IID_IDispatch;
        lrc_IOptional_QueryInterface(IOptional_0, =;riid, (void**)=;IOptional_0,
CHECK_HRES);
    }
    {
        VARIANT P1 = lrc_variant_short("47");
        VARIANT P2 = lrc_variant_short("37");
        VARIANT P3 = lrc_variant_date("3/19/1901");
        VARIANT var3 = lrc_variant_scode("4");
        lrc_DispMethod((IDispatch*)IOptional_0, "in_out_optional_args", /*locale
*/1024, =;P1, =;P2, OPTIONAL_ARGS, "#3", =;P3, "var3", =;var3, LAST_ARG, CHECK_H
RES);
}
```

Die Function Reference enthält eine detaillierte Auflistung der verschiedenen **lrc_Disp**-Methoden, welche die Schnittstelle **IDispatch** verwenden.

Typenkonvertierungen und Datenextraktion

Wie aus dem Beispiel oben deutlich wird, werden viele COM-Parameter als Varianten definiert. Zur Extraktion dieser Werte verwendet VuGen eine Reihe von Konvertierungsfunktionen, die aus vergleichbaren COM-Funktionen abgeleitet werden. Eine vollständige Liste finden Sie in der Function Reference. Weiter oben wurde gezeigt, wie der Aufruf **lrc_DispMethod1** eingesetzt wurde, um ein Array aus Namenzeichenfolgen abzurufen:

```
VARIANT retValue = lrc_variant_empty();
retValue = lrc_DispMethod1((IDispatch*)IDispatch_0, "GetAgentsArray", /*locale*/
1033, LAST_ARG, CHECK_HRES);
```

Das folgende Beispiel zeigt, wie VuGen die Zeichenfolgen aus **retValue** abruft, eine Variante, die als Zeichenfolgenarray interpretiert wird.

In einem ersten Schritt extrahiert VuGen das BSTR-Array aus der Variante:

```
BstrArray array0 = 0;
array0 = lrc_GetBstrArrayFromVariant(=;retValue);
```

VuGen stellt für alle Werte in **array0** Code zur Verfügung, mit dem Sie die Elemente aus dem Array zur späteren Parametrierung wie im Beispiel unten gezeigt extrahieren können:

```
//GetElementFrom1DBstrArray(array0, 0); // value: Alex
//GetElementFrom1DBstrArray(array0, 1); // value: Amanda
....
```

VuGen verfügt über eine Reihe von Funktionen für die Typenkonvertierung und Extraktion von konventionellen Typen aus Varianten. Ausführliche Informationen zu diesen Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Auswählen der aufzuzeichnenden COM-Objekte

Die zu testende Anwendung verwendet möglicherweise zahlreiche COM-Objekte. Nur einige davon erzeugen tatsächlich Last und sind für den Lasttest relevant. Aus diesem Grund sollten Sie vor dem Aufzeichnen einer COM-Anwendung die gewünschten Objekte auswählen, die für den Lasttest erfasst werden sollen. VuGen stellt eine Suchfunktion für COM-Objekte in Typbibliotheken zur Verfügung, die sich entweder auf dem lokalen Computer oder auf anderen Computern im Netzwerk befinden können.

Entscheiden, welche Objekte relevant sind

Es gibt verschiedene Möglichkeiten zu entscheiden, welche COM-Objekte in den Test einbezogen werden sollen. Versuchen Sie herauszufinden, welche Remoteobjekte von der Software verwendet werden. Wenn Sie sich nicht sicher sind, welche diese sind, verwenden Sie den Standardfilter. Die Verzweigung **Environments** enthält Aufrufe der drei Objektsätze (ADO, RDS und Remote), die wahrscheinlich Last auf den Remoteservern erzeugen werden.

Zur Verfeinerung des Filters können Sie aber auch die tatsächlich erfolgten Aufrufe überprüfen. Nachdem Sie den Test aufgezeichnet haben, können Sie die Datei speichern und in dem von

VuGen erstellen Ordner **data** nach einer Datei namens **lrc_debug_list_<nnn>.log** suchen, wobei **nnn** die Prozessnummer darstellt. Diese Protokolldatei enthält eine Liste aller COM-Objekte, die von der aufgezeichneten Anwendung aufgerufen wurden, egal, ob diese durch den Aufzeichnungsfilter erfasst wurden oder nicht. Es sollten nur Aufrufe, die Last auf dem Server erzeugen, für die Aufzeichnung berücksichtigt werden.

Das folgende Beispiel zeigt einen lokalen COM-Aufruf der Visual Basic-Bibliothek:

```
Class JetES {039EA4C0-E696-11D0-878A-00A0C91EC756}
was loaded from type library "JET Expression Service Type Library"
({2358C810-62BA-11D1-B3DB-00600832C573} ver 4.0)
```

Der Aufruf sollte nicht hinzugefügt werden, da er keine Last auf dem Server erzeugt.

Ähnlich verhält es sich in den folgenden Beispielen. Da es sich bei der OLE DB und den allgemeinen Microsoft Windows-Steuerelementen um lokale Objekte handelt, erzeugen die Klassen und Bibliotheken keinerlei Last auf dem Server und sollten deshalb auch nicht aufgezeichnet werden:

```
Class DataLinks {2206CDB2-19C1-11D1-89E0-00C04FD7A829}
was loaded from type library "Microsoft OLE DB Service Component 1.0 Type Librar
y"
({2206CEB0-19C1-11D1-89E0-00C04FD7A829} ver 1.0)
Class DataObject {2334D2B2-713E-11CF-8AE5-00AA00C00905}
was loaded from type library "Microsoft Windows Common Controls 6.0 (SP3)"
({831FDD16-0C5C-11D2-A9FC-0000F8754DA1} ver 2.0)
```

Im Gegensatz dazu enthält die folgende Auflistung eine Klasse, die aufgezeichnet werden sollte, da sie Last auf dem Server erzeugt:

```
Class Order {B4CC7A90-1067-11D4-9939-00105ACECF9A}
was loaded from type library "FRS"
({B4CC7A8C-1067-11D4-9939-00105ACECF9A} ver 1.0)
```

Aufrufe von Klassen aus der Bibliothek **FRS**, wie sie u. a. im **flight_sample**-Beispiel verwendet werden, das mit der Installation von VuGen bereitgestellt wird, belegen Serverkapazität und sollten aufgezeichnet werden.

Wenn ein COM-Objekt selbst andere COM-Objekte aufruft, werden alle Aufrufe in der Protokolldatei aufgelistet. Bei jedem Aufruf der **FRS**-Klassenfunktion ruft die Bibliothek **FRS** beispielsweise die Bibliothek **ActiveX Data Object (ADO)** auf. Wenn mehrere Funktionen innerhalb einer solchen Kette im Filter enthalten sind, zeichnet VuGen lediglich den ersten Aufruf auf, der die Kette auslöst. Wenn Sie sowohl **FRS**- als auch **ADO**-Aufrufe ausgewählt haben, werden nur die **FRS**-Aufrufe aufgezeichnet.

Wenn Sie andererseits nur die Bibliothek **ADO** im Filter auswählen, werden Aufrufe der Bibliothek **ADO** aufgezeichnet. Es ist oft einfacher, den Aufruf des ersten Remoteobjekts in der Kette aufzuzeichnen. Es gibt jedoch auch Fälle, in denen eine Anwendung Methoden mehrerer unterschiedlicher COM-Objekte verwendet. Wenn diese alle ein einzelnes Objekt verwenden, das Last auf dem Server erzeugt, reicht es aus, das gemeinsame Objekt aufzuzeichnen.

Auswählbare Objekte

VuGen kann nur dann Objekte aufzeichnen, wenn es die zugehörigen Typbibliotheken lesen kann.

Wenn diese nicht im System installiert sind oder VuGen sie nicht finden kann, werden die zugehörigen COM-Objekte nicht im Dialogfeld **Aufzeichnungsoptionen** aufgeführt. Wenn die Objekte von der Anwendung verwendet werden, ist VuGen nicht in der Lage, sie zu identifizieren. In den Dateien werden sie als **INoTypeInfo** gekennzeichnet.

Auszuschließende Schnittstellen

Im Dialogfeld **Aufzeichnungsoptionen** werden Ihnen für jedes Objekt alle in der Typbibliothek enthaltenen Schnittstellen angezeigt, die Sie einzeln ein- oder ausschließen können. **ADO**, **RDS** und Remoteobjekte können jedoch als Gruppe in den Filter aufgenommen werden. Der Filter zeigt nicht die einzelnen Objekte für diese Umgebungen oder deren Schnittstellen an. Objekte, die Sie aus den Typbibliotheken zum Filter hinzugefügt haben, verfügen möglicherweise über Schnittstellen, die dort nicht enthalten sind, und werden deshalb nicht im Dialogfeld **Aufzeichnungsoptionen** angezeigt. Nachdem das VuGen-Skript erzeugt wurde, können Sie diese Schnittstellen identifizieren und die zugehörigen GUIDs aus der von VuGen erstellten Datei **interfaces.h** entnehmen. Mithilfe dieser Information können Sie die Schnittstellen dann wie unter "Knoten "COM/DCOM > Filter"" auf Seite 169 beschrieben aus dem Filter ausschließen.

Datenbankprotokolle

Datenbankprotokolle – Übersicht

Nehmen wir einmal an, dass Sie über eine Kundendatenbank verfügen, auf die alle Kundendienstmitarbeiter in einem Land zugreifen können. In diesem Fall verwenden Sie Datenbank-Vuser zum Simulieren einer Situation, in der der Datenbankserver zahlreiche Informationsanfragen bedienen muss. Ein Datenbank-Vuser hat folgende Möglichkeiten:

- Herstellen einer Verbindung zum Server
- Übermitteln einer SQL-Abfrage
- Abrufen und Verarbeiten von Informationen
- Trennen der Verbindung zum Server

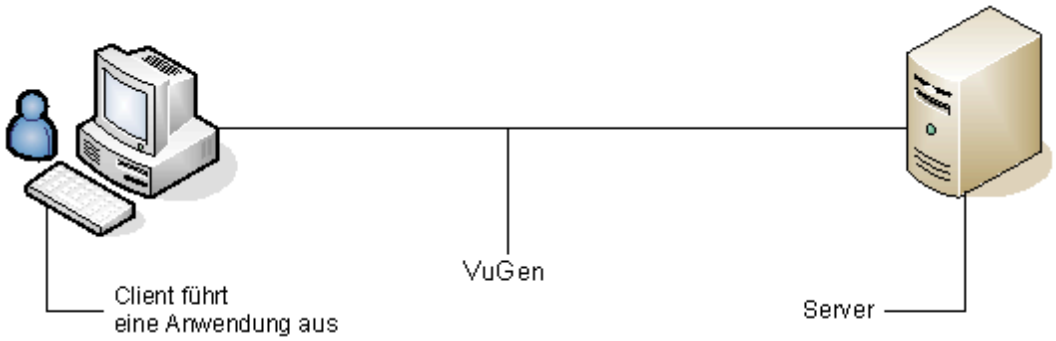
Sie verteilen mehrere hundert Datenbank-Vuser auf die verfügbaren Lastgeneratoren, wobei jeder Vuser über die Server-API Zugriff auf die Datenbank hat. Auf diese Weise können Sie die Leistung Ihres Servers unter der Last eine großen Anzahl von Benutzern messen.

Bei dem Programm, das die Aufrufe an die Server-API enthält, handelt es sich um ein Datenbank-Vuser-Skript. Das Skript emuliert die Clientanwendung sowie alle von der Anwendung durchgeführten Aktionen. Die Vuser führen das Skript aus und emulieren die Benutzerlast auf dem Client-/Server-System. Die Vuser generieren die Leistungsdaten, die Sie anschließend in einem Bericht- und Grafikformat analysieren können.

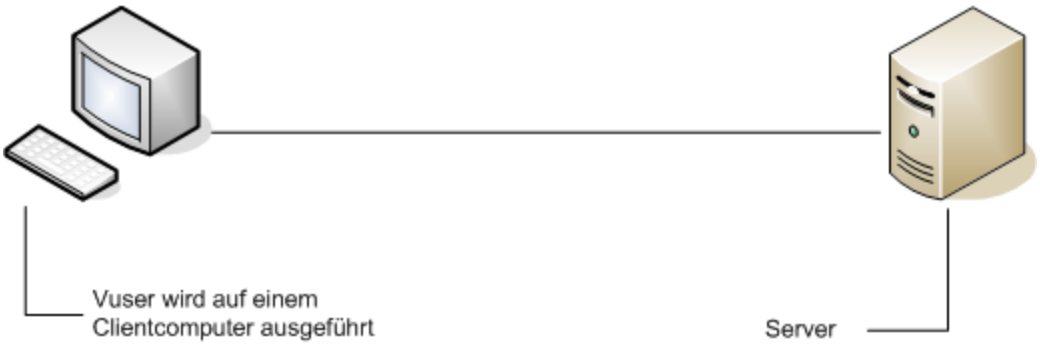
VuGen unterstützt die folgenden Datenbanktypen: CtLib, DbLib, Oracle, ODBC und DB2-CLI. Das daraus resultierende Skript enthält LRD-Funktionen, mit denen die Datenbankaktivität beschrieben wird.

VuGen-Aufzeichnungstechnologie für Datenbanken

VuGen erstellt Datenbank-Vuser-Skripts durch Aufzeichnen aller Aktivitäten zwischen einem Datenbankclient und einem Server. VuGen überwacht die Clientseite der Datenbank und protokolliert alle Anfragen, die vom Datenbankserver gesendet oder empfangen werden.



Wie alle anderen mit VuGen erstellten Vuser kommunizieren Datenbank-Vuser mit dem Server, ohne Clientsoftware hinzuzuziehen. Stattdessen führt jeder Datenbank-Vuser ein Skript aus, das die Aufrufe direkt in den Server-API-Funktionen ausführt.



Sie verwenden VuGen zum Erstellen von Datenbank-Vuser-Skripts in einer Windows-Umgebung. Nachdem Sie ein Skript erstellt haben, können Sie es Vusern in einer Windows- oder einer Linux-Umgebung zuweisen.

Benutzer, die ausschließlich in einer Linux-Umgebung arbeiten, können Datenbank-Vuser-Skripts erstellen, indem sie VuGen-Vorlagen als Skriptbasis für die Programmierung verwenden. Weitere Informationen zur Programmierung von Datenbank-Vuser-Skripts unter Linux finden Sie unter "Erstellen und Ausführen von Skripts unter Linux" auf Seite 1082.

Datenbanktabellen

Wenn Sie ein Vuser-Skript aufzeichnen oder wiedergeben, werden die von jeder Abfrage zurückgegebenen Daten in einer Datentabelle angezeigt. In einem Vuser-Skript wird das Vorhandensein einer Datentabelle durch eine **GRID**-Anweisung angezeigt. VuGen zeigt Datentabellen entweder im Datenraster-Bereich oder im Snapshot-Bereich an.

- Eine Einführung zum Snapshot-Bereich finden Sie unter "Snapshot-Bereich - Übersicht" auf Seite 62.
- Details zum Arbeiten mit dem Snapshot-Bereich finden Sie unter "Arbeiten mit Snapshots" auf Seite 65.
- Details zur Benutzeroberfläche des Snapshot-Bereichs finden Sie unter "Snapshot-Bereich" auf Seite 109.

So korrelieren Sie einen Wert in einer Datentabelle:

Zeigen Sie die Datentabelle im Snapshot-Bereich an, klicken Sie mit der rechten Maustaste in eine Zelle der Datentabelle und wählen Sie **Korrelation erstellen** aus.

So speichern Sie die Daten einer Datentabelle in einer Datei:

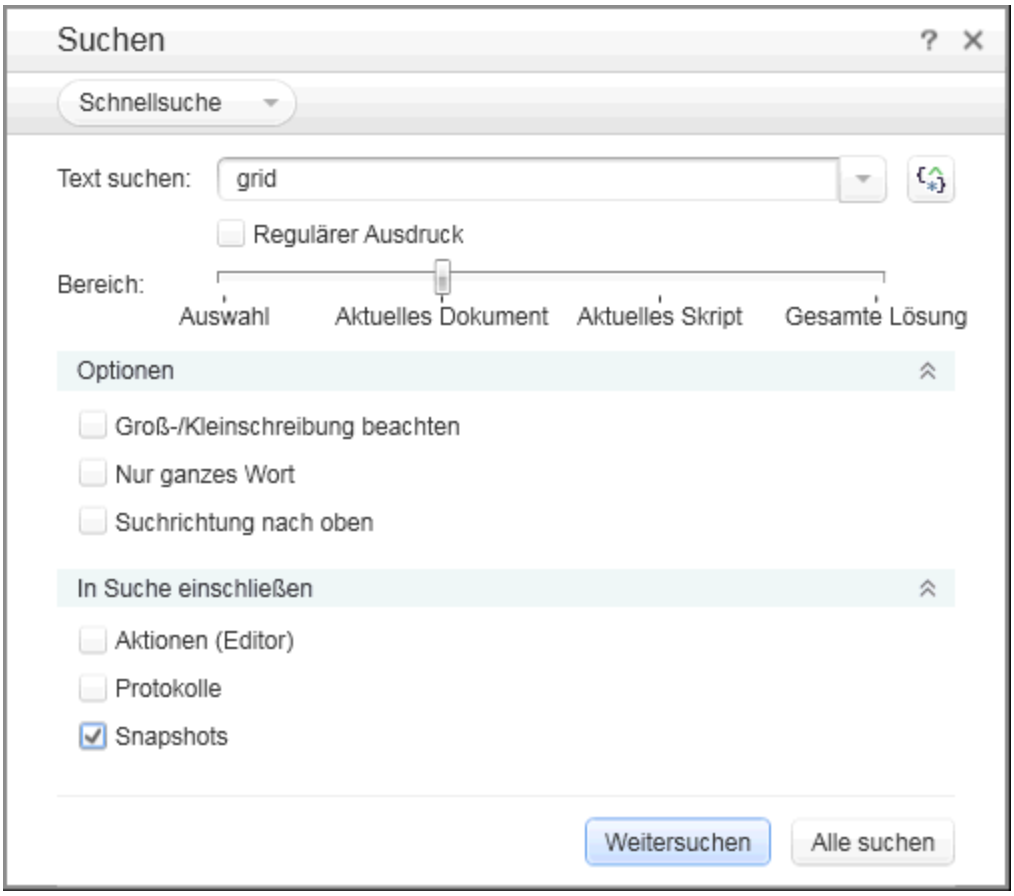
Zeigen Sie die Datentabelle im Snapshot- oder im Datenraster-Bereich an, klicken Sie mit der rechten Maustaste in eine Zelle der Datentabelle und wählen Sie **Raster in Datei speichern** aus.

So kopieren Sie den Text aus einer Zelle der Datentabelle in die Zwischenablage:

Zeigen Sie die Datentabelle im Snapshot- oder im Datenraster-Bereich an, klicken Sie mit der rechten Maustaste in eine Zelle der Datentabelle und wählen Sie **Auswahl kopieren** aus.

So suchen Sie in einer Datentabelle nach Daten:

1. Um die Datentabelle im Snapshot-Bereich anzuzeigen, klicken Sie auf **Suchen > Schnellsuche**, um das Dialogfeld **Suchen** zu öffnen.
2. Klicken Sie auf **In Suche einschließen** und aktivieren Sie anschließend das Kontrollkästchen **Snapshots**.



Protokoll "Flex (RTMP/AMF)"

Flex – Übersicht

Hinweis: Dieses Thema bezieht sich nur auf Flex-Vuser-Skripts.

Das Flex-Vuser-Protokoll emuliert die Kommunikation zwischen einer Client-Server-Anwendung, die die Flex-Technologien nutzt.

Hier finden Sie einen Überblick über die verschiedenen Themen zu Flex-Vuser-Skripts.

Was ist Flex?

Flex ist ein Framework zum Erstellen von RIAs (Rich Internet Applications), die auf Flash Player basieren.

RIAs sind schlanke webbasierte Programme, die dem Benutzer eine dynamischere Interaktion ermöglichen als herkömmliche Webseiten. Ähnlich wie Webanwendungen, die mit Ajax erstellt wurden, bieten Flex-Anwendungen allgemein eine erhöhte Reaktionsgeschwindigkeit, da die

Webseiten bei Benutzereingaben nicht jedes Mal komplett neu geladen werden müssen. Im Gegensatz zu Ajax ist Flex jedoch von Browserimplementierungen wie JavaScript oder CSS unabhängig. Zur Ausführung nutzt das Framework Adobes plattformunabhängige Flash Player-Anwendung.

Flex-Anwendungen bestehen aus einer Vielzahl von MXML- und ActionScript-Dateien. Sie werden zu einer einzigen SWF-Filmdatei kompiliert, die im Clientbrowser von Flash Player abgespielt werden kann.

Hinweis: Verwenden Sie für Flex-Anwendungen, die mit SOAP-Daten arbeiten, das Vuser-Protokoll **Web Services**.

Flex-Technologien

In der folgenden Tabelle werden die speziellen Technologien beschrieben, die von der VuGen-Aufzeichnungsengine unterstützt werden.

Technologie	Beschreibung
AMF0	Action Message Format
AMF3	Action Message Format - Komprimiertes Format
RTMP	Real Time Message Protocol: Messaging und Streaming über TCP
RTMPS	Real Time Message Protocol: Messaging und Streaming über TCP/SSL
RTMPT	Real Time Message Protocol Tunneled: Messaging und Streaming über HTTP

VuGen unterstützt die folgenden Entwicklungslösungen:

Entwicklungsplattformen	Beschreibung
BlazeDS	Open-Source-Lösung für Remoting und Messaging
GraniteDS	Open-Source-Entwicklungs- und -Integrationslösung zum Erstellen von Flex-Anwendungen.
LiveCycle	Adobe-Entwicklungs- und -Integrationslösung zum Erstellen von Flex-Anwendungen.

Die nachfolgende Tabelle enthält eine Liste der LoadRunner-Dokumentation in Bezug auf die Entwicklung eines Flex-Vuser-Skripts.

Thema	Beschreibung
Erstellen eines Flex-Vuser-Skripts	Allgemeine VuGen-Dokumentation zum Erstellen von Vuser-Skripts ["Erstellen von über Vuser-Skripts – Übersicht" auf Seite 148].

Thema	Beschreibung
Aufzeichnen	Zusätzlich zur allgemeinen Dokumentation zum Aufzeichnen von Vuser-Skripts ["Aufzeichnen – Übersicht" auf Seite 162] siehe: • "Aufzeichnen eines Flex-Skripts" auf Seite 697 • "Festlegen des Flex-Aufzeichnungsmodus" auf Seite 698 • "RTMP/RTMPT-Streaming" auf Seite 688 • "RTMP Tunneled" auf Seite 696 Möglicherweise müsse Sie Aufzeichnungsoptionen für das Flex-Skript konfigurieren: • " Flex > RTMP (Knoten)" auf Seite 189 • "Flex > Externalisierbare Objekte (Knoten) (Aufzeichnen)" auf Seite 209
Korrelieren	Zusätzlich zur VuGen-Dokumentation zum Korrelieren von Vuser-Skripts ["Korrelieren – Übersicht" auf Seite 281] siehe: "Flex-Korrelationen" auf Seite 703
Wiedergeben	Zusätzlich zur VuGen-Dokumentation zum Wiedergeben von Vuser-Skripts ["Übersicht über die Wiedergabe" auf Seite 329] siehe: • "Abfragen einer XML-Struktur" auf Seite 706 Möglicherweise müsse Sie Laufzeiteinstellungen für das Flex-Skript konfigurieren: • "Flex > RTMP (Knoten)" auf Seite 348 • "Flex > Externalisierbar (Knoten) (Laufzeiteinstellungen)" auf Seite 349 • "Flex > Konfiguration (Knoten) (Laufzeiteinstellungen)" auf Seite 348
Debuggen	Allgemeine Dokumentation zum Debuggen von Vuser-Skripts ["Debuggen – Übersicht" auf Seite 413]. • "Externalisierbare Objekte in Flex-Skripts" auf Seite 700 • "Serialisieren von Flex-Skripts " auf Seite 704
Anzeigen von Testergebnissen	Allgemeine Dokumentation zum Anzeigen von Testergebnissen ["Anzeigen von Wiedergabeergebnissen – Übersicht" auf Seite 513].

Aufzeichnen von Flex-Skripts

Beim Aufzeichnen einer Flex-Anwendung generiert VuGen Flex-Vuser-Skriptfunktionen, die die Anwendung emulieren. In der folgenden Tabelle werden die Funktionen beschrieben, die vom Flex-Protokoll unterstützt werden.

AMF

Mit dem **Flex**-Protokoll können Sie in VuGen Skripts zur Emulation von Flex-Anwendungen erstellen, die mit AMF0 und AMF3-arbeiten.

Funktionsname	Beschreibung
flex_amf_call	Sendet eine AMF-Anfrage.
flex_amf_define_envelope_header_set	Definiert einen Satz von Umschlagkopfzeilen.
flex_amf_define_header_set	Definiert einen Satz von AMF-Kopfzeilen.
flex_login	Meldet sich bei einer kennwortgeschützten Flex-Anwendung an.
flex_logout	Meldet sich bei einer kennwortgeschützten Flex-Anwendung ab.
flex_ping	Überprüft, ob eine Flex-Anwendung verfügbar ist.
flex_remoting_call	Ruft eine oder mehrere Methoden auf einem serverseitigen Remoteobjekt (RPC) auf.

AMF-Beispiel 1**:**

Im folgenden Beispiel wird die Verfügbarkeit eines Dienstes mithilfe der Funktion **flex_ping** überprüft. Die Funktion **flex_remoting_call** ruft den Service remote auf.

```
flex_ping("1",
    "URL=http://<HOST>/weborb.aspx",
    "Snapshot=t6.inf",
    LAST);
flex_remoting_call("getProductEdition::GenericDestination",
    "URL=http://testlab1/weborb30/console/weborb.aspx",
     "Snapshot=t1.inf",
    INVOCATION,
    "Target=/2",
    "Operation=getProductEdition",
    "Destination=GenericDestination",
    "DSEndpoint=my-amf",
    "Source=Weborb.Management.LicenseService",
    "Argument=<arguments/>",
    LAST);
```

AMF-Beispiel 2:

In dem folgenden AMF0-Beispiel greift die Funktion **flex_amf_call** auf ein Gateway zu und sendet eine Nachricht an den Server.

```
flex_amf_call("EchoAny",
       "Gateway=http://<host>/gateway.aspx",
       "Snapshot=t1.inf",
       "IsParseResponse=No",
       MESSAGE,
       "Method=EchoAMF.EchoAMF.EchoAny",
       "TargetObjectId=/1",
       BEGIN_ARGUMENTS,
             "<boolean>true</boolean>",
       END_ARGUMENTS,
       LAST);
```

AMF-Beispiel 3:

In dem folgenden AMF3-Beispiel sendet die Funktion **flex_remoting_call** einen AMF-Aufruf an den Server, der serialisiert werden kann.

```
flex_remoting_call(
"product::getProductsByName",
 URL=http://<HOST>:<PORT>/amf;jsessionid={CorrelationParameter}",
 "Snapshot=t1.inf",
 "IsParseResponse=No",
 INVOCATION,
 "Target=/2",
 "Operation=getProductsByName",
 "Destination=product",
 "DSEndpoint=my-amf",
 "DSId=8E3759E5-E51A-3906-0EAB-6119CD1E26BF",
 "Argument="
       "<arguments>"
              "<string>A</string>"
       "</arguments>",
 LAST);
```

RTMP-Funktionen

Funktionsname	Beschreibung
flex_rtmp_ connect	Stellt eine Verbindung zwischen einem Client und einem RTMP-Server her und legt die Verbindungsoptionen fest.
flex_rtmp_ disconnect	Trennt die Verbindung eines Clients mit einem RTMP-Server.

Funktionsname	Beschreibung
flex_rtmp_ receive_stream	Empfängt Streamingdaten von einem RTMP-Server.
flex_rtmp_ receive	Empfängt Antworten von einem RTMP-Server.
flex_rtmp_send	Sendet eine Anforderung an einen RTMP-Server.

RTMP-Beispiel

Im folgenden Beispiel empfängt **flex_rtmp_receive** Daten.

```
     flex_rtmp_receive("recv_step0",
"ConnectionID=19",
"Snapshot=tRTMP44.inf",
CHANNEL,
"ChunkStreamID=2",
CHANNEL,
"ChunkStreamID=2",
LAST);
```

RTMP-Tunneled-Funktionen

Funktionsname	Beschreibung
flex_rtmp_tunneled_ connect	Verbindet einen Client über HTTP mit einem RTMP-Server.
flex_rtmp_tunneled_ disconnect	Trennt die Verbindung eines Clients von der Sitzung über HTTP mit einem RTMP-Server.
flex_rtmp_tunneled_ send	Sendet eine Anforderung über HTTP an einen RTMP-Server.

Beispiel für RTMP Tunneled

Im folgenden Beispiel sendet **flex_rtmp_tunneled_send** eine RTMP-Tunneled-Anforderung.

```
flex_rtmp_tunneled_send(
 "send_step0",
 "SessionID=0",
 "Snapshot=t30.inf",
 MESSAGE,
 "DataType=command message amf3",
 "ChunkStreamID=3",
 "MessageStreamID=0",
 "Argument="
       "<arguments>"
                          ...
```

```
        "</arguments>",
LAST);
```

Detaillierte Informationen zur Syntax sämtlicher Flex-Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

RTMP/RTMPT-Streaming

Das Flex-Protokoll von LoadRunner unterstützt die Aufzeichnung und Wiedergabe von Streaming für das RTMP- und das RTMPT-Protokoll. Sie können im normalen Aufzeichnungsmodus oder im vereinfachten Aufzeichnungsmodus aufzeichnen. Der vereinfachten Modus ermöglicht VuGen, eine einzelne Funktion anstelle mehrerer Funktionen zu generieren, die generiert werden, wenn der normale Aufzeichnungsmodus verwendet wird. Wenn Sie den vereinfachten Modus zum Aufzeichnen verwenden, geschieht Folgendes:

- RTMP-basierter Stream: VuGen generiert einen einzelnen **flex_rtmp_receive_stream**-Schritt anstelle von vielen **flex_rtmp_receive**- und **flex_rtmp_send**-Schritten.
- RTMPT-basierter Stream: VuGen generiert einen einzelnen geänderten **flex_rtmp_tunneled_send**-Schritt anstelle von vielen **flex_rtmp_tunneled_send**-Schritten.

Der einzelne generierte **flex_rtmp_receive_stream**- oder **flex_rtmp_tunneled_send**-Schritt vereinfacht die Lesbarkeit des Vuser-Skripts (durch Einsparung mehrerer Zeilen von Code) und erhöht die Zuverlässigkeit der Skriptwiedergabe. Es wird empfohlen, den vereinfachten Modus für die Aufzeichnung Ihrer Vuser-Skripts zu verwenden, es sei denn, die Vuser-Aktivität beinhaltet ein asynchrones Verhalten, wie nachfolgend beschrieben.

Der vereinfachte Aufzeichnungsmodus ist der Standardaufzeichnungsmodus für Streaming. Zum Aktivieren des vereinfachten Modus öffnen Sie das Dialogfeld **Aufzeichnungsoptionen**, klicken Sie auf **Flex > RTMP**, und aktivieren Sie das Kontrollkästchen **Einzelnen Schritt zur Verarbeitung von RTMP/T-Datenstrom generieren**.

Die Unterschiede zwischen dem vereinfachten und dem normalen Aufzeichnungsmodus werden nachfolgend aufgeführt:

	Vereinfachter Modus	**Normaler Modus**
(Aufzeichnungsoption) Kontrollkästchen **Einzelnen Schritt zur Verarbeitung von RTMP/T-Datenstrom generieren**	Aktiviert	Nicht aktiviert

	Vereinfachter Modus	Normaler Modus
Generierte Funktionen	RTMP: Generiert **flex_rtmp_receive_stream**-Funktionen. RTMPT: Generiert **flex_rtmp_tunneled_send**-Funktionen.	Generiert **flex_rtmp_receive**- und **flex_rtmp_send**-Schritte.
Anzahl der generierten Funktionen pro Stream	Eine	Mehrere
Unterstützt asynchrones Verhalten	Nein	Ja
Standardmodus	Ja	Nein

Hinweis: Der vereinfachte Aufzeichnungsmodus wird für Flash Media Server, Versionen 3.5 und 4, unterstützt.

Synchrones Vuser-Verhalten

Die **flex_rtmp_receive_stream**- und **flex_rtmp_tunneled_send**-Funktionen, die generiert wenn, wenn der vereinfachte Aufzeichnungsmodus ausgewählt ist, sind synchrone Funktionen. Das bedeutet, dass keine anderen Vuser-Funktionen ausgeführt werden können, während eine dieser Funktionen ausgeführt wird. Gehen Sie beispielsweise von einem Vuser-Skript aus, das eine **flex_rtmp_receive_stream**-Funktion enthält, die 5 Minuten lang ein Video streamt. In den 5 Minuten, in denen das Video gestreamt wird, kann der Vuser keine anderen Aktionen durchführen, z. B. auf die Schaltfläche **Anhalten** klicken oder an eine andere Position im Video springen. Das Klicken auf eine Schaltfläche während des Streamens eines Videos ist ein Beispiel für asynchrones Verhalten.

Obwohl ein einzelner erzeugter Schritt die Skriptwiedergabe zuverlässiger macht, können so keine asynchronen Aktionen (z. B. Anhalten und Suchvorgänge) wiedergegeben werden, die Sie möglicherweise während der Aufzeichnung des Skripts ausgeführt haben. Der einzelne generierte Schritt gibt auch die automatischen Anforderungen nicht wieder, die beim dynamischen Streaming vom Client ausgeführt werden. Wenn es ist wichtig ist, dass die asynchronen Aktionen wiedergegeben werden, müssen Sie das Vuser-Skript unter Verwendung des normalen Aufzeichnungsmodus anstelle des vereinfachten Aufzeichnungsmodus aufzeichnen und dann das generierte Skript wie nachfolgend beschrieben ändern.

Ändern von Skripts für die Wiedergabe von asynchronen Benutzeraktionen

Wenn Ihr Vuser-Skript in der Lage sein muss, asynchrone Aktionen wiederzugeben, die durchgeführt werden, während eine Streamingaktion ausgeführt wird, müssen Sie das Vuser-Skript unter Verwendung des normalen Aufzeichnungsmodus anstelle des vereinfachten Aufzeichnungsmodus aufzeichnen und dann das generierte Skript manuell ändern. Das geänderte Skript enthält eine Kombination aus einzelnen Streamingschritten und den ausführlicheren Schritten, die im normalen Aufzeichnungsmodus generiert werden.

Hinweis: In diesem Abschnitt verweist der Begriff *erforderliche Benutzeraktionen* auf die Aktionen, die durchgeführt werden müssen, während ein Video gestreamt wird.

Zum Erstellen eines Skripts, das asynchrones Verhalten wiedergeben kann, zeichnen Sie zuerst das Skript im normalen Aufzeichnungsmodus, nicht im vereinfachten Aufzeichnungsmodus auf. Anschließend bestimmen Sie die **flex_rtmp_send**-Schritte, die die *erforderlichen Benutzeraktionen* darstellen. Ersetzen Sie dann die Schritte zwischen den *erforderlichen Benutzeraktionen* durch einzelne Streamingfunktionen. Einzelheiten dazu enthalten die nachfolgenden Abschnitte.

Hinweis: Das Änderungsverfahren zwischen RTMP- und RTMPT-Schritten unterscheidet sich geringfügig.

Ändern von aufgezeichneten Flex-RTMP-Schritten

Wenn Sie den normalen Aufzeichnungsmodus verwenden, generiert VuGen **flex_rtmp_receive**- und **flex_rtmp_send**-Schritte für die gesamte Kommunikation mit dem Server. Dadurch ist sichergestellt, dass Benutzeraktionen wie Anhalten und Suchen sowie automatische Anforderungen, die beim dynamischen Streaming vom Client ausgeführt werden, im Skript enthalten sind. Bei dieser Methode werden jedoch auch nicht unbedingt erforderliche Codezeilen aufgezeichnet, welche während der Wiedergabe von Streamingaktionen die Lesbarkeit und die Zuverlässigkeit verschlechtern.

Hinweis: Zur Aktivierung des normalen Aufzeichnungsmodus deaktivieren Sie das Kontrollkästchen **Einzelnen Schritt zur Verarbeitung von RTMP/T-Datenstrom generieren** im Bereich **Flex > RTMP** des Dialogfelds **Aufzeichnungsoptionen**.

Folgen Sie den Anleitungen weiter unten, um die unnötigen **flex_rtmp_receive**- und **flex_rtmp_send**-Schritte aus dem Skript zu entfernen.

1. Suchen Sie nach dem **flex_rtmp_send**-Schritt, der das erste play-Argument enthält. Beispiel:

```
flex_rtmp_send("send_step2",
"ConnectionID=10",
"Snapshot=tRTMP6.inf",
MESSAGE,
...
...
MESSAGE,
...
...
"Argument=<arguments><string>play</string><number>0</number><null/>"
...
LAST);
```

2. Löschen Sie die während des Streamings auftretenden **flex_rtmp_receive**-Schritte, oder

kommentieren Sie sie aus. Beispiel:

```
//this is the start of the stream:
//      flex_rtmp_receive("recv_step2",
//              "ConnectionID=10",
//              "Snapshot=tRTMP7.inf",
//              CHANNEL,
//              "ChunkStreamID=2",
//              CHANNEL,
//              "ChunkStreamID=2",
//              CHANNEL,
//              "ChunkStreamID=4",
//              CHANNEL,
//              "ChunkStreamID=2",
//              LAST);
//
//      flex_rtmp_receive("recv_step3",
//              "ConnectionID=10",
//              "Snapshot=tRTMP8.inf",
//              CHANNEL,
//              "ChunkStreamID=5",
//              CHANNEL,
//              ...
//              ...
```

3. Entfernen Sie die **flex_rtmp_send**-Schritte, die nicht mit erforderlichen Benutzeraktionen verknüpft sind, z. B. Schritte vom Typ "user control message". Beispiel:

```
//      flex_rtmp_send("send_step3",
```

```
//              "ConnectionID=10",
//              "Snapshot=tRTMP9.inf",
//              MESSAGE,
//              "DataType=user control message",
//              "EventType=set buffer length",
//              "MessageStreamID=1",
//              "BufferLength=100",
//              LAST);
```

4. Wenn Sie einen **flex_rtmp_send**-Schritt finden, der eine erforderliche Benutzeraktion darstellt, gehen Sie folgendermaßen vor:

   a. Fügen Sie vor dem Sende-Schritt manuell einen **flex_rtmp_receive_stream**-Schritt ein.

   - Stellen Sie sicher, dass das Argument **ConnectionID** denselben Wert hat wie die Schritte, die Sie oberhalb davon entfernt haben.

   - Das Argument **Snapshot** hat für den manuell hinzugefügten Schritt keine Relevanz.

   - Über das Argument **ContinueToNexStepAfter = <ms>** können Sie festlegen, wie lange der herunterzuladende Stream minimal wiedergegeben werden muss, bevor mit dem nächsten Schritt fortgefahren wird.

   b. Bestimmen Sie die **flex_rtmp_send**-Schritte, die die erforderlichen Benutzeraktionen darstellen. Diese beinhalten wahrscheinlich Argumente wie z. B. **pauseRaw**, **pause**, **seek** und **play2** (für Dynamic Stream). Beispiel:

```
flex_rtmp_send("send_step5",
"ConnectionID=10",
"Snapshot=tRTMP62.inf",
MESSAGE,
"DataType=command message amf3",
"ChunkStreamID=8",
"MessageStreamID=1",
"Argument=<arguments><string>pauseRaw</string><number>0</number><null/>"
"<boolean>true</boolean><number>12000</number></arguments>",
LAST);
```

c. Ermitteln Sie, welche **flex_rtmp_send**-Schritte überflüssig sind und entfernt werden können. Wenn Sie eine Komponente, z. B. eine Schaltfläche, innerhalb des Datenstroms ziehen, könnten minimale Zuckungen oder Unterbrechungen der Bewegung als Pausen- bzw. getrennte Suchaktion aufgezeichnet werden. In diesem Fall werden nicht alle aufgezeichneten Aktionen benötigt. Behalten Sie nur diejenigen im Skript, die zur Beschreibung der gewünschten Operation erforderlich sind.

d. Identifizieren Sie den **flex_rtmp_receive**-Schritt, der angibt, dass der Server das Ende der Benutzeraktion empfangen hat. Beispiel:

```
//this is the confirmation from the server on the "seek" command.
flex_rtmp_receive("recv_step55",
"ConnectionID=10",
"Snapshot=tRTMP68.inf",
CHANNEL,
"ChunkStreamID=2",
CHANNEL,
"ChunkStreamID=2",
LAST);
```

5. Wiederholen Sie die Schritte 2–4 für alle unnötigen Empfangsdaten und erforderlichen Benutzeraktionen im Skript.

Weitere Details zu **flex_rtmp_receive_stream** einschließlich eines vollständigen Beispiels finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Ändern von aufgezeichneten Flex-RTMPT-Schritten

Wenn Sie den normalen Aufzeichnungsmodus verwenden, generiert VuGen einen **flex_rtmp_tunneled_send**-Schritt für die gesamte Kommunikation mit dem Server. Dadurch ist sichergestellt, dass Benutzeraktionen wie Anhalten und Suchen sowie automatische Anforderungen, die beim dynamischen Streaming vom Client ausgeführt werden, im Skript enthalten sind. Bei dieser Methode werden jedoch auch nicht unbedingt erforderliche Codezeilen aufgezeichnet, welche während der Wiedergabe von Streamingaktionen die Lesbarkeit und die Zuverlässigkeit verschlechtern.

Hinweis: Zur Aktivierung des normalen Aufzeichnungsmodus deaktivieren Sie das Kontrollkästchen **Einzelnen Schritt zur Verarbeitung von RTMP/T-Datenstrom generieren** im Bereich **Flex > RTMP** des Dialogfelds **Aufzeichnungsoptionen**.

Folgen Sie den Anleitungen weiter unten, um die unnötigen Schritte aus dem Skript zu entfernen.

1. Suchen Sie nach dem **flex_rtmp_tunneled_send**-Schritt, der das erste play-Argument enthält. Beispiel:

```
flex_rtmp_tunneled_send("send_step2",
"SessionID=1",
"Snapshot=t36.inf",
MESSAGE,
...
...
MESSAGE,
...
...
"Argument=<arguments><string>play</string><number>0</number><null/>"
...
LAST);
```

2. Entfernen Sie **flex_rtmp_tunneled_send**-Schritte entfernen, die nicht mit erforderlichen Benutzeraktionen verknüpft sind, z. B. Schritte vom Typ "user control message". Beispiel:

```
//     flex_rtmp_tunneled_send("send_step3",
//         "SessionID=10",
//         "Snapshot=t15.inf",
//         MESSAGE,
//         "DataType=user control message",
//         "EventType=set buffer length",
//         "MessageStreamID=1",
//         "BufferLength=100",
//         LAST);
```

3. Wenn Sie einen **flex_rtmp_tunneled_send**-Schritt finden, der eine erforderliche Benutzeraktion darstellt, gehen Sie folgendermaßen vor:

   a. Fügen Sie ein Argument **ContinueToNexStepAfter = <ms>** zum vorherigen Schritt hinzu. Über das Argument **ContinueToNexStepAfter = <ms>** können Sie die minimale Wiedergabedauer für den Stream festlegen, der heruntergeladen werden soll, bevor mit dem nächsten Schritt fortgefahren wird. Beispiel:

```
flex_rtmp_tunneled_send("send_step2",
"SessionID=1",
"Snapshot=t36.inf",


//Read the stream until at least 15 seconds of media have been downloaded
"ContinueToNexStepAfter = 15000",
MESSAGE,
```

```
...
...
MESSAGE,
...
...
"Argument=<arguments><string>play</string><number>0</number><null/>"
...
LAST);
```

b. Bestimmen Sie die **flex_rtmp_tunneled_send**-Schritte, die die erforderlichen Benutzeraktionen darstellen. Diese beinhalten in der Regel Argumente wie z. B. **pauseRaw**, **pause**, **seek** und **play2** (für Dynamic Stream). Beispiel:

```
flex_rtmp_tunneled_send("send_step5",
"SessionID=10",
"Snapshot=t16.inf",
MESSAGE,
"DataType=command message amf3",
"ChunkStreamID=8",
"MessageStreamID=1",
"Argument=<arguments><string>pauseRaw</string><number>0</number><null/>"
"<boolean>true</boolean><number>12000</number></arguments>",
LAST);
```

c. Ermitteln Sie, welche **flex_rtmp_tunneled_send**-Schritte überflüssig sind und entfernt werden können. Wenn Sie eine Komponente, z. B. eine Schaltfläche, innerhalb des Datenstroms ziehen, könnten minimale Zuckungen oder Unterbrechungen der Bewegung als Pausen- bzw. getrennte Suchaktion aufgezeichnet werden. In diesem Fall werden nicht alle aufgezeichneten Aktionen benötigt. Behalten Sie nur diejenigen im Skript, die zur Beschreibung der gewünschten Operation erforderlich sind.

4. Wiederholen Sie die Schritte 2 - 3 für alle unnötigen Datensendeschritte und erforderlichen Benutzeraktionen im Skript.

Weitere Details zur **flex_rtmp_tunneled_send**-Funktion einschließlich eines vollständigen Beispiels finden Sie in derFunction Reference (**Hilfe > Funktionsreferenz**).

Livestreaming

Das Flex-Protokoll von VuGen unterstützt die Adobe-Funktion für Livestreaming. Wenn VuGen einen Livestream erkennt, während Sie ein Vuser-Skript aufzeichnen, fügt VuGen der generierten **flex_rtmp_receive_stream**- oder **flex_rtmp_tunneled_send**-Funktion **ContinueToNextStepAfter**- und **ContinueMode**-Argumente hinzu. Diese zusätzlichen Argumente ermöglichen die präzise Wiedergabe des Livestreams. Weitere Informationen zu diesen Argumenten finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Hinweis: Der Standardwert des generierten **ContinueToNextStepAfter**-Arguments ist die Zeitdauer (in Millisekunden), für die das Video gestreamt wurde, während das Vuser-Skript aufgezeichnet wurde.

RTMP Tunneled

VuGen unterstützt die Aufzeichnung von RTMP Tunneled-Schritten in der Flex-Anwendung; diese bestehen aus folgenden Schritttypen:

- **Messaging-Unterstützung**. Das Flex-Protokoll unterstützt die erweiterte Aufzeichnung und Wiedergabe von Messaging und wurde für Adobe LiveCycle Data Services ES2, Version 3.1, zertifiziert.

- **Streaming-Unterstützung**. Das Flex-Protokoll unterstützt die erweiterte Aufzeichnung und Wiedergabe von Streaming. Weitere Informationen finden Sie unter "RTMP/RTMPT-Streaming" auf Seite 688.

  Bei der Aufzeichnung eines Flex-Datenstroms generiert LoadRunner standardmäßig einen einzelnen **flex_rtmp_tunneled_send**-Schritt anstelle von mehreren **flex_rtmp_tunneled_send**-Schritten. Der Schritt **flex_rtmp_receive_stream** erhöht die Lesbarkeit Ihres Skripts (es können hierdurch potentiell Hunderte von Zeilen eingespart werden) und macht die Wiedergabe stabiler.

  Hinweis: Der neue **flex_rtmp_tunneled_send**-Schritt wird generiert, wenn die Option zum **Generieren eines einzelnen RTMP/T-Schritts** im Bereich **Flex:RTMP** des Dialogfelds **Aufzeichnungsoptionen** aktiviert wurde. Obwohl das Skript durch diesen Schritt zuverlässiger wird, werden bestimmte Aktionen, die Sie möglicherweise während der Aufzeichnung des Skripts ausführen, z. B. Anhalten und Suchen, nicht wiedergegeben. Auch die automatischen Anforderungen werden nicht wiedergegeben, die beim dynamischen Streaming vom Client ausgeführt werden.

  Wenn es wichtig ist, dass diese Aktionen ebenfalls wiedergegeben werden, deaktivieren Sie die Option zum Generieren des einzelnen RTMP/T-Schritts (**Generate single RTMP/T step**) im Bereich **Flex > RTMP** des Dialogfelds **Aufzeichnungsoptionen**. LoadRunner generiert nun die Schritte für alle gestreamten Rohdaten.

  Um eine ordnungsgemäße Wiedergabe sicherzustellen, müssen Sie jedoch manuell das generierte Skript wie unter "RTMP/RTMPT-Streaming" auf Seite 688 beschrieben ändern.

  Diese Funktionalität wurde für Flash Media Server, Version 3.5 und 4, getestet und verifiziert.

- **Externalisierbare Objekte**. LoadRunner unterstützt die Übertragung externalisierbarer Objekte über RTMP Tunneled. Weitere Informationen finden Sie unter "Externalisierbare Objekte in Flex-Skripts" auf Seite 700.

- **Benutzerdatenpunkte**. LoadRunner generiert eine Reihe neuer Datenpunkte, die nützliche Informationen für die Analyse liefern.
- Das Flex RTMP-Tunneled-Protokoll unterstützt manuelle Korrelation unter Verwendung der **web_reg_save_param_xpath-API**.

Weitere Details zu **flex_rtmp_tunneled_send** einschließlich eines vollständigen Beispiels finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Aufzeichnen eines Flex-Skripts

In dieser Aufgabe wird beschrieben, wie ein Flex-Vuser-Skript aufgezeichnet wird.

Erstellen Sie ein neues Skript oder öffnen Sie ein vorhandenes.

Wählen Sie **Neues Skript und neue Lösung > Flex-Protokoll**

Weitere Informationen finden Sie unter "Erstellen oder Öffnen eines Vuser-Skripts" auf Seite 148.

Konfigurieren der Aufzeichnungsoptionen

Zu den Aufnahmeoptionen gehören Optionen, die sich darauf auswirken, wie ein Vuser-Skript nach Aufzeichnung oder Neugenerierung des Skripts erzeugt wird.

Bei einem Flex-Skript müssen Sie die folgenden Aufzeichnungsoptionen konfigurieren::

- **Aufzeichnungsoptionen > Flex > RTMP > Einzelnen Schritt für RTMPT/Streaming generieren**

  Diese Option ist standardmäßig aktiviert und ermöglicht VuGen, einen Datenstrom in einem einzigen Schritt aufzuzeichnen. Wenn Sie jedoch nur einen einzelnen Schritt generieren, werden bestimmte Aktionen wie Pause und Suchen nachher nicht wiedergegeben. Wenn es Ihnen darauf ankommt, diese Aktionen wiederzugeben, sollten Sie diese Option deaktivieren.

- **Aufzeichnungsoptionen > Flex > Konfiguration > Externe JVM verwenden**

  Aktivieren Sie diese Option, wenn Sie eine externe Java Virtual Machine verwenden möchten, und geben Sie den JVM-Pfad in dem dafür vorgesehenen Feld an.

- **Aufzeichnungsoptionen > Flex > Konfiguration > GraniteDS verwenden**

  Aktivieren Sie diese Option, wenn Sie GraniteDS als serverseitige Datendienstkonfiguration verwenden.

- **Aufzeichnungsoptionen > Flex > Externalisierbare Objekte**

  Mithilfe dieser Option können Sie zusätzliche .jar-Dateien angeben, die für die Aufzeichnung des Skript erforderlich sind.

  Weitere Informationen finden Sie unter "Externalisierbare Objekte in Flex-Skripts" auf Seite 700.

Informationen zum Konzept und zur Benutzeroberfläche finden Sie unter "Aufzeichnungsoptionen" auf Seite 164.

Starten der Aufzeichnungssitzung

Bei der Erstellung eines neuen Skripts erfolgt dies automatisch. Zum manuellen Starten der Aufzeichnung klicken Sie auf die Schaltfläche **Aufzeichnen** in der VuGen-Symbolleiste, vervollständigen Sie die Angaben im Dialogfeld **Aufzeichnung starten** und klicken Sie dann auf **Aufzeichnung starten**. Die freischwebende VuGen-Symbolleiste wird eingeblendet und VuGen öffnet Ihre Anwendung und ist bereit, Ihre Aktionen aufzuzeichnen.

- Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Aufzeichnung starten"" auf Seite 272.
- Weitere Informationen zu den Skriptabschnitten, in die Sie aufzeichnen können, finden Sie unter "Abschnitte des Vuser-Skripts" auf Seite 260.

Durchführen von Geschäftsprozessen mit Ihrer Anwendung

Führen Sie die gewünschten Geschäftsprozesse, die aufgezeichnet werden sollen, durch. Die freischwebende Symbolleiste gibt Ihnen die Möglichkeit, Transaktionen, Rendezvous-Punkte und Kommentare einzufügen. Darüber hinaus können Sie festlegen, in welchen Abschnitten des Skripts aufgezeichnet werden soll. Informationen zur Benutzeroberfläche finden Sie unter "Freischwebende Aufzeichnungssymbolleiste" auf Seite 276.

Klicken Sie in der unverankerten Symbolleiste auf die Schaltfläche **Anhalten** , wenn Sie mit der Aufzeichnung fertig sind.

Generieren Sie den Code neu.

Generiert das Skript nach der Aufzeichnung neu, um zu ermitteln, ob alle Schritte korrekt analysiert wurden.

Weitere Informationen finden Sie unter "Codegenerierung im Flex-Protokoll" auf der nächsten Seite.

Festlegen des Flex-Aufzeichnungsmodus

Sie können in VuGen festlegen, wie ein Skript für eine Flash Remoting-Sitzung mit Flex- und Webprotokollen generiert wird.

Beispiel

Verwenden Sie die Web HTTP-Technologie, um die **web_custom_request**-Funktionen mit den Flash Remoting-Informationen zu generieren.

```
web_url("flash",
    "URL=http://<HOST>:<PORT>/flash/",
     "Resource=0",
     "RecContentType=text/html",
     "Referer=",
```

```
     "Snapshot=t1.inf",
     "Mode=HTML",
    EXTRARES,
    "Url=movies/XMLExample.swf", "Referer=", ENDITEM,
    "Url=movies/JavaBeanExample.swf", "Referer=", ENDITEM,
    LAST);
web_link("Sample JavaBean Movie Source",
    "Text=Sample JavaBean Movie Source",
    "Snapshot=t2.inf",
    EXTRARES,
    "Url=XMLExample.swf", "Referer=", ENDITEM,
    "Url=JavaBeanExample.swf", "Referer=", ENDITEM,
    LAST);
web_custom_request("gateway",
    "URL=http://<HOST>:<PORT>/flashservices/gateway",
    "Method=POST",
     "Resource=0",
    "RecContentType=application/x-amf",
     "Referer=",
    "Snapshot=t3.inf",
     "Mode=HTML",
    "EncType=application/x-amf",
    "BodyBinary=\\x00\\x00\\x00\\x01\\x00\\x10amf_server_debug\\x01
                \\x00\\x00\\x00`\\x03\\x00\ncoldfusion\\x01\\x01\\x00
                \namfheaders\\x01\\x00\\x00\\x03amf\\x01\\x00\\x00
                \\x0Bhttpheaders\\x01\\x00\\x00\trecordset\\x01\\x01
                \\x00\\x05error\\x01\\x01\\x00\\x05trace\\x01\\x01
                \\x00\\x07m_debug\\x01\\x01\\x00\\x00\t\\x00\\x01
                \\x00/flashgateway.samples.FlashJavaBean.testDocument
                \\x00\\x02/1\\x00\\x00\\x004\n\\x00\\x00\\x00\\x01
                \\x0F\\x00\\x00\\x00*<TEST message=\"test\"><INSIDETEST /></TES
T>",
        LAST);
```

Codegenerierung im Flex-Protokoll

Benachrichtigung für die Codegenerierung

Wenn für ein Flex-, Silverlight- oder Java over HTTP-Skript ein Fehler während der Codegenerierung festgestellt wurde, gibt VuGen eine Warnung aus. Diese Warnung wird im Fehlerbereich angezeigt, wenn die Schaltfläche **Warnungen** ausgewählt wurde und der Filter **Verfügbare Kategorien definieren** auf **Alle** oder auf **Benachrichtigungen für die Codegenerierung** gesetzt ist. In der Liste mit den Warnungen werden Details zu jedem Fehler sowie die empfohlenen Maßnahmen zur Behebung des Problems angezeigt. Befolgen Sie diese Anweisungen und generieren Sie das Skript neu.

Wenn sich der Fehler auf externalisierbare Objekte in einem Flex-Skript bezieht, siehe "Externalisierbare Objekte in Flex-Skripts" auf der nächsten Seite.

Sie können den Fehlerbereich jederzeit manuell öffnen, indem Sie **Ansicht > Fehler** auswählen.

Analysieren von Antworten in Flex-Skripts

Wenn Sie ein Flex-Skript generieren, versucht LoadRunner, die Antworten für die folgenden Schritte zu analysieren:

• flex_amf_call

• flex_remoting_call

• flex_login

• flex_logout

• flex_ping

Wenn die Analyse fehlschlägt, wird das folgende Attribut dynamisch zum Schritt hinzugefügt:

IsParseResponse = No

Dadurch wird LoadRunner angewiesen, die Antworten für diesen Schritt bei der Skriptwiedergabe nicht zu analysieren. Bei jeder erneuten Generierung des Skripts versucht LoadRunner, die Analyse erneut durchzuführen, und dieser Parameter wird auf "False" gesetzt, wenn der Vorgang fehlschlägt. Bei Bedarf können Sie diese Zeile löschen oder den Wert auf "Yes" setzen, um die Analyse der Antworten für diesen Schritt bei der Wiedergabe durch LoadRunner zu erzwingen.

Darüber hinaus können Sie das Attribut manuell hinzufügen und den Wert in einem generierten Skript auf "No" setzen, selbst wenn die Analyse erfolgreich war, da dadurch die Wiedergabeleistung möglicherweise verbessert wird.

Externalisierbare Objekte in Flex-Skripts

Beim Aufzeichnen einer Flex-Anwendung werden die Informationen zwischen Client und Server normalerweise mithilfe bekannter Serialisierungsmethoden (AMF) ausgetauscht. In diesem Fall erstellt VuGen einen **flex_amf_call**, und sowohl die Anforderung als auch die Antwort werden analysiert.

Wenn ein angegebenes AMF-Objekt jedoch eine benutzerdefinierte Serialisierungsmethode (externalisierbar) verwendet, gibt VuGen automatisch eine Warnung aus. In dieser Warnung werden Details zu der Ausnahme sowie empfohlene Maßnahmen zur Behebung des Problems angezeigt.

Nachstehend finden Sie einige Beispiele für Ausnahmen, die das generierte Skript enthalten kann, wenn ein AMF-Objekt eine benutzerdefinierte Serialisierungsmethode verwendet:

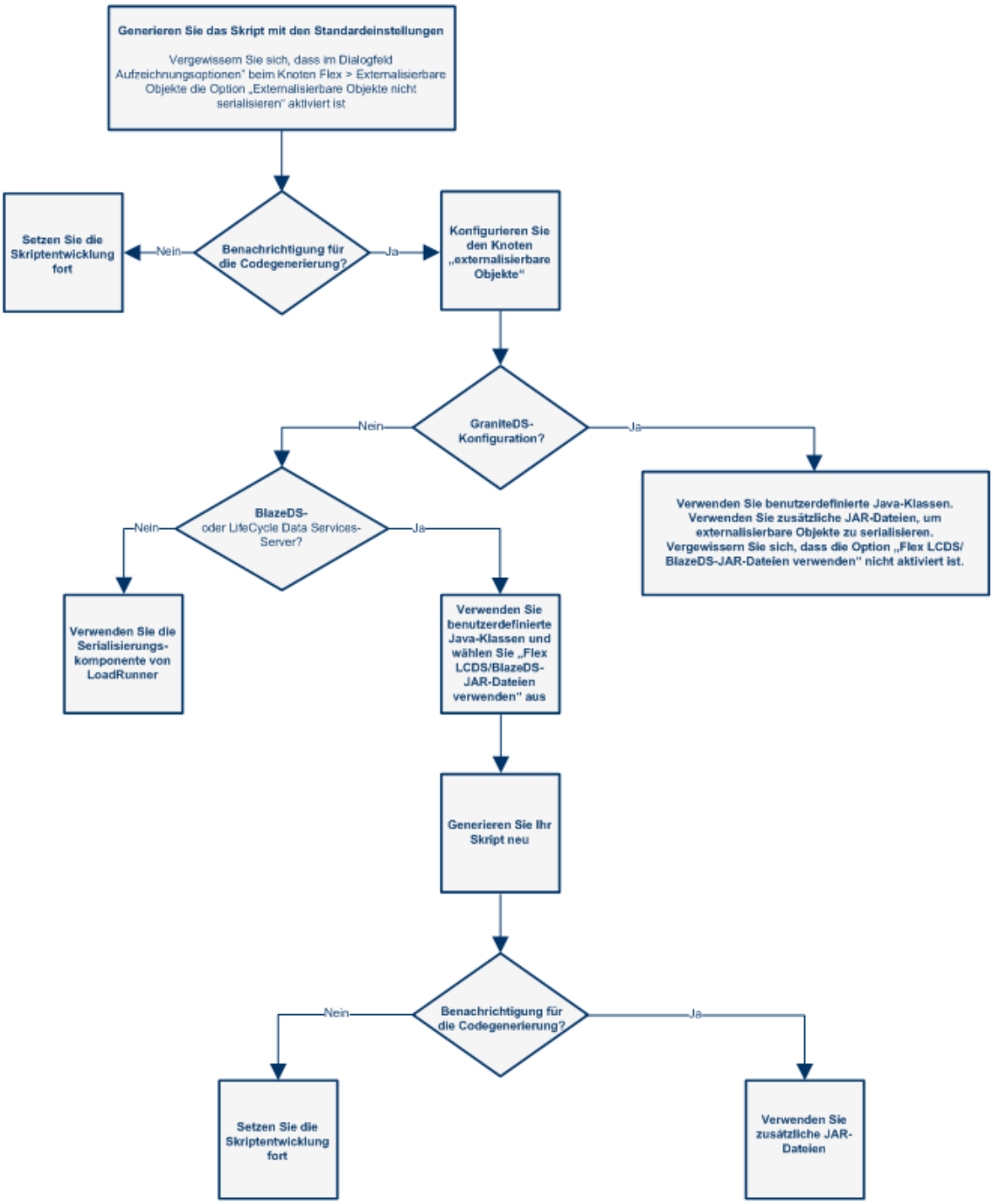
- **Anforderung und Antwort nicht analysiert.** Diese Ausnahme wird automatisch im Fehlerbereich angezeigt, wenn die Schaltfläche **Warnungen** ausgewählt und der Filter **Verfügbare Kategorien definieren** auf **Alle** oder auf **Benachrichtigungen für die Codegenerierung** gesetzt wurde. Details über die Ausnahme sowie empfohlene Maßnahmen werden aufgelistet. Weitere Informationen finden Sie unter "Bereich "Fehler"" auf Seite 116.

- **Anforderung analysiert, Antwort jedoch nicht analysiert.** VuGen generiert eine Anweisung

`IsParseResponse=No`. Zudem gibt VuGen eine Warnung aus, die automatisch im Fehlerbereich angezeigt wird, wenn die Schaltfläche **Warnungen** ausgewählt und der Filter **Verfügbare Kategorien definieren** auf **Alle** oder auf **Benachrichtigungen für die Codegenerierung** gesetzt wurde. In der Liste mit Warnungen werden Details zur Ausnahme zusammen mit empfohlenen Maßnahmen angezeigt. Weitere Informationen finden Sie unter "Bereich "Fehler"" auf Seite 116.

Weitere Informationen zum Konfigurieren des Knotens **Aufzeichnungsoptionen > Flex > Externalisierbare Objekte** finden Sie unter "Flex > Externalisierbare Objekte (Knoten) (Aufzeichnen)" auf Seite 209.

Im folgenden Flussdiagramm sind die Schritte zum Auflösen externalisierbarer Objekte in Flex-Skripts dargestellt:



Weitere Informationen zum Serialisieren externalisierbarer Objekte finden Sie unter:

- "Flex > Externalisierbare Objekte (Knoten) (Aufzeichnen)" auf Seite 209
- "Serialisieren von Flex-Skripts " auf Seite 704

Flex-Korrelationen

VuGen unterstützt die Korrelation in Flex-Skripts.

Die Unterstützung für Korrelationen gilt für folgende Flex-Schritte:

- flex_login
- flex_logout
- flex_ping
- flex_amf_call
- flex_remoting_call
- flex_rtmp_tunneled_connect
- flex_rtmp_tunneled_send

Die Flex-Korrelation enthält die Integration mit folgenden Features:

- **Korrelationsregeln**

  DSid, **jsessionid** und **RTMPT ID**

- **Design Studio**
- **Manuelle Korrelation** mithilfe der API **web_reg_save_param_xpath**.

Allgemeine Informationen finden Sie unter "Registerkarte "Korrelation" [Design Studio] - Übersicht" auf Seite 282.

Weitere Informationen über die Aufgabe finden Sie unter "Korrelieren von Skripts mit Design Studio" auf Seite 288.

Flex-Snapshots

Auf dem Flex-Protokoll basierende Vuser-Skripts verwenden den Snapshot-Bereich von VuGen.

- Eine Einführung zum Snapshot-Bereich finden Sie unter "Snapshot-Bereich - Übersicht" auf Seite 62.
- Details zum Arbeiten mit dem Snapshot-Bereich finden Sie unter "Arbeiten mit Snapshots" auf Seite 65.
- Details zur Benutzeroberfläche des Snapshot-Bereichs finden Sie unter "Snapshot-Bereich" auf Seite 109.

Zudem wurde ein neuer Snapshot entworfen, um Flex-Daten in verschiedenen Ansichten zu zeigen:

- Rohdaten

  Die vom Server empfangenen Daten, die weder formatiert noch analysiert sind.

- Antworttext

  Vom Server empfangene Datenentität

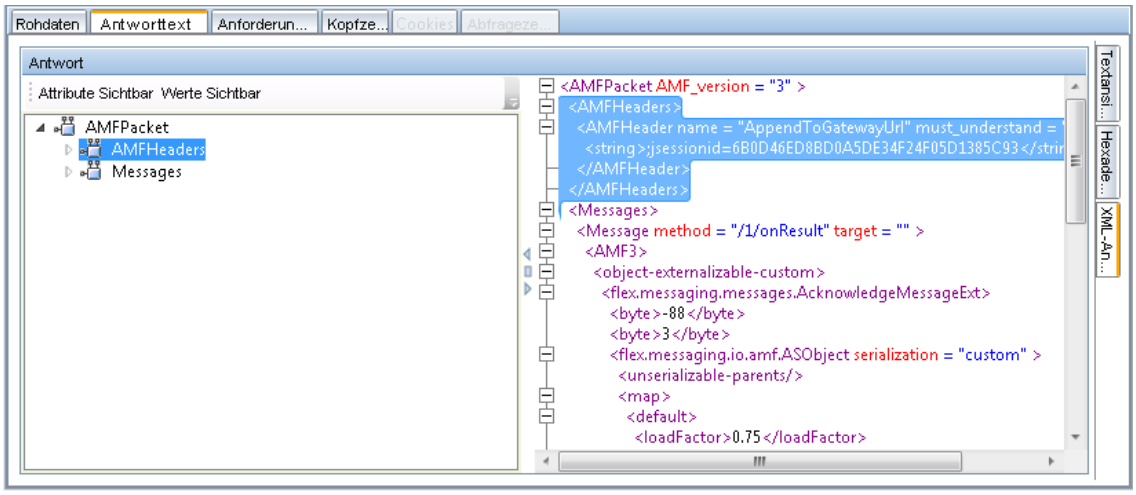
- Anforderungstext

  An den Server gesendete Datenentität

- Kopfzeilen

Sind Antwort- und Anforderungstext im XML-Format, können die Daten in folgenden Formaten angezeigt werden:

- Text
- Hex
- XML



Serialisieren von Flex-Skripts

Serialisieren mit der externen Java-Serialisierungskomponente

Sie können die Java-Klassen des Flex-Servers zur Serialisierung von AMF-Nachrichten in Ihrem Skript verwenden. Dieser Prozess wurde so vereinfacht, dass Sie die JAR-Anwendungsdateien nur einschließen müssen, wenn die AMF-Objekte eine externalisierbare Schnittstelle implementieren.

1. Wählen Sie in unter **Aufzeichnungsoptionen > Flex > Externalisierbare Objekte** die Option **Serialisieren von Objekten mit** und dann aus der Dropdownliste **Benutzerdefinierte Java-Klassen** aus.

2. Wählen Sie die entsprechenden Dateien mit den Schaltflächen **Alle Klassendateien im Ordner hinzufügen** bzw. **JAR- oder ZIP-Datei hinzufügen** aus. Fügen Sie die folgenden Dateien hinzu:

   a. Fügen Sie **für Adobe BlazeDS oder Adobe LCDS** die folgenden JAR-Dateien hinzu:

      - flex-messaging-common.jar
      - flex-messaging-core.jar

   b. Generieren Sie das Skript neu, und notieren Sie alle auftretenden Fehler. Öffnen Sie das Dialogfeld **Aufzeichnungsoptionen** über die Schaltfläche **Gen.optionen** und fügen Sie die erforderlichen JAR-Anwendungsdateien hinzu.

3. Vergewissern Sie sich, dass die hinzugefügten Dateien sowohl auf dem VuGen-Computer als auch auf allen Lastgeneratoren an demselben Speicherort vorhanden sind.

   Weitere Informationen finden Sie unter "Externalisierbare Objekte in Flex-Skripts" auf Seite 700.

Hinweise und Einschränkungen zur Java-Serialisierungskomponente

- Unterstützte JDK-Versionen: 1,6 und früher.
- Unterstützte Server: Adobe BlazeDS und Adobe Livecycle DS.
- Microsoft .NET-Klassen werden nicht unterstützt.
- Während der Codegenerierung überprüft VuGen die Funktionalität der Anforderungspuffer. Hierzu wird geprüft, ob ein Puffer mithilfe der bereitgestellten jar-Dateien gelesen und geschrieben werden kann. Verläuft die Prüfung negativ, bedeutet dies, dass die Klassen nicht mit LoadRunner kompatibel sind.

Verwenden der LoadRunner-Serialisierungskomponente

Sie können versuchen, externalisierbare Objekte mit der Serialisierungskomponente von LoadRunner zu serialisieren. Vergewissern Sie sich vorher, dass alle geöffneten Skripts gespeichert wurden, da bei dieser Option unerwartete Fehler oder ungültige Schritte auftreten könnten.

1. Speichern Sie alle in VuGen geöffneten Skripts.

2. Wählen Sie in unter **Aufzeichnungsoptionen > Flex > Externalisierbare Objekte** die Option **Serialisieren von Objekten mit** und dann aus der Referenz **LoadRunner AMF Serializer**

aus.

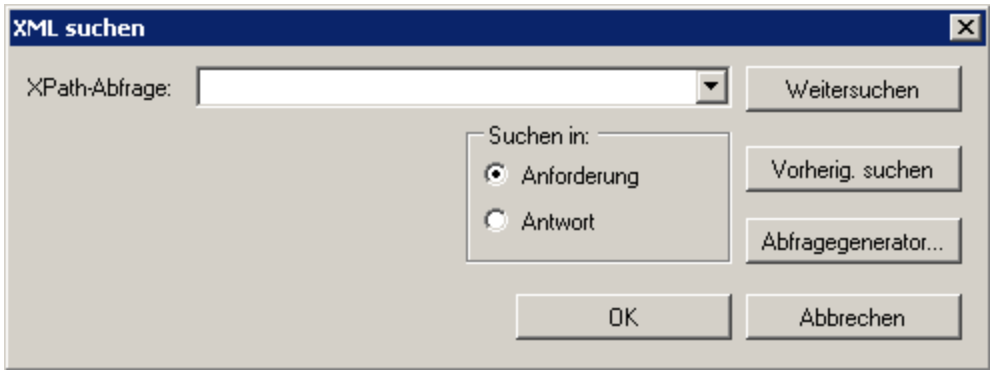
Abfragen einer XML-Struktur

VuGen enthält einen Abfragegenerator, mit dessen Hilfe Sie XML-Abfragen erstellen und ausführen können.

VuGen stellt den XML-Code innerhalb einer erweiterbaren Struktur dar. Sie können eine Abfrage für Ihr XML-Dokument ausführen und so nach einer bestimmten Namespace-URI, einem Wert oder Attribut suchen. Bei allen Abfragen ist zwischen Groß- und Kleinschreibung zu unterscheiden.

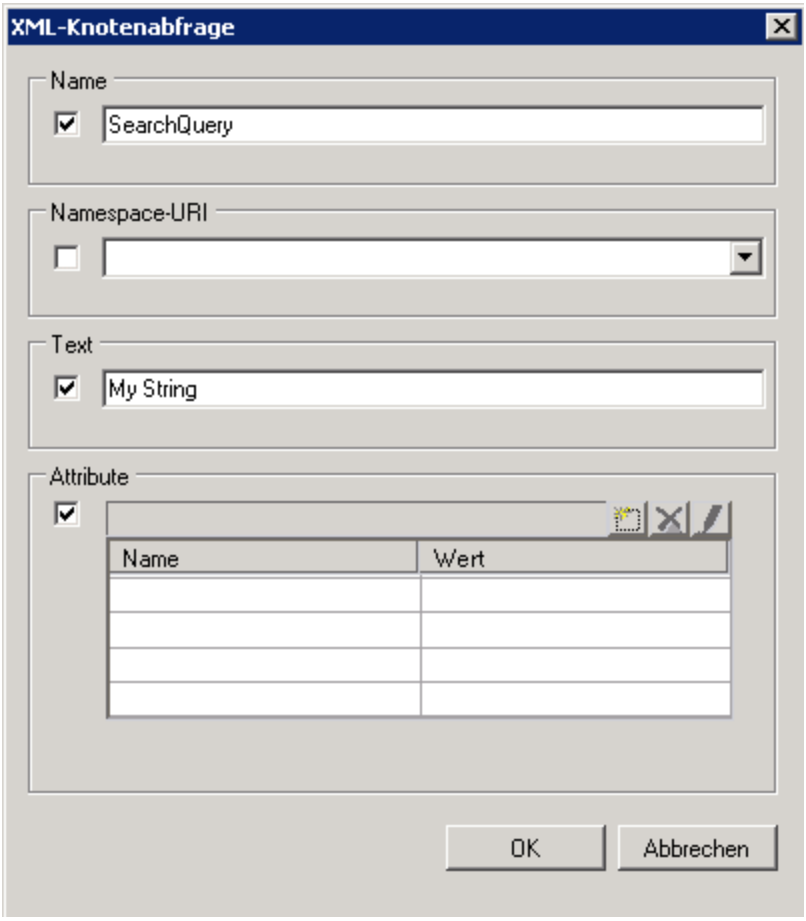
Ausführen einer Abfrage

1. Wählen Sie im Snapshot-Bereich den Knoten aus, der durchsucht werden soll. Klicken Sie auf die Schaltfläche **XML suchen**. Das Dialogfeld **XML suchen** wird geöffnet.



2. Wählen Sie **Anforderung** oder **Antwort** aus. Geben Sie unter **XPath-Abfrage** eine Abfrage ein und klicken Sie auf **OK**. Klicken Sie zum Erstellen einer Abfrage auf die Schaltfläche **Abfragegenerator**. Das Dialogfeld **XML-Knotenabfrage** wird geöffnet.

3. Wählen Sie die gewünschten Optionen für die Suche aus.



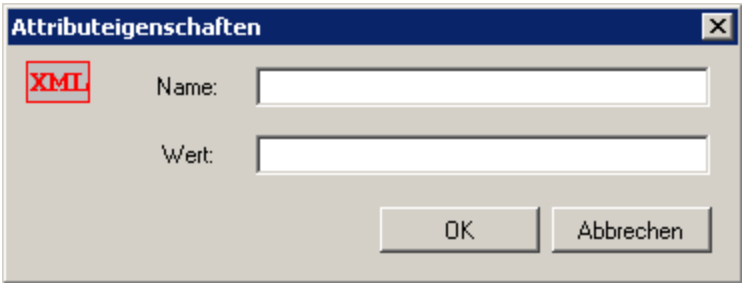
4. Aktivieren Sie den Abschnitt **Name**, um nach dem Namen eines Knotens oder Elements zu suchen.

5. Aktivieren Sie den Abschnitt **Namespace-URI**, um nach einem Namespace zu suchen.

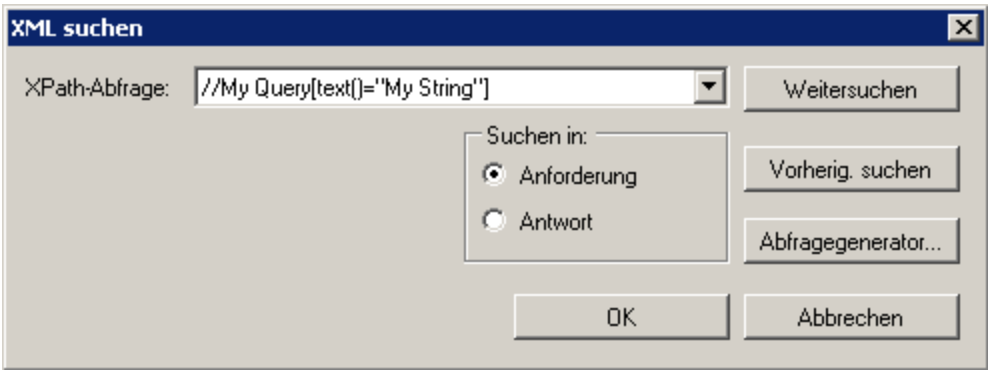
6. Aktivieren Sie den Abschnitt **Text**, um nach dem Wert des Elements zu suchen, das Sie im Feld **Name** angegeben haben.

7. Aktivieren Sie den Abschnitt **Attribute**, um nach einem Attribut zu suchen.

8. Geben Sie den Suchtext in die entsprechenden Felder ein. Klicken Sie zum Hinzufügen eines Attributs auf die Schaltfläche **Hinzufügen**. Das Dialogfeld **Attributeigenschaften** wird geöffnet. Geben Sie einen Namen und einen Wert für das Attribut ein. Klicken Sie auf **OK**.



9. Klicken Sie im Dialogfeld **XML-Knotenabfrage** auf **OK**. VuGen fügt den Text der Abfrage im Feld **XML suchen** ein.



10. Klicken Sie auf **Weitersuchen**, um die Suche zu starten.

Flex - Fehlerbehebung und Einschränkungen

In diesem Abschnitt werden die Fehlerbehebung und die für das Flex-Protokoll geltenden Einschränkungen beschrieben.

- Es ist nicht möglich, ein Flex-Skript mithilfe einer externen Java Virtual Machine (JVM), Version 1.4 oder niedriger, zu erstellen.
- Wenn Ihr Skript mehrere **flex_RTMP_tunneled_connect**-Schritte mit demselben Gateway-Parameter enthält, müssen Sie für den vorherigen **flex_RTMP_tunneled_connect**-Schritt einen Schritt zum Trennen einfügen, bevor Sie die Verbindung erneut herstellen. Beispiel:

```
Flex_rtmp_tunneled_connect("connect_step0",
"SessionId=0",
"Gateway=http://123.123.123.123:1935",
.
LAST);
.
Flex_rtmp_tunneled_disconnect("disconnect_step0",
"SessionId=0")
flex_rtmp_tunneled_connect("connect_step1",
"SessionId=1",
"Gateway=http://123.123.123.123:1935",
.
LAST);
.
Flex_rtmp_tunneled_disconnect("disconnect_step0",
"SessionId=1")
```

- Wenn ein nachfolgender **flex_rtmp_tunneled_connect**-Befehl über denselben Gateway-Parameter verfügt wie der vorherige **flex_trtmp_tunneled_connect**-Schritt, und der **flex_rtmp_tunneled_disconnect**-Schritt wird ausgelassen, dann wird das Skript unbegrenzt angehalten.

GraniteDS (Data Services)

- Wenn Sie die Datei **granite-config.xml** geändert haben, kopieren Sie diese in das Verzeichnis **<LoadRunnerInstallation>\dat**.
- Wenn Sie zwischen der Analyse unter BlazeDS und GraniteDS wechseln (unter **Aufzeichnungsoptionen > Flex > Konfiguration**), muss VuGen neu gestartet werden.
- LoadRunner kann GraniteDS- und BlazeDS/LCDS-Meldungen nicht gleichzeitig im selben Skript serialisieren.
- Flex RTMP-Messaging-Unterstützung. Alle Einschränkungen für die AMF3-Analyse gelten auch für die Übertragung von externalisierbaren Objekten über RTMP.

- **flex_rtmp_receive_stream**-Schritt. Wenn die Option zum Generieren des **flex_rtmp_receive_stream**-Schritts aktiviert ist, werden alle über die Aufzeichnungssymbolleiste hinzugefügten Transaktionen, Kommentare und Rendezvous-Punkte nach dem **flex_rtmp_receive_stream**-Schritt in das Skript eingefügt.
- Die Webdiagnose funktioniert nicht bei RTMP- und RTMPT-Schritten (selbst, wenn die Aufschlüsselung aktiviert ist).
- Es ist nicht möglich, zwei RTMPT-Schritte gleichzeitig auszuführen.

Java-Protokolle

Übersicht über die Aufzeichnung von Java-Protokollen

VuGen ermöglicht Ihnen das Aufzeichnen einer Java-Anwendung bzw. eines Java-Applets. VuGen erstellt ein reines Java-Skript, das mit Java-spezifischen Vuser-API-Funktionen erweitert wird. Nach dem Aufzeichnen können Sie das Skript unter Verwendung von JDK-Bibliotheken oder angepassten Klassen mit Java-Standardcode erweitern oder ändern.

Sobald Sie Ihr Skript vorbereitet haben, können Sie es im Einzelmodus über VuGen ausführen. Der standardmäßige Java-Compiler von Sun, **javac.exe**, prüft das Skript auf Fehler und kompiliert es. Sobald Sie das Skript auf seine fehlerfreie Ausführung geprüft haben, integrieren Sie es in ein LoadRunner -Szenario oder eine Business Process Monitor-Konfiguration.

Wenn Sie ein Skript mittels der Aufzeichnung von Code und manueller Erweiterungen erstellen, gelten alle auf Java-Vuser-Skripts anwendbare Richtlinien und Einschränkungen. Außerdem müssen alle spezifischen, im Skript verwendeten Klassen auf dem Computer vorhanden sein, der die Vuser ausführt und über den Klassenpfad in der Umgebungsvariablen **classpath** auffindbar sein. Weitere wichtige Informationen über die Funktionssyntax und die Systemkonfiguration finden Sie unter "Java-Vuser-Protokoll - Manuelles Programmieren von Skripts" auf Seite 730.

Vor dem Aufzeichnen einer CORBA-Sitzung müssen Sie prüfen, ob sich Ihre Anwendung bzw. Ihr Applet auf dem für die Aufzeichnung verwendeten Computer fehlerfrei ausführen lässt.

Überprüfen Sie die ordentliche Installation einer JDK-Version von Sun auf dem Computer, auf dem VuGen ausgeführt wird. JRE allein ist nicht ausreichend. Die Installation muss vor dem Aufzeichnen eines Skripts abgeschlossen sein. Vergewissern Sie sich, dass die Umgebungsvariablen **classpath** und **path** gemäß der JDK-Installationsanweisungen definiert wurden.

Hinweis: Wenn Sie während des Aufzeichnens ein Applet oder eine Anwendung aus VuGen laden, nimmt dies u. U. einige Sekunden mehr in Anspruch, als das Laden eines von VuGen unabhängigen Elements.

VuGen enthält eine Funktion zum Konvertieren eines für das Internet erstellten Vuser-Skripts in Java. Weitere Informationen finden Sie unter "Konvertieren eines Web (HTTP/HTML)-Vuser-Skripts in ein Java-Vuser-Skript" auf Seite 892.

Nach dem Aufzeichnen können Sie das Skript unter Verwendung von JDK-Bibliotheken oder angepassten Klassen mit Java-Standardcode erweitern oder ändern.

Sobald Sie Ihr Skript vorbereitet haben, können Sie es im Einzelmodus über VuGen ausführen. Der standardmäßige Java-Compiler von Sun, **javac.exe**, prüft das Skript auf Fehler und kompiliert es.

Die fertigen Skripts werden in Ihre Umgebung integriert: ein LoadRunner-Szenario, ein Performance Center-Lasttest oder eine Business Process Monitor-Konfiguration. Weitere Informationen finden Sie in der Function Reference oder in der *Performance Center*-Dokumentation.

Hinweis: Java 7 aktiviert standardmäßig das Java-Überprüfungstool für die Aufteilung (Java Split Verifier). Dadurch wird verhindert, dass Java in LoadRunner aufzeichnet. LoadRunner verwendet den Schlüssel **-XX:-UseSplitVerifier** während der Aufzeichnung bei gleichzeitiger Initialisierung der JVM, um das Überprüfungstool zu deaktivieren. Diese Anpassung erfordert keine Benutzereingaben.

Übersicht über Java-Vuser-Skripts

Wenn Sie eine Sitzung aufzeichnen, protokolliert VuGen alle Aufrufe an den Server und generiert ein Skript mit entsprechenden Funktionen. Diese Funktionen beschreiben alle Aktionen innerhalb der Anwendung bzw. des Applets. Ferner enthält das Skript den zusätzlichen Code, der für eine fehlerfreie Wiedergabe erforderlich ist, beispielsweise für die Einstellung von Eigenschaften und die Initialisierung des Benennungsdienstes (JNDI).

Das aufgezeichnete Skript besteht aus drei Abschnitten:

- Imports
- Code
- Variables

Das Skript beginnt mit dem Abschnitt **Imports**. Dieser Abschnitt referenziert alle Pakete, die zum Kompilieren des Skripts erforderlich sind. Der Abschnitt **Code** enthält die Aktionsklasse sowie den in den Methoden **init**, **actions** und **end** aufgezeichneten Code. Der im Anschluss an die Methode **end** erscheinende Abschnitt **Variables** enthält alle Typdeklarationen für die im Code verwendeten Variablen.

Nach dem Aufzeichnen können Sie die Funktionen in Ihrem Skript ändern oder weitere Java- bzw. LoadRunner-Funktionen hinzufügen, um das Skript zu ergänzen. Beim Ausführen der Java-Vuser als Thread müssen Sie darauf achten, dass der Java-Code, den Sie in Ihrem Skript hinzufügen, auch tatsächlich threadsicher ist. Zusätzliche Informationen zur Syntax von Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**). Darüber hinaus können Sie Ihr Skript dahingehend ändern, dass es als Teil eines anderen Pakets ausgeführt wird. Weitere Informationen finden Sie unter "Kompilieren und Ausführen eines Skripts als Teil eines Pakets" auf Seite 734.

Übersicht über RMI over IIOP

Die **IIOP**-Technologie (Internet Inter-ORB Protocol) wurde entwickelt, um die Implementierung von CORBA-Lösungen über das Internet zu ermöglichen. Mit der IIOP-Technologie können Browser und Server komplexe Objekte, wie z. B. Arrays, austauschen, während HTTP lediglich die Übertragung von Text unterstützt.

Die **RMI over IIOP**-Technologie ermöglicht einem einzelnen Client den Zugriff auf Dienste, die in der Vergangenheit nur über RMI- oder CORBA-Clients zur Verfügung standen. Es handelt sich hierbei um eine Hybrid-Technologie des JRMP-Protokolls, das zusammen mit RMI und IIOP verwendet wird, während diese Technologien wiederum zusammen mit CORBA eingesetzt werden. Mit **RMI over IIOP** können CORBA-Clients auf neue Technologien, wie z. B. **Enterprise Java Beans** (EJB) und andere J2EE-Standards, zugreifen.

VuGen bietet eine vollständige Unterstützung für die Aufzeichnung und Wiedergabe von Vusers unter Verwendung des **RMI over IIOP**-Protokolls. Je nach den Elementen, die Sie aufzeichnen, können Sie den in VuGen enthaltenen RMI-Recorder zum Erstellen eines Skripts verwenden, das einen physischen Benutzer optimal emuliert:

- **Reiner RMI-Client.** Aufzeichnen eines Clients, der das native JRMP-Protokoll für Remote-Aufrufe verwendet.
- **RMI over IIOP-Client.** Aufzeichnen einer Clientanwendung, die mit Hilfe des IIOP-Protokolls anstelle des JRMP-Protokolls (zur Gewährleistung der Kompatibilität mit CORBA-Servern) kompiliert wurde.

Corba-Aufzeichnungsoptionen

Für das Aufzeichnen einer CORBA-Sitzung müssen Sie die folgenden Aufzeichnungsoptionen definieren:

- JNDI
- DLL-Hooking verwenden, um VuGen-Unterstützung anzufügen

Anbieterklassen für CORBA-Anwendungen

Beim Ausführen von CORBA-Anwendungen mit JDK 1.2 oder höher werden u. U. JDK-interne CORBA-Klassen anstelle der spezifischen CORBA-Anbieterklassen geladen. Um einen virtuellen Computer zur Verwendung der Anbieterklassen zu zwingen, geben Sie die folgenden java.exe-Befehlszeilenparameter ein:

```
Visigenic 3.4
-Dorg.omg.CORBA.ORBClass=com.visigenic.vbroker.orb.ORB
-Dorg.omg.CORBA.ORBSingletonClass=com.visigenic.vbroker.orb.
     ORBSingleton
Visigenic 4.0
-Dorg.omg.CORBA.ORBClass=com.inprise.vbroker.orb.ORB
```

```
-Dorg.omg.CORBA.ORBSingletonClass=com.inprise.vbroker.orb.ORBSingleton
OrbixWeb 3.x
-Dorg.omg.CORBA.ORBClass=IE.Iona.OrbixWeb.CORBA.ORB
-Dorg.omg.CORBA.ORBSingletonClass=IE.Iona.OrbixWeb.CORBA.
     singletonORB
OrbixWeb 2000
-Dorg.omg.CORBA.ORBClass=com.iona.corba.art.artimpl.ORBImpl
-Dorg.omg.CORBA.ORBSingletonClass=com.iona.corba.art.artimpl.
     ORBSingleton
```

Aufzeichnen von RMI

Vor dem Aufzeichnen einer RMI-Sitzung müssen Sie prüfen, ob sich Ihre Anwendung bzw. Ihr Applet auf dem für die Aufzeichnung verwendeten Computer fehlerfrei ausführen lässt.

Bevor Sie mit der Aufzeichnung beginnen, müssen Sie die ordnungsgemäße Konfiguration Ihrer Umgebung überprüfen. Vergewissern Sie sich, dass sich die erforderlichen Klassen im Klassenpfad befinden, und dass JDK vollständig installiert wurde. Weitere Informationen zu den erforderlichen Umgebungseinstellungen finden Sie unter "Manuelles Erstellen eines Java-Skripts" auf Seite 734.

Aufzeichnen eines Jacada-Vusers

Der Jacada Interface Server bietet eine Oberflächenschicht für Mainframe-Anwendungen. Diese Ebene trennt die Benutzeroberfläche von der Anwendungslogik, um das Unternehmen von Änderungen in Standards und Technologien unabhängig zu machen. Um Benutzern das Arbeiten mit Green-Screen-Anwendungen zu ersparen, konvertiert der Jacada-Server die Umgebung in eine benutzerfreundliche Oberfläche.

VuGen zeichnet den in Jacada enthaltenen Java-Thin-Client auf. Verwenden Sie einen Vuser vom Typ Web HTTP/HTML zum Aufzeichnen der Kommunikation mit dem Jacada-Server über den HTML-Thin-Client. Weitere Informationen finden Sie unter "Webprotokolle (Allgemein)" auf Seite 898.

Vor der Wiedergabe müssen Sie die Datei **clbase.jar** vom Jacada-Server herunterladen. Alle vom Java-Vuser verwendeten Klassen müssen im Klassenpfad enthalten sein. Die Definition erfolgt entweder in der Umgebungsvariablen **classpath** des Computers oder in der Liste **Klassenpfadeinträge** im Knoten **Classpath** der Laufzeiteinstellungen.

Bei der Wiedergabe muss der Jacada-Server Bildschirme aus dem vorhandenen System zurückgeben, und zwar in einer anderen Reihenfolge als sie im aufgezeichneten Skript erscheinen. Bei der Wiedergabe kann dies eine Ausnahme verursachen. Bei Fragen zur Handhabung dieser Ausnahmen wenden Sie sich an den Support.

Verwenden von CORBA

CORBA-spezifische Skripts weisen in der Regel gut definierte Muster auf. Der erste Abschnitt enthält die Initialisierung und die Konfiguration des ORB. Der nächste Abschnitt enthält den Pfad der CORBA-Objekte. Der darauf folgende Abschnitt besteht aus Serveraufrufen von CORBA-Objekten. Der letzte Abschnitt beinhaltet eine Abschaltprozedur zum Beenden des ORB. Beachten Sie, dass das Muster keineswegs erforderlich ist und jeder dieser Abschnitte innerhalb eines Skripts mehrmals erscheinen kann.

In dem nun folgenden Segment initialisiert das Skript eine ORB-Instanz und führt eine Bindungsoperation durch, um ein CORBA-Objekt zu erhalten. Hier sehen Sie, wie VuGen alle notwendigen Klassen importiert.

```
import org.omg.CORBA.*;
import org.omg.CORBA.ORB.*;
import lrapi.lr;
public class Actions {
   // Public function: init
   public int init() throws Throwable {
   // Initialize Orb instance...
   MApplet mapplet = new MApplet("http://chaos/classes/", null);
   orb = org.omg.CORBA.ORB.init(mapplet, null);

   // Bind to server...
   grid = grid_dsi.gridHelper.bind("gridDSI", "chaos");
   return lr.PASS;
}
```

Die Funktion `org.omg.CORBA.ORB` stellt die Verbindung zum ORB her. Aus diesem Grund sollte diese Funktion nur einmal aufgerufen werden. Wenn Sie mehrere Iterationen ausführen möchten, platzieren Sie diese Funktion im Abschnitt **init**.

Im folgenden Abschnitt hat VuGen die Aktionen aufgezeichnet, die mit einem CORBA-Rasterobjekt durchgeführt wurden.

```
// Public function: action
public int action() throws Throwable {

   grid.width();
   grid.height();
   grid.set(2, 4, 10);
   grid.get(2, 4);
   return lr.PASS;
}
```

Am Ende der Sitzung hat VuGen die Abschaltung des ORB aufgezeichnet. Die im aufgezeichneten Code durchgehend verwendeten Variablen erscheinen im Anschluss an die Methode **end** und vor der mit geschweiften Klammern abgegrenzten Aktionsklasse.

```
// Public function: end
    public int end() throws Throwable {
```

```
        if (lr.get_vuser_id() == -1)
            orb.shutdown();
        return lr.PASS;
}
// Variable section
    org.omg.CORBA.ORB orb;
    grid_dsi.grid ;
}
```

Beachten Sie, dass die Anweisung zum Abschalten für dieses Produkt angepasst wurde. Mit dieser Anpassung wird verhindert, dass durch das Abschalten eines einzelnen Vusers auch alle anderen Vuser abschaltet werden.

Arbeiten mit RMI

In diesem Abschnitt werden die Elemente des Java-Vuser-Skripts beschrieben, die sich speziell auf RMI beziehen. RMI enthält keine Konstrukte, wie sie in CORBA zu finden sind, sondern verwendet stattdessen serialisierbare Java-Objekte. Der erste Abschnitt führt die Initialisierung und Konfiguration einer Naming Registry durch. Der nächste Abschnitt wird generiert, sobald Java-Objekte (Remote und Serializable) gefunden und bestimmt wurden. Der folgende Abschnitt besteht aus Serveraufrufen von Java-Objekten. In RMI gibt es im Gegensatz zu CORBA keinen Abschnitt mit spezifischen Abschaltanweisungen. Beachten Sie, dass die Objekte innerhalb des Skripts mehrmals erscheinen können.

Das folgende Segment sucht nach einer Naming Registry. Darauf folgt eine Suchoperation, um ein spezifisches Java-Objekt zu erhalten. Sobald Sie das Objekt erhalten haben, können Sie mit diesem Objekt arbeiten und Aufrufe, wie z. B. **set_sum**, **increment** und **get_sum** durchführen. Die folgenden Segmente zeigen, wie VuGen alle erforderlichen RMI-Klassen importiert.

```
Import java.rmi.*;
Import java.rmi.registry.*;
:
:
// Public function: action
public int action() throws Throwable {

    _registry = LocateRegistry.getRegistry("localhost",1099);
    counter = (Counter)_registry.lookup("Counter1");
    counter.set_sum(0);
    counter.increment();
    counter.increment();
    counter.get_sum();
    return lr.PASS;
    }
:
```

Beim Aufzeichnen von RMI-Java enthält Ihr Skript u. U. mehrere Aufrufe an **lr.deserialize**, über den alle relevanten Objekte deserialisiert werden. Die **lr.deserialize**-Aufrufe werden generiert, weil das Objekt, das an den nächsten Aufruf weitergeleitet wird, nicht mit einem der Rückgabewerte aus

früheren Aufrufen korreliert werden kann. Daher zeichnet VuGen den Zustand des Werts auf und verwendet den **lr.deserialize**-Aufruf zur Darstellung dieser Werte bei der Wiedergabe. Die Deserialisierung wird durchgeführt, bevor VuGen die Objekte als Parameter an Aufrufe weiterleitet. Weitere Informationen finden Sie unter "Korrelieren von Skripts - Java-Skripts - Serialisierung" auf Seite 305.

Verwenden von Jacada

Die Actions-Methode eines Java-Vuser-Skripts, das von Jacada verwendet wird, ist in zwei Teile aufgegliedert: Eigenschaften und Körper. Der Abschnitt mit den Eigenschaften ruft die Servereigenschaften ab. Anschließend definiert VuGen die Systemeigenschaften und stellt eine Verbindung zum Jacada-Server her.

```
 // Set system properties...
        _properties = new Properties(System.getProperties());
        _properties.put("com.ms.applet.enable.logging", "true");
        System.setProperties(_properties);

        _jacadavirtualuser = new cst.client.manager.JacadaVirtualUser();

        lr.think_time(4);
        _jacadavirtualuser.connectUsingPorts("localhost", 1100, "LOADTEST", "",
"", "");
     ...
```

Der Skriptkörper enthält die Benutzeraktionen zusammen mit den Ausnahmehandhabungsblocks für die Methoden **checkFieldValue** und **checkTableCell**.

```
     1...
/*
try {
            _jacadavirtualuser.checkFieldValue(23, "S44452BA");
        }catch(java.lang.Exception e) {
             lr.log_message(e.getMessage());
        }
*/     1...
/*
try {
    _jacadavirtualuser.checkTableCell(41, 0, 0, "");
        }catch(java.lang.Exception e) {
             lr.log_message(e.getMessage());
        }
*/     1...
```

Die Methode **checkField** enthält zwei Argumente: **field ID number** und **expected value**. Die Methode **checkTableCell** enthält vier Argumente: **table ID**, **row**, **column** und **expected value**. Wenn eine falsche Zuordnung zwischen dem erwarteten Wert und dem empfangenen Wert erfolgt, wird eine Ausnahme generiert.

Die Try-Catch-Wrapper-Blocks sind standardmäßig auskommentiert. Um sie in Ihrem Skript verwenden zu können, müssen Sie die Kommentarmarkierungen entfernen.

Zusätzlich zu dem aufgezeichneten Skript können Sie beliebige Java-Vuser-API-Funktionen hinzufügen. Eine Liste dieser Funktionen und Anweisungen zum Hinzufügen dieser Funktionen zu Ihrem Skript finden Sie unter "Java-Vuser-Protokoll - Manuelles Programmieren von Skripts" auf Seite 730.

Übersicht über benutzerdefinierte Java-Filter

In diesem Abschnitt werden die Hintergrundinformation beschrieben, die für die Erstellung von benutzerdefinierten Java-Filtern erforderlich sind. Weitere Informationen über die Aufgabe finden Sie unter "Erstellen eines benutzerdefinierten Java-Filters" auf Seite 725.

Durch das Testen Ihrer Java-Anwendung können Sie feststellen, wie der Server auf Clientanfragen reagiert. Die Lasttests zeigen, wie der Server auf die Belastung durch zahlreiche Benutzer reagiert. Mit dem in VuGen enthaltenen Java-Vuser können Sie ein Skript zum Emulieren einer Clientkommunikation mit Ihrem Server erstellen.

VuGen enthält Filterdateien, die die Hakeneigenschaften für allgemein verwendete Methoden definieren. Dazu zählen Filterdefinitionen für RMI-, CORBA-, JMS- und JACADA-Protokolle. Sie können auch benutzerdefinierte Filter definieren.

Beim Aufzeichnen einer Methode werden die Methoden, die entweder direkt oder indirekt aus einer aufgezeichneten Methode aufgerufen werden, nicht aufgezeichnet.

Um eine Methode aufzuzeichnen zu können, muss VuGen das Objekt erkennen, für das die Methode aufgerufen wird, und zwar zusammen mit den Argumenten der jeweiligen Methode. VuGen erkennt ein Objekt, wenn es von einer anderen aufgezeichneten Methode zurückgegeben wird, sofern folgende Bedingungen erfüllt sind:

- Die Konstruktionsmethode dieses Objekts nutzt den Hook-Mechanismus.
- Es handelt sich um ein elementares oder ein integriertes Objekt.
- Das Objekt unterstützt eine serialisierbare Oberfläche.

Sie können einen angepassten Filter erstellen, um unerwünschte Methoden auszuschließen. Beim Aufzeichnen einer Java-Anwendung berücksichtigt Ihr Skript möglicherweise Aufrufe von Methoden, die keine Auswirkung auf den Server haben, wie z. B. ein lokales Dienstprogramm oder die grafische Benutzeroberfläche. Diese Aufrufe sind für die Testziele normalerweise nicht relevant und sollten daher herausgefiltert werden.

Die integrierten Filter für RMI-, CORBA-, JMS- und JACADA-Protokolle wurden dahingehend definiert, dass sie nur den Serververkehr aufzeichnen, der für Ihre Testziele relevant ist. In einigen Instanzen benötigen Sie jedoch u. U. Filter zum Erfassen der Aufrufe Ihrer JAVA-Anwendung bzw. zum Ausschließen unnötiger Aufrufe. Für angepasste JAVA-Protokolle, proprietäre Ergänzungen und Erweiterungen von Standardprotokollen und die Datenabstraktion ist grundsätzlich die Definition einer angepassten Filterdefinition erforderlich.

Wir empfehlen Ihnen, sich vor dem Erstellen eines Test mit Ihrer Anwendung vertraut zu machen und die primären Klassen und Methoden festzulegen, so dass Sie wissen, welche Klassen und Methoden beim Aufzeichnen berücksichtigt werden sollen.

Sollten Sie die Klassen Ihrer Anwendung nicht kennen, ermöglicht Ihnen VuGen das Aufzeichnen eines Stacktraces zum Protokollieren aller Methoden, die von Ihrer Anwendung aufgerufen wurden. Für das Aufzeichnen mit der Stapelverfolgung müssen Sie für die Protokollebene den Wert **Detailliert** einstellen.

Benutzerdefinierte Java-Filter - Ermitteln der einzuschließenden Elemente

Beim Erstellen eines anpassten Filters empfehlen wie Ihnen, mit der Auswahl eines geeigneten integrierten Filters als Basisfilter zu beginnen. Anschließend können Sie den Filter auf verschiedenen Weisen anpassen:

- **Top-Down-Konzept.** Bei diesem Konzept wird das relevante Paket eingeschlossen, während die spezifischen Klassen, die nicht Teil der Client-Server-Aktivität sind, ausgeschlossen werden. Wir empfehlen dieses Konzept, wenn Sie mit Ihrer Anwendung vertraut sind und eine korrekt definierte Schicht identifizieren können, die alle Client-Server-Aktivitäten implementiert, ohne dabei GUI-Elemente einzubeziehen.

- **Bottom-up-Konzept.** Bei diesem Konzept verwenden Sie den Standardfilter und passen diesen durch Hinzufügen individueller Methoden und Klassen an. Verwenden Sie dieses Konzept, wenn Sie keine gut definierte Schicht identifizieren können oder mit der Handhabung Ihrer Anwendung nicht besonders vertraut sind. Fügen Sie nicht alle AUT-Pakete hinzu, und versuchen Sie, zusätzliche Komponenten einzeln zu entfernen.

Der folgende Abschnitt enthält Richtlinien zum Ein- bzw. Ausschließen von Elementen.

- Wenn Ihr Skript nach dem Einschließen einer Klasse zahlreiche Aufrufe unabhängiger Methoden enthält, müssen Sie den Filter dahingehend ändern, dass irrelevante Methoden ausgeschlossen werden.

- Sollten Sie einen Nicht-Client/Server-Aufruf in Ihrem Skript identifizieren, müssen Sie die entsprechende Methode im Filter ausschließen.

- Beim Aufzeichnen erkennt VuGen u. U. ein unbekanntes Eingabeargument, wie z. B. ein Argument mit einer bis dahin noch nie verwendeten Konstruktion. Wenn dieses Argument die Serialisierung unterstützt, wird es von VuGen durch Speichern in einem besonderen Format in einer Datei serialisiert. Bei der Wiedergabe wird das Argument von VuGen mittels Deserialisierung rekonstruiert.

- VuGen serialisiert nur Objekte, die erfolgreich in Form von Argumenten ausgeführt wurden, die nicht von dem Filter eingeschlossen wurden. Wir empfehlen Ihnen das Einschließen dieses Objekts im Filter, um seine Konstruktion und Aktivität protokollieren zu können, anstatt das

Objekt in seiner serialisierten Form zu verwenden. Sie können serialisierte Objekte im Skript identifizieren, indem Sie in Ihrem Skript nach Aufrufen der Methode **lr.deserialize()** suchen. Weitere Informationen finden Sie unter "Korrelieren von Skripts - Java-Skripts - Serialisierung" auf Seite 305.

- Schließen Sie alle Aktivitäten aus, die GUI-Elemente betreffen.
- Fügen Sie Klassen für Dienstprogramme hinzu, die für das Kompilieren des Skripts erforderlich sein können.

Aufzeichnen eines Java-Vuser-Skripts

In dieser Aufgabe wird beschrieben, wie ein Java-Vuser-Skript aufgezeichnet wird.

Java-Protokolke können nur 32-Bit-Anwendungen aufzeichnen. Daher muss die ausführbare Datei für 32-Bit verwendet werden.

Der Benutzer kann zudem direkt eine aufzuzeichnende BAT-Datei angeben.

1. ### Voraussetzungen

   Überprüfen Sie, ob eine JDK-Version von Sun ordnungsgemäß auf dem Computer installiert ist, auf dem der Vuser ausgeführt wird. JRE allein ist nicht ausreichend. Vergewissern Sie sich, dass die Umgebungsvariablen **classpath** und **path** gemäß der JDK-Installationsanweisungen definiert wurden. Vor der Wiedergabe eines Vuser-Skripts müssen Sie sicherstellen, dass Ihre Umgebung ordnungsgemäß für die JDK- und relevante Java-Klassen konfiguriert wurde.

2. ### Erstellen eines neuen Skripts für das Java Record Replay-Protokoll

   Wählen Sie **Datei** > **Neu** aus und wählen Sie **JAVA Record Replay** in der **Java**-Kategorie aus.

3. ### Vervollständigen der Angaben im Dialogfeld "Aufzeichnung starten"

   Geben Sie die Details zu Ihrer Anwendung im Dialogfeld **Aufzeichnung starten** ein. Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Aufzeichnung starten"" auf Seite 272.

   Hinweis: Java-Protokolke können nur 32-Bit-Anwendungen aufzeichnen. Stellen Sie bei Angabe einer Anwendung daher sicher, dass es sich um die 32-Bit-Version handelt. Sie können auch eine Batchdatei (BAT) als aufzuzeichnende Anwendung angeben.

4. ### Festlegen der Aufzeichnungsoptionen

   Klicken Sie im Dialogfeld **Aufzeichnung starten** auf **Optionen**, um das Dialogfeld **Aufzeichnungsoptionen** zu öffnen. Im Knoten **Recorder-Optionen** wird über das Feld **Aufgezeichnetes Protokoll** das aufzuzeichnende Hauptprotokoll konfiguriert. Wenn Sie mehr

als ein Java-Protokoll aufzeichnen, geben Sie die zusätzlichen Protokolle im Feld mit **Erweiterungsliste** ein.

5. **Aufzeichnung starten**

   Zeichnen Sie die Anwendung oder den Batch durch Ausführen der zu testenden Aktionen auf.

Aufzeichnen von Java-Skripts bei Verwendung von Windows XP und Windows 2000 Server

Beim Aufzeichnen auf Windows XP- und Windows 2000-Servern ist das Java Plug-In u. U. nicht mit dem VuGen-Recorder kompatibel. Um eine fehlerfreie Funktionsweise zu gewährleisten, führen Sie im Anschluss an die Installation des Java Plug-ins und vor dem Aufzeichnen eines Skripts folgende Prozedur durch.

Konfigurieren des Computers für das Aufzeichnen von CORBA- oder RMI-Sitzungen

1. Öffnen Sie das Java Plug-in über die Systemsteuerung. Wählen Sie **Start > Settings > Control Panel** aus und öffnen Sie die **Java Plug-in**-Komponente. Die Registerkarte **Basic** wird geöffnet.

2. Löschen Sie das Kontrollkästchen **Enable Java Plug-In** und klicken Sie auf **Apply**. Markieren Sie erneut das Kontrollkästchen **Enable Java Plug-In** und klicken Sie auf **Apply**.

3. Wechseln Sie zur Registerkarte **Browser**. Deaktivieren Sie das Kontrollkästchen **Microsoft Internet Explorer**, und klicken Sie auf **Apply**. Markieren Sie erneut Sie das Kontrollkästchen **Microsoft Internet Explorer** und klicken Sie auf **Apply**.

Ausführen eines Skripts als Teil eines Pakets

Dieser Abschnitt ist für Skripts vom Typ Jacada nicht relevant.

Beim Erstellen oder Aufzeichnen eines Java-Skripts müssen Sie u. U. Methoden aus Klassen verwenden, in denen die Methode oder Klasse geschützt ist. Bei dem Versuch, ein solches Skript zu kompilieren, erhalten Sie einen Kompilationsfehler mit dem Hinweis, dass der Zugriff auf die Methoden nicht möglich ist.

Um die geschützten Methoden zu verwenden, fügen Sie den Vuser zu dem Paket mit den erforderlichen Methoden hinzu. Fügen Sie am Anfang Ihres Skripts folgende Zeile ein:

```
package a.b.c;
```

wobei **a.b.c** eine Ordnerhierarchie darstellt. VuGen erstellt die Ordnerhierarchie a/b/c im Benutzerverzeichnis und kompiliert dort die Datei **Actions.java**, die somit Teil des Pakets ist. Beachten Sie, dass die Anweisung **package** nicht aufgezeichnet wird und Sie diese Anweisung manuell einfügen müssen.

Manuelles Einfügen von Java-Methoden

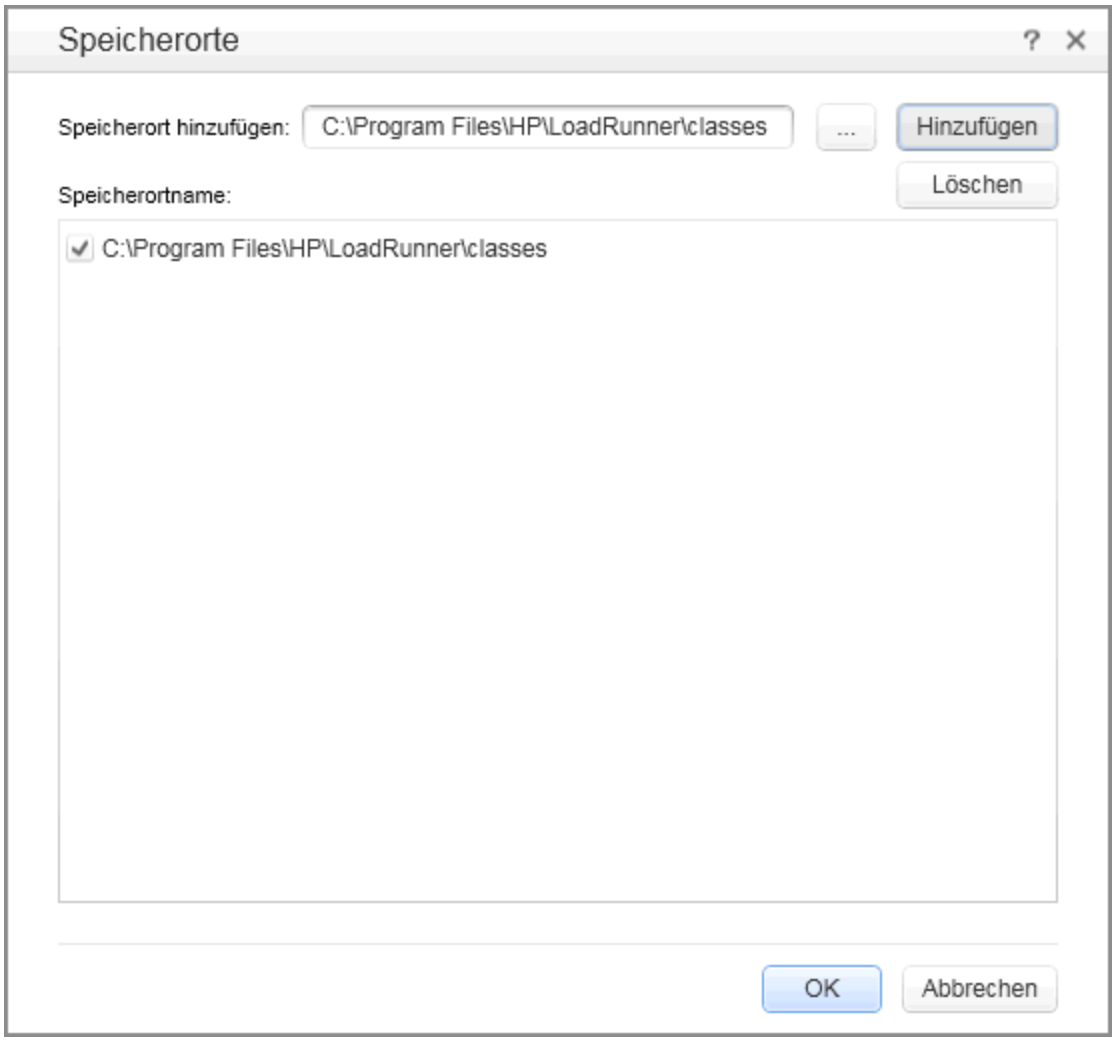
Verwenden Sie den Java Function-Navigator zum Anzeigen und Hinzufügen von Java-Funktionen in Ihrem Skript. Sie können die Funktionsgenerierungseinstellungen mittels entsprechender Änderungen in der Konfigurationsdatei anpassen. Weitere Informationen finden Sie unter "Allgemein > Skript (Knoten)" auf Seite 196.

Einfügen von Java-Funktionen

1. Klicken Sie in Ihrem Skript auf den Punkt, an dem Sie eine Funktion einfügen möchten.

2. Wählen Sie **Einfügen > Java-Funktion einfügen** aus. Das Dialogfeld **Java-Funktion einfügen** wird geöffnet. Im unteren Teil des Dialogfeld wird eine Beschreibug des Java-Objekts angezeigt.

3. Klicken Sie auf **Speicherorte**. Das Dialogfeld **Speicherorte** wird geöffnet. VuGen führt standardmäßig die Pfade in einer Liste auf, die in der Umgebungsvariablen CLASSPATH definiert sind.



4. Klicken Sie auf **Durchsuchen**, um einen weiteren Pfad oder ein Archiv in der Liste hinzuzufügen. Um einen Pfad hinzuzufügen, wählen Sie **Durchsuchen > <Ordner>** aus. Um ein Archiv (**jar** oder **zip**) hinzuzufügen, wählen Sie **Durchsuchen > <Datei>** aus. Wenn Sie einen Ordner oder eine Datei auswählen, werden diese von VuGen im Feld **Speicherort hinzufügen** eingefügt.

5. Klicken Sie auf **Hinzufügen**, um das Element in der Liste hinzuzufügen.

6. Wiederholen Sie die Schritte 4 und 5 für jeden Pfad bzw. jedes Archiv, das Sie hinzufügen möchten.

7. Markieren Sie die Kontrollkästchen links neben den Elementen in der Liste oder heben Sie die Auswahl auf. Wenn ein Element markiert ist, werden die Mitglieder im Java Class-Navigator aufgeführt.

8. Klicken Sie auf **OK**, um das Dialogfeld **Speicherorte** zu schließen und zeigen Sie die verfügbaren Pakete an.

9. Klicken Sie auf die Plus- und Minuszeichen links neben den Elementen im Navigator, um die Strukturen zu erweitern oder auszublenden.

10. Wählen Sie ein Objekt und klicken Sie auf **Einfügen**. VuGen platziert das Objekt an der Cursorposition. Um alle Methoden einer Klasse in Ihrem Skript einzufügen, wählen Sie die Klasse und klicken auf **Einfügen**.

11. Wiederholen Sie den vorherigen Schritt für alle gewünschten Methoden oder Klassen.

12. Ändern Sie die Parameter der Methoden. Wenn die Einstellung zur Skriptgenerierung **DefaultValues** den Wert **true** aufweist, können sie die von VuGen eingefügten Standardwerte verwenden. Wenn für die Einstellung **DefaultValues** der Wert **false** definiert wird, müssen Sie Parameter für alle Methoden hinzufügen, die Sie in Ihr Skript einfügen.

    Zusätzlich können Sie beliebige Rückgabewerte ändern. Wenn Ihr Skript beispielsweise die Anweisung `"(String)=LavaVersion.getVersionId();"` generiert, ersetzen Sie (String) mit einer Variablen vom Typ string.

13. Fügen Sie alle notwendigen Anweisungen ein, wie z. B. Importe oder Vuser-API-Java-Funktionen (beschrieben unter "Java-Vuser-Protokoll - Manuelles Programmieren von Skripts" auf Seite 730).

14. Beschreiben Sie das Skript und führen Sie es über VuGen aus.

Manuelles Konfigurieren von Skriptgenerierungseinstellungen

Sie können die Art und Weise anpassen, wie der Navigator Methoden zu Ihrem Skript hinzufügt.

Um die Konfigurationseinstellungen anzuzeigen, öffnen Sie im dat-Ordner von VuGen die Datei **jquery.ini**.

```
[Display]
FullClassName=False
[Insert]
AutoTransaction=False
DefaultValues=True
CleanClassPaste=False
```

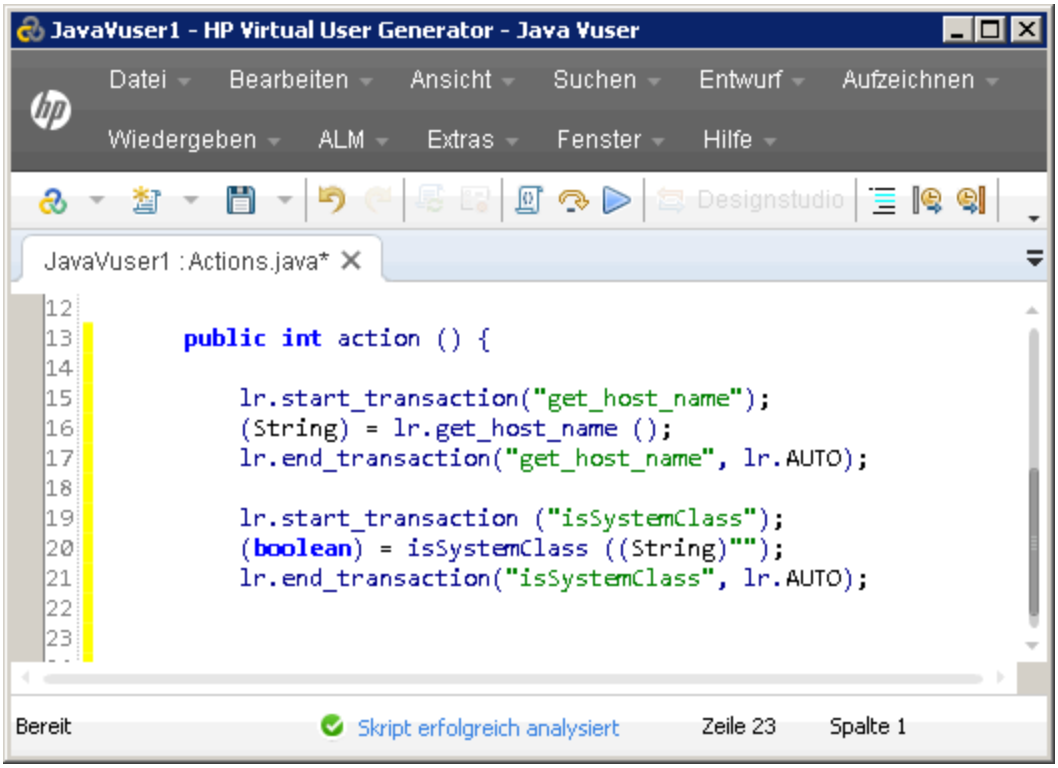
Klassennamepfad

Die Option **FullClassName** zeigt im Java Function-Navigator ein komplettes Paket mitsamt der Klassennamen an. Diese Option hat keine Auswirkung auf die im Skript hinzugefügten Funktionen, sondern beeinflusst lediglich die Anzeige der Klassen im Navigator. Standardmäßig wird diese Option auf **false** gesetzt. Wenn Ihr Paket zahlreiche Klassen enthält und Sie das Paket mitsamt der Klassennamen nicht gleichzeitig anzeigen können, sollten Sie diese Option aktivieren.

FullClassName aktiviert	FullClassName deaktiviert

Automatische Transaktionen

Die Einstellung **AutoTransaction** erstellt eine Vuser-Transaktion für alle Methoden. Wenn Sie diese Option aktivieren, schließt VuGen automatisch alle Java-Methoden mit **lr.start_transaction**- und **lr.end_transaction**-Funktionen ein. Auf diese Weise können Sie die Leistung jeder einzelnen Methode verfolgen. Diese Option ist standardmäßig deaktiviert.



Standardparameterwerte

Die Einstellung **DefaultValues** bezieht sich auf die Standardwerte aller Methoden, die Sie in Ihr Skript einfügen. Diese Option ist standardmäßig aktiviert und fügt eine Null für alle Objekte ein. Wenn Sie diese Option deaktivieren, müssen Sie die Parameterwerte für alle Funktionen im Skript manuell einfügen. Die folgende Tabelle zeigt die aktivierte und deaktivierte DefaultValues-Option.

DefaultValues aktiviert	DefaultValues deaktiviert

`lr.message((String)"");`
`lr.think_time((int)0);`
`lr.enable_redirection((boolean)false);`
`lr.save_data((byte[])null, (String)"");`
	`lr.message((String));`
	`lr.think_time((int));`
	`lr.enable_redirection((boolean));`
	`lr.save_data((byte[]), (String));`

Einfügen von Klassen

Die **CleanClassPaste**-Einstellung fügt eine Klasse ein, die eine ordentliche Kompilierung gewährleistet: mit einer vom Konstruktor zurückgegebenen Instanz, mit Standardwerten als Parameter und ohne die Notwendigkeit eines Imports von Anweisungen. Wenn Sie diese Option verwenden, werden Sie Ihr Skript höchstwahrscheinlich ohne weitere Änderungen ausführen können. Wenn Sie diese Option deaktivieren (Standard), müssen Sie die Parameter u. U. manuell definieren und die Importanweisungen einbeziehen. Beachten Sie, dass diese Einstellung nur gültig ist, wenn Sie eine ganze Klasse in Ihr Skript einfügen, jedoch nicht beim Einfügen einer einzelnen Methode.

Der folgende Abschnitt zeigt, wie die toString-Methode bei aktivierter CleanClassPaste-Option im Skript eingefügt wird.

```
_class.toString();
     // Returns: java.lang.String
```

Bei einer deaktivierten CleanClassPaste-Option wird die gleiche Methode folgendermaßen eingefügt:

```
(String) = toString();
```

Der folgende Abschnitt zeigt, wie die Konstruktormethode **NumInserter** bei aktivierter CleanClassPaste-Option im Skript eingefügt wird.

```
utils.NumInserter _numinserter = new utils.NumInserter
                ((java.lang.String)"", (java.lang.String)"", (java.lang.String)"
"...);
// Returns: void
```

Mit einer deaktivierten CleanClassPaste-Option wird die gleiche Methode folgendermaßen eingefügt:

```
new utils.NumInserter((String)"", (String)"", (String)"",...);
```

Erstellen eines benutzerdefinierten Java-Filters

Im Rahmen dieser Aufgabe wird beschrieben, wie ein benutzerdefinierter Java-Filter erstellt wird. Hintergrundinformationen finden Sie unter "Übersicht über benutzerdefinierte Java-Filter" auf Seite 717.

Weitere Informationen zur Struktur der Hook-Datei finden Sie unter "Struktur der Hook-Datei" auf Seite 727.

Bei der Vorbereitung eines Skripts müssen Sie den Filter möglicherweise wiederholt anpassen, um ihn optimal einzustellen. Ein optimaler Filter zeichnet die relevanten Methoden auf, ohne eine allzu große Zahl irrelevanter Aufrufe im Skript einzufügen.

Hinweis: Wenn Sie manuellen Code, wie z. B. Kontrollstrukturen oder Meldungsanweisungen, in Ihr Skript einfügen möchten, dürfen Sie das erst nach dem Einrichten eines funktionalen Skripts tun, das in VuGen ausgeführt wird. Das erklärt sich folgendermaßen: Wenn Sie ein Skript nach dem Ändern der Filter erneut aufzeichnen, werden alle manuell vorgenommenen

Änderungen überschrieben.

Definieren eines benutzerdefinierten Java-Filters

1. Erstellen Sie einen neuen Filter anhand eines integrierten Filters und ändern Sie dazu die Datei **user.hooks**, die im Ordner **classes** des Produkts gespeichert ist. Weitere Informationen zur Datei **user.hook** finden Sie unter "Struktur der Hook-Datei" auf der nächsten Seite.

2. Öffnen Sie das Dialogfeld **Aufzeichnungsoptionen** (Strg + F7) und wählen Sie den Knoten **Protokolloptionen** aus. Wählen Sie für die Protokollebene den Wert **Detailliert** aus.

3. Zeichnen Sie eine Anwendung auf. Klicken Sie zum Starten auf **Aufzeichnen** (Strg + R) und zum Beenden auf **Aufzeichnung beenden** (Strg + F5).

4. Zeigen Sie die Schritte des Skripts an. Wenn Sie die Geschäftslogik anhand der Schritte feststellen und die Korrelation übernehmen können, müssen Sie möglicherweise keine angepassten Filter erstellen. Wenn das Skript sehr lang oder schwer zu aktualisieren und zu korrelieren ist, müssen Sie den Filter des Skripts anpassen.

5. Versuchen Sie, eine Methode auf einer hohen Ebene in dem Aufruf zu identifizieren, der einen oder mehrere Clientserveraufrufe erfasst oder umfasst. Dazu können Sie die AUT-Quelldateien (sofern verfügbar) öffnen oder ein Stacktrace des Skripts anzeigen.

6. Stellen Sie den Filter so ein, dass alle relevanten Methoden eingeschlossen werden. Weitere Informationen finden Sie unter "Benutzerdefinierte Java-Filter - Ermitteln der einzuschließenden Elemente" auf Seite 718.

7. Zeichnen Sie die Anwendung erneut auf. Nach einer Änderung des Filters sollte die Anwendung grundsätzlich neu aufgezeichnet werden.

8. Wiederholen Sie die Schritte 4 bis 7, bis Sie ein einfaches Skript erhalten, das sich verwalten und korrelieren lässt.

9. Korrelieren Sie das Skript. Um die fehlerfreie Ausführung Ihres Tests zu gewährleisten, müssen Sie eine Korrelation einfügen, um einen Wert zu erfassen und diesen zu einem späteren Zeitpunkt im Skript verwenden zu können. Weitere Informationen über integrierte Korrelationsmechanismen finden Sie unter "Korrelieren von Skripts - Java-Skripts - Serialisierung" auf Seite 305.

Hinweis: Nehmen Sie keine Änderungen in den anderen .Hook-Dateien vor, da es dadurch zu einer Beschädigung des VuGen-Recorders kommen kann.

Das Hinzufügen angepasster Hooks im standardmäßig verwendeten Recorder ist eine komplizierte Aufgabe und sollte gründlich bedacht werden, da dieser Aufzeichnungsmechanismus sowohl funktions- als auch leistungsspezifische Auswirkungen hat.

Fehlerhafte Hook-Definitionen führen zu fehlerhaften Skripts, zu Verzögerungen beim Aufzeichnen und zum Einfrieren der Anwendung.

Struktur der Hook-Datei

Im folgenden Abschnitt wird die Struktur einer typischen Hook-Datei (.hooks) beschrieben:

```
[Hook-Name]
class      = MyPackage.MyClass
method     = MyMethod
signature  = ()V
ignore_cl =
ignore_mtd =
ignore_tree =
cb_class   = mercury.ProtocolSupport
cb_mtd =
general_cb = true
deep_mode = soft | hard
make_methods_public = true | false
lock = true | false
```

Die Hook-Dateien sind als .ini-Dateien strukturiert, wobei jeder Abschnitt einer Hook-Definition entspricht. In einigen Einträgen werden reguläre Ausdrücke unterstützt. Einträge, die reguläre Ausdrücke verwenden, müssen mit einem '!' beginnen.

Hook-Name

Gibt den Namen dieses Abschnitts in der Hook-Datei an. Der Hook-Name muss in allen Hook-Dateien eindeutig sein. Es ist immer ratsam, einen vollständig qualifizierte Klassennamen und Methoden anzugeben. Beispiel:

[javax.jms.Queue.getQueueName]

Class

Ein vollständig qualifizierter Klassenname. Sie können einen regulären Ausdruck zum Einschließen mehrerer Klassen aus einem Paket, zum Einschließen eines ganzen Pakets, mehrerer Pakete oder einer beliebigen Klasse verwenden, die mit einem Namen übereinstimmt. Beispiel:

Class = !javax\.jms\.*

Methode

Der Name einer einzuschließenden Methode. Sie können reguläre Ausdrücke verwenden, um mehr als eine Methode aus einer Klasse einzuschließen. Beispiel:

Method = getQueueName

Signature

Die Java-interne Standardsignatur der Methode. Um die Signatur einer Methode festzulegen, führen Sie den Befehl `javap -s Klassenname` aus, wobei `Klassenname` dem vollständig qualifizierten

Namen der Klasse entspricht. Sie können reguläre Ausdrücke verwenden, um mehrere Methoden mit dem gleichen Namen, aber unterschiedlichen Argumenten einzuschließen. Beispiel:

Signature = !.*

ignore_cl

Eine spezifische Klasse, die in den Klassen ignoriert werden muss, die diesem Haken zugeordnet sind. Dabei kann es sich um eine kommagetrennte Liste mit Klassennamen handeln. Jedes Element in der Liste kann einen regulären Ausdruck enthalten. Wenn ein Element in der Liste einen regulären Ausdruck enthält, müssen Sie dem Klassennamen ein '!' voranstellen. Beispiel:

Ignore_cl = !com.hp.jms.Queue,!com\.hp\..*

ignore_mtd

Eine spezifische Methode, die ignoriert werden soll. Wenn eine geladene Klassenmethode mit dieser Hook-Definition übereinstimmt, wird die entsprechende Methode nicht eingehakt. Bei dem Methodenname muss es sich um den einfachen Methodenname handelt, gefolgt von der Signatur (wie oben erklärt). Um mehrere Methoden zu ignorieren, müssen Sie diese in einer kommagetrennten Liste aufführen. Wenn Sie einen regulären Ausdruck verwenden möchten, müssen Sie dem Methodennamen ein '!' voranstellen. Beispiel:

Ignore_cl = open, close

ignore_tree

Eine spezifische Struktur, die ignoriert werden soll. Wenn der Klassenname mit dem Ausdruck für die ignorierte Struktur übereinstimmt, werden alle von dieser Struktur erbenden Klassen nicht eingehakt, wenn eine Übereinstimmung mit dieser Hakendefinition gegeben ist. Um mehrere Strukturen zu ignorieren, müssen Sie diese in einer kommagetrennten Liste aufführen. Wenn Sie einen regulären Ausdruck verwenden möchten, müssen Sie dem Klassennamen ein '!' voranstellen. Diese Option ist nur für Haken relevant, die als tief definiert sind.

cb_class

Die Callback-Klasse, die einen Aufruf von der eingehakten Methode erhält. Für diese Klasse sollte grundsätzlich der Wert **mercury.ProtocolSupport** festgelegt werden.

cb_mtd

Eine Methode der der Callback-Klasse, die einen Aufruf von der eingehakten Methode erhält. Wenn Sie diese Methode nicht vorgegeben, gilt standardmäßig die Methode **general_rec_func**. In den Fällen, in denen Sie nur die untergeordnete Aufrufstruktur sperren müssen, verwenden Sie stattdessen **general_func**.

general_cb

Die allgemeine Callback-Methode. Für diesen Wert sollte grundsätzlich **true** gelten.

Deep_mode

Der Deep-Modus bezieht sich auf Klassen und Schnittstellen, die die Klasse bzw. Schnittstelle erben bzw. implementieren, für die der Haken aufgeführt ist. Das Einhaken der ererbten Klassen erfolgt unter Berücksichtigung des Hakentyps: **Hard**, **Soft** oder **Off**.

- **Hard.** Hakt die aktuelle Klasse und alle von dieser Klasse erbenden Klassen ein. Wenn reguläre Ausdrücke existieren, werden diese allen Klassen zugeordnet, die von der Klasse in der Hakendefinition erben. Die Schnittstellenvererbung wird ebenso gehandhabt wie die Klassenvererbung.
- **Soft.** Hakt die aktuelle Klasse und alle von dieser Klasse erbenden Klassen ein, allerdings nur, wenn die Methoden in der erbenden Klasse überschrieben werden. Wenn der Haken eine Schnittstelle auflistet und eine Klasse diese Schnittstelle implementiert, werden die entsprechenden Methoden eingehakt. Wenn sie in den Klassen existieren, die direkt von dieser Klasse erben, werden sie ebenfalls eingehakt. Wenn der Haken allerdings eine Schnittstelle aufführt und eine Klasse eine zweite Schnittstelle implementiert, die von dieser ersten Schnittstelle erbt, wird die Klasse nicht eingehakt.

Hinweis: Reguläre Ausdrücke werden nicht vererbt, sondern in Methoden konvertiert.

- **Off.** Nur die in der Hook-Definition aufgeführte Klasse und die direkt erbende Klasse wird eingehakt. Wenn der Haken eine Schnittstelle aufführt, werden die Klassen, die diese Schnittstelle direkt implementieren, eingehakt.

make_methods_public:

Alle Methoden, die mit dieser Hakendefinition übereinstimmen, werden in die Methode **public** konvertiert. Dies ist beim Anpassen von Haken oder zum Sperren einer untergeordneten Aufrufstruktur von einer nicht öffentlichen Methode hilfreich.

Beachten Sie, dass dies nur beim Aufzeichnen gilt. Bei der Wiedergabe verwendet die Methode die ursprünglichen Zugriffskennzeichen. Bei der Verwendung nicht öffentlicher Methoden wird `java.lang.VerifyError` ausgelöst.

Lock

Wenn Sie den Wert **true** einstellen, wird die untergeordnete Struktur gesperrt, so dass keine Methode aus der ursprünglichen Methode aufgerufen werden kann.

Wenn Sie den Wert **false** einstellen, wird die untergeordnete Struktur freigegeben, es werden alle aus der aktuellen Methode stammenden Methoden aufgezeichnet, sofern die aktuelle Methode eingehakt ist, und es wird ein Callback durchgeführt.

Referenzliste für Java-Symbole

Die folgende Tabelle beschreibt die Symbole, die die verschiedenen Java-Objekte darstellen:

Symbol	Element	Beispiel
	Paket	java.util
	Klasse	public class Hashtable extends java.util.Dictionary implements java.lang.Cloneable, java.io.Serializable

Symbol	Element	Beispiel
	Schnittstellenklasse (graues Symbol)	public interface Enumeration
	Methode	public synchronized java.util.Enumeration keys ()
	Statische Methode (gelbes Symbol)	public static synchronized java.util.TimeZone getTimeZone
	Konstruktormethode	public void Hashtable ()

Java-Vuser-Protokoll - Manuelles Programmieren von Skripts

Übersicht über das manuelle Programmieren von Java-Skripts

Verwenden Sie zum Vorbereiten von Vuser-Skripts mithilfe von Java-Code den Vuser-Typ **Java**. Dieser Vuser-Typ unterstützt Java auf Protokollebene. Das Vuser-Skript wird von einem Java-Compiler kompiliert und unterstützt alle Java-Standardkonventionen. So können Sie beispielsweise einen Kommentar einfügen, indem Sie dem Text zwei Schrägstriche ("//") voranstellen.

Der erste Schritt beim Erstellen eines Java-kompatiblen Vuser-Skripts besteht im Anlegen einer neuen Vuser-Skriptvorlage vom Typ **Java Vuser**. Danach können Sie den gewünschten Java-Code direkt in der Skriptvorlage eingeben oder dort einfügen. Sie können das Skript durch Java-Vuser-Funktionen erweitern und die Argumente parametrieren, um bei Iterationen unterschiedliche Werte verwenden zu können.

Das Java-Vuser-Skript wird als skalierbare Multithreadanwendung ausgeführt. Bei Verwendung von benutzerdefinierten Klassen in Ihrem Skript sollten Sie sicherstellen, dass diese threadsicher sind. Code, der nicht threadsicher ist, kann zu ungenauen Ergebnissen führen. Führen Sie bei Code, der nicht threadsicher ist, die Java-Vuser als Prozesse aus. Hierdurch wird eine separate JVM (Java Virtual Machine) für jeden Prozess ausgeführt. Dies hat jedoch ein weniger skalierbares Skript zur Folge.

Sobald Sie Ihr Skript vorbereitet haben, können Sie es im Einzelmodus über VuGen ausführen. Ein Java-Compiler (javac) prüft das Skript auf Fehler und kompiliert es.

Integrieren Sie das Skript nach der Erstellung in Ihre Umgebung: ein LoadRunner-Szenario, ein Performance Center-Lasttest oder eine Business Process Monitor-Konfiguration.

Tipps zur Programmierung des Java-Protokolls

Beim Programmieren von Java-Vuser-Skripts können Sie entweder vorgefertigte Codesegmente einfügen oder vorgefertigte Klassen importieren, um deren Methoden aufzurufen. Wenn Vuser aus Skalierungsgründen unter dem Controller als Threads ausgeführt werden müssen, gilt es sicherzustellen, dass der gesamte importierte Code threadsicher ist.

Ob Code wirklich threadsicher ist oder nicht, lässt sich oft nur schwer feststellen. Ein Java-Vuser wird bei einer begrenzten Anzahl von Vusern in VuGen und unter dem Controller möglicherweise fehlerlos ausgeführt. Möglicherweise treten dann jedoch Probleme bei einer großen Anzahl an Vusern auf. Nicht threadsicherer Code resultiert in der Regel aus der Verwendung von statischen Klassenmitgliedern:

```
import lrapi.*;
public class Actions
{
   private static int iteration_counter = 0;
    public int init() {
        return 0;
    }
    public int action() {
        iteration_counter++;
        return 0;
    }
    public int end() {
        lr.message("Number of Vuser iterations: "+iteration_counter);
        return 0;
    }
}
```

Bei Ausführung nur eines Vusers wird die Anzahl der ausgeführten Iterationen vom Element **iteration_counter** ermittelt. Wenn mehrere Vuser zusammen als Threads auf einer einzelnen Virtual Machine ausgeführt werden, wird das statische Klassenmitglied **iteration_counter** von allen Threads genutzt, was zu inkorrekten Ergebnissen führt. Es wird die Gesamtzahl aller Vuser-Iterationen ermittelt.

Wenn Code als nicht threadsicher bekannt ist, Sie den Code aber trotzdem in Ihrem Skript verwenden möchten, können Sie die Vuser als Prozesse ausführen. Weitere Informationen zum Ausführen von Vusern als Threads oder Prozesse finden Sie unter "Laufzeiteinstellungen" auf Seite 330

Einfache Java-Vuser-Skripts bestehen aus einem einzigen Thread, dem Hauptthread. Nur der Hauptthread hat Zugriff auf das Java-Vuser-API. Wenn ein Java-Vuser sekundäre Arbeitsthreads erzeugt, kann die Verwendung des Java-APIs zu unvorhergesehenen Ergebnissen führen. Es wird daher empfohlen, das Java-Vuser-API nur innerhalb des Hauptthreads zu verwenden. Beachten Sie, dass diese Einschränkung auch Auswirkung auf die Funktion **lr.enable_redirection** hat.

Das folgende Beispiel verdeutlicht, in welchen Fällen das LR-API verwendet werden darf und in welchen nicht. Die erste Meldung im Ausführungsprotokoll besagt, dass der Wert des Kennzeichens falsch ist. Die VM erzeugt daraufhin einen neuen Thread, **set_thread**. Dieser Thread wird ausgeführt und setzt das Kennzeichen auf **true**. Es wird jedoch keine Meldung an das

Protokoll ausgegeben, obwohl der Aufruf von **lr.message** erfolgt ist. Die letzte Protokollmeldung besagt, dass der Code innerhalb des Threads ausgeführt und das Kennzeichen auf **true** gesetzt wurde.

```
boolean flag = false;
public int action() {
   lr.message("Flag value: "+flag);
   Thread set_thread = new Thread(new Runnable();{
        public void run() {
           lr.message("LR-API NOT working!");
           try {Thread.sleep(1000);} catch(Exception e) {}
           flag = true;
        }
   });
   set_thread.start();
   try {Thread.sleep(3000);} catch(Exception e) {}
   lr.message("Flag value: "+flag);
   return 0;
}
```

Ausführen von Java-Vuser-Skripts

Java-Vuser-Skripts unterscheiden sich von C-Vuser-Skripts dadurch, dass sie vor der Ausführung kompiliert werden müssen. C-Vuser-Skripts werden interpretiert. VuGen sucht in der JDK-Installation nach dem **javac**-Compiler und kompiliert den Java-Code innerhalb des Skripts. Dieser Vorgang wird durch die Statusmeldung **Compiling...** am unteren Rand des VuGen-Fensters angegeben. Während der Kompilierung auftretende Fehler werden im Ausführungsprotokoll aufgelistet. Um zu dem Code in Ihrem Skript zu springen, der den Fehler verursacht hat, doppelklicken Sie auf die Meldung mit der Zeilennummer des Fehlers. Beheben Sie den Fehler und führen Sie das Skript erneut aus.

Sobald die Kompilierung erfolgreich war, ändert sich die Statusmeldung von **Compiling...** in **Running...** und VuGen beginnt mit der Skriptausführung. Bei einer erneuten Ausführung des Skripts wird dieses nicht erneut kompiliert, vorausgesetzt, es wurden keine Änderungen an dem Skript vorgenommen. Zur weiteren Fehlerbehebung können Sie Breakpoints setzen und die Ausführung im animierten Modus unter Verwendung der Schrittoption nutzen.

Hinweis: Wenn Sie Aufrufe von JNDI-Erweiterungen in Ihrem Skript verwenden, könnten Probleme entstehen, wenn Sie versuchen, Vuser als **Threads** auszuführen. Der Grund hierfür ist, dass bei JNDI jeder Thread über ein eigenes Kontextklassen-Ladeprogramm verfügen muss. Weisen Sie deshalb für die Ausführung als Thread jeden Vuser an, mit einem eigenen Kontextklassen-Ladeprogramm ausgeführt zu werden. Fügen Sie hierzu folgende Zeile am Anfang des **init**-Abschnitts ein:

```
DummyClassLoader.setContextClassLoader();
```

Bearbeiten und Ausführen von Skripts in Eclipse

Eclipse bietet zusätzliche Tools zum Anzeigen, Bearbeiten und Debuggen Ihrer Java-Vuser-Skripts

(Java Record Replay und Java over HTTP). Sie können Haltepunkte hinzufügen, Variablenwerte anzeigen, Referenzen hinzufügen und das Skript mit IntelliSense bearbeiten. Außerdem können Sie das Skript zum Debuggen schrittweise ausführen.

Wenn Sie das Skript speichern, erstellt VuGen Java-Quelldateien im Ordner des Skripts. Sie können die Lösungsdatei in Eclipse öffnen und all ihre Komponenten im Project Explorer öffnen.

Zum Öffnen des Vuser-Skripts in Eclipse klicken Sie in der VuGen-Symbolleiste auf die Taste **Skript in Eclipse öffnen** . Wenn Sie Eclipse zum ersten Mal über VuGen verwenden, wird das Eclipse-Plug-In automatisch installiert.

> Vor dem Öffnen eines Skripts in Eclipse müssen Sie den Speicherort der Eclipse-IDE im Knoten **Java** der Skriptoptionen von VuGen festlegen. Wenn Sie diesen Wert nicht festlegen, werden Sie zur Auswahl des Speicherorts aufgefordert. Weitere Informationen finden Sie unter "Skriptoptionen" auf Seite 83.

Eine zusätzliche LoadRunner-Symbolleiste ermöglicht den Zugriff auf allgemeine VuGen-Befehle, z. B. "Laufzeiteinstellungen", "Parameterliste", "Ausführen", "Anhalten" und "Szenario erstellen".

LoadRunner enthält auch ein Add-In für Eclipse-Entwickler, mit dem Sie JUnit-Tests erstellen können, die direkt vom Controller aus aufgerufen werden können, ohne dass sie in VuGen geöffnet werden müssen. Die Add-Ins befinden sich im Ordner "DVD/Additional Components". Weitere Informationen zu Add-Ins finden Sie im *LoadRunner-Installationshandbuch*.

Weitere Informationen finden Sie in "Erstellen von Vuser-Skripts oder LoadRunner-Tests in Visual Studio oder Eclipse" auf Seite 1068.

Öffnen von Java-Vuser-Skripts in Eclipse

Eclipse bietet zusätzliche Tools zum Anzeigen, Bearbeiten und Debuggen Ihrer Java-Vuser-Skripts (beispielsweise Java Record Replay und Java over HTTP). Sie können Haltepunkte hinzufügen, Variablenwerte anzeigen, Referenzen hinzufügen und das Skript mit IntelliSense im Eclipse-Editor bearbeiten.

Mit der Integration von VuGen und Eclipse können Sie das Skript, wie aus VuGen gewohnt, über die Eclipse-IDE konfigurieren. Ein Menü **Vuser** in der Eclipse-IDE bietet Ihnen Zugriff auf die Parameterliste, die Laufzeiteinstellungen, die Funktionen zum Ausführen und Anhalten sowie auf die Szenarioerstellung.

So öffnen Sie das Vuser-Skript in Eclipse:

1. Stellen Sie sicher, dass auf Ihrem Computer Eclipse 4.2 oder höher mit JDK 1.7 oder höher ausgeführt wird.

2. Legen Sie den Speicherort der Eclipse-IDE im Knoten **Skripterstellung > Java** im VuGen-Dialogfeld **Optionen** fest. Weitere Informationen finden Sie unter "Skriptoptionen" auf Seite 83.

3. Erstellen Sie ein Java-Skript (Java-Vuser, Java Record Replay, Java over HTTP usw.).

4. Klicken Sie auf die Schaltfläche **Skript in Eclipse öffnen** in der VuGen-Symbolleiste. Wenn Sie Eclipse zum ersten Mal über VuGen verwenden, wird das Eclipse-Plug-In für VuGen automatisch installiert.

5. Doppelklicken Sie auf den entsprechenden Abschnitt, z. B. **Actions.java**, um den Code zu bearbeiten.

6. Mit dem **Vuser**-Menü können Sie Parameter definieren, Laufzeiteinstellungen konfigurieren und das Skript direkt über die Eclipse-IDE ausführen.

Kompilieren und Ausführen eines Skripts als Teil eines Pakets

Zum Erstellen eines Java-Vuser-Skripts ist es u. U. erforderlich, Methoden aus anderen Klassen zu verwenden, in denen die Klasse bzw. die Methode geschützt ist. Wenn Sie ein solches Skript zu kompilieren versuchen, enthalten Sie die Fehlermeldung, dass auf die Methoden nicht zugegriffen werden kann. Um sicherzustellen, dass Ihr Skript auf diese Methoden zugreifen kann, fügen Sie den Namen des Pakets, in dem sie enthalten sind, am Anfang des Skripts in dem Format ein, wie sie es von herkömmlichen Java-Programmen gewohnt sind – <Paket_Name>. Im Beispiel unten wird das Paket **just.do.it** durch eine Pfadangabe definiert:

```
package my.test;
import lrapi.*;
public class Actions
{
     :
}
```

Im Beispiel oben erstellt VuGen automatisch die Ordnerstruktur **my/test** unterhalb des Vuser-Ordners und kopiert die Datei **Actions.java** in **my/test/Actions.java**, wodurch diese mit dem entsprechenden Paket kompiliert werden kann. Denken Sie daran, dass – genau wie bei Java – die Paketanweisung die erste Zeile des Skripts sein muss. Davon ausgenommen sind Kommentare.

Manuelles Erstellen eines Java-Skripts

Im Rahmen dieser Aufgabe wird beschrieben, wie ein benutzerdefiniertes Java-Skript manuell erstellt und bearbeitet wird.

1. **Erstellen eines neuen Skripts**
   a. Öffnen Sie VuGen.

   b. Wählen Sie **File > New** aus oder klicken Sie auf die Schaltfläche **New**. Das Dialogfeld **New Virtual User** wird geöffnet.

c. Wählen Sie **Custom > Java Vuser** in der Liste **Select Vuser type** aus und klicken Sie auf **OK**. VuGen öffnet eine leeres Java-Vuser-Skript.

d. Klicken Sie im Bereich links in den Abschnitt **Actions**, um die Klasse **Actions** anzuzeigen.

2. Einfügen von Code in das Skript

Nach dem Erstellen einer leeren Vorlage können Sie nun den gewünschten Java-Code eingeben bzw. einfügen. Beim Arbeiten mit dieser Art von Vuser-Skript fügen Sie Ihren gesamten Code in die Klasse **Actions** ein. Klicken Sie zum Anzeigen der Klasse **Actions** im Bereich auf der linken Seite auf **Actions**. VuGen zeigt den zugehörigen Inhalt im Bereich auf der rechten Seite an.

```
import lrapi.*;
public class Actions
{
    public int init() {
        return 0;
    }
    public int action() {
        return 0;
    }
    public int end() {
        return 0;
    }
}
```

Die Aktionsklasse enthält drei Methoden: **init**, **action** und **end**. Die Tabelle unten zeigt, was in jede Methode einbezogen werden kann, und wann jede Methode ausgeführt wird.

Skriptmethode	Verwendung zum Emulieren...	Wird ausgeführt, wenn...
init	einer Anmeldung an einen Server.	der Vuser initialisiert (geladen) wird.
action	Clientaktivität.	sich der Vuser im Status **Running** befindet.
end	einer Abmeldungsprozedur	der Vuser endet oder angehalten wird.

Init-Methode

Platzieren Sie alle Anmeldeverfahren und einmalig verwendeten Konfigurationseinstellungen in der Methode **init**. Die Methode **init** wird nur einmal ausgeführt, nämlich wenn der Vuser die Skriptausführung startet. Die folgende **init**-Beispielmethode initialisiert ein Applet. Stellen Sie sicher, dass Sie die Funktion **org.omg.CORBA.ORB** in diesen Abschnitt importieren, so dass sie nicht für jede Iteration wiederholt wird.

```
import org.omg.CORBA.*;
import org.omg.CORBA.ORB.*;
import lrapi.lr;
// Public function: init
    public int init() throws Throwable {
    // Initialize Orb instance...
    MApplet mapplet = new MApplet("http://chaos/classes/", null);
    orb = org.omg.CORBA.ORB.init(mapplet, null);
...
```

Action-Methode

Platzieren Sie alle Vuser-Aktionen innerhalb der Action-Methode. Die Methode **action** wird basierend auf der in den Laufzeiteinstellungen konfigurierten Anzahl von Iterationen ausgeführt. Weitere Informationen zu den Iterationseinstellungen finden Sie unter "Laufzeiteinstellungen" auf Seite 330. Die folgende action-Beispielmethode ruft die Vuser-ID ab und gibt sie zurück.

```
public int action() {
   lr.message("vuser: " + lr.get_vuser_id() + " xxx");
        return 0;
  }
```

End-Methode

Platzieren Sie in der Methode **end** den Code, den der Vuser am Ende des Skripts ausführen soll, z. B. für die Abmeldung beim Server oder zum Bereinigen der Umgebung etc.

Die Methode **end** wird nur einmal ausgeführt, und zwar wenn der Vuser die Skriptausführung beendet. Im folgenden Beispiel beendet die Methode das Skript und schreibt eine Beendigungsmeldung in das Ausführungsprotokoll.

```
public int end() {
       lr.message("End");
       return 0;
   }
```

3. ## Einfügen zusätzlicher LoadRunner-API-Funktionen

   VuGen stellt ein spezielles Java-API für Java-Vuser-Skripts zur Verfügung. Die zugehörigen Funktionen sind alle statische Methoden der Klasse **lrapi.lr**.

   Die Java-API-Funktionen werden in mehrere Kategorien untergliedert: Transaktion, Befehlszeilenanalyse, Information, Zeichenfolge, Nachricht sowie Laufzeitfunktionen.

   Weitere Informationen zu diesen Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**). Wenn Sie ein neues Java-Vuser-Skript erstellen, ist die Klasse import lrapi.* bereits vorhanden.

4. ## Einfügen zusätzlicher Java-Funktionen

Wenn Sie weitere Java-Klassen verwenden möchten, importieren Sie diese wie unten dargestellt am Anfang des Skripts.

Denken Sie daran, den Klassenordner oder die entsprechende jar-Datei zum Klassenpfad hinzuzufügen. Vergewissern Sie sich, dass die zusätzlich verwendeten Klassen threadsicher und skalierbar sind.

```
import java.io.*;
import lrapi.*;
public class Actions
{
...
}
```

5. **Hinzufügen von Skripterweiterungen**

   Sie fügen Skripterweiterungen wie Rendezvous-Punkte, Transaktionen und Ausgabemeldungen hinzu. Weitere Informationen finden Sie unter "Erweitern eines Java-Skripts" unten.

6. **Festlegen der Java-Umgebung**

   Vergewissern Sie sich vor dem Ausführen Ihres Java-Vuser-Skripts, dass die Umgebungsvariablen (path und classpath) auf allen Computern, auf denen Vuser ausgeführt werden, ordnungsgemäß konfiguriert sind:

   - Für die Kompilierung und Wiedergabe der Skripts muss das JDK (Version 1.1, 1.2 oder 1.3) vollständig installiert sein. Die Installation der JRE alleine reicht nicht aus. Es wird nicht empfohlen, mehr als eine JDK- oder JRE-Version auf einem Computer zu installieren. Deinstallieren Sie nach Möglichkeit alle nicht benötigten Versionen.
   - Die Umgebungsvariable **PATH** muss einen Eintrag für **JDK/bin** enthalten.
   - Für JDK 1.1.x muss die Umgebungsvariable **CLASSPATH** den Pfad **classes.zip** (Unterordner **JDK/lib**) und sämtliche VuGen-Klassen (Unterordner **classes**) enthalten.
   - Alle von Java-Vusern verwendeten Klassen müssen im Klassenpfad enthalten sein. Die Definition erfolgt entweder in der Umgebungsvariable **CLASSPATH** des Computers oder in der Liste **Klassenpfadeinträge** im Knoten **Classpath** der Laufzeiteinstellungen.

Erweitern eines Java-Skripts

Im Rahmen dieser Aufgabe wird beschrieben, wie benutzerdefinierte Java-Skripts erweitert werden.

Einfügen von Transaktionen

Mithilfe von Transaktionen lässt sich auch die Serverleistung messen. Bei jeder Transaktion wird die Zeit gemessen, die der Server benötigt, um auf bestimmte Anfragen zu antworten. Bei diesen Anfragen kann es sich um kurze oder komplexe Tasks handeln. Wenn Sie mit LoadRunner

arbeiten, können Sie das Leistungsverhalten transaktionsspezifisch während und nach der Szenarioausführung mithilfe von Onlinemonitoren und -Diagrammen analysieren.

Sie können auch einen Transaktionsstatus angeben: **lr.PASS** oder **lr.FAIL**. Sie können den Vuser automatisch ermitteln lassen, ob die Transaktion erfolgreich war, oder diese in eine Bedingungsschleife integrieren. Innerhalb Ihres Codes könnten Sie beispielsweise auf einen bestimmten Rückgabecode prüfen lassen. Wenn der Code korrekt ist, wird der Status **lr.PASS** ausgegeben. Wenn der Code nicht korrekt ist, wird der Status **lr.FAIL** ausgegeben.

Markieren einer Transaktion

1. Fügen Sie **lr.start_transaction** an dem Punkt in das Skript ein, an dem die Messung der Transaktionsausführung gestartet werden soll.

2. Fügen Sie **lr.end_transaction** an dem Punkt in das Skript ein, an dem die Messung der Transaktionsausführung beendet werden soll. Verwenden Sie den Transaktionsnamen, so wie er in der Funktion **lr.start_transaction** angezeigt wird.

3. Geben Sie den gewünschten Status für die Transaktion an: **lr.PASS** oder **lr.FAIL**.

```
public int action() {
for(int i=0;i<10;i++)
   {
    lr.message("action()"+i);
    lr.start_transaction("trans1");
    lr.think_time(2);
    lr.end_transaction("trans1",lr.PASS);
   }
    return 0;
}
```

Einfügen von Rendezvous-Punkten

Wenn Sie eine hohe Benutzerlast auf dem Client/Server-System simulieren möchten, haben Sie die Möglichkeit, einen Rendezvous-Punkt zu erstellen, an dem alle Vuser eine Task zeitsynchron ausführen. Wenn ein Vuser den Rendezvous-Punkt erreicht, wird er vom Controller so lange angehalten, bis alle andern Vuser ebenfalls dort eintreffen.

Sie legen den Treffpunkt fest, indem Sie eine Rendezvous-Funktion in Ihr Vuser-Skript einfügen.

Einfügen eines Rendezvous-Punkts

- Fügen Sie die Funktion **lr.rendezvous** an der Stelle in das Skript ein, an der die Vuser das Rendezvous ausführen sollen.

```
public int action() {
  for(int i=0;i<10;i++)
   {
    lr.rendezvous("rendz1");
    lr.message("action()"+i);
    lr.think_time(2);
```

```
    }
    return 0;
}
```

Abrufen von Vuser-Informationen

Sie können zum Abrufen von Vuser-Informationen die folgenden Funktionen zu Ihren Vuser-Skripts hinzufügen:

lr.get_attrib_string	Gibt eine Zeichenfolge mit Befehlszeilenargumentenwerten oder Laufzeitinformationen zurück, z. B. die Vuser-ID oder den Namen des Lastgenerators.
lr.get_group_name	Gibt den Namen der Vuser-Gruppe zurück.
lr.get_host_name	Gibt den Namen des Lastgenerators zurück, der das Vuser-Skript ausführt.
lr.get_master_host_name	Gibt den Namen des Computers zurück, auf dem LoadRunner Controller oder Business Process Monitor ausgeführt wird.
lr.get_scenario_id	Gibt die ID des aktuellen Szenarios zurück (nur für LoadRunner).
lr.get_vuser_id	Gibt die ID des aktuellen Vusers zurück (nur für LoadRunner).

Im folgenden Beispiel ruft die Funktion **lr.get_host_name** den Namen des Computers ab, auf dem der Vuser ausgeführt wird.

```
String my_host = lr.get_host_name();
```

Weitere Informationen zu den o. g. Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Ausgeben von Meldungen

Während der Ausführung eines Szenarios werden im Ausgabefenster des Controllers Informationen zur Skriptausführung angezeigt. Sie können Anweisungen zum Senden von Benachrichtigungsmeldungen an den Controller in ein Vuser-Skript einfügen. Der Controller zeigt diese Meldungen im Ausgabefenster an. Beispielsweise können Sie eine Meldung einfügen, die den aktuellen Status der Clientanwendung anzeigt. Sie können diese Meldungen auch in einer Datei speichern.

Hinweis: Vermeiden Sie es, Meldungen innerhalb von Transaktionen zu senden. Hierdurch wird die Ausführungszeit der Transaktion verlängert und die Transaktionsergebnisse möglicherweise verfälscht.

Sie können die folgenden Meldungsfunktionen in Ihren Vuser-Skripts verwenden:

lr.debug_ message	Sendet eine Debugmeldung an das Ausgabefenster.
lr.log_ message	Sendet eine Meldung an die Vuser-Protokolldatei.
lr.message	Sendet eine Meldung an das Ausgabefenster.
lr.output_ message	Sendet eine Meldung mit Informationen zum Speicherort an die Protokolldatei und das Ausgabefenster.

Im folgenden Beispiel sendet **lr.message** eine Meldung mit Informationen zur Schleife an das Ausgabefenster:

```
for(int i=0;i<10;i++)
   {
    lr.message("action()"+i);
    lr.think_time(2);
   }
```

Weitere Informationen zu den Meldungsfunktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Sie können die Vuser anweisen, die Java-Standardausgabe und -Standardfehlerstreams an das VuGen-Ausführungsprotokoll umzuleiten. Dies ist besonders hilfreich, wenn Sie vorhandenen Java-Code einfügen müssen oder vorgefertigte Klassen in Ihren Vuser-Skripts nutzen, die die Aufrufe **System.out** und **System.err** enthalten. Im Ausführungsprotokoll werden Standardausgabemeldungen blau dargestellt, Standardfehler rot.

Das folgende Beispiel zeigt, wie bestimmte Meldungen mithilfe von **lr.enable_redirection** an die Standardausgabe umgeleitet werden:

```
lr.enable_redirection(true);
System.out.println("This is an informatory message...");  // Redirected
System.err.println("This is an error message...");  // Redirected
lr.enable_redirection(false);
System.out.println("This is an informatory message...");  // Not redirected
System.err.println("This is an error message...");  // Not redirected
```

Hinweis: Wenn Sie **lr.enable_redirection** auf **true** setzen, werden alle vorherigen Umleitungen außer Kraft gesetzt. Um die vorherigen Umleitungen wiederherzustellen, setzen Sie diese Funktion auf **false**.

Zusätzliche Informationen zu dieser Funktion finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Emulieren von Benutzerlatenzzeit

Die Zeit, die ein Benutzer zwischen dem Ausführen von aufeinander folgenden Aktionen wartet, wird als die Latenzzeit bezeichnet. Vuser verwenden die Funktion **lr.think_time** zur Emulation der Benutzerlatenzzeit. Im folgenden Beispiel wartet Vuser zwei Sekunden zwischen den Schleifen:

```
for(int i=0;i<10;i++)
   {
    lr.message("action()"+i);
    lr.think_time(2);
   }
```

Sie können die Latenzzeiteinstellungen so verwenden, wie sie im Skript vorkommen, bzw. ein Vielfaches dieser Werte. Die Verarbeitung der Latenzzeitfunktionen durch Vuser kann in den Laufzeiteinstellungen konfiguriert werden. Weitere Informationen finden Sie unter "Laufzeiteinstellungen" auf Seite 330.

Weitere Informationen zur Funktion **lr.think_time** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Verarbeiten von Befehlszeilenargumenten

Sie können zur Laufzeit Werte an ein Vuser-Skript übergeben, indem Sie Befehlszeilenargumente beim Ausführen des Skripts festlegen. Befehlszeilenoptionen werden im Controller oder Business Process Monitor nach dem Skriptpfad und dem Dateinamen eingefügt. Es stehen drei Funktionen zum Lesen der Befehlszeilenargumente und anschließenden Weitergabe der Werte an ein Vuser-Skript zur Verfügung:

lr.get_attrib_double	Ruft die Argumente vom Typ Gleitpunkt mit doppelter Präzision ab.
lr.get_attrib_long	Ruft die Argumente vom Typ lange Ganzzahl ab.
lr.get_attrib_string	Ruft Zeichenfolgen ab.

Ihre Befehlszeile muss das folgende Format haben, wobei die Argumente und ihre Werte paarweise nach dem Skriptnamen aufgelistet werden:

```
script_name   - argument argument_value   -argument argument_value
```

Das folgende Beispiel zeigt die Befehlszeilen-Zeichenfolge, die verwendet wird, damit Skript1 fünfmal auf PC4 wiederholt wird:

```
script1  -host pc4  -loop 5
```

Weitere Informationen zu Befehlszeilen-Parserfunktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**). Weitere Informationen zum Einfügen der Befehlszeilenoptionen finden Sie in den Dokumentationen zu *LoadRunner Controller*, *Performance Center* bzw. *HP Business Service Management*.

Fehlerbehebung und Einschränkungen – Java-Protokoll]

In diesem Abschnitt werden die Fehlerbehebung und die für das Java-Protokoll geltenden Einschränkungen beschrieben.

- Bei Aufzeichnung auf Internet Explorer 8 mit aktiviertem Java-Protokoll müssen zunächst alle IE-Instanzen geschlossen werden, bevor LoadRunner die Explorer-Instanz für die

Aufzeichnungssitzung öffnet.

- Java Record Replay-Protokoll: Für die Aufzeichnung von JMS-Anwendungen ist JDK, Version 1.7 oder 1.6u32 oder niedriger erforderlich.

Java über HTTP-Protokoll

Übersicht über das Java über HTTP-Protokoll

Das Java über HTTP-Protokoll ist auf die Aufzeichnung Java-basierter Anwendungen und Applets ausgerichtet. Es generiert ein Java-Skript, in dem Webfunktionen eingesetzt werden. Dieses Protokoll unterscheidet sich von anderen Java-Protokollen dadurch, dass es Java-Remoteaufrufe über HTTP aufzeichnen und wiedergegeben kann.

Hinweis: Java über HTTP unterstützt asymmetrischen Java-Objektdatenverkehr. Das bedeutet, dass der Datenverkehr bei der Objektserialisierung auch dann erfasst wird, wenn es sich nur um eine einseitige Kommunikation handelt. Dieser Fall tritt ein, wenn eine Serialisierung angefordert wird, die Antwort jedoch einfaches HTTP ist, oder umgekehrt.

Anzeigen von Antworten und Anforderungen im XML-Format

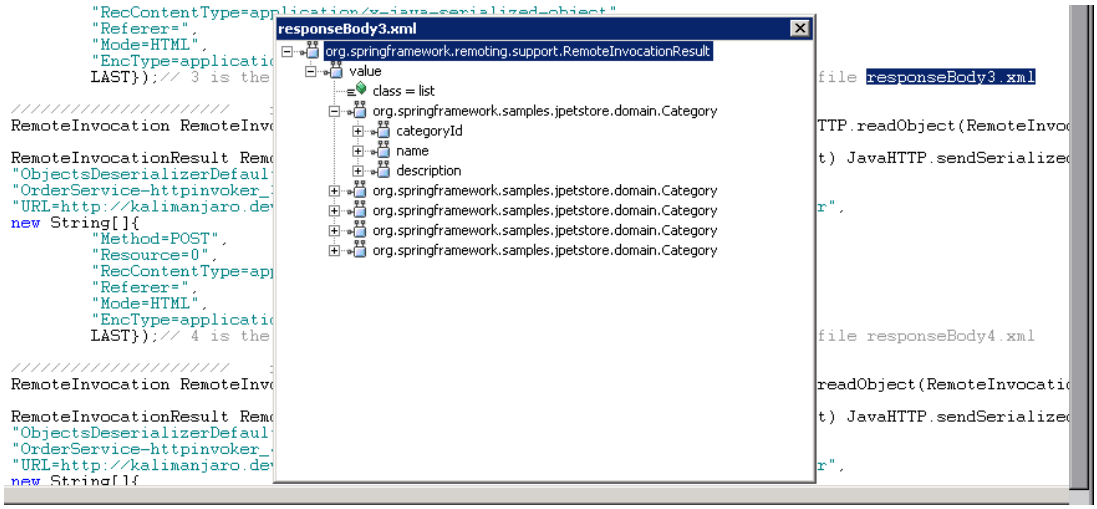
Beachten Sie, dass dieser Punkt nur für das Java over HTTP-Protokoll gilt.

Sie können für jede Anforderung und Antwort die entsprechenden XML-Daten anzeigen, die das binäre Java-Objekt während der Aufzeichnungsphase darstellen.

Anzeigen von XML-Daten

1. Suchen Sie nach dem Zielanforderungs- oder Antwortabschnitt im Code. Klicken Sie mit der rechten Maustaste auf die kommentierte Datei **RequestBodyX.xml** oder **ResponseBodyX.xml**.

2. Wählen Sie **XML anzeigen** aus. Die XML-Daten werden in einem separaten Fenster angezeigt.



Aufzeichnen mit Java über HTTP

Zum Aufzeichnen mit Java over HTTP müssen Sie angeben, welche JAR-Dateien verwendet werden sollen, um die aufgezeichneten Daten zu deserialisieren.

In diesem Abschnitt wird beschrieben, wie Sie die relevanten JAR-Dateien finden und dem Klassenpfad hinzufügen.

Aufzeichnen von Java-Applets

Wenn Ihre Anwendung Java-Applets verwendet, müssen Sie die relevanten JAR-Dateien finden und sie im Klassenpfad aktivieren.

1. Leeren Sie den JAR-Cache erneut, indem Sie **Systemsteuerung > Java** > Registerkarte **Allgemein > Temporäre Internet-Dateien > Einstellungen > Dateien löschen** auswählen.

2. Öffnen Sie Ihre Anwendung und führen Sie einige Geschäftsprozesse aus, um den JAR-Cache erneut mit JAR-Dateien aus Ihrer Anwendung zu füllen. Schließen Sie Ihre Anwendung, wenn Sie fertig sind.

3. Wählen Sie **Systemsteuerung > Java** > Registerkarte **Allgemein > Temporäre Internet-Dateien > Einstellungen > Applets anzeigen** aus. Daraufhin wird der JAR-Cache angezeigt. Dieser sollte nur die von Ihrer Anwendung verwendeten JAR-Dateien enthalten.

4. Laden Sie alle Dateien herunter. Probieren Sie die nachstehenden Optionen in der angezeigten Reihenfolge aus. Wenn diese Aktion erfolgreich ist, fahren Sie mit dem nächsten Schritt fort, um die JAR-Dateien zum Klassenpfad hinzuzufügen.

   a. Möglichkeit 1: Rufen Sie für jede JAR-Datei den aufgeführten URL auf und laden Sie die Datei herunter. Wenn Sie eine oder mehrere der JAR-Dateien nicht herunterladen können, fahren Sie mit der nächsten Option fort.

b. Möglichkeit 2: Leeren Sie den JAR-Cache erneut, indem Sie **Systemsteuerung > Java** > Registerkarte **Allgemein > Temporäre Internet-Dateien > Einstellungen > Dateien löschen** auswählen. Öffnen Sie Ihre Anwendung erneut und führen Sie einige Geschäftsprozesse aus. Schließen Sie die Anwendung nicht. Öffnen Sie die Java-Konsole . Es sollte eine Nachricht für jede JAR-Datei angezeigt werden, der Sie den Speicherort entnehmen können, an dem sie sich in einer temporären Datei auf Ihrem Computer befindet. Es handelt sich normalerweise um Dateien mit Hash, die keine JAR-Erweiterungen aufweisen. Ändern Sie den Namen (einschließlich der Erweiterung - ändern Sie diese in JAR) und kopieren Sie die Datei an einen bekannten Speicherort.

c. Möglichkeit 3: Wenn die Dateien nicht in der Java-Konsole angezeigt werden, suchen Sie den temporären Ordner wie unter **Systemsteuerung > Java** > Registerkarte **Allgemein > Temporäre Internet-Dateien > Einstellungen > Adresse** aufgeführt. Wechseln Sie zum angegebenen Speicherort und benennen Sie die Erweiterungen aller Dateien in den Unterordnern in JAR um. Benennen Sie keine der Dateien im Hauptordner um.

5. Fügen Sie die JAR-Dateien zum Klassenpfad im Knoten **Aufzeichnungsoptionen > Java-Umgebungseinstellungen > Classpath** hinzu. Weitere Informationen finden Sie unter Java > Classpath (Knoten).

Aufzeichnen lokaler Java-Anwendungen

Wenn Sie eine lokale Java-Anwendung aufzeichnen (kein Applet), sind alle JAR-Dateien bereits auf Ihrem Computer vorhanden.

1. Suchen Sie in die Batchdatei, die die Anwendung gestartet hat. Alle JAR-Dateien, auf die verwiesen wird, sollten dem Klassenpfad hinzugefügt werden.

2. Wenn Sie die Batchdatei nicht finden oder verstehen, fügen Sie dem Klassenpfad alle JAR-Dateien aus dem Anwendungsordner und den Unterordnern hinzu.

3. Fügen Sie die JAR-Dateien zum Klassenpfad im Knoten **Aufzeichnungsoptionen > Java-Umgebungseinstellungen > Classpath** hinzu. Weitere Informationen finden Sie unter Java > Classpath (Knoten).

Debuggen von Java über HTTP-Skripts

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie Java over HTTP-Vuser-Skripts debuggen, indem Sie die Anforderungs- und Antwortdaten aus den Aufzeichnungs- und Wiedergabephasen vergleichen.

1. ### Hinzufügen von Argumenten zum VM Param-Knoten

   Wählen Sie den Knoten **Wiedergabe > Laufzeiteinstellungen > Java VM** aus. Geben Sie im Feld **Zusätzliche VM-Parameter** die folgende Zeichenfolge ein:

   ```
   -DdumpServerRequests=true -DdumpServerResponses=true
   ```

2. **Vergleichen von Aufzeichnungs- und Wiedergabedaten**

   Klicken Sie im Lösungs-Explorer mit der rechten Maustaste auf den Skriptnamen, und wählen Sie **Skriptordner öffnen**. Die Daten aus der Aufzeichnungsphase befinden sich im Hauptordner. Die Daten aus der Wiedergabephase befinden sich im Wiedergabeordner.

   Die Dateien, die das Format **RequestBodyX** aufweisen, enthalten die Anforderungsdaten. Die Dateien, die das Format **ResponseBodyX**aufweisen, enthalten die Antwortdaten.

   Für einen Vergleich der Aufzeichnungs- und Wiedergabedaten für Debuggingzwecke vergleichen Sie die Dateien mit identischen Namen aus den Aufzeichnungs- und Wiedergabephasen. Vergleichen Sie beispielsweise die Datei **RequestBody1** aus dem Hauptordner (Aufzeichnungsphase) mit der Datei **RequestBody1** aus dem Wiedergabeordner. Normalerweise sollte es sich um identische Dateien handeln. Fälle, in denen die Dateien nicht identisch sind, weisen möglicherweise auf Probleme im Skript hin.

3. **Entfernen von Argumenten vor Lasttests**

   Kehren Sie zum Knoten **Java VM** und zu den Elementen zurück, die Sie im Feld **Zusätzliche VM-Parameter** hinzugefügt haben.

Einfügen von Parametern in Java über HTTP-Skripts

Parameterfunktionen können für alle Antwort- oder Anforderungstexte an einem bestimmten Speicherort hinzugefügt werden. Dieser Speicherort wird durch eine leere Zeile angegeben, die sich normalerweise ein oder zwei Zeilen unterhalb des Beginns des Antwort- oder Anforderungstexts befindet. Im unten aufgeführten Beispiel können Parameterfunktionen in den leeren Zeilen in jedem requestBody-Abschnitt hinzugefügt werden.

```
//////////////////////    requestBody2.xml    //////////////////////
RemoteInvocation RemoteInvocation_getUsernameList2 =
    (RemoteInvocation) JavaHTTP.readObject(RemoteInvocationBA0);
//INSERT PARAMETERIZATION AND CORRELATION CODE HERE
RemoteInvocationResult RemoteInvocationResult_ArrayList2 =
    (RemoteInvocationResult) JavaHTTP.sendSerialized(RemoteInvocation_getUsernameList2, 2,
"ObjectsDeserializerDefaultImpl",
"OrderService-httpinvoker",
"URL=http://kalimanjaro.devlab.ad:8080/jpetstore/remoting/OrderService-httpinvoker",
new String[]{
        "Method=POST",
        "Resource=0",
        "RecContentType=application/x-java-serialized-object",
        "Referer=",
        "Mode=HTML",
        "EncType=application/x-java-serialized-object",
        LAST});// 2 is the number of the header file, record time response is at file responseBody2.xml

//////////////////////    requestBody3.xml    //////////////////////
RemoteInvocation RemoteInvocation_getCategoryList3 =
    (RemoteInvocation) JavaHTTP.readObject(RemoteInvocationBA1);
//INSERT PARAMETERIZATION AND CORRELATION CODE HERE
RemoteInvocationResult RemoteInvocationResult_ArrayList3 =
    (RemoteInvocationResult) JavaHTTP.sendSerialized(RemoteInvocation_getCategoryList3, 3,
"ObjectsDeserializerDefaultImpl",
"OrderService-httpinvoker_2",
"URL=http://kalimanjaro.devlab.ad:8080/jpetstore/remoting/OrderService-httpinvoker",
new String[]{
        "Method=POST",
        "Resource=0",
        "RecContentType=application/x-java-serialized-object",
        "Referer=",
        "Mode=HTML",
        "EncType=application/x-java-serialized-object",
        LAST});// 3 is the number of the header file, record time response is at file responseBody3.xml
```

Java over HTTP - Fehlerbeseitigung und Beschränkungen

In diesem Abschnitt werden die Fehlerbehebung und die für das Java über HTTP-Protokoll geltenden Einschränkungen beschrieben.

Einschränkungen

- JDK 1.5 oder höher ist erforderlich.
- Lazy Evaluation-Objekte werden nicht unterstützt, etwa der Ruhezustand im Verzögerungsmodus.
- Bei zustandsorientierten Serialisierungsmechanismen auf dem Anwendungsserver kann es zu Beeinträchtigungen der LoadRunner-Deserialisierung kommen. Dies kann zum Auftreten unserialisierter Daten und unerwarteter Fehlern führen.
- Die folgenden Menüelemente stehen für dieses Protokoll nicht zur Verfügung:
  - Neuen Schritt einfügen/Transaktion starten/Transaktion beenden/Rendezvous

Deaktivieren der Ausnahmefehlerüberprüfung

Wenn Ihnen Ausnahmefehler angezeigt werden und Sie sicher sind, dass diese Fehler nicht relevant sind, ermöglicht VuGen die Deaktivierung sämtlicher Fehlermeldungen dieser Art. Wählen Sie hierzu den Knoten **Wiedergeben > Laufzeiteinstellungen > Java VM** aus. Fügen Sie im Feld **Zusätzliche VM-Parameter** die folgende Zeichenfolge an das Ende des aktuellen Eintrags an:

```
-DvalidateServerResponse=false
```

Darüber hinaus können Sie das Fehlerüberprüfungsverhalten während eines bestimmten Schritts ändern, indem Sie in der Skriptansicht ein schließendes Argument zur sendSerialized-Funktion hinzufügen. Weitere Informationen finden Sie im Function Reference.

Korrelieren von PrivateObject-Membern nicht möglich

Wenn Sie Daten korrelieren oder parametrisieren müssen, bei denen es sich nicht um einen privaten Member eines Objekts handelt, können Sie die Funktionen **lrapi.lr2.fieldSetter** und **lrapi.lr2.fieldGetter** verwenden.

```
 RemoteInvocation RemoteInvocation2 = (RemoteInvocation) JavaHTTP.readObject(Rem
oteInvocationBA0);
            RemoteInvocation.methodName="applyToSchool";
            Student student=RemoteInvocation.arguments[0];

            Map grades=lr2.fieldGetter(student,"grades");//grades is a private m
ember of Student
            grades.put("Math","95");
            lr2.fieldSetter(student,"super.name","Tom");
             //Student class inherits the name field from Person. name field is
a string
```

```
            lr2.fieldSetter(student,"super.ID","98764321");
             //Student class inherits the ID field from Person. ID field is an i
nt
            RemoteInvocationResult RemoteInvocationResult_ArrayList2 =
(RemoteInvocationResult) JavaHTTP.sendSerialized(RemoteInvocation2, 2,
             "ObjectsDeserializerDefaultImpl",....
```

LDAP-Protokoll

Übersicht über das LDAP-Protokoll

LDAP (Lightweight Directory Access Protocol) ist ein Protokoll für den Zugriff auf eine Ordnerliste. Der LDAP-Ordner setzt sich aus zahlreichen LDAP-Einträgen zusammen. Jeder LDAP-Eintrag besteht aus einer Sammlung von Attributen mit einem DN-Namen (Distinguished Name). Weitere Informationen zur Erstellung von DN-Namen finden Sie unter "Definieren von Distinguished Name-Einträgen " auf der nächsten Seite.

Die LDAP-Ordnereinträge sind in einer hierarchischen Struktur angeordnet, die politische, geografische und/oder organisatorische Grenzen widerspiegelt. Die Einträge, die einem Land entsprechen, erscheinen in der Struktur ganz oben. Im Anschluss an die Länder erscheinen die Einträge für Staaten oder nationale Organisationen. Danach kommen die Einträge für Personen, Organisationseinheiten, Drucker, Dokumente u. ä.

VuGen zeichnet die Kommunikation über LDAP-Server auf. Dabei wird ein Vuser-Skript mit den Funktionen erstellt, die die Aktionen des Anwenders emulieren. Zu diesen Aktionen zählt das An- und Abmelden vom Server, das Hinzufügen und Löschen von Einträgen und das Abfragen eines Eintrags.

Beispielskript für ein LDAP-Protokoll

Bei den meisten LDAP-Funktionen gibt es zwei unterschiedliche Versionen – eine für globale Sitzungen und eine, bei der Sie eine bestimmte Sitzung angegeben können. Um eine Aktion auf alle Sitzungen anzuwenden, verwenden Sie die Version ohne das Suffix **ex**. Um die Aktion auf eine bestimmte Sitzung anzuwenden, verwenden Sie die Version mit der Sitzungs-ID und dem Suffix **ex**. **mldap_logon** führt beispielsweise eine globale Anmeldung bei dem LDAP-Server durch, während mit **mldap_logon_ex** die Anmeldung beim LDAP-Server für eine bestimmte Sitzung erfolgt.

Im folgenden Beispiel meldet sich der Benutzer bei einem LDAP-Server, ldap1, an. Dabei wird ein Eintrag hinzugefügt und das OU-Attribut von **Sales** in **Marketing** umbenannt.

```
Action()
{
// Anmelden beim LDAP-Server

mldap_logon("Login",
                "URL=ldap://johnsmith:tiger@ldap1:80",
```

```
                LAST);
// Hinzufügen eines Eintrags für Sally R. Jones
mldap_add("LDAP Add",
          "DN=cn=Sally R. Jones,OU=Sales, DC=com",
          "Name=givenName", "Value=Sally", ENDITEM,
          "Name=initials", "Value=R", ENDITEM,
          "Name=sn", "Value=Jones", ENDITEM,
          "Name=objectClass", "Value=contact", ENDITEM,
          LAST);
// Rename Sally's OU to Marketing
mldap_rename("LDAP Rename",
             "DN=CN=Sally R. Jones,OU=Sales,DC=com",
             "NewDN=OU=Marketing",
             LAST);
// Abmelden vom LDAP-Server
mldap_logoff();
    return 0;
}
```

Definieren von Distinguished Name-Einträgen

Die LDAP-API referenziert Objekte nach ihrem **Distinguished Name** (DN). Ein DN setzt sich aus einer Reihe von kommagetrennten **Relative Distinguished Names** (RDN) zusammen.

Ein RDN ist ein Attribut mit einem zugehörigen Wert in der Form Attribut=Wert. Bei den Attributnamen wird keine Groß-/Kleinschreibung beachtet. In der folgenden Tabelle sind die häufigsten RDN-Attributtypen aufgeführt.

Zeichenfolge	Attributtyp
DC	domainComponent
CN	commonName
OU	organizationalUnitName
O	organizationName
STREET	streetAddress
L	localityName

ST	stateOrProvinceName
C	countryName
UID	userid

Im Folgenden finden Sie Beispiele für Distinguished Names:

```
DN=CN=John Smith,OU=Accounting,DC=Fabrikam,DC=COM
DN=CN=Tracy White,CN=admin,DC=corp,DC=Fabrikam,DC=COM
```

In der folgenden Tabelle sind reservierte Zeichen aufgeführt, deren Verwendung in einem Attributwert nicht möglich ist.

Zeichen	Beschreibung
	Leerzeichen oder #-Zeichen zu Beginn einer Zeichenfolge
	Leerzeichen am Ende einer Zeichenfolge
,	Komma
+	Plus-Zeichen
"	Doppelte Anführungszeichen
\	Rückwärtsschrägstrich
<	Kleiner-Zeichen
>	Größer-Zeichen
;	Semikolon

Um ein reserviertes Zeichen als Teil eines Attributwerts verwenden zu können, müssen Sie ihm ein Umschaltzeichen für einen Codewechsel, also einen Rückwärtsschrägstrich (\) voranstellen. Wenn ein Attributwert andere reservierte Zeichen enthält, wie z. B. das Gleichzeichen (=) oder Nicht-UTF-8-Zeichen, müssen Sie den Wert in einem Hexadezimalformat verschlüsseln, also einen Rückwärtsschrägstrich gefolgt von zwei Hexadezimalzahlen schreiben.

Die folgenden Zeichen zeigen DNs, in denen Umschaltzeichen enthalten sind. Bei dem ersten Beispiel handelt es sich um den Namen einer Organisationseinheit mit einem eingebetteten Komma. Das zweite Beispiel ist ein Wert, in dem ein Wagenrücklaufzeichen enthalten ist.

```
DN=CN=Bitwise,OU=Docs\, Support,DC=Fabrikam,DC=COM
DN=CN=Before\0DAfter,OU=Test,DC=North America,DC=Fabrikam,DC=COM
```

Angeben von LDAP-Verbindungsoptionen

Mit der Funktion **mldap_logon[_ex]** können Sie festlegen, wie bei der Anmeldung beim LDAP-Server verfahren werden soll.

Wenn Sie den URL des LDAP-Servers angeben, legen Sie gleichzeitig fest, wie und mit welchen Anmeldeinformationen eine Verbindung hergestellt werden soll.

Beim Angeben des Server-URL verwenden Sie das folgende Format:

```
ldap[s][username:[password]@][server[:port]]
```

Die folgende Tabelle zeigt mehrere Beispiele für Verbindungen mit LDAP-Servern.

Syntax	Beschreibung
ldap://a:b@server.com:389	Stellt eine Verbindung zum Server (Port 389) her und bindet anschließend den Benutzernamen "a", Passwort "b".
ldap://:@server.com	Stellt eine Verbindung zum Server (standardmäßig ungesicherter Port 389) her und bindet anschließend den NULL-Benutzernamen und das Passwort.
ldaps://a:@server.com	Stellt eine Verbindung zum Server (standardmäßig gesicherter Port 636) her und bindet anschließend den Benutzernamen "a", Passwort "".
ldap://@server.com, ldap://server.com	Stellt eine Verbindung zum Server ohne Bindung her.
ldap://a:b@	Bindet den Benutzernamen "a", Passwort "b", und führt dazu eine Bindung der vorhandenen Sitzung durch, ohne erneut eine Verbindung herzustellen.
ldap://:@	Stellt eine anonyme Bindung zu einem NULL-Benutzernamen und Passwort her (führt die Bindung mit der vorhandenen Sitzung durch, ohne erneut eine Verbindung herzustellen).

Mit den folgenden optionalen Argumenten können Sie zusätzlich LDAP-Modi bzw. SSL-Zertifikate angeben:

- **Mode.** Der LDAP-Aufrufmodus: *Sync* oder *Async*
- **Zeitüberschreitung.** Die zur Suche nach dem LDAP-Server maximal zur Verfügung stehende Zeit.
- **Version.** Die LDAP-Protokollversion 1, 2 oder 3.
- **SSLCertDir.** Der Pfad zu der Datenbankdatei mit den SSL-Zertifikaten (cert8.db).
- **SSLKeysDir.** Der Pfad zu der Datenbankdatei mit den SSL-Schlüsseln (key3.db).
- **SSLKeyNickname.** Der Spitzname des SSL-Schlüssels in der Datenbankdatei mit den Schlüsseln.
- **SSLKeyCertNickname.** Der Zertifikat-Spitzname des SSL-Schlüssels in der Datenbankdatei

mit den Zertifikaten.

- **SSLSecModule.** Der Pfad zur Datei mit dem SSL-Sicherheitsmodul (secmod.db).
- **StartTLS.** Erfordert, dass der spezifische StartTLS-Erweiterungsbefehl ausgegeben muss, um die Verbindung in den TLS-Modus (SSL) zu schalten.

Ausführliche Informationen zu diesen Argumenten finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Fehlerbehebung und Einschränkungen – LDAP

In diesem Abschnitt werden die Fehlerbehebung und die für das LDAP-Protokoll geltenden Einschränkungen beschrieben.

- Wenn beim Wiedergeben eines LDAP (Version 3)-Skripts ein Fehler auftrifft, müssen Sie die Anweisung **mldpa_logon_ex** ändern, um die Versionsnummer anzugeben. Fügen Sie nach **"URL=.."** den Text **"Version=3"** hinzu.
- Beim Aufzeichnen von LDAP-Skripts werden die Binärparameterwerte für bestimmte LDAP-Funktionen (z. B. **mldap_add** oder **mldap_modify**) nicht aufgezeichnet. Das Aufzeichnen binärer Parameter ist Teil der erweiterten Funktionalität des Protokolls und wird von VuGen nicht unterstützt.

Mailingdienstprotokolle

Mailingdienstprotokolle - Übersicht

Mailingdienstprotokolle emulieren Benutzer, die mit E-Mail-Clients arbeiten und E-Mails anzeigen und senden. Die folgenden Mailingdienste werden unterstützt:

- Internet Messaging (IMAP)
- MS Exchange (MAPI)
- Post Office Protocol (POP3)
- Simple Mail Transfer Protocol (SMTP)

Mit Ausnahme des MAPI-Protokolls, das nur die Wiedergabe unterstützt, unterstützen die Mailprotokolle sowohl das Aufzeichnen als auch die Wiedergabe.

IMAP-Protokoll - Übersicht

Funktionen von IMAP-Vuser-Skripts zeichnen das Internet Mail Application Protocol auf.

Alle IMAP-Funktionen beginnen mit dem Präfix **imap**. Ausführliche Informationen zur Syntax dieser Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Im folgenden Beispiel erstellt die Funktion **imap_create** mehrere neue Postfächer: **Products**, **Solutions** und **FAQs**.

```
Actions()
{
    imap_logon("ImapLogon",
            "URL=imap://johnd:letmein@exchange.mycompany.com",
            LAST);
    imap_create("CreateMailboxes",
            "Mailbox=Products",
            "Mailbox=Solutions",
            "Mailbox=FAQs",
            LAST);
    imap_logout();
    return 1;
}
```

Hinweis: VuGen unterstützt derzeit nur die IMAP LOGIN-Authentifizierungsmethode, nicht aber die AUTHENTICATE-Methode.

MAPI-Protokoll - Übersicht

Funktionen von MAPI-Vuser-Skripts generieren die Kommunikation, die auf einem MS Exchange-Server ein- und ausgeht. Alle MAPI-Funktionen beginnen mit dem Präfix **mapi**. Ausführliche Informationen zur Syntax dieser Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**). Das Aufzeichnen von Vuser-Skripts wird für das MAPI-Protokoll nicht unterstützt.

Hinweis: Um MAPI-Skripts auszuführen, müssen Sie auf dem Computer, auf dem das Skript ausgeführt wird, ein E-Mailprofil definieren. Installieren Sie beispielsweise Outlook Express, legen Sie die Anwendung als standardmäßigen Mailclient fest und erstellen ein E-Mailkonto. Alternativ können Sie auch Microsoft Outlook installieren, die Anwendung als standardmäßigen Mailclient festlegen und ein E-Mailkonto und ein E-Mailprofil erstellen. Um ein E-Mailprofil in Microsoft Outlook zu erstellen, wählen Sie **Settings > Control Panel > Mail** > **Show Profiles** aus und fügen ein E-Mailprofil hinzu.

Im folgenden Beispiel sendet die Funktion **mapi_send_mail** eine Kurznotiz über einen MS Exchange-Server.

```
Actions()
{
    mapi_logon("Logon",
            "ProfileName=John Smith",
            "ProfilePass=Tiger",
            LAST);
    //Send a Sticky Note message
    mapi_send_mail("SendMail",
```

```
            "To=user1@techno.merc-int.com",
            "Cc=user0002t@techno.merc-int.com",
            "Subject=<GROUP>:<VUID> @ <DATE>",
            "Type=Ipm.StickyNote",
            "Body=Please update your profile today.",
            LAST);
    mapi_logout();
    return 1;
}
```

POP3-Protokoll - Übersicht

Funktionen von POP3-Vuser-Skripts emulieren Aktionen unter Verwendung des POP3-Protokolls (Post Office Protocol). Alle diese Funktionen beginnen mit dem Präfix **pop3**. Ausführliche Informationen zur Syntax dieser Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Im folgenden Beispiel ruft die Funktion **pop3_retrieve** fünf Nachrichten vom POP3-Server ab.

```
Actions()
{
pop3_logon("Login", "
        URL=pop3://user0004t:my_pwd@techno.merc-int.com",
        LAST);
// List all messages on the server and receive that value
totalMessages = pop3_list("POP3", LAST);
// Display the received value (It is also displayed by the pop3_list function)
lr_log_message("There are %d messages.\r\n\r\n", totalMessages);
// Retrieve 5 messages on the server without deleting them
pop3_retrieve("POP3", "RetrieveList=1:5", "DeleteMail=false", LAST);
pop3_logoff();
    return 1;
}
```

SMTP-Protokoll - Übersicht

Funktionen von SMTP-Vuser-Skripts emulieren den SMTP-Datenverkehr (Single Mail Transfer Protocol). Alle SMTP-Funktionen beginnen mit dem Präfix **smtp**. Ausführliche Informationen zur Syntax dieser Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Im folgenden Beispiel sendet die Funktion **smtp_send_mail** eine E-Mailnachricht über den SMTP-Mailserver **techno**.

```
Actions()
{
    smtp_logon("Logon",
               "URL=smtp://user0001t@techno.merc-int.com",
               "CommonName=Smtp Test User 0001",
               NULL);
```

```
    smtp_send_mail(    "SendMail",
                    "To=user0002t@merc-int.com",
                    "Subject=MIC Smtp: Sample Test",
                    "MAILOPTIONS",
                    "X-Priority: 3",
                    "X-MSMail-Priority: Medium",
                    "X-Mailer: Microsoft Outlook Express 5.50.400\r\n",
                    "X-MimeOLE: By Microsoft MimeOLE V5.50.00\r\n",
                    "MAILDATA",
                    "MessageText="
                        "Content-Type: text/plain;\r\n"
                        "\tcharset=\"iso-8859-1\"\r\n"
                        "Test,\r\n"
                        "MessageBlob=16384",
                NULL);
    smtp_logout();
    return 1;
}
```

Protokolle für mobile Anwendungen

Protokolle für mobile Anwendungen - Übersicht

Die Protokolle für mobile Anwendungen von VuGen erweitern den Funktionsumfang von LoadRunner, sodass Benutzeraktivitäten von mobilen Anwendungen sowohl **systemeigen**[1] als auch **browserbasiert**[2] aufgezeichnet werden können. Mit dieser Lösung können Sie folgende Aktionen durchführen:

- Simulieren von Benutzern, die mit Mobilgeräten arbeiten
- Erstellen von Skripts anhand der Aufzeichnung der Mobilgeräte oder Emulatoren

Protokolloptionen für Mobilgeräte

Sie können mit einem der beiden folgenden Protokolle Benutzeraktivitäten auf einem Mobilgerät aufzeichnen und Skripts in VuGen generieren:

- **Mobile Application - HTTP/HTML:** Ein Protokoll, mit dem Sie Skripts anhand von Mobilgeräten oder Geräteemulatoren entwickeln können, die über HTTP mit Servern

[1]Eine mobile Anwendung, zum Beispiel ein neuer Dienst. Die Anwendung befindet sich auf dem Gerät, kommuniziert jedoch in unterschiedlichen Intervallen mit dem Server.
[2]Eine browserbasierte Anwendung, die für die Anzeige des Mobilgeräts konfiguriert wurde.

kommunizieren. Sie können den Netzwerkdatenverkehr in einer Aufzeichnungsdatei (PCAP-Datei) aufzeichnen und ein VuGen-Skript erstellen. Zudem können Sie Ihre Skripts mit einem Mobilgeräteemulator auf dem VuGen-Computer entwickeln.

- **TruClient Ajax - Mobile:** Ein Protokoll, mit dem Sie Benutzeraktivitäten anhand der Ajax TruClient-Technologie in browserbasierten mobilen Anwendungen aufzeichnen können. Der Ajax TruClient-Browser wird so modifiziert, dass die Anzeige Ihres mobilen Browsers emuliert wird.

Anhand der folgenden Tabelle können Sie feststellen, welches Protokoll Ihren Anforderungen am besten entspricht:

Anwendungstyp	TruClient Ajax - Mobile	Mobile Application - HTTP/HTML
Browserbasierte Anwendungen	✓	✓
Systemeigene Anwendungen		✓

Weitere Informationen finden Sie unter "Auswählen einer Aufzeichnungsmethode für mobile Anwendungen" unten.

Auswählen einer Aufzeichnungsmethode für mobile Anwendungen

Sie können mit einem der folgenden Protokolle Benutzeraktivitäten auf einem Mobilgerät aufzeichnen und Skripts in VuGen generieren:

- **Mobile Application - HTTP/HTML:** Ein Protokoll, mit dem Sie Skripts anhand von Mobilgeräten oder Geräteemulatoren entwickeln können, die über HTTP mit Servern kommunizieren. Sie können den Netzwerkdatenverkehr in einer Aufzeichnungsdatei (PCAP-Datei) aufzeichnen und ein VuGen-Skript erstellen. Zudem können Sie Ihre Skripts mit einem Mobilgeräteemulator auf dem VuGen-Computer entwickeln.

- **TruClient Ajax - Mobile:** Ein Protokoll, mit dem Sie Benutzeraktivitäten anhand der Ajax TruClient-Technologie in browserbasierten mobilen Anwendungen aufzeichnen können. Der Ajax TruClient-Browser wird so modifiziert, dass die Anzeige Ihres mobilen Browsers emuliert wird.

- **SMP (SAP Mobile Platform):** Ein Protokoll, mit dem Sie .NET-basierte Skripts mit von SMP generierten Dateien erstellen können. Weitere Informationen finden Sie unter ".NET-Protokoll" auf Seite 786.

In den folgenden Tabellen sind die Anforderungen für die verschiedenen Optionen für die Aufzeichnung von mobilen Anwendungen zusammengefasst.

Weitere Informationen zu einer bestimmten Aufzeichnungsmethode erhalten Sie durch Klicken auf die verknüpfung.

TruClient Ajax - Protokoll für mobile Anwendungen

Clientanwendung	zu verwendendes Protokoll
• browserbasierte mobile Version einer Webseite • unterstützt Firefox	Mobile TruClient

SMP (SAP Mobile Platform)

Clientanwendung	zu verwendendes Protokoll
• Wird erstellt auf SMP (SAP Mobile Platform)	SMP (SAP Mobile Platform)

Mobile Application - HTTP/HTMLProtokoll

Clientanwendung	und	zu verwendendes Protokoll
• Kommunikationsprotokoll = HTML/HTTP • browserbasierte oder systemeigene Anwendung	• Ihr Gerät ist im selben Netzwerk wie der VuGen Computer. • Ihr Gerät lässt die Proxykonfiguration zu.	LoadRunner-Proxy

Mobile Application - HTTP/HTMLProtokollForts.

Clientanwendung	und	zu verwendendes Protokoll
• Kommunikationsprotokoll = HTML/HTTP • browserbasierte oder systemeigene Anwendung	• Sie möchten (oder können) nicht über das eigentliche Gerät aufzeichnen. • Das Gerät, die Serveranwendung und der VuGen-Computer sind im selben Netzwerk. • Sie können den mobilen Sniffer-Agent auf dem Servercomputer installieren.	Mobile HTTP/ HTML > Aufzeichnen & Datenverkehr analysieren
• Kommunikationsprotokoll = HTML/HTTP • browserbasierte oder systemeigene Anwendung	• Sie haben eine vorhandene Aufzeichnungsdatei.	Mobile HTTP/ HTML > Datenverkehr analysieren
• Kommunikationsprotokoll = HTML/HTTP • browserbasierte oder systemeigene Anwendung	• Sie möchten (oder können) nicht über das eigentliche Gerät aufzeichnen. • Ihr mobiles BS ist Android. • Sie haben einen Geräteemulator.	Mobile HTTP/HTML > Emulator aufzeichnen
• Kommunikationsprotokoll = HTML/HTTP • browserbasierte oder systemeigene Anwendung	• Sie haben ein **Root**[1]-Android-Gerät. • Sie können die Anwendung **LoadRunner Mobile Recorder** installieren.	Mobiles Aufzeichnen

Netzwerk – Geschwindigkeitssimulation

Maximale Bandbreite

Diese Option wird in Fällen bereitgestellt, in denen Sie ein Mobilfunknetz emulieren möchten. Dies ist die Standardeinstellung.

[1]Der Vorgang, einem Benutzer privilegierten Zugriff auf das Gerätesubsystem zu ermöglichen.

Standardbandbreite

Die Simulation der Netzwerkgeschwindigkeit für Protokolle für mobile Anwendungen bietet vordefinierte Einstellungen, die für bekannte Mobilfunknetze geeignet sind:

- General Packet Radio Service (GPRS)
- Enhanced Data rates for GSM Evolution (EDGE)
- Universal Mobile Telecommunications System (UMTS)
- High-Speed Downlink Packet Access (HSDPA)
- High-Speed Downlink Packet Access Phase 2 (HSDPA phase 2)
- High-Speed Uplink Packet Access (HSUPA)

Jeder Netzwerktyp weist eine maximale und eine erwartete Rate auf. Die maximale Rate stellt die optimale Leistungsrate der Technologie dar, während die erwartete Rate die Echtzeitleistung exakter wiedergibt.

Benutzerdefinierte Bandbreite

Anhand der Simulation der Netzwerkgeschwindigkeit können Sie benutzerdefinierte Download- und Upload-Geschwindigkeiten in Bit definiert festlegen. Sie können entweder einen einzelnen Wert oder einen Bereich für die Upload- bzw. Download-Geschwindigkeit festlegen. Diese Option ist beispielsweise sinnvoll, wenn Sie für einen bestimmten Bereich eine erwartete Netzwerkgeschwindigkeit von Ihrem Mobilfunkanbieter zugesagt bekommen haben.

Weitere Informationen zum Festlegen eines benutzerdefinierten Bandbreite finden Sie unter . "Netzwerk > Geschwindigkeitssimulation (Knoten)" unten

Netzwerk > Geschwindigkeitssimulation (Knoten)

In diesem Dialogfeld können Sie die Bandbreite für das mobile Protokoll konfigurieren.

Zugriff	**VuGen > Laufzeiteinstellungen > Netzwerk > Geschwindigkeitssimulation**
Wichtige Informationen	Diese Option steht nur für mobile Protokolle zur Verfügung. "Netzwerk – Geschwindigkeitssimulation" auf der vorherigen Seite

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Maximale Bandbreite verwenden	Vuser werden mit der maximalen Bandbreite ausgeführt, die das Netzwerk ermöglicht. **Standardwert**: Aktiviert.
Standardbandbreite verwenden	Wählen Sie eine für Mobilgeräte verfügbare Bandbreite aus.
Benutzerdefinierte Bandbreite	**Downloadgeschwindigkeit** Geben Sie eine benutzerdefinierte Downloadgeschwindigkeit in Bits an. Sie können einen Bereich oder einen Einzelwert angeben. **Uploadgeschwindigkeit** Geben Sie eine benutzerdefinierte Uploadgeschwindigkeit in Bits an. Sie können einen Bereich oder einen Einzelwert angeben. **Hinweis:** • Wenn Sie **Benutzerdefinierte Bandbreite** auswählen, aber keine Upload- und Downloadgeschwindigkeit angeben, wird die Standardeinstellung – maximale Bandbreite – verwendet. • Wenn Sie **Benutzerdefinierte Bandbreite** auswählen, aber keine Upload- oder Downloadgeschwindigkeit angeben, wird für die nicht angegebene Einstellung automatisch der definierte Wert verwendet.

Mobile Application - HTTP/HTML

Aufzeichnungsmethoden

Erstellen eines Skripts mit der LoadRunner Mobile Recorder-Anwendung

Sie können mit der LoadRunner Mobile Recorder-Anwendung eine Aufzeichnungsdatei auf Ihrem mobilen Gerät aufzeichnen.

Weitere Informationen finden Sie unter:

- "Aufzeichnen mit Mobile Recorder" auf Seite 774
- "LoadRunner Mobile Recording-Anwendung - Benutzeroberfläche" auf Seite 781

Erstellen eines Skripts mit dem LoadRunner-Proxy

Der VuGen-Computer fungiert als Proxyserver, der den gesamten Datenverkehr vom mobilen Gerät zum Zielserver erfasst. Nach der Aufzeichnung des Geschäftsprozesses erstellt VuGen ein Skript.

Weitere Informationen finden Sie unter:

- "Aufzeichnung über einen Proxy - Übersicht" auf Seite 264
- "Aufzeichnen eines Skripts über einen Proxy" auf Seite 266

Aufzeichnen von Datenverkehr in eine Aufzeichnungsdatei (Sniffer)

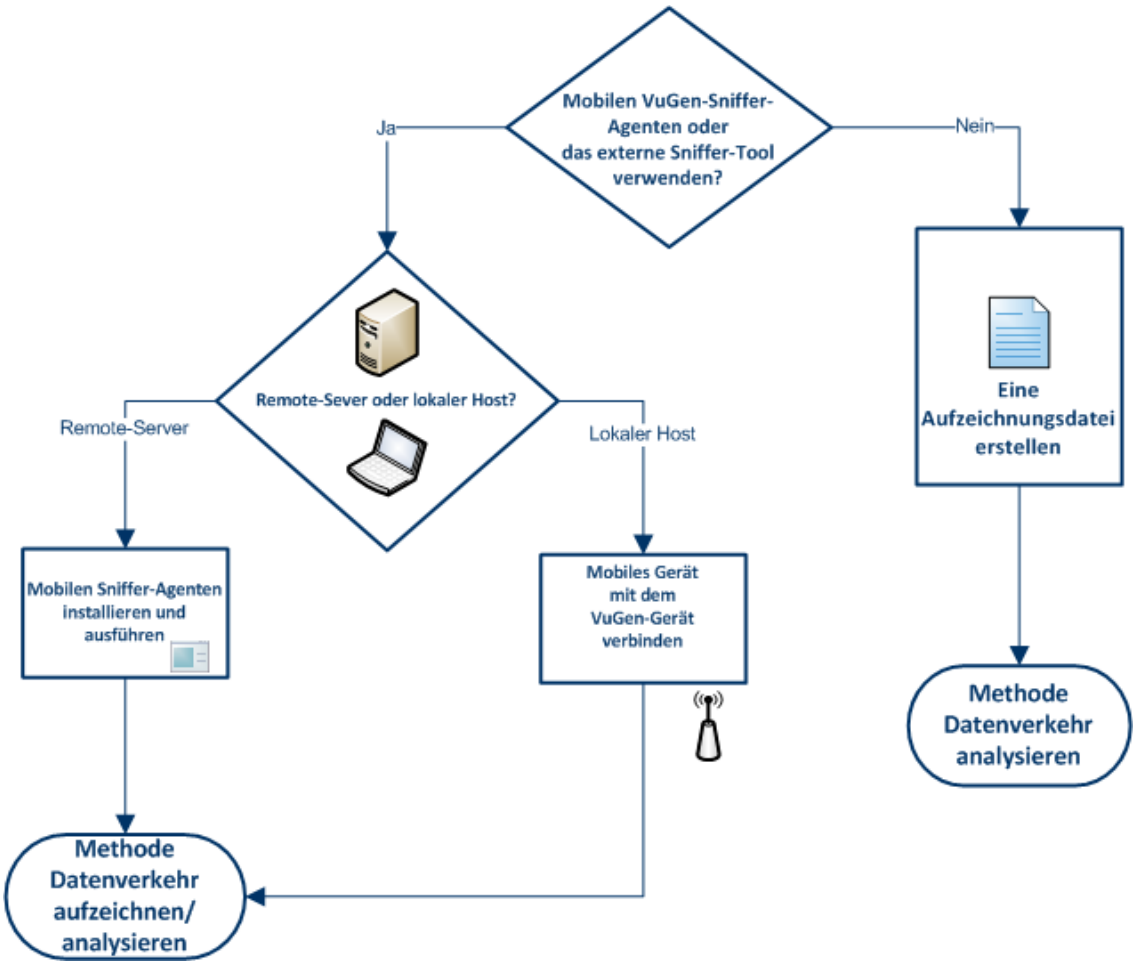
Das Aufzeichnen von Anwendungsverkehr in einer Aufzeichnungsdatei ist effizient, wenn es nicht möglich ist, eine Anwendung mit VuGen aufzuzeichnen, was bei mobilen Anwendungen der Fall ist. Eine Aufzeichnungsdatei ist eine Verfolgungsdatei, in der der gesamte TCP-Verkehr im Netzwerk protokolliert wird. Mithilfe einer Sniffer-Anwendung erhalten Sie ein Abbild des gesamten Netzwerkverkehrs. Der Sniffer zeichnet alle Ereignisse innerhalb des Netzwerks auf und speichert diese in einer Aufzeichnungsdatei. Um ein kleineres, besser handhabbares Skript zu generieren, versuchen Sie, die Aufzeichnung des Netzwerkverkehrs auf den Zeitraum zu beschränken, in dem Sie Aktionen innerhalb der betreffenden Anwendung ausführen.

Je nach Betriebssystem und Gerät gibt es verschiedene Optionen zum Aufzeichnen einer Aufzeichnungsdatei. In der nachfolgenden Tabelle werden einige Speicherorte sowie deren Vor- und Nachteile aufgeführt:

Quelle der Aufzeichnungs-datei	Unterstützt durch mobilen Sniffer-Agent	Vorteile	Nachteile
Mobilgerät, das die LR Mobile Recording-Anwendung verwendet		Der tatsächliche Datenverkehr vom Gerät wird aufgezeichnet.	Administratorrechte erforderlich
VuGen mit konfiguriertem **"Hot Spot**[1]**"**	✓	• Einfach und intuitiv • Keine Administratorrechte erforderlich	• Nicht alle Geräte unterstützen "Ad-hoc" • Nur WiFi wird erfasst (kein mobiler Netzwerkverkehr)
Server	✓	Der tatsächliche Datenverkehr wird aufgezeichnet (sowohl WiFi als auch mobiler Netzwerkverkehr)	• Administratorrechte erforderlich • Installation einer Software auf dem Server erforderlich

[1] Eine WiFi-Verbindung

Sie können Datenverkehr ganz flexibel mit dem mobilen Sniffer-Agenten von LoadRunner oder mit einem externen Sniffer-Agenten aufzeichnen. Das folgende Flussdiagramm zeigt den Ablauf für beide Methoden:



Aufzeichnen von Datenverkehr mit dem mobilen Sniffer-Agenten von VuGen

Aufzeichnen von Datenverkehr auf Remoteservern

Wenn Sie Datenverkehr auf einem Remoteserver aufzeichnen möchten, müssen Sie zunächst den mobilen Sniffer-Agenten auf Ihrem Server installieren. Dazu kopieren Sie den relevanten Ordner aus **<LR-Installationsverzeichnis>\mobileRemoteAgent** an einen beliebigen Speicherort auf Ihrem Server. In der folgenden Tabelle werden die unterstützten Betriebssysteme und die entsprechenden Dateiverzeichnisse aufgelistet:

OS	Ordner
Windows-Betriebssystem	Win32

Linux	LinuxRH3
Mac	Mac

Nach dem Kopieren des Ordners starten Sie den Prozess, indem Sie ein Befehlszeilenargument eingeben. Beim Aufzeichnen in einer Windows-Umgebung sieht die Befehlszeilen-Zeichenfolge ungefähr so aus:

```
mongoose-2.11.exe -e errorLog.txt -r "C:\Program
Files\HP\LoadRunner\mobileRemoteAgent\win32" -C ".cgi" -p 80
```

Allgemeine Befehlszeilenargumente und ihre Beschreibungen:

-c	Identifizieren der Erweiterungen des remoten Sniffer-Agenten. Immer "cgi".
-p	Identifizieren des Ports und optional der IP-Adresse, die der mobile Sniffer-Agent abhören soll.
-r	Speicherort des mobilen Sniffer-Agenten. **Der Pfad muss absolut sein, da der Prozess sonst fehlschlägt.**
-l	Schränkt den Zugriff auf den mobilen Sniffer-Agenten auf einen bestimmten Client ein, der durch die IP identifiziert wird.
-e	Name des Fehlerprotokolls.

Hinweis: Verwenden Sie nicht die Option "–g" ("digest user/password"). Damit kommt es zu einem Fehler.

Nachdem ein Sniffer-Agent installiert wurde, können Sie den Agenten manuell starten und mit der Methode **Aufzeichnen/Analysieren** des **Aufzeichnungs-Assistenten** fortfahren.

Weitere Informationen finden Sie unter "Aufzeichnungs-Assistent" auf Seite 776.

Optionen für CGI-Konfigurationsdatei

Die CGI-Konfigurationsdatei befindet sich im **<LR-Installationsverzeichnis>\mobileRemoteAgent\<win32/Mac/LinuxRH3>/cgi-bin/mobileCGI.conf**.

Sie können folgende Optionen konfigurieren:

- **timeout-seconds**

  Standard = 1800

Die Überprüfung der Zeitüberschreitungssitzung wird angehalten, wenn der Prozess aufgrund von fehlendem Datenverkehr inaktiv ist. Sobald der Adapter Ereignisse aufzeichnet, wird die Sitzung fortgesetzt.

- **max-pcap-file-size-kb**

  Standard = 102400

  Die Überprüfung der maximalen Dateigröße wird angehalten, wenn der Prozess aufgrund von fehlendem Datenverkehr inaktiv ist. Sobald der Adapter Ereignisse aufzeichnet, wird die Sitzung fortgesetzt.

  Da der Datenverkehr zuerst in eine Pufferdatei geschrieben und dann im Flash-Speicher der Festplatte gespeichert wird, kann die maximale Größe überschritten werden.

- **log-file-enable**

  Standard = 0

Beispiel der CGI-Konfigurationsdatei

timeout-seconds 1800

max-pcap-file-size-kb 102400

log-file-enable 0

Aufzeichnen von Datenverkehr auf lokalen Hosts

In bestimmten Fällen ist das direkte Aufzeichnen von Datenverkehr auf dem Anwendungsserver nicht praktisch. In dieser Instanz können Sie Datenverkehr auf Ihrem lokalen Computer aufzeichnen, indem Sie Ihr Mobilgerät mit dem lokalen Host verbinden. Durch Klicken auf **Verbinden** wird der mobile Sniffer-Agent gestartet. Fahren Sie fort mit der Methode **Aufzeichnen/Analysieren** des **Aufzeichnungs-Assistenten.**

Konfigurieren eines Hotspot auf dem lokalen Host

Wenn Sie mit einem lokalen Host aufzeichnen möchten, müssen Sie zunächst Ihren Computer als Hotspot konfigurieren. Dazu können Sie Adhoc oder SoftApp verwenden.

Informationen zu Adhoc und SoftAP finden Sie auf vielen öffentlich zugänglichen Websites.

Hinweise:

- Adhoc wird nicht von allen Geräten unterstützt. Beispielsweise wird es von einigen Android-Versionen nicht unterstützt.
- Da unser Agent libpcap verwendet (gleiches Paket wie Wireshark), werden nicht alle Netzwerkkonfigurationen vom mobilen Sniffer-Agenten unterstützt.
- Damit ein Hotspot funktioniert, müssen Sie Ihre Firewall entsprechend konfigurieren.

- Der Hotspot kann mit NAT implementiert werden, also ist die Adresse des Mobilgeräts mit dem Hotspot-Computer identisch.

Nachdem ein Hotspot auf dem lokalen Host konfiguriert wurde, führen Sie die Methode **Aufzeichnen/Analysieren** des **Aufzeichnungs-Assistenten** aus.

Weitere Informationen finden Sie unter "Aufzeichnungs-Assistent" auf Seite 776.

Aufzeichnen von Datenverkehr mit einem externen Sniffer-Tool

Die meisten UNIX-Betriebssysteme verfügen über ein integriertes Aufzeichnungstool. Darüber hinaus gibt es viele Aufzeichnungstools wie Wireshark/tcpdump, die zum Download zur Verfügung stehen. Achten Sie bei der Verwendung externer Tools darauf, dass sämtliche Paketdaten vollständig aufgezeichnet werden. Bei einigen Dienstprogrammen müssen zusätzliche Argumente angegeben werden. Bei tcpdump muss beispielsweise das Argument **-s 0** angegeben werden, damit die Datenpakete vollständig ohne Trunkierungen aufgezeichnet werden. Sie können eine Aufzeichnungsdatei auch manuell über das Befehlszeilenprogramm erfassen.

Das VuGen-Befehlszeilendienstprogramm **lrtcpdump** befindet sich im Ordner **LoadRunnerInstallationDVD\Additional Components\lrtcpdump**. Für jede Betriebssystemplattform ist ein eigenes Befehlszeilenprogramm vorhanden:

- lrtcpdump.exe (Windows)
- lrtcpdump.hp9
- lrtcpdump.ibm
- lrtcpdump.Linux
- lrtcpdump.solv4

Weitere Informationen finden Sie unter "Erstellen eines Skripts durch Analysieren des Datenverkehrs" auf Seite 945.

Darüber hinaus gibt es viele Aufzeichnungstools wie Wireshark/tcpdump, die zum Download zur Verfügung stehen. Nachdem die Aufzeichnungsdatei erstellt wurde, können Sie das Skript mit der Methode **Datenverkehr analysieren** des **Aufzeichnungs-Assistenten** generieren.

Weitere Informationen finden Sie unter "Aufzeichnungs-Assistent" auf Seite 776.

Fehlerbehebung für fehlende Pakete

Problem: In Ihrem Skript fehlen Schritte, die Sie in einer Aufzeichnungsdatei aufgezeichnet haben.

Unter **Ausgabebereich** > **Codegenerierung** wird folgende Warnung angezeigt:

```
Warning: One or more responses are missing or have missing packets. Therefore, a
step may appear to be missing in the script.
This issue can be caused if the recording was stopped before all the responses w
ere received.
```

```
If the script is generated from a .pcap file, check if the file has missing pack
ets.
```

Dieser Fehler kann durch unnötige Netzwerkaktivität auf dem aufgezeichneten Computer verursacht werden, was dazu führen kann, dass die aufzeichnende Anwendung Pakete löscht.

Schritte für die Behebung: Stellen Sie sicher, dass auf dem aufzeichnenden Computer im Hintergrund kein unnötiger Netzwerkverkehr stattfindet.

Hinweis: Um dieses Problem zu umgehen, wählen Sie **Aufzeichnungsoptionen > HTTP-Eigenschaften > Erweitert > Schritte mit fehlenden Antworten generieren**, um Schritte für HTTP-Anforderungen zu erstellen, für die Serverantworten fehlen.

Analysieren von Datenverkehr

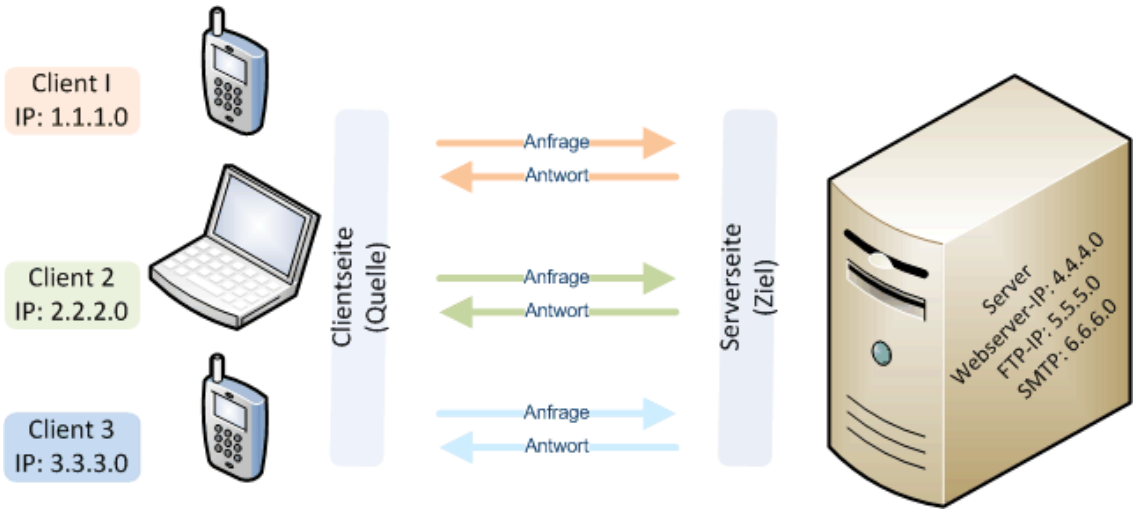
Sie müssen angeben, welche IP-Adresse der Aufzeichnungs-Assistent analysieren soll. Die Option **Datenverkehr analysieren** des **Aufzeichnungs-Assistenten** bietet zwei Methoden für die Angabe der IP-Adresse an, deren Datenverkehr in das von VuGen generierte Skript einbezogen werden soll. Wie die Grafik zeigt, wird der Datenverkehr entweder von der Serverseite oder von der Clientseite ermittelt. Sie müssen die IP-Adresse angeben, deren server- oder clientseitiger Datenverkehr in das Skript einbezogen werden soll.

Serverseite (Ziel)

Sie können mehrere IP-Adressen auf einem Server konfigurieren. Sie können beispielsweise einen SMTP- , FTP- und Webserver auf demselben Computer mit eindeutigen IP-Adressen ausführen. Wenn der Sniffer-Agent die Aufzeichnungsdatei erstellt, enthält sie sämtliche Ereignisse aller Dienste auf dem Server. Sie können allerdings die IP-Adresse und den Port der getesteten Anwendung angeben, indem Sie den serverseitigen Datenverkehr konfigurieren. Diese Methode bietet sich an, wenn Sie sich sicher sind, dass nur ein Client mit dem Server kommuniziert.

Clientseite (Quelle)

Sie können den Client, der mit dem Server kommuniziert hat, als die zu analysierende IP-Adresse angeben. In diesem Fall werden nur die Ereignisse der IP-Adresse des Clients in das von VuGen generierte Skript einbezogen.



Filteroptionen

Sie können zusätzliche IP-Adressen, die in die Codegenerierung eingeschlossen oder von der Codegenerierung ausgeschlossen werden sollen, unter **Aufzeichnungsoptionen > Datenverkehrsanalyse > Filter für Datenverkehr** angeben. Weitere Informationen finden Sie unter "Datenverkehrsanalyse > Filter für Datenverkehr" auf Seite 247.

Aufzeichnen mit Emulation

Mobiles BS	Link für SDK-Download	Aufzu-zeichnen-der Emulator (Beispiel)	Befehlszeile (Beispiel)	Arbeits-verzeichnis (Beispiel)
Android	Download für Android-Emulator	C:\Programme\Android\android-sdk-windows\tools\emulator.exe	@Android_v2.2	C:\Programme\Android\android-sdk-windows\tools

Mobiles BS	Link für SDK-Download	Aufzu-zeichnen-der Emulator (Beispiel)	Befehlszeile (Beispiel)	Arbeits-verzeichnis (Beispiel)
Blackberry	Download für Blackberry-Emulator	C:\Programme\Research In Motion\BlackBerry Smartphone Simulators 6.0.0\6.0.0.337 (9800)\fledge.exe	/app=Jvm.dll /handheld=9800 /session=9800 /app-param=DisableRegistration /app-param=JvmAlxConfigFile:9800.xml /data-port=0x4d44 /data-port=0x4d4e /pin=0x2100000A (Siehe "Aufzeichnen mit Emulation" auf der vorherigen Seite)	C:\Programme\Research In Motion\BlackBerry Smartphone Simulators 6.0.0\60.0.337 (9800)
Windows 8	Download für Windows Mobile-Emulator	C:\Programme\Microsoft SDKs\Windows Phone\v7.0\Tools\XDE Launcher\XdeLauncher.exe	"Windows Phone 7" "Windows Phone 7-Emulator"	C:\Programme\Microsoft SDKs\Windows Phone\v7.0\Tools\XDE Launcher

Aufzeichnen mit einem Mac iPhone-Emulator

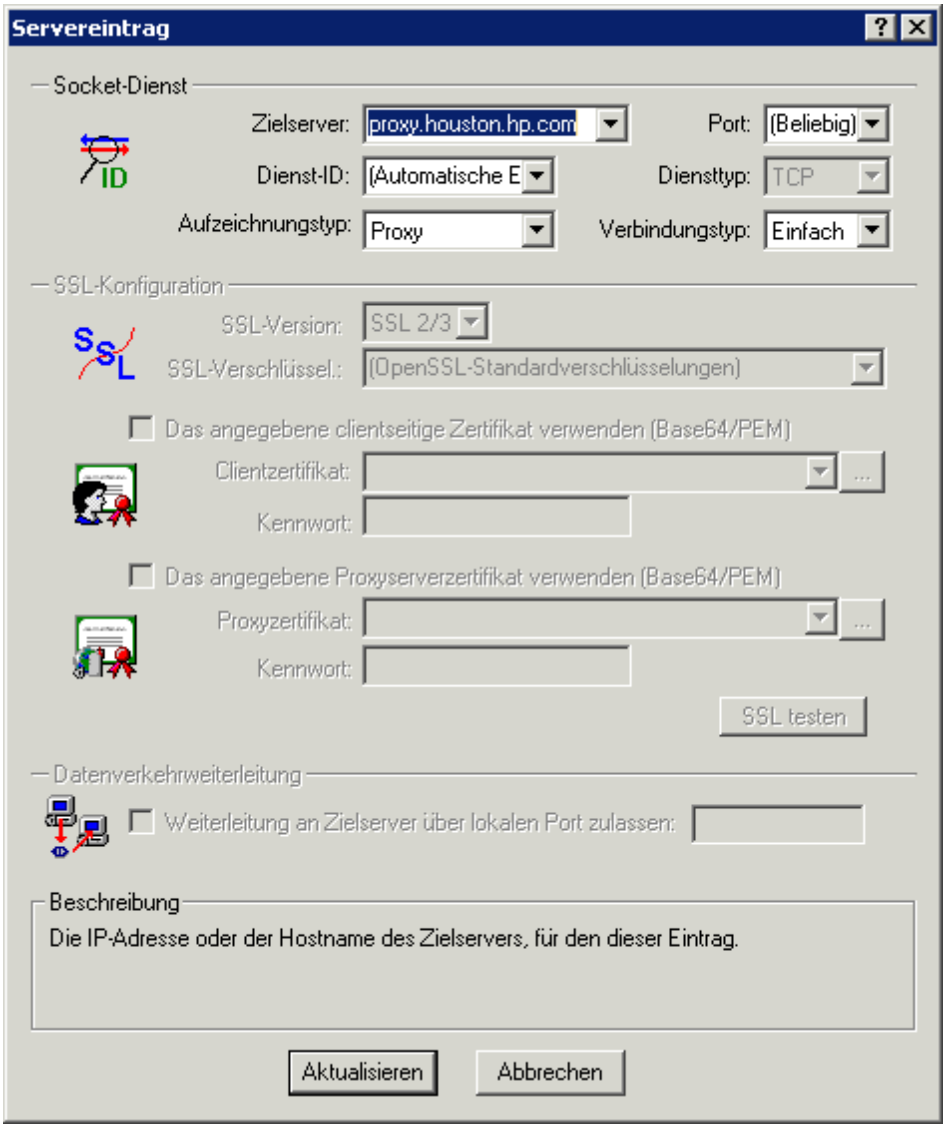
Sie können ein iPhone-Skript mithilfe eines Emulators aufzeichnen, indem Sie die Methode **Aufzeichnen und Analysieren** des Aufzeichnungs-Assistenten verwenden.

- Laden Sie den iPhone-Emulator herunter und installieren Sie ihn auf Ihrem Mac. Download für iPhone-Emulator.
- Verwenden Sie die Methode **Aufzeichnen und Analysieren** des Aufzeichnungs-Assistenten, um eine Aufzeichnungsdatei zu erstellen.

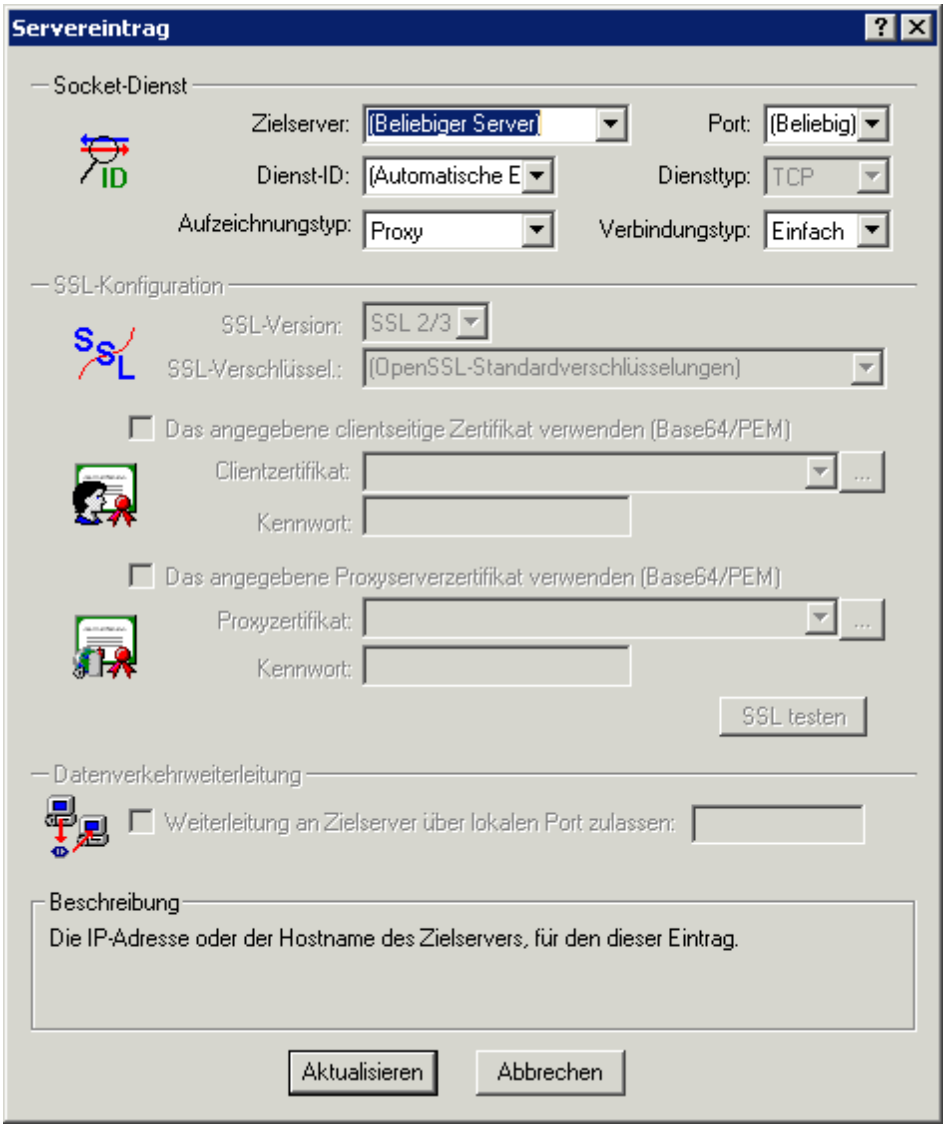
Aufzeichnen mit einem Google Android-Emulator

- Wenn Sie mit einem Google Android-Emulator Version 2.0 oder höher aufzeichnen, wenden Sie die folgende Umgehungslösung an, wenn Sie Probleme beim Aufzeichnen haben:

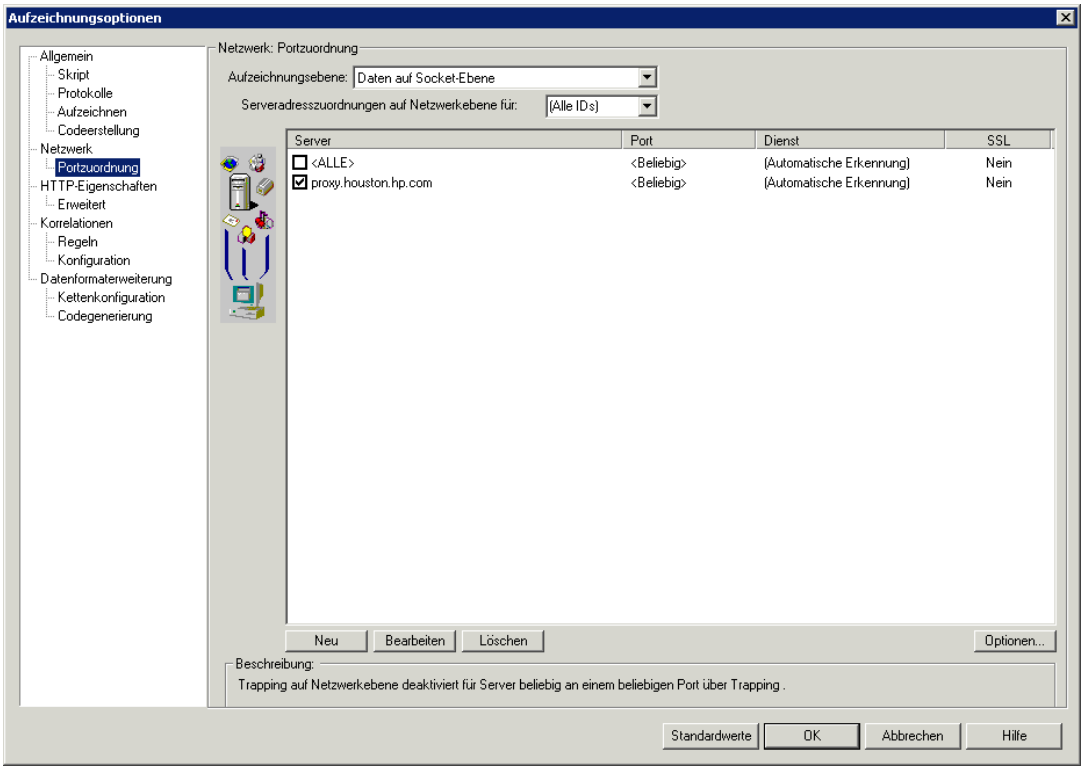
1. Geben Sie eine neue Portzuordnung an, indem Sie **Aufzeichnungsoptionen > Netzwerk > Portzuordnung** und anschließend **Neuer Eintrag** auswählen.
2. Geben Sie einen Zielserver und -port an.



3. Geben Sie einen zweiten Portzuordnungseintrag an, ohne irgendwelche Angaben zu ändern.



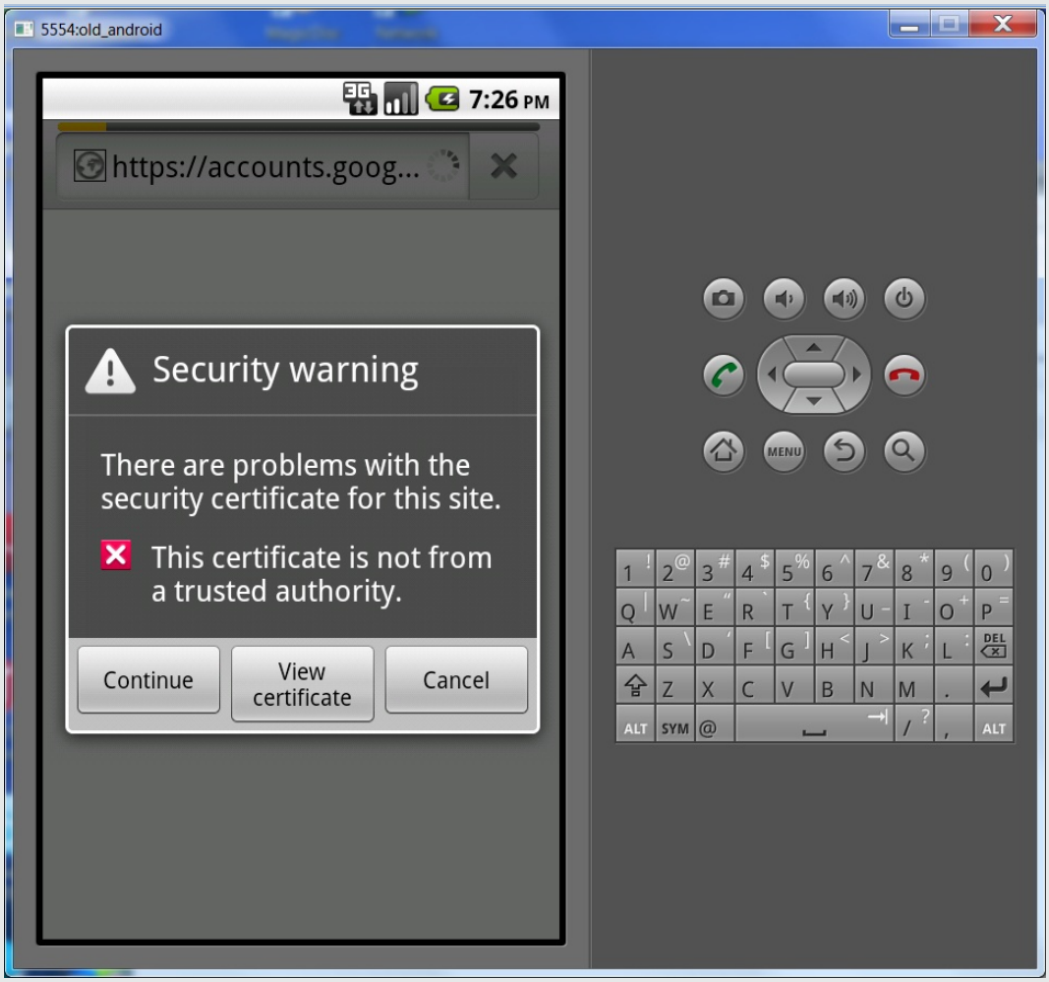
4. Deaktivieren Sie den zweiten Eintrag, damit sämtlicher Datenverkehr vom ersten Eintrag in der Portzuordnung abgewickelt wird.



Hinweis:

- Diese Einschränkung wurde im Google Android-Emulator Version 4.03 behoben.
- Während Sie eine SSL-Site aufzeichnen, wird möglicherweise eine Warnmeldung ausgegeben, die Sie darauf hinweist, dass es Probleme mit dem Zertifikat für die Site gibt.

Klicken Sie auf **Weiter**, um mit der Aufzeichnung fortzufahren.



Erstellen eines Skripts durch Analysieren des Datenverkehrs

In dieser Aufgabe wird die Erstellung eines Vuser-Skripts mithilfe einer Datei für den Netzwerkverkehr (Aufzeichnungsdatei) beschrieben. Sie können eine der folgenden Methoden verwenden, um eine Aufzeichnungsdatei zu erstellen, oder Sie können eine Aufzeichnungsdatei verwenden, die Sie mit einem externen Tool wie Fiddler oder Wireshark erzeugt haben.

1. **Erstellen einer Aufzeichnungsdatei auf einer Windows-Plattform - optional**

   Erstellen Sie eine Aufzeichnungsdatei, in der der gesamte TCP-Verkehr im Netzwerk auf einer Windows-Plattform protokolliert wird. Verwenden Sie ein Aufzeichnungstool (als Download

verfügbar) oder das im Ordner **bin** des Produkts bereitgestellte Tool **lrtcpdump.<Plattform>**.

a. Führen Sie das Erfassungsdienstprogramm in einem Befehlsfenster **lrtcpdump -f<Dateiname>.cap** aus. **lrtcpdump** fordert Sie auf, eine Netzwerkkarte auszuwählen.

b. Geben Sie die Nummer der gewünschten Schnittstellenkarte ein (falls mehrere vorhanden) und drücken Sie die EINGABETASTE.

c. Führen Sie typische Aktionen innerhalb Ihrer Anwendung aus.

d. Wechseln Sie wieder zur Eingabeaufforderung zurück und drücken Sie erneut die EINGABETASTE, um die Aufzeichnung zu beenden.

e. Legen Sie die Aufzeichnungsdatei im Netzwerk an einem Speicherort ab, auf den von VuGen aus zugegriffen werden kann.

2. **Erstellen einer Aufzeichnungsdatei mit HP LoadRunner Mobile Recorder (optional)**

Weitere Informationen finden Sie unter "Aufzeichnen mit Mobile Recorder" auf Seite 774.

3. **Erstellen einer Aufzeichnungsdatei auf einer Linux Redhat-Plattform (optional)**

a. Kopieren Sie den Ordner "LinuxRH3" vom LoadRunner-Installationsdatenträger **DVD\Additional Components\mobileRemoteAgent** auf den Linux-Computer.

b. Führen Sie folgende Befehle in der Shell aus, um den beiden Dateien Ausführungsberechtigungen unter Linux zuzuweisen.

i. chmod +x mongoose

ii. chmod +x cgi-bin/mobileCGI.cgi

c. Führen Sie einen **Verzeichniswechsel** in den Ordner **cgi-bin/** aus, und führen Sie folgende Befehle aus:

i. export SCRIPT_FILENAME=<Full path of mobileCGI.cgi>, wobei <Full path of mobileCGI.cgi> der Pfad zur Datei "mobileCGI.cgi" ist.

ii. `export QUERY_STRING=STARTRECORD=0`, wobei 0 die 0-basierte Sequenznummer der Netzwerkschnittstelle im Dialogfeld "Aufzeichnung starten" in VuGen ist.

iii. `./mobileCGI.cgi` Hiermit wird die Datei **currentPCAP.pcap** generiert, in der der gesamte Datenverkehr enthalten ist, der über die angegebene Ethernet-Schnittstelle erfasst wird.

Hinweis: Schritt iii dient in erster Linie zur Fehlerbeseitigung, um

sicherzustellen, dass die Linux-Umgebung korrekt konfiguriert wurde. Sobald dies sichergestellt ist, muss dieser Befehl nicht mehr ausgeführt werden.

d. Führen Sie den Befehl "Mongoose" mit den folgenden Optionen aus:

`mongoose -e errorLog.txt -r "/root/LinuxRH3" -C ".cgi" -p <port>` , wobei `/root/LinuxRH3` der Pfad ist, unter dem sich die Datei "mongoose" befindet.

Der Parameter "-p <port>" stellt den Port dar, den Mongoose abhört.

e. Sie können dann von VuGen aus eine Verbindung mit diesem Port herstellen.

4. **Erstellen einer Aufzeichnungsdatei auf einer UNIX-Plattform - optional**

Erstellen Sie eine Aufzeichnungsdatei, in der der gesamte TCP-Verkehr im Netzwerk auf einer UNIX-Plattform protokolliert wird.

a. Suchen Sie im Produktverzeichnis **bin** nach dem Dienstprogramm **lrtcpdump** für Ihre Plattform. Kopieren Sie es in ein Verzeichnis, auf das Sie von der UNIX-Maschine aus zugreifen können. Kopieren Sie z. B. bei einer HP-Plattform die Datei **lrtcpdump.hp9**. Stellen Sie bei Verwendung von FTP sicher, dass der binäre Übertragungsmodus aktiviert ist.

b. Wechseln Sie zum Erteilen der Ausführungsberechtigungen zum Root-Benutzer: chmod 755 lrtcpdump.<Plattform>

c. **lrtcpdump** verwendet bei Vorhandensein mehrerer Schnittstellenkarten die erste in alphabetischer Reihenfolge. Verwenden Sie den Befehl **ifconfig**, um alle vorhandenen Schnittstellenkarten aufzulisten.

d. Führen Sie das Dienstprogramm mit der kompletten Syntax unter Angabe von Schnittstelle und Dateiname aus. Beispielsweise **lrtcpdump.hp9 -ietho -f<Dateiname>.cap**. Die Aufzeichnung des Netzwerkverkehrs wird gestartet.

e. Führen Sie typische Aktionen innerhalb Ihrer Anwendung aus.

f. Wechseln Sie zum Befehlszeilenfenster zurück, in dem **lrtcpdump** ausgeführt wird, und folgen Sie den Anweisungen zum Beenden der Aufzeichnung.

g. Legen Sie die Aufzeichnungsdatei im Netzwerk an einem Speicherort ab, auf den von VuGen aus zugegriffen werden kann.

5. **Option 1 - Erstellen eines Skripts von Windows-Explorer aus**

Wählen Sie in Windows-Explorer die Datei **pcap**, **lrcap** oder **saz** (Fiddler) auf dem Computer aus, auf dem VuGen installiert ist. Wählen Sie im Kontextmenü die Option **VuGen-Skript**

erstellen aus. VuGen konvertiert die Aufzeichnungsdatei in ein Skript und öffnet dieses in VuGen.

6. **Option 2 - Erstellen eines Skripts in VuGen**
   a. Klicken Sie auf die Schaltfläche **Aufzeichnung starten**, um den Aufzeichnungs-Assistenten zu starten. Wählen Sie die Methode **Datenverkehr analysieren** aus.

   b. Wechseln Sie zur Aufzeichnungsdatei, und geben Sie den Abschnitt des Skripts an, in den Sie den Datenverkehr laden möchten: *vuser_init*, *Action* oder *vuser_end*.

   c. Geben Sie an, ob Sie server- oder clientseitigen Datenverkehr analysieren möchten. Weitere Informationen finden Sie unter "Analysieren von Datenverkehr" auf Seite 765.

   d. Um den Datenverkehr von einem sicheren Server zu analysieren, klicken Sie auf die Schaltfläche **SSL-Konfiguration**, um die zugehörigen Zertifikate hinzuzufügen. Weitere Informationen finden Sie unter "Dialogfeld "SSL-Konfiguration"" auf Seite 966.

   Weitere Details finden Sie unter "Aufzeichnungs-Assistent" auf Seite 776.

Hinweis: Wenn der Server Last aufweist, die verhindert, dass zeitnah Antworten empfangen werden, können Sie ein Skript erstellen, das auf Anforderungen basiert, indem Sie einfach die Registrierung bearbeiten.

Rufen Sie **regedit** auf, und fügen Sie an folgender Stelle einen DWORD-Schlüssel hinzu:

```
Pfad:
Software\\Mercury Interactive\\LoadRunner\\Protocols\\HTTP\\Analyzer\
Schlüssel:
AllowAutomaticOutApiEvents = 1
```

Aufzeichnen und Analysieren eines Skripts für mobile Anwendungen

In dieser Aufgabe wird beschrieben, wie Sie ein Skript mit dem Aufzeichnungs-Assistenten aufzeichnen und analysieren.

Aufzeichnen mit Mobile Recorder

In dieser Aufgabe wird beschrieben, wie Sie eine .lrcap-Datei für den Anwendungsverkehr direkt auf Ihrem mobilen Android-Gerät erstellen können.

1. **Vorbereiten des Mobilgeräts für die Aufzeichnung**
   a. Damit Sie den LoadRunner Mobile Recorder verwenden können, muss Ihr Mobilgerät einen **Stamm** erhalten.

Hinweis: Während die entsprechenden Geräte hierfür vom United States Copyright Office in den USA für rechtmäßig erklärt wurden, erlischt möglicherweise in bestimmten Fällen die Garantie. Es wird empfohlen, ein speziell dafür gedachtes Testgerät zu verwenden.

b. Herunterladen und Installieren von LoadRunner Mobile Recorder.

2. **Aufzeichnen des Geschäftsprozesses**

a. Tippen Sie auf dem Hauptbildschirm auf das -Symbol zum Öffnen der LoadRunner Mobile Recorder-Anwendung.

b. Tippen Sie auf die -Schaltfläche, um die Aufzeichnungssitzung zu starten.

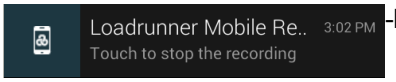
Hinweis: Die Schaltfläche wechselt zur Schaltfläche , und in der Benachrichtigungsleiste des Telefons wird ein Symbol für den LoadRunner Mobile Recorder geöffnet.

c. Drücken Sie auf die Startschaltfläche des Geräts, um zum Startbildschirm zu wechseln.

d. Öffnen Sie die gewünschte Anwendung für die Aufzeichnung und führen Sie die Schritte des Geschäftsprozesses durch.

e. Um die Aufzeichnung zu beenden, führen Sie einen der folgenden Schritte aus:

i. Drücken Sie auf die Startschaltfläche, und tippen Sie auf das -Symbol. Tippen Sie dann auf die Schaltfläche , um die Aufzeichnungssitzung anzuhalten.

ii. **Alternative:** Tippen Sie in der Benachrichtigungsleiste auf das Symbol für LoadRunner Mobile Recorder und wischen Sie über die 

-Benachrichtigung. Dadurch gelangen Sie automatisch zum Startbildschirm von LoadRunner Mobile Recorder. Tippen Sie auf die Schaltfläche .

3. **Exportieren der aufgezeichneten Sitzung**

a. Wählen Sie die zu exportierende Datei aus.

Sie können die standardmäßige .lrcap-Datei senden oder auf ... tippen, um eine zuvor aufgezeichnete .lrcap-Datei auszuwählen.

b. Tippen Sie auf die Schaltfläche Export, um eine Anwendung auszuwählen:

c. Wählen Sie im angezeigten Fenster die Anwendung aus, in die Sie die Datei exportieren möchten.

4. **Erstellen eines Vuser-Skripts**

a. Öffnen Sie die E-Mail, laden Sie die .lrcap-Datei herunter und doppelklicken Sie dann darauf. VuGen öffnet, erstellt und generiert das Skript automatisch.

Aufzeichnungs-Assistent

Mit diesem Assistenten können Sie Aufzeichnungen für das Protokoll Mobile Application - HTTP/HTML konfigurieren und Einstellungen analysieren.

Zugriff	Klicken Sie auf die Schaltfläche **Aufzeichnung starten**.
Wichtige Informationen	"Aufzeichnen von Datenverkehr in eine Aufzeichnungsdatei (Sniffer)" auf Seite 760 "Analysieren von Datenverkehr" auf Seite 765
Assistentennavigation: Datenverkehr aufzeichnen/analysieren	Im Folgenden wird die Assistentennavigation dargestellt, wenn Sie auf der Seite **Aufzeichnungsmethode** die Option **Aufzeichnen und Analysieren** auswählen. "Dialogfeld "Aufzeichnungsmethode"" auf der nächsten Seite > "Dialogfeld "Konfigurieren und Aufzeichnen"" auf Seite 779 > "Dialogfeld "Datenverkehr analysieren"" auf Seite 778
Assistentennavigation: Datenverkehr analysieren	Im Folgenden wird die Assistentennavigation dargestellt, wenn Sie auf der Seite **Aufzeichnungsmethode** die Option **Datenverkehr analysieren** auswählen. "Dialogfeld "Aufzeichnungsmethode"" auf der nächsten Seite > "Dialogfeld "Datenverkehr analysieren"" auf Seite 778
Assistentennavigation: Emulator aufzeichnen	Im Folgenden wird die Assistentennavigation dargestellt, wenn Sie auf der Seite **Aufzeichnungsmethode** die Option **Emulator aufzeichnen** auswählen. "Dialogfeld "Aufzeichnungsmethode"" auf der nächsten Seite > "Dialogfeld "Aufzeichnungsemulation"" auf Seite 779

Assistentennavigation: Proxy-Aufzeichnung	Im Folgenden wird die Assistentennavigation dargestellt, wenn Sie auf der Seite **Aufzeichnungsmethode** die Option **Proxy-Aufzeichnung** auswählen: "Dialogfeld "Aufzeichnungsmethode"" unten > "Dialogfeld "Proxy-Aufzeichnung"" auf Seite 780
Relevante Aufgaben	"Erstellen eines Skripts durch Analysieren des Datenverkehrs" auf Seite 771 "Aufzeichnen und Analysieren eines Skripts für mobile Anwendungen" auf Seite 774

Dialogfeld "Aufzeichnungsmethode"

In diesem Dialogfeld können Sie eine neue Aufzeichnungsmethode auswählen.

Zugriff	Schaltfläche **VuGen > Aufzeichnung starten**
Relevante Aufgaben	"Erstellen eines Skripts durch Analysieren des Datenverkehrs" auf Seite 771 "Aufzeichnen und Analysieren eines Skripts für mobile Anwendungen" auf Seite 774

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Datenverkehr analysieren	Erstellen Sie ein Skript, indem Sie eine vorhandene Aufzeichnungsdatei in ein Vuser-Skript konvertieren. Weitere Informationen finden Sie unter "Aufzeichnen von Datenverkehr in eine Aufzeichnungsdatei (Sniffer)" auf Seite 760.
Verkehr aufzeichnen und analysieren	Erstellen Sie ein Vuser-Skript, indem Sie Ereignisse mit einem auf dem VuGen-Computer installierten Geräteemulator aufzeichnen.
Emulator aufzeichnen	Erstellen Sie ein Vuser-Skript, indem Sie Ereignisse mit einem auf dem VuGen-Computer installierten Geräteemulator aufzeichnen.
Proxy-Aufzeichnung	Erstellen Sie ein Vuser-Skript, indem Sie eine Clientanwendung über den LoadRunner-Proxy aufzeichnen.

Dialogfeld "Datenverkehr analysieren"

In diesem Dialogfeld können Sie die Einstellungen für die Skripterstellung aus einer vorhandenen Aufzeichnungsdatei konfigurieren.

Zugriff	**VuGen > Aufzeichnungsmethode > Datenverkehr analysieren > Weiter**

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Aufzeichnungsdatei	Der Name der Aufzeichnungsdatei, die den Serverdatenverkehr enthält. VuGen unterstützt folgende Typen von Aufzeichnungsdateien: .cap, .pcap, .lrcap, .saz. Außerdem werden nur Dateien, bei denen der Kopfzeilentyp der Verbindungsschicht entweder `LINKTYPE_ETHERNET` oder `LINKTYPE_LINUX_SLL` lautet, von VuGen unterstützt. Wenn die Kopfzeile der Verbindungsschicht nicht einer der oben genannten ist, wird Ihre Datei möglicherweise nicht korrekt analysiert. Sie können den Kopfzeilentyp der Verbindungsschicht ermitteln, indem Sie die Aufzeichnungsdatei in WireShark öffnen und dann Statistics > Summary > Encapsulation aus dem Hauptmenü auswählen.
Durchsuchen	Ermöglicht es Ihnen, eine Aufzeichnungsdatei auszuwählen, die importiert werden soll.
Serverseitiger Filter	IP-Adresse und Port des Servers, dessen Verkehr analysiert werden soll. Es können mehrere Server angegeben werden. Sie können sowohl nach server- als auch client-seitigem Verkehr filtern.
Clientseitiger Filter	IP-Adresse des Clients, dessen Verkehr analysiert werden soll. Sie können sowohl nach server- als auch client-seitigem Verkehr filtern.
In Aktion aufzeichnen	Der Abschnitt für die Aufzeichnung: **vuser_init**, **Aktion** oder **vuser_end**. Verwenden Sie für zu wiederholende Aktionen den Abschnitt **Aktion**. Verwenden Sie für Initialisierungsschritte **vuser_init**.
Filteroptionen	Öffnet den Knoten **Filter für Datenverkehr** im Dialogfeld **Aufzeichnungsoptionen**. Auf diese Weise können Sie angeben, welche IP-Adressen bei der Skripterstellung einbezogen und welche ausgeschlossen werden sollen.
SSL-Konfiguration	Öffnet das Dialogfeld **SSL-Konfiguration**, in dem Sie SSL-Zertifikate zum Analysieren des Datenverkehrs von einem sicheren Server hinzufügen können.

Dialogfeld "Konfigurieren und Aufzeichnen"

Über dieses Dialogfeld können Sie die Einstellungen konfigurieren, um Datenverkehr in einer pcap-Datei aufzuzeichnen.

Zugriff	**Aufzeichnungs-Assistent > Aufzeichnungsmethode > Weiter**
Relevante Aufgaben	"Erstellen eines Skripts durch Analysieren des Datenverkehrs" auf Seite 771 "Aufzeichnen und Analysieren eines Skripts für mobile Anwendungen" auf Seite 774

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Speicherort des mobilen Sniffer-Agenten	**Host-URL.** URL, unter dem der mobile Sniffer-Agent gespeichert ist. **Port.** Geben Sie den Port an, den der mobile Sniffer-Agent abhören soll.
Konfiguration des mobiles Sniffers:	**Netzwerkschnittstelle aufzeichnen.** Server und Computer können über mehrere Netzwerkadapter verfügen. Mit dieser Einstellung können Sie den Netzwerkadapter auswählen, den der mobile Agent abhören soll. **In Aktion aufzeichnen.** Der Abschnitt für die Aufzeichnung: **vuser_init**, **Aktion** oder **vuser_end**. Verwenden Sie für zu wiederholende Aktionen den Abschnitt **Aktion**. Verwenden Sie für Initialisierungsschritte **vuser_init**.
Aufzeichnen	**Aufzeichnung starten.** Startet die Aufzeichnung von Datenverkehr. **Aufzeichnung beenden: Beendet die Aufzeichnung von Datenverkehr.**

Dialogfeld "Aufzeichnungsemulation"

In diesem Dialogfeld können Sie die Einstellungen konfigurieren, die ein mobiler Emulator zum Aufzeichnen eines Skripts verwenden soll.

Zugriff	**Aufzeichnungs-Assistent > Aufzeichnungsmethode > Weiter**

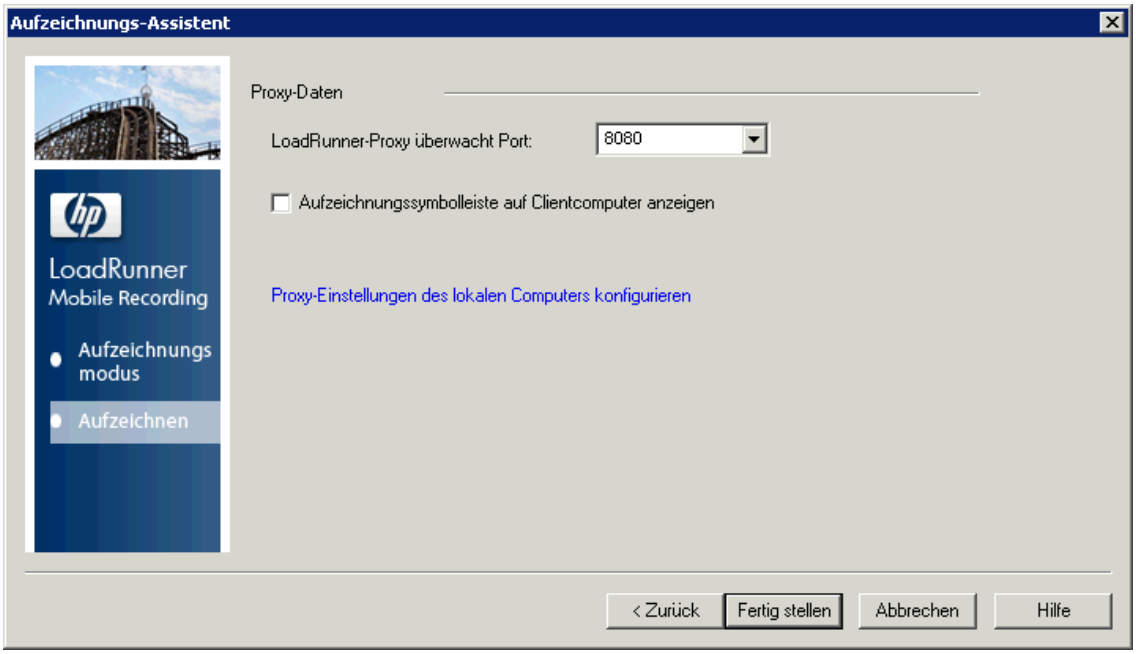
Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Aufzuzeichnender Emulator	Gibt den Speicherort der ausführbaren Datei der Emulatoranwendung an.

Element der Oberfläche	Beschreibung
Befehlszeile	Gibt die Argumente an, die die Emulatoranwendung zum Ausführen des Emulators benötigt.
Arbeitsverzeichnis	Gibt den Speicherort der Emulatoranwendung an.
Aktion für die Aufzeichnung	Geben Sie den Abschnitt für die Aufzeichnung an: **vuser_init**, **Aktion** oder **vuser_end**. Verwenden Sie für zu wiederholende Aktionen den Abschnitt **Aktion**. Verwenden Sie für Initialisierungsschritte **vuser_init**.
Schaltfläche "Optionen"	Greift auf den Bereich **Aufzeichnungsoptionen** zu.

Dialogfeld "Proxy-Aufzeichnung"

In diesem Dialogfeld können Sie Proxy-Einstellungen konfigurieren.



Zugriff	**VuGen > Aufzeichnungsmethode > Proxy-Aufzeichnung > Weiter**
Wichtige Informationen	"Aufzeichnung über einen Proxy - Übersicht" auf Seite 264
Relevante Aufgaben	"Aufzeichnen eines Skripts über einen Proxy" auf Seite 266

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
LoadRunner-Proxy überwacht Port:	Geben Sie den Port an, den der LoadRunner-Proxy überwachen soll, z. B. Port 8080.
Aufzeichnungssymbolleiste auf Clientcomputer anzeigen	Mit der Aufzeichnungssymbolleiste können Sie die Aufzeichnung Ihres Skripts auf dem Clientcomputer steuern.

LoadRunner Mobile Recording-Anwendung - Benutzeroberfläche

Diese Anwendung ermöglicht das Aufzeichnen des Anwendungsdatenverkehrs in einer Aufzeichnungsdatei (.lrcap) auf Ihrem mobilen Android-Gerät.

Zugriff	Tippen Sie auf das Symbol auf dem Hauptbildschirm Ihres Geräts.
Wichtige Informationen	• Ihr Mobilgerät muss einen **Stamm** aufweisen. • Herunterladen und Installieren von LoadRunner Mobile Recorder.
Relevante Aufgaben	"Aufzeichnen mit Mobile Recorder" auf Seite 774

Hauptbildschirm

Element der Oberfläche	Beschreibung
Aufzeichnen des Geschäftsprozesses	
	Aufzeichnung starten. Starten Sie eine Aufzeichnungssitzung.
	Aufzeichnung beenden. Beenden Sie eine Aufzeichnungssitzung.
Aufgezeichnete Sitzung senden	
	Durchsuchen. Wählen Sie eine zuvor aufgezeichnete Sitzung aus.

Element der Oberfläche	Beschreibung
Send to: John.Doe@gmail.com	**Senden an.** Geben Sie eine E-Mail-Adresse an, an die Sie die Datei senden möchten.
Export	**Senden.** Senden Sie die .lrcap-Datei an die angegebene E-Mail-Adresse.

Erweiterte Optionen

Element der Oberfläche	Beschreibung
E-Mail-Einstellungen	
Send automatically after recording	Option zum Senden der .lrcap-Datei an die angegebene E-Mail, nachdem Sie auf die Schaltfläche Stop Recording getippt haben.
Delete files automatically after sending	Option zum Löschen der .lrcap-Datei, nachdem Sie auf die Schaltfläche Export getippt haben.
Arbeitsordner	
Capture files are saved to this location	Option zum Auswählen eines Ordners auf dem Gerät, um die .lrcap-Datei zu speichern.
...	Wechseln Sie zum Ordner, in dem die .lrcap-Datei gespeichert werden soll.

Mobile TruClient

Mobile TruClient-Protokoll

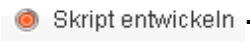
Mit TruClient Ajax - Mobile, das auf der innovativen TruClient-Technologie von LoadRunner basiert, können Sie Webanwendungen testen, die speziell für Mobilgeräte entwickelt wurden.

Mit diesem Protokoll können Sie folgende Aktionen durchführen:

- Simulieren verschiedener Browser für Mobilgeräte
- Entwickeln von Skripts, die auf der Benutzerebene aufgezeichnet werden, sodass sie klar und einfach verwaltet werden können

Folgende Abbildung zeigt den Workflow für die Verwendung des Mobile TruClient-Protokolls:

Aufzeichnen eines Skripts mit Mobile TruClient

1. Erstellen eines neuen Mobile TruClient-Skripts
2. **Auswählen der**  **-Schaltfläche**

   Definieren Sie Mobilgerät, Benutzer-Agent und Displaygröße im Dialogfeld **Mobile Einstellungen**, das automatisch geöffnet wird.

   Weitere Informationen finden Sie unter " Dialogfeld "TruClient-Mobilgeräte-Manager"" auf der nächsten Seite.
3. **Aufzeichnen des Geschäftsprozesses**

   Details zur Verwendung der TruClient-Funktionalität finden Sie unter Ajax TruClient Protocol.

Hinzufügen, Entfernen und Importieren von Mobilgeräteeinstellungen für Mobile TruClient

Erstellen eines benutzerdefinierten Geräts mit dem Mobilgeräte-Manager

Der TruClient Ajax - Mobile-Geräte-Manager enthält Einstellungen für viele verbreitete Mobilgeräte; Sie können aber problemlos benutzerdefinierte Geräte hinzufügen.

1. Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **TruClient Ajax - Mobile-Geräte-Manager**.
2. Wählen Sie **Mobilgerät hinzufügen** aus. Das Dialogfeld **Mobilgerät hinzufügen** wird geöffnet.
3. Geben Sie den Namen des Geräts ein, das Sie hinzufügen möchten, oder wählen Sie ein vorhandenes Gerät in der Dropdownliste aus, um es anschließend anzupassen.
4. Geben Sie den Benutzer-Agenten an. Weitere Informationen finden Sie unter " Dialogfeld "TruClient-Mobilgeräte-Manager"" unten.
5. Geben Sie das Display an. Weitere Informationen finden Sie unter " Dialogfeld "TruClient-Mobilgeräte-Manager"" unten.

Entfernen eines Mobilgeräts

1. Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **TruClient Ajax - Mobile-Geräte-Manager**.
2. Wählen Sie **Mobilgerät entfernen** aus. Das Dialogfeld **Mobilgerät entfernen** wird geöffnet.
3. Markieren Sie das nicht benötigte Gerät und klicken Sie auf **Entfernen**.

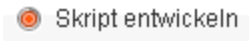
Importieren von Mobilgeräteeinstellungen in Ihr Skript

Sie können die Importfunktion verwenden, um in anderen Benutzerskripts erstellte Mobilgeräte zu importieren.

1. Öffnen Sie ein Skript, in dem die benutzerdefinierten Geräteeinstellungen enthalten sind.
2. Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **TruClient Ajax - Mobile-Geräte-Manager**.
3. Wählen Sie **Geräteeinstellungen importieren** aus. Das Dialogfeld **Geräteeinstellungen importiere** wird geöffnet.
4. Markieren Sie das benutzerdefinierte Gerät und wählen Sie **Importieren** aus.

Dialogfeld "TruClient-Mobilgeräte-Manager"

Mit diesem Dienstprogramm können Sie mobile Geräteeinstellungen, die mit dem TruClient Ajax - Mobile-Protokoll verwendet werden, konfigurieren, hinzufügen und importieren.

Zugriff	Führen Sie einen der folgenden Schritte durch: • Schaltfläche **Skriptentwicklung**  • **Extras > TruClient-Mobilgeräte-Manager** • Schaltfläche **Mobile Einstellungen**
Relevante Aufgaben	"Hinzufügen, Entfernen und Importieren von Mobilgeräteeinstellungen für Mobile TruClient" auf Seite 783 "Aufzeichnen eines Skripts mit Mobile TruClient" auf Seite 783

Element der Oberfläche	Beschreibung
Mobilgerät	Wählen Sie den mobilen Gerätetyp aus, den Sie testen möchten.
Benutzer-Agent:	Geben Sie die Kopfzeilenzeichenfolge an, die an den Server gesendet wird, um das Mobilgerät zu identifizieren. Nach der Auswahl eines Geräts wird der Standard-Kopfzeilenwert angezeigt. Diese Kopfzeile kann jedoch geändert werden.
Anzeigen	Geben Sie die Höhe und Breite des Mobilgerätebildschirms an. Mobile TruClient öffnet ein Browserfenster entsprechend den Displayeinstellungen.

SAP Mobile Platform

Erstellen eines SMP (SAP Mobile Platform)-Skripts

Mithilfe des SMP (SAP Mobile Platform)-Protokolls können Sie .NET-basierte Skripts mit von SMP (zuvor SUP) generierten Dateien erstellen und wiedergeben. In dieser Aufgabe werden die zum Erstellen eines SMP-Skripts auszuführenden Schritte beschrieben.

Voraussetzungen

Zum Erstellen eines SMP-Skripts müssen Sie einen Geschäftsprozess mit SMP aufzeichnen. SMP ist eine von dem SAP-Unternehmen Sybase bereitgestellte Plattform. Beim Aufzeichnen werden folgende Dateien generiert:

- Action.cs
- Vuser_init.cs
- Vuser_end.cs

Erstellen eines SMP (SAP Mobile Platform)-Skripts

1. Wählen Sie **Neues Skript und neue Lösung > SMP (SAP Mobile Platform)** aus.
2. Kopieren Sie die generierten CS-Dateien (die unter "Erstellen eines SMP (SAP Mobile Platform)-Skripts" auf der vorherigen Seite beschrieben werden) in den Skriptordner.

   Hinweis: Der Aufzeichnungsmechanismus für SMP-Skripts ist deaktiviert.

3. Speichern und schließen Sie das Skript, und öffnen Sie es dann erneut.
4. Aktualisieren Sie den Speicherort der "SAP.Mobile.LoadRunner.dll" unter **Laufzeiteinstellungen > .Net > Gemeinsam genutzte DLLs**.

   Hinweis: Die "SAP.Mobile.LoadRunner.dll" wurde erstellt und wird von dem SAP-Unternehmen Sybase verwaltet.

5. Generierte .cs-Dateien können Objekte aus externen DLLs enthalten. Für eine einwandfreie Skriptwiedergabe müssen Sie diese .dlls unter **Laufzeiteinstellungen > .Net > Gemeinsam genutzte DLLs** referenzieren.
6. Spielen Sie das Skript ab. Weitere Informationen finden Sie unter "Debuggen von .NET-Vuser-Skripts" auf Seite 795.

.NET-Protokoll

Übersicht über das .NET-Protokoll

Microsoft .NET Framework dient als Grundlage für Entwickler, um verschiedene Arten von Anwendungen zu erstellen, z. B. ASP.NET, Windows Forms, Webdienste, verteilte Anwendungen oder Anwendungen, bei denen mehrere dieser Modelle kombiniert werden.

VuGen unterstützt .NET als Protokoll auf Anwendungsebene. Mit VuGen können Sie Vuser-Skripts erzeugen, mit denen Benutzer von Microsoft .NET-Clientanwendungen, die im .NET-Framework erstellt wurden, emuliert werden. VuGen zeichnet alle Clientaktionen durch Methoden und Klassen auf und erstellt Vuser-Skripts in C Sharp oder VB .NET.

Die VuGen-Umgebung ist standardmäßig für .NET Remoting-, ADO.NET-, Enterprise Services- und WCF-Anwendungen (Windows Communication Foundation) konfiguriert. Informationen zum Konfigurieren von VuGen für die Aufzeichnung von Anwendungen mit anderer Client-Server-Aktivität erhalten Sie vom Customer Support.

Hinweise zum Arbeiten mit dem .NET-Protokoll

Das .NET-Protokoll ermöglicht Ihnen Lasttests durch die Wiedergabe von Methodenaufrufen der Anwendung.

Sie können ein Lasttestskript manuell erstellen oder eines generieren, indem Sie einen Geschäftsprozess aufzeichnen.

Im Gegensatz zu anderen transportbasierten LoadRunner-Protokollen zeichnet das .NET-Protokoll die im Filter angegebenen Methodenaufrufe der Anwendung auf. Methodenaufrufe, die nicht im Filter definiert sind, werden nicht während der Aufzeichnung der Anwendung in das generierte Skript aufgenommen.

In der Regel kann ein Benutzer anhand des standardmäßigen Umgebungsfilters ein Skript generieren, das für Lasttests verwendet werden kann. Für bestimmte komplexe Anwendungen kann es jedoch schwierig sein, ein funktionierendes Skript zu generieren, da im Filter die falsche Methode angegeben wurde. Die schwierigste Aufgabe beim Erstellen eines Lasttestskripts mit dem .NET-Protokoll ist die Behebung von Aufzeichnungs- oder Codegenerierungsfehlern.

Wenn Sie die folgenden Anforderungen erfüllen, können Sie problemlos den richtigen Filter für den Aufzeichnungsprozess definieren und ein funktionierendes Lasttestskript generieren:

- Sie sollten mit dem .NET-Framework vertraut sein.
- Sie sollten in der Lage sein, Code mit C# oder VB.NET zu generieren.
- Sie sollten mit XML vertraut sein.
- Sie sollten Visual Studio 2010 bereits installiert haben.
- Sie sollten über Kenntnisse der Architektur und der Kommunikationstechniken der Anwendung verfügen, damit Sie feststellen können, welche Funktionen oder Klassen für das Lasttestskript relevant sind.

Folgende Punkte können den Vorgang zum Erstellen des richtigen Filters optimieren:

Sie sollten Zugriff auf den Anwendungscode haben oder bestimmte .NET-Reflektortools besitzen, mit denen Sie den dekompilierten Code anzeigen können.

Sie sollten Zugang zu den Entwicklern der Anwendung haben, die Ihnen helfen können, die für den Lasttest erforderlichen Methoden zu ermitteln.

Weitere Informationen über .NET und die oben genannten Umgebungen finden Sie auf der MSDN-Website.

Anzeigen von Daten und Datentabellen

Beim Aufzeichnen einer Methode, die Daten, eine Datentabelle oder eine Datenleseaktion zurückgibt, generiert VuGen eine Tabelle zum Anzeigen der Daten.

Bei Verwendung einer Datenleseaktion erfasst VuGen die Daten, die durch jeden Lesevorgang abgerufen werden, und konvertiert sie in die Wiedergabehilfefunktion **DoDataRead**.

Beispiel: Der folgende Anwendungscode wird aufgezeichnet.

```
SqlDataReader reader = command.ExecuteReader();
     while( reader.Read() )
       {
           // read the values, e.g., get the string located in column 1
           string str = reader.GetString(1)
       }
```

Anschließend generiert VuGen im Skript die folgenden Zeilen.

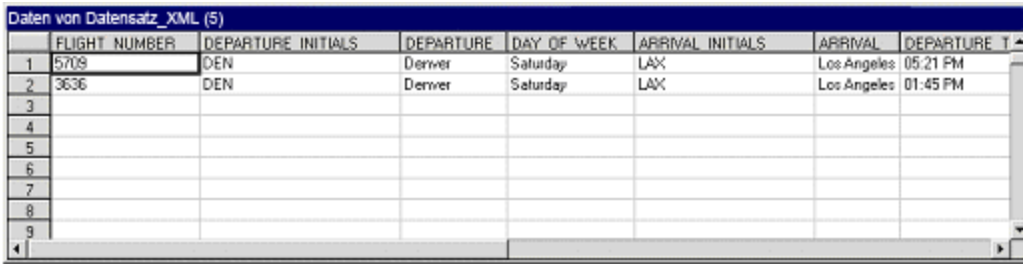
```
SqlDataReader_1 = SqlCommand_1.ExecuteReader();
LrReplayUtils.DoDataRead(SqlDataReader_1, out valueTable_1, true, 27);
```

Die beiden Parameter zeigen an, dass die Anwendung während der Aufzeichnung alle 27 verfügbaren Datensätze gelesen hat. Daher liest das Skript bei der Wiedergabe alle verfügbaren Datensätze.

Darüber hinaus generiert VuGen eine Datentabelle mit allen Informationen, die durch die Lesevorgänge abgerufen werden.

bei der Wiedergabe können Sie die Ausgabedatentabelle mit den tatsächlichen abgerufenen Werten für Korrelation und Überprüfung verwenden. Weitere Informationen zur Funktion **DoDataRead** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Gegebenenfalls zeigt VuGen die Rasterschritte im Schrittenavigator und die zugehörigen Raster im Snapshot-Bereich an.



Die Daten werden in einer XML-Datei gespeichert. Diese XML-Datei können Sie im Datenordner des Skripts anzeigen. Die Datendateien weisen die Bezeichnung <*Indexname*>.xml auf, z. B. 20.xml. Da jede Datei mehrere Datentabellen enthalten kann, können Sie in der Datei **datasets.grd** nachsehen, welcher Skriptindex welchem Dateiindex zugeordnet ist, um die XML mit den entsprechenden Daten zu bestimmen.

Aufzeichnen von WCF-Duplexkommunikation

Bei WCF (Windows Communication Foundation) handelt es sich um ein Programmierungsmodell, das Webdienste, .NET Remoting, Microsoft Distributed Transaction Coordinator und Microsoft Message Queuing in einem einzigen dienstorientierten Programmierungsmodell für verteilte Verarbeitung vereint.

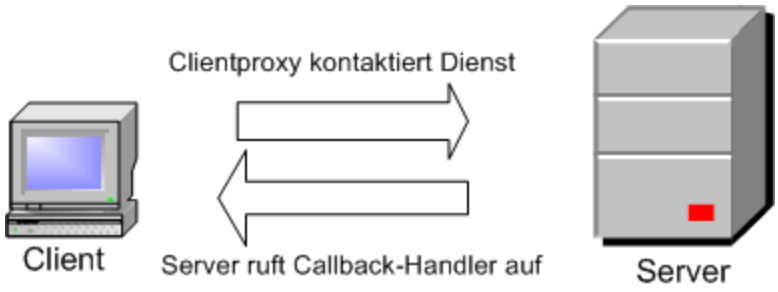
WCF erzeugt ein Proxy-Objekt, um Daten für den Dienst bereitzustellen. Außerdem sorgt WCF dafür, dass die vom Dienst zurückgegebenen Daten in das vom Aufrufer erwartete Formular eingetragen werden.

Neben der allgemeinen Unterstützung für die WCF-Umgebung bietet VuGen auch spezielle Unterstützung für Anwendungen, die WCF-Duplexkommunikation nutzen. Bei Duplexkommunikation kontaktiert der Clientproxy den Dienst und der Dienst ruft den Callback-Handler auf dem Clientcomputer auf. Der Callback-Handler implementiert eine vom Server definierte Callbackschnittstelle. Der Server muss nicht synchron antworten, sondern bestimmt unabhängig, wann die Antwort erfolgt und der Callback-Handler aufgerufen wird.

Kommunikation zwischen Client und Server

Die Kommunikation zwischen Client und Server läuft wie folgt ab:

- Der Server definiert den Dienstvertrag und eine Schnittstelle für das Callback.
- Der Client implementiert die vom Server definierte Callbackschnittstelle.
- Der Server ruft den Callback-Handler auf dem Client bedarfsabhängig auf.



Beim Aufzeichnen und Wiedergeben von Duplexkommunikation treten möglicherweise Probleme auf, wenn das Skript die ursprünglichen Callbackmethoden aufruft. Die Callback-Handler sind standardmäßig nicht im Filter enthalten. Sie könnten die Filter anpassen, um die Callback-Handler darin aufzunehmen. Dadurch würde jedoch die Standardwiedergabe für Lasttests ineffektiv, weil es sich bei vielen Callbacks um lokale Operationen wie z. B. GUI-Aktualisierungen handelt. Bei einem effektiven Lasttest können Sie die ursprüngliche, vom Server aufgerufene Callbackmethode nicht wiedergeben.

Bei der VuGen-Lösung wird der ursprüngliche Callback-Handler durch eine Dummy-Implementierung ersetzt. Diese Implementierung führt eine typische Reihe von Aktionen aus, die Sie für die Anwendung weiter anpassen können.

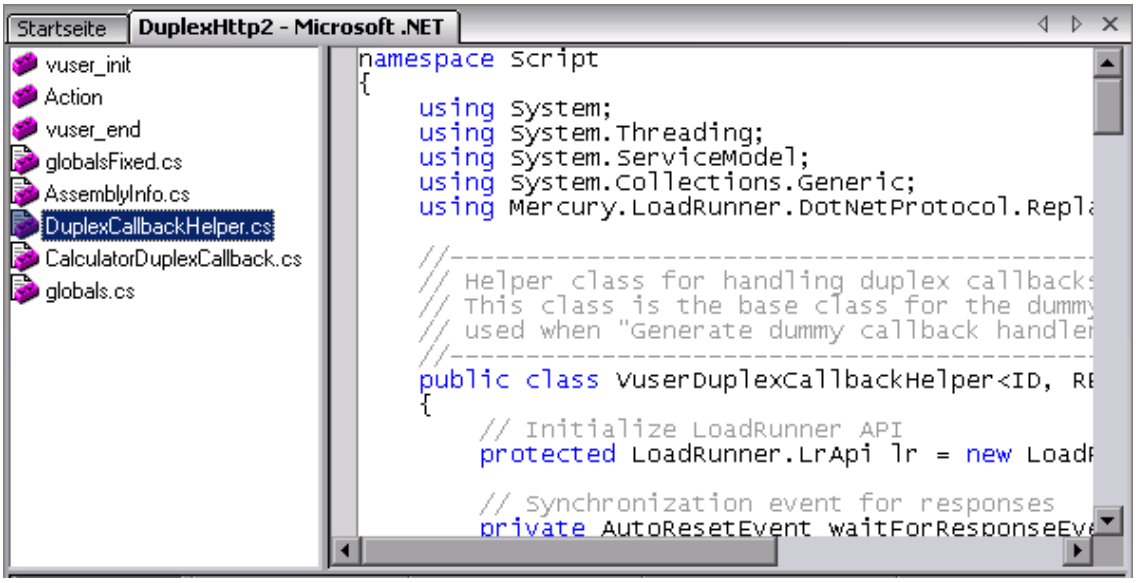
Damit VuGen die ursprünglichen Callbacks ersetzt, aktivieren Sie die Aufzeichnungsoption **Dummy-Callback-Handler generieren**. Weitere Informationen finden Sie unter "Eigenschaft "Remoteobjekte"" auf Seite 221.

VuGen-Implementierung eines Duplex-Callbacks

Bei der Lösung für Duplexkommunikation generiert VuGen zwei Unterstützungsdateien:

- `DuplexCallbackHelper.<`*Sprache*`>`
- *Rückrufname*`.<`*Sprache*`>`

Das folgende Beispiel zeigt die generierten Dateien für eine Calculator-Anwendung mit Duplexkommunikation:



Die Datei mit der Bezeichnung **Helper** dient als allgemeine Vorlage für das Arbeiten mit Duplex-Callback-Handlern. Sie stellt eine Basisklasse für die Implementierung des Callbacks bereit.

Die zweite Datei **Rückrufname** enthält die Implementierung des Callbacks. Der Name der Callback-Implementierungsklasse lautet **Vuser**<*xxxx*>. Dabei steht *xxxx* für den Namen der Callbackschnittstelle und die Klasse erbt von der in der Datei **Helper** definierten Klasse **VuserDuplexCallbackHelper**. VuGen erstellt separate Implementierungsdateien für jede Schnittstelle.

Diese Datei übernimmt zwei primäre Aufgaben:

Festlegen der Antwort. Die vom Server erhaltenen Daten werden in einem Adressraum gespeichert. Beim Speichern werden sequentielle IDs zugewiesen, um das Abrufen zu vereinfachen. Diese Methode wird von der Implementierung der Callbackschnittstelle aufgerufen. Der folgende Beispielcode zeigt die Dummy-Implementierung einer Rückrufmethode mit der Bezeichnung **Ergebnis**. Die Argumente der Methode werden im Adressraum als Objektarray gespeichert.

```
//  ---------------------------------------------------------------------------
        public virtual void Result(string operation, double result) {
            // Add here your own callback implementation and set the response d
ata
            SetResponse(responseIndex++, new object[] {
                        operation,
                        result});
}
```

- **Abrufen der Antwort.** Warten auf das Eintreffen der nächsten Antwort. Durch die Implementierung von **GetNextResponse** wird die nächste im Adressraum gespeicherte Antwort über den sequentiellen Index abgerufen; alternativ wird auf das Eintreffen der nächsten Antwort gewartet.

GetNextResponse steht im Skript an der Stelle, an der der ursprüngliche Callback-Handler während der Aufzeichnung aufgerufen wurde. An dieser Stelle gibt das Skript eine Warnung aus:

```
// Wait here for the next response.
// The original callback during record was:…
```

Ersetzen des Callbacks im Skript

Wenn Sie die Dummy-Rückruf-Option aktivieren (standardmäßig aktiviert), ersetzt VuGen die ursprünglichen Duplex-Rückrufhandler durch Dummy-Implementierungen. Die Dummy-Implementierung erhält die Bezeichnung **Vuser <*Rückrufname*>**. An der Position des ursprünglichen Callback-Handlers wird im Skript mit einer Warnung auf das Ersetzen hingewiesen.

Anpassen der Dummy-Implementierung

Sie können die Implementierungsdatei so ändern, dass sie Ihrer Umgebung entspricht. Dieser Abschnitt enthält mehrere Vorschläge für Anpassungen.

Timeouts

Die standardmäßige Zeitüberschreitung für das Warten des Callbacks auf die nächste Antwort beträgt 60000 Millisekunden bzw. eine Minute. Wenn Sie eine bestimmte Zeitüberschreitung verwenden möchten, ersetzen Sie den Aufruf an **GetNextResponse** durch die Überlastungsmethode, die die Zeitüberschreitung wie unten gezeigt als Argument abruft. Diese Methode ist in der Callback-Implementierungsdatei <*Rückrufname*> enthalten, die im linken Bereich nach der Datei **DuplexCallbackHelper** aufgelistet ist.

```
// Get the next response.
        // This method waits until receiving the response from the server
        // or when the specified timeout is exceeded.
        public virtual object GetNextResponse(int millisecondsTimeout) {
            return base.GetResponse(requestIndex++, millisecondsTimeout);
        }
```

Zum Ändern des standardmäßigen Schwellenwerts für alle Callbacks bearbeiten Sie die Datei **DuplexCallbackHelper**.

```
 // Default timeout threshold while waiting for response
        protected int millisecondsTimeoutThreshold = 60000;
```

Schlüsselkennung

Viele Anwendungen weisen den Daten bestimmte Schlüsselkennungen zu, um Anfrage und Antwort zu verbinden. Dadurch können Sie die Daten anhand ihrer ID aus dem Adressraum abrufen, anstatt den integrierten inkrementellen Index zu nutzen. Wenn Sie eine Schlüsselkennung statt des Index verwenden möchten, ändern Sie die Datei <*Rückrufname*>, indem Sie den ersten Parameter **named ID** der Basisvorlage durch den Typ Ihrer Schlüsselkennung ersetzen. Wenn Sie z. B. als Schlüsselkennung eine Zeichenfolge nutzen, ändern Sie das erste Vorlagenargument von **int** in **string**:

```
public class VuserXXX : VuserDuplexCallbackHelper<string, object>
```

Zusätzlich können Sie die Implementierung von **GetNextResponse()** entfernen und durch Aufrufe an **GetResponse(ID)** ersetzen, wie in der Basisklasse definiert.

Rückgabewerte

Da VuGen *One-Way*-Kommunikation unterstützt, gibt das Implementierungscallback standardmäßig keinen Wert zurück und aktualisiert keinen Ausgabeparameter, wenn es aufgerufen wird.

```
 public virtual void Result(string operation, double result) {
            // Add here your own callback implementation and set the response da
ta
```

Wenn die Anwendung die Rückgabe eines Werts durch das Callback erfordert, fügen Sie die Implementierung an dieser Stelle ein.

Reihenfolge des Antwortempfangs

In der VuGen-Implementierung sorgt eine Sperrmethode dafür, dass auf jede Antwort gewartet wird. Dadurch wird die tatsächliche Reihenfolge der Ereignisse während der Aufzeichnung wiedergegeben, d. h., der Server hat mit Daten geantwortet. Sie können dieses Verhalten so ändern, dass das Warten auf eine Antwort entfällt oder die Sperre erst nach Abschluss des Geschäftsprozesses wirksam wird.

Portsuche

Bei der Methode **FindPort** in der Datei **Helper** handelt es sich um ein praktisches Dienstprogramm, das sich für verschiedenste Implementierungen eignet. Die Klasse **Helper** sucht mit dieser Methode nach eindeutigen Ports für die Ausführung mehrerer Skriptinstanzen. Sie können diese Dienstprogrammmethode für weitere benutzerdefinierte Implementierungen verwenden.

Aufzeichnen von Anwendungen, wenn der Server auf einem Client gehostet wird

Wenn die Kommunikation in Ihrem System über einen Server erfolgt, der auf einem Client gehostet wird, bleibt die VuGen-Standardlösung für Duplexkommunikation wirkungslos. In Umgebungen, in denen der Server auf einem Client gehostet wird, findet keine echte Duplexkommunikation statt, da der Client den Dienst eröffnet und nicht durch das Framework kommuniziert. Bei Warteschlangen sendet der Client z. B. eine Meldung an den Dienst und eröffnet eine Antwortwarteschlange, um die Antworten zu erfassen.

Zum Emulieren eines Servers, der auf einem Client gehostet wird, verwenden Sie das Muster aus der zuvor beschriebenen Lösung. Ersetzen Sie dabei die ursprünglichen Antwortwarteschlangen durch Dummy-Callbacks und führen Sie bei Bedarf eine Synchronisierung durch. Weitere Informationen erhalten Sie vom HP Support.

Asynchrone Aufrufe

Wenn VuGen asynchrone Aufrufe von Remoteobjekten aufzeichnet, können Sie angeben, wie die Aufrufe im ".NET > Aufzeichnen - Knoten "Aufzeichnen"" auf Seite 219 verarbeitet werden sollen. Diese Optionen sind insbesondere für .NET Remoting- und WCF-Umgebungen relevant.

In diesem Fall haben Sie folgende Optionen für die Konfiguration von VuGen:

- **Ursprüngliche Callbacks standardmäßig aufrufen.** Beim Generieren und Wiedergeben des Skripts wird das ursprüngliche Callback der aufgezeichneten Anwendung verwendet. Wenn die Callback-Methode durch einen Filter explizit ausgeschlossen wird, gilt dieser Ausschluss selbst bei Aktivieren dieser Option. Wenn die Callbacks zu Aktionen führen, die nicht direkt mit dem Geschäftsprozess verbunden sind (z. B. Aktualisieren der Benutzeroberfläche), müssen Sie diese Option deaktivieren.

- **Asynchrone Callbacks generieren.** Diese Option bestimmt, wie Callbacks von VuGen verarbeitet werden, wenn die ursprünglichen Callbacks nicht aufgezeichnet wurden. Dies spielt eine Rolle, wenn die zuvor genannte Option **Ursprüngliche Callbacks aufrufen** deaktiviert ist oder wenn die Callbacks explizit ausgeschlossen wurden.

  Durch Aktivieren dieser Option wird eine Dummy-Methode erstellt, die bei der Wiedergabe statt dem ursprünglichen Callback aufgerufen wird. Dieses Dummy-Callback wird im Abschnitt **callbacks.cs** des Skripts generiert.

  Durch Deaktivieren dieser Option fügt VuGen einen NULL-Wert für das Callback ein und zeichnet die Ereignisse so auf, wie sie eintreten.

Das folgende Segment zeigt die Skriptgenerierung für einen Calculator-Client bei aktivierter Option **Asynchrone Rückrufe generieren**.

Zum Anzeigen der Callback-Methode **OnComplete1** klicken Sie im linken Bereich auf die Datei **callback.cs**.

Das folgende Segment zeigt die Skriptgenerierung, wenn die Option deaktiviert ist. VuGen generiert anstelle des Callbacks eine NULL und zeichnet die Ereignisse des Callbacks auf, wie sie eintreten.

Hinweis: Wenn Sie ein Skript mit bestimmten Aufzeichnungsoptionen erfasst haben und diese ändern möchten, müssen Sie das Skript nicht erneut aufzeichnen. Stattdessen können Sie das Skript mit den neuen Einstellungen erneut generieren.

Weitere Informationen finden Sie unter ".NET > Aufzeichnen - Knoten "Aufzeichnen"" auf Seite 219.

Aufzeichnen von dualen HTTP-Bindungen

Wenn die Anwendung duale HTTP-Bindung nutzt, gilt Folgendes: Da HTTP an sich kein Duplexprotokoll ist, verwendet das Framework einen Standardport für den Empfang von Antwortdaten, die an das Callback übergeben werden. Das Ausführen mehrerer Instanzen der Anwendung über dieselbe Portnummer schlägt möglicherweise fehl. VuGen bietet eine Option, mit der Sie die ursprüngliche Portnummer der Client-Basisadresse durch einen eindeutigen Port ersetzen können.

Wenn Sie die Aufzeichnungsoption **Eindeutige Client-Basisadresse generieren** aktivieren, prüft VuGen, welchen Kommunikationstyp die Anwendung nutzt. Falls duale HTTP-Kommunikation bzw. **WSDualHttpBinding** erkannt wird, führt VuGen das Dienstprogramm **FindPort** (in LrReplayUtils bereitgestellt) in der Datei **Helper** aus, um eindeutige Ports für jede Callbackinstanz zu ermitteln.

Diese Option ist standardmäßig aktiviert. Sie ist nur relevant, wenn Sie die oben genannte Option **Dummy-Rückrufhandler generieren** aktivieren.

Durch das Aktivieren dieser Option generiert VuGen den folgenden Code im Skript:

```
#warning:  Code Generation Warning
    // Override the original client base address with a unique port number
    DualProxyHelper.SetUniqueClientBaseAddress<XXXX>(YYYYY);
```

Weitere Informationen finden Sie unter ".NET > Aufzeichnen - Knoten "Aufzeichnen"" auf Seite 219.

Verbindungspooling

ADO.NET-Anbieter stellen eine Funktion für **Verbindungspooling** bereit, die die Genauigkeit von Lasttests erheblich beeinflussen kann. Wird nur eine Anwendungsdomäne für alle Vuser verwendet, ist das Verbindungspooling eingeschaltet, d. h., .NET Framework sorgt dafür, dass die Datenbankverbindungen geöffnet bleiben und wiederverwendet werden, wenn eine neue Verbindung angefragt wird. Da viele Vuser im Kontext einer einzigen Anwendungsdomäne ausgeführt werden, entstehen möglicherweise Konflikte zwischen den Vusern. Die mangelnde Linearität ihres Verhaltens kann die Genauigkeit beeinträchtigen.

In den .NET-Laufzeiteinstellungen ermöglicht die Eigenschaft **Anwendungsdomäne pro Vuser** die Ausführung der einzelnen Vuser in einer gesonderten Anwendungsdomäne (standardmäßig aktiviert). Dies ermöglicht Verbindungspooling im Bereich jedes Vusers, ohne dass Konflikte zwischen Vusern auftreten können. Durch diese Einstellung steigt die Genauigkeit, doch die Skalierbarkeit sinkt.

Wenn Sie diese Option deaktivieren, müssen Sie das Verbindungspooling für die Datenbank manuell ausschalten.

In der folgenden Tabelle wird das manuelle Deaktivieren von Verbindungspooling beschrieben:

Provider	Option
.NET Framework-Datenanbieter für SQL Server	"Pooling=false" oder "Pooling=no"
.NET Framework-Datenanbieter für Oracle	"Pooling=false" oder "Pooling=no"

.NET Framework-Datenanbieter für ODBC	Das Verbindungspooling wird über einen ODBC-Treibermanager verwaltet. Zum Aktivieren oder Deaktivieren von Verbindungspooling verwenden Sie den ODBC-Datenquellenadministrator (Zugriff über die Systemsteuerung oder den Ordner für Administrationstools). Auf der Registerkarte für das Verbindungspooling können Sie für jeden installierten ODBC-Treiber die Parameter für das Verbindungspooling festlegen.
.NET Framework-Datenanbieter für OLE DB	"OLE DB Services=-2"
Oracle-Datenanbieter für .NET	"Pooling=false"
ADO.NET-Datenanbieter für Adaptive Server Enterprise	"Pooling=False"

Debuggen von .NET-Vuser-Skripts

Sie können ein .NET-Vuser-Skript kompilieren, um seine Syntax ohne Ausführung des Skripts zu überprüfen. Um das Skript direkt in VuGen zu kompilieren, drücken Sie Umschalttaste + F5 oder wählen **Vuser > Kompilieren** aus. Die von VuGen erkannten Kompilierungsfehler werden im Ausgabefenster angezeigt. Doppelklicken Sie auf den Fehler, um die problematische Zeile im Skript anzuzeigen.

Um das Skript direkt in VuGen auszuführen, drücken Sie F5 oder wählen Sie **Wiedergeben > Ausführen** aus. Breakpoints und Wiedergabe in Einzelschritten werden für .NET-Vuser im Editorfenster von VuGen nicht unterstützt. Zum Debuggen eines Skripts und um das Skript mit Breakpoints oder schrittweise abzuspielen, müssen Sie es wie unten beschrieben in Visual Studio .NET ausführen.

Anzeigen von Skripts in Visual Studio

Visual Studio stellt zusätzliche Tools zum Anzeigen, Bearbeiten und Debuggen der Vuser-Skripts bereit. Sie können Breakpoints hinzufügen, Variablenwerte anzeigen, Assemblyreferenzen hinzufügen und das Skript mithilfe von Visual Studio IntelliSense bearbeiten. Außerdem können Sie das Skript zum Debuggen schrittweise ausführen.

Beim Speichern des Skripts erstellt VuGen eine Visual Studio 2010-Lösungsdatei mit der Bezeichnung **Script.sln** im Skriptordner. Sie können die Lösungsdatei in Visual Studio .NET öffnen und alle ihre Komponenten im Solution Explorer anzeigen.

Zum Öffnen des Vuser-Skripts in Visual Studio 2010 wählen Sie **Entwurf > Skript in Visual Studio öffnen** aus oder klicken in der VuGen-Symbolleiste auf die Schaltfläche **Visual Studio** .

Hinweis:

- Wenn das Vuser-Skript mit einer früheren Version von LoadRunner (Version 11 oder niedriger) erstellt wurde, öffnet LoadRunner das Skript in Visual Studio 2008.
- Wenn Visual Studio 2010 ebenfalls installiert ist, wird das Skript in eine Visual Studio 2010-Lösung konvertiert.
- Wenn das Skript mit LoadRunner 11.50 oder höher erstellt wurde, ist Visual Studio 2010 erforderlich.

Doppelklicken Sie im Solution Explorer auf den jeweiligen Abschnitt, z. B. **vuser_init.cs**, um den Skriptinhalt anzuzeigen.

Hinweis: VuGen lädt automatisch alle Referenzen, die während der Aufzeichnung benötigt wurden. Sie können weitere Referenzen hinzufügen, die der Solution Explorer beim Kompilieren und Wiedergeben verwendet. Wählen Sie den Knoten **Referenz** und dann im Kontextmenü die Option **Verweis hinzufügen** aus.

Klicken Sie im Solution Explorer auf **globals.cs** oder **globals.vb**, um eine Liste der im Skript definierten und verwendeten Variablen anzuzeigen.

Übersicht über .NET-Filter

Die Aufzeichnungsfilter verweisen auf die Assemblys, Schnittstellen, Namespaces, Klassen oder Methoden, die beim Aufzeichnen und Generieren eines Skripts ein- oder ausgeschlossen werden sollen.

VuGen enthält standardmäßig integrierte Systemfilter für .NET Remoting, ADO.NET, Enterprise Services und WCF (Windows Communication Foundation). Diese Filter wurden so erstellt, dass sie die relevanten Schnittstellen für ADO.NET, Remoting Enterprise Services und WCF einschließen. Außerdem ermöglicht Ihnen VuGen das Erstellen angepasster Filter.

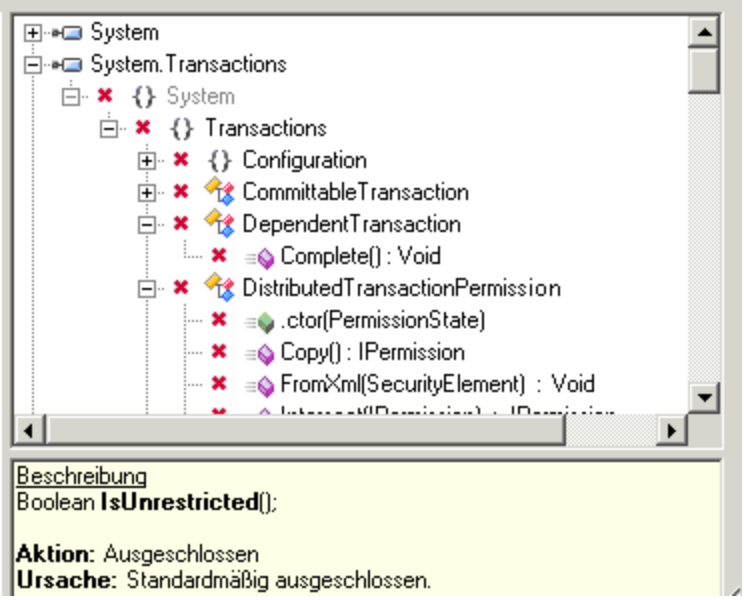
Angepasste Filter bieten mehrere Vorteile:

- **Remoting.** Beim Arbeiten mit .NET Remoting müssen Sie unbedingt bestimmte Klassen einschließen, die Ihnen das Aufzeichnen von Argumenten ermöglichen, die an die Remotemethode übergeben wurden.
- **Fehlende Objekte.** Wenn das aufgezeichnete Skript kein spezifisches Objekt in Ihrer Anwendung aufgezeichnet hat, können Sie einen Filter verwenden, um die fehlende Schnittstelle, Klasse oder Methode einzuschließen.
- **Debugging.** Wenn Sie einen Fehler erhalten und die Fehlerursache unklar ist, können Sie Filter

zum Ausschließen von Methoden, Klassen oder Schnittstellen verwenden, um die problematische Operation zu identifizieren.

- **Wartbarkeit.** Sie können Skripts auf einer höhere Ebene aufzeichnen und die Aktualisierung sowie das Erstellen von Korrelationen in einem Skript erleichtern.

Ein Filtermanager ermöglicht das Bearbeiten vorhandener angepasster Filter. Dieser Manager zeigt Assemblys, Namespaces, Klassen, Methoden und Eigenschaften in einer farbcodierten hierarchischen Struktur an.



Der untere Fensterbereich enthält eine Beschreibung des Assemblys, des Namespaces, der Klasse, der Methode, der Eigenschaft oder des Ereignisses. Ferner wird hier angezeigt, ob das Element ein- oder ausgeschlossen ist, sowie der Grund für den Ein- bzw. Ausschluss.

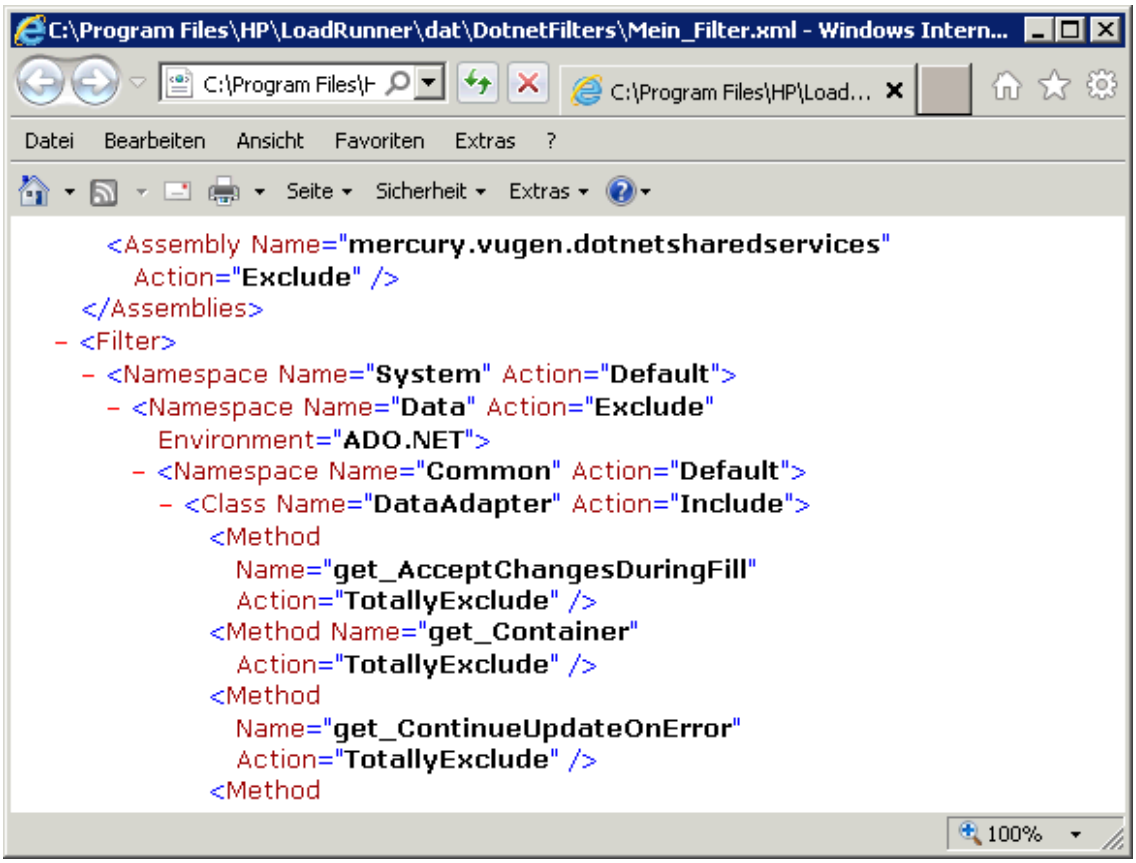
.NET-Filter - Weiterführende Themen

In der hierarchischen Struktur des Filter-Managers werden ausschließlich öffentliche Klassen und Methoden angezeigt. Nicht öffentliche Klassen bzw. Delegierte werden nicht angezeigt.

Durch manuelle Eingabe in der Filterdefinitionsdatei können Sie nicht öffentliche Klassen oder Methoden hinzufügen.

Die Filterdefinitionsdateien, **<Filtername>.xml** sind im Installationsordner dat\DotnetFilters gespeichert. Im Folgenden sind die für jedes Element verfügbaren Aktionseigenschaften aufgeführt: **Include**, **Exclude** oder **Totally Exclude**. Weitere Informationen finden Sie unter "Filter-Manager [.NET Protocol]" auf Seite 212.

Wenn Sie eine **Klasse** ausschließen, wendet der Filtermanager standardmäßig die Eigenschaft **Exclude** an und schließt die Klasse aus. Die von der ausgeschlossenen Klasse generierte Aktivität wird dabei allerdings eingeschlossen. Wenn Sie eine **Methode** ausschließen, wird die Eigenschaft **Totally Exclude** angewendet, so dass alle referenzierten Methoden ausgeschlossen werden.



Nehmen wir beispielsweise an, dass die Funktion A die Funktion B aufruft. Wenn für die Funktion A die Eigenschaft **ausgeschlossen** gilt und der Service die Funktion A aufruft, schließt das Skript den Aufruf der Funktion B ein. Gilt für die Funktion A jedoch die Eigenschaft **Totally Excluded**, schließt das Skript einen Aufruf der Funktion B nicht ein. Die Funktion B wird nur bei einem direkten Aufruf und nicht bei einem Aufruf über die Funktion A aufgezeichnet.

VuGen speichert eine Sicherungskopie des Filters, wie er bei der Aufzeichnung konfiguriert war. Die Kopie heißt **RecordingFilterFile.xml** und wird im Datenordner des Skripts gespeichert. Das ist sinnvoll, wenn Sie seit der letzten Aufzeichnung Änderungen am Filter vorgenommen haben und die Umgebung neu konstruieren müssen.

Richtlinien für das Festlegen von .NET-Filtern

Durch Testen Ihrer .NET-Anwendung können Sie feststellen, wie der Server auf Clientanfragen reagiert. Lasttests zeigen, wie der Server auf die Belastung durch mehrere Benutzer reagiert.

Beim Aufzeichnen einer .NET-Anwendung berücksichtigt Ihr Skript möglicherweise Aufrufe von Methoden, die keine Auswirkung auf den Server haben, wie z. B. die Aufrufe eines lokalen Dienstprogramms oder einer grafischen Benutzeroberfläche. Diese Aufrufe sind für die Testziele normalerweise nicht relevant und sollten daher herausgefiltert werden.

Die integrierten Filter für .NET Remoting, ADO.NET, Enterprise Services und WCF wurden dahingehend definiert, dass sie nur den Serververkehr aufzeichnen, der für Ihre Testziele relevant ist. In einigen Fällen müssen Sie die Filter u. U. anpassen, um die Aufrufe Ihrer .NET-Anwendung

zu erfassen oder unnötige Aufrufe auszuschließen. Mit dem Filtermanager können Sie angepasste Filter erstellen, die irrelevante Aufrufe ausschließen und serverspezifische Aufrufe erfassen.

Wir empfehlen Ihnen, sich vor dem Erstellen eines Test mit Ihrer Anwendung vertraut zu machen und die primären Klassen und Methoden festzulegen, so dass Sie wissen, welche Klassen und Methoden beim Aufzeichnen berücksichtigt werden sollen.

Sollten Sie mit den Klassen Ihrer Anwendung nicht vertraut sein, können Sie Visual Studio oder eine Stapelverfolgung verwenden, um die von Ihrer Anwendung aufgerufenen Methoden zu ermitteln und diese in den Filter einzuschließen. VuGen erlaubt das Aufzeichnen mit einem Stacktrace, der alle Methoden protokolliert, die von Ihrer Anwendung aufgerufen wurden.

Sobald Sie die erforderlichen Methoden und Klassen ermittelt haben, schließen Sie sie mithilfe des Filtermanagers ein. Bei der Vorbereitung eines Skripts müssen Sie den Filter möglicherweise wiederholt anpassen, um ihn optimal einzustellen. Ein optimaler Filter zeichnet die relevanten Methoden auf, ohne eine allzu große Zahl irrelevanter Aufrufe im Skript einzufügen.

Tipp: Ändern Sie den Filter dahingehend, dass Ihr Skript in VuGen kompiliert wird (Umschalt+F5) - ein Test mit einer korrekten Syntax. Fahren Sie mit der Anpassung des Filters fort und erstellen Sie ein funktionales Skript, das in VuGen ausgeführt wird.

Wenn Sie manuellen Code, wie z. B. Kontrollstrukturen oder Meldungsanweisungen, in Ihr Skript einfügen möchten, dürfen Sie das erst nach dem Einrichten eines funktionalen Skripts tun, das in VuGen ausgeführt wird. Der Grund hierfür ist, dass manuelle Änderungen verloren gehen, wenn Sie ein Skript erneut aufzeichnen oder generieren.

Feststellen ein- bzw. auszuschließender Elemente

Beim Erstellen eines anpassten Filters empfehlen wie Ihnen, mit der Auswahl eines geeigneten integrierten Filters als Basisfilter zu beginnen. Anschließend können Sie den Filter auf verschiedenen Weisen anpassen:

- **Top-Down-Konzept.** Bei diesem Konzept wird der relevante Namespace eingeschlossen, während die spezifischen Klassen, die nicht Teil der Client-Server-Aktivität sind, ausgeschlossen werden. Wir empfehlen dieses Konzept, sofern Sie mit Ihrer Anwendung vertraut sind und ein korrekt definiertes Assembly identifizieren können, das alle Client-Server-Aktivitäten implementiert, ohne dabei GUI-Elemente, wie z. B. MyDataAccessLayer.dll, einzubeziehen.

- **Bottom-up-Konzept.** Bei diesem Konzept verwenden Sie den Standardfilter und passen diesen durch Hinzufügen individueller Methoden und Klassen an. Verwenden Sie dieses Konzept, wenn Sie keine gut definierte Schicht identifizieren können oder mit der Handhabung Ihrer Anwendung nicht besonders vertraut sind. Fügen Sie nicht alle AUT-Assemblys hinzu, und versuchen Sie, zusätzliche Komponenten einzeln zu entfernen.

Der folgende Abschnitt enthält Richtlinien zum Ein- bzw. Ausschließen von Elementen.

- Wenn Ihr Skript nach dem Einschließen einer Klasse zahlreiche Aufrufe unabhängiger Methoden enthält, müssen Sie den Filter dahingehend ändern, dass irrelevante Methoden ausgeschlossen werden.

- Sollten Sie einen Nicht-Client/Server-Aufruf in Ihrem Skript identifizieren, müssen Sie die entsprechende Methode im Filter ausschließen.

- Beim Aufzeichnen erkennt VuGen u. U. ein unbekanntes Eingabeargument, wie z. B. ein Argument mit einer bis dahin noch nie verwendeten Konstruktion. Wenn dieses Argument die Serialisierung unterstützt, wird es von VuGen durch Speichern in einem besonderen Format in einer Datei **serialisiert**. Bei der Wiedergabe wird das Argument von VuGen mittels einer **Deserialisierung** rekonstruiert.

- VuGen serialisiert nur Objekte, die erfolgreich in Form von Argumenten ausgeführt wurden, die nicht von dem Filter eingeschlossen wurden. Wir empfehlen Ihnen das Einschließen dieses Objekts im Filter, um seine Konstruktion und Aktivität protokollieren zu können, anstatt das Objekt in seiner serialisierten Form zu verwenden. Sie können serialisierte Objekte im Skript identifizieren, indem Sie in Ihrem Skript nach Aufrufen der Methode **LrReplayUtils.GetSerializedObject** oder in einer WCF-Umgebung nach Aufrufen der Methode **LrReplayUtils.GetSerializedDataContract** suchen. VuGen speichert die serialisierten Objekte im Skriptordner **\data\SerializedObjects** als XML-Datei mit Index: **Serialization_1.xml**, **Serialization_2.xml** und so weiter.

- Methoden, für die keine Regeln vorgegeben wurden, sind standardmäßig ausgeschlossen. Ist die Remotingumgebung aktiviert, werden alle Remoteaufrufe standardmäßig eingeschlossen, auch wenn dies nicht explizit geschieht. Um das Standardverhalten zu ändern, können Sie eine angepasste Regel hinzufügen und auf diese Weise spezifische Aufrufe ausschließen, die an den Remoteserver gerichtet sind.

- Die Bearbeitung der im Rahmen von Remotingaufrufen übergebenen Argumente, deren Typen von dem Filter nicht eingeschlossen werden, erfolgt über den Serialisierungsmechanismus. Wenn Sie die Serialisierung der Argumente verhindern möchten, können Sie diese Typen explizit einschließen, um die Konstruktion und die Aktivität dieser Argumente aufzuzeichnen.

- Schließen Sie alle Aktivitäten aus, die GUI-Elemente betreffen.

- Fügen Sie die Assemblys für die Dienstprogramme hinzu, die für das Kompilieren des Skripts u. U. erforderlich sind.

Weitere Informationen zum Ein- und Ausschließen von Elementen finden Sie unter "Filter-Manager [.NET Protocol]" auf Seite 212.

Definieren eines wirksamen Filters

Bei der Vorbereitung eines Skripts müssen Sie den Filter möglicherweise wiederholt anpassen, um ihn optimal einzustellen. Ein optimaler Filter zeichnet die relevanten Methoden auf, ohne eine allzu große Zahl irrelevanter Aufrufe im Skript einzufügen.

Definieren eines wirksamen Filters

1. Erstellen Sie einen neuen Filter anhand eines der integrierten Filter. Wenn Sie wissen, dass die zu testende Anwendung (AUT, Application Under Test) weder ADO.NET noch Remoting, WCF oder Enterprise Services verwendet, müssen Sie diese Option löschen, da unnötige Filter u. U. die Aufzeichnungsgeschwindigkeit beeinträchtigen.

2. Setzen Sie die Option **Stapelverfolgung** für das Aufzeichnen und Generieren von Code auf **true**. Öffnen Sie das Dialogfeld **Aufzeichnungsoptionen** (Strg + F7) und wählen Sie den Knoten **Aufzeichnen** aus. Aktivieren Sie **Debugoptionen: Stapelverfolgung** und **Codegenerierung: Stapelverfolgung anzeigen**.

3. Zeichnen Sie eine Anwendung auf. Klicken Sie zum Starten auf **Aufzeichnen** (Strg + R) und zum Beenden auf **Aufzeichnung beenden** (Strg + F5).

4. Zeigen Sie die Schritte des Skripts an. Wenn Sie die Geschäftslogik anhand der Schritte feststellen und die Korrelation übernehmen können, müssen Sie möglicherweise keine angepassten Filter erstellen. Wenn das Skript sehr lang oder schwer zu aktualisieren und zu korrelieren ist, sollten Sie den Skriptfilter anpassen.

5. Versuchen Sie, eine Methode auf einer hohen Ebene in dem Aufruf zu identifizieren, der einen oder mehrere Clientserveraufrufe erfasst oder umfasst. Dazu können Sie die AUT-Quelldateien (sofern verfügbar) in Visual Studio öffnen oder ein Stacktrace des Skripts anzeigen.

6. Richten Sie den Filter so ein, dass alle relevanten Methoden eingeschlossen werden. Dazu müssen Sie möglicherweise ein Assembly hinzufügen. Tipps zum Ein- und Ausschließen von Elementen in einem Filter finden Sie unter "Übersicht über .NET-Filter" auf Seite 796.

7. Zeichnen Sie die Anwendung erneut auf. Nach einer Änderung des Filters sollte die Anwendung grundsätzlich neu aufgezeichnet werden.

8. Wiederholen Sie die Schritte 4 bis 7, bis Sie ein einfaches Skript erhalten, das sich verwalten und korrelieren lässt.

9. Nach dem Erstellen eines optimalen Skripts deaktivieren Sie die Optionen für **Stapelverfolgung** und regenerieren das Skript. Öffnen Sie das Dialogfeld **Aufzeichnungsoptionen** (Strg + F7) und wählen Sie den Knoten **Aufzeichnen** aus. Deaktivieren Sie **Debugoptionen: Stapelverfolgung** und **Codegenerierung: Stapelverfolgung anzeigen**. Damit verbessern Sie die Leistung der nachfolgenden Aufzeichnungen.

10. Korrelieren Sie das Skript. Um die fehlerfreie Ausführung Ihres Tests zu gewährleisten, müssen Sie eine Korrelation einfügen, um einen Wert zu erfassen und diesen zu einem späteren Zeitpunkt im Skript verwenden zu können. Weitere Informationen finden Sie unter "Korrelieren von Skripts - Microsoft .NET" auf Seite 303.

Konfigurieren von Anwendungssicherheit und Berechtigungen

Sicherheitsausnahmen während der Aufzeichnung einer Anwendung werden meist durch fehlende Berechtigungen verursacht, d. h., die Berechtigungen des Aufzeichnungscomputers reichen nicht aus, um die Anwendung aufzuzeichnen. Dies ist häufig der Fall, wenn sich eine Anwendung nicht lokal auf dem Computer, sondern im Intranet oder in einem Netzwerk befindet.

Zur Lösung dieses Problems müssen Sie dem Aufzeichnungscomputer einen Zugriff auf die Anwendung und das Skript mit dem Berechtigungssatz **Vollständig vertrauenswürdig** gewähren.

Eine Möglichkeit besteht darin, die Anwendung zu kopieren und das Skript lokal zu speichern, da die Benutzer standardmäßig über den Berechtigungssatz **Vollständig vertrauenswürdig** für alle lokalen Anwendungen und Ordner verfügen.

Als weitere Möglichkeit können Sie neue Codegruppen erstellen, durch die **Vollständig vertrauenswürdig** für jeden Anwendungsordner und den Skriptordner gilt.

Gewähren des Berechtigungssatzes "Vollständig vertrauenswürdig", wenn Visual Studio NICHT installiert ist

1. Führen Sie über die Eingabeaufforderung die Anwendung **caspol.exe** aus.
2. Legen Sie die gewünschte Berechtigung fest.

Gewähren des Berechtigungssatzes "Vollständig vertrauenswürdig", wenn Visual Studio installiert ist

1. Öffnen Sie die .NET-Konfigurationseinstellungen. Wählen Sie **Start > Programme > Verwaltung > Microsoft .NET Framework 2.0-Konfiguration** aus. Das Fenster für die .NET-Konfiguration wird geöffnet.
2. Erweitern Sie den Knoten **Laufzeitsicherheitsrichtlinie**, um die Codegruppen des Computers anzuzeigen.
3. Wählen Sie den Knoten **All_Code** aus.
4. Wählen Sie **Aktion > Neu** aus. Das Dialogfeld **Neue Codegruppe erstellen** wird geöffnet.
5. Geben Sie einen Namen für eine neue Codegruppe für die Anwendung oder das Skript ein. Klicken Sie auf die Schaltfläche **Weiter**.
6. Wählen Sie den Bedingungstyp **URL** aus. Geben Sie im Feld **URL** den vollständigen Pfad der Anwendung oder des Skripts im Format `file://…` ein und klicken Sie auf **Weiter**.
7. Wählen Sie den Berechtigungssatz **Vollständig vertrauenswürdig** aus. Klicken Sie auf **Weiter**.
8. Klicken Sie im Dialogfeld zum Abschließen des Assistenten auf **Fertig stellen**. Die Codegruppe wird durch das Konfigurationstool zur Liste der bestehenden Gruppen hinzugefügt.
9. Wiederholen Sie die zuvor beschriebene Prozedur für alle .NET-Anwendungen, die Sie aufzeichnen möchten.
10. Wiederholen Sie die zuvor beschriebene Prozedur für den Vuser-Skriptordner.

Hinweis: Stellen Sie sicher, dass auf allen am Test beteiligten Lastgeneratorcomputern für den Skriptordner der Berechtigungssatz **Vollständig vertrauenswürdig** gewährt wurde

(nur LoadRunner).

.NET - Fehlerbehebung und Einschränkungen

In diesem Abschnitt werden die Fehlerbehebung und die für .NET-Vuser-Skripts geltenden Einschränkungen beschrieben.

Die folgenden Beschränkungen gelten für die VuGen-Aufzeichnung einer Microsoft .NET-Anwendung:

- Bei .NET-Vuser-Skripts wird in VuGen nur die Einzelprotokollaufzeichnung unterstützt.
- Der direkte Zugriff auf öffentliche Felder wird nicht unterstützt; die zu testende Anwendung muss über Methoden oder Eigenschaften auf Felder zugreifen.
- VuGen zeichnet keine statischen Felder in den Anwendungen auf, sondern nur Methoden innerhalb von Klassen.
- Multithread-Unterstützung hängt von der Clientanwendung ab. Wird Multithreading von der aufgezeichneten Anwendung unterstützt, gilt dies auch für das Vuser-Skript.
- In bestimmten Fällen können Sie mehrere Iterationen nur ausführen, wenn Sie das Skript ändern. Objekte, die bereits in einer vorherigen Iteration initialisiert wurden, können nicht erneut initialisiert werden. Daher müssen Sie zum Ausführen mehrerer Iterationen alle offenen Verbindungen oder Remotingkanäle am Ende jeder Iteration schließen.
- Die Aufzeichnung von Enterprise Services-Kommunikation auf der Basis von MSMQ und Enterprise Services mit IIS-Hosting wird nicht unterstützt.
- Die Aufzeichnung von WCF-Diensten, die von der Clientanwendung gehostet werden, wird in VuGen teilweise unterstützt.
- Die Aufzeichnung von Remotingaufrufen mithilfe eines benutzerdefinierten Proxys wird nicht unterstützt
- Die Aufzeichnung der Eigenschaft **ExtendedProperties** von ADO.NET-Objekten wird nicht unterstützt, wenn der standardmäßige ADO.NET-Filter verwendet wird.
- Mit .NET Framework 1.1 erstellte Anwendungen, die nicht mit Framework 2.0 kompatibel sind, können nicht aufgezeichnet werden. Um die Kompatibilität einer Framework 1.1-Anwendung zu überprüfen, fügen Sie die folgenden XML-Tags zur Konfigurationsdatei der Anwendung hinzu:

```
 <configuration>
     <startup>
          <supportedRuntime version="v2.0.50727"/>
     </startup>
</configuration>
```

Rufen Sie die Anwendung (ohne VuGen) auf und testen Sie ihre Funktionalität. Wenn die Anwendung ordnungsgemäß funktioniert, kann VuGen sie aufzeichnen. Entfernen Sie die oben angegebenen Tags, bevor Sie die zu testende Anwendung mit VuGen aufzeichnen. Weitere Informationen zu dieser Lösung finden Sie in der MSDN Knowledge Base.

- Anwendungen, die .NET Remoting Framework verwenden und in CLR 2 (.NET Framework 2/3/3.5) ausgeführt werden, stürzen während der Aufzeichnung möglicherweise ab. Bei einem Absturz erhalten Sie eine Meldung mit den Zeichenfolgen "Version=4.x.x.x" und "ist nicht für die Aktivierung registriert".
  Mögliche Problemumgehung: Wählen Sie in der Microsoft .NET- Aufzeichnungsoberfläche unter **Unterstützung für frühere .NET-Versionen** die Option **Frühere .NET-Versionen in Transportebene emulieren** aus, und führen Sie dann die Aufzeichnung erneut aus.

- Wenn die getestete Anwendung ein serveraktiviertes Objekt durch den Aufruf von **new RemoteObject()** empfängt, erzeugt VuGen eine RemotingServices.Connect-Funktion.

- Anwendungen, die mehrere Prozesse oder mehrere Anwendungsdomänen verwenden, werden nur teilweise unterstützt.

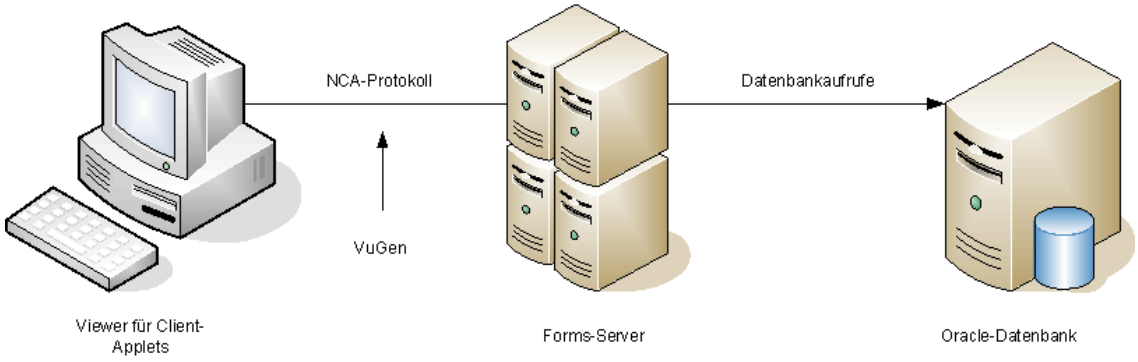
- Gemeinsam genutzte DLLs dürfen nur in **Aufzeichnungsoptionen** angegeben werden. In **Laufzeiteinstellungen** an der Liste gemeinsam genutzter DLLs vorgenommene Änderungen haben keine Auswirkungen.

Oracle NCA-Protokoll

Übersicht über das Oracle NCA-Protokoll

Bei Oracle NCA handelt es sich um ein Protokoll, das für die Kommunikation mit dem Oracle Forms-Server zuständig ist. Mit dem Browser starten Sie den Datenbankclient, einen Applet-Viewer. Über diesen Applet-Viewer führen Sie Aktionen für die NCA-Datenbank durch. Dadurch benötigen Sie keine Clientsoftware und können Datenbankaktionen von allen Plattformen ausführen, die den Applet-Viewer unterstützen.

Die NCA-Umgebung ist eine 3-Tier-Umgebung. Der Benutzer sendet als Erstes einen HTTP-Aufruf vom Browser an einen Webserver. Mit diesem Aufruf wird auf die Start-HTML-Seite zugegriffen, die wiederum das Applet für Oracle-Anwendungen aufruft. Der Client (Applet-Viewer) kommuniziert über das proprietäre NCA-Protokoll mit dem Anwendungsserver (Oracle Forms-Server), der dann die Informationen an den Datenbankserver weitergibt.



VuGen zeichnet die NCA-Kommunikation zwischen dem Client und dem Formularserver (Anwendungsserver) auf und spielt sie ab.

Das Oracle NCA-Protokoll wird in der Regel als Mehrfachprotokoll in Kombination mit Web (HTTP/HTML) verwendet. Dies wird für die Aufzeichnung mit Oracle NCA empfohlen. Verwenden Sie Oracle NCA als Einzelprotokoll, werden standardmäßig Webereignisse aufgezeichnet, aber keine Schritte generiert (oder wiedergegeben).

Wenn Sie zunächst ein Skript mit einem Einzelprotokoll für Oracle NCA erstellt haben und zu einem späteren Zeitpunkt die Webfunktionen für Tests benötigen, können Sie das Skript in VuGen erneut erzeugen, um die Webfunktionen hinzuzufügen. Dabei müssen Sie die Sitzung nicht erneut aufzeichnen. Sie können dies über den Knoten **Protokolle** in den Aufzeichnungsoptionen angeben.

Beispielskripts für das Oracle NCA-Protokoll

Im folgenden Beispiel hat der Benutzer ein Element aus einer Liste ausgewählt (**nca_list_activate_item**), auf eine Schaltfläche geklickt (**nca_button_press**), einen Listenwert abgerufen (**nca_lov_retrieve_items**) und in ein Bearbeitungsfeld geklickt (**nca_edit_click**). Die logischen Namen der Objekte sind die Parameter dieser Funktionen.

```
...
nca_lov_select_item("Responsibilities","General Ledger, Vision Operations");
nca_list_activate_item("FNDSCSGN.NAVIGATOR.LIST.0","+ Journals");
nca_list_activate_item("FNDSCSGN.NAVIGATOR.LIST.0","       Enter");
nca_button_press("GLXJEENT.TOOLBAR.LIST.0");
nca_lov_find_value("Batches","");
nca_lov_retrieve_items("Batches",1,9);
nca_lov_select_item("Batches","AR 1020 Receivables 2537: A 1020");
nca_edit_click("GLXJEENT.FOLDER_QF.BATCH_NAME.0");
...
```

In bestimmten Tests, die beispielsweise für die Oracle Configurator-Anwendungen durchgeführt werden, müssen während der Sitzung Informationen von mindestens einer Funktion zurückgegeben werden. VuGen speichert die dynamischen Informationen automatisch in einem Parameter, indem eine Funktion **web_reg_save_param** in das Skript eingefügt wird. Im folgenden Beispiel werden die Verbindungsinformationen in einem Parameter mit der Bezeichnung **NCAJServSessionID** gespeichert.

```
web_reg_save_param ("NCAJServSessionId", "LB=\r\n\r\n", "RB=\r",
       LAST);
web_url("f60servlet",
         "URL=http://usscifforms05.sfb.na/servlet/f60servlet\?config
         =mult",         LAST);
```

Im obigen Beispiel ist die rechte Begrenzung **\r**. Die tatsächliche rechte Begrenzung ist möglicherweise je nach System unterschiedlich.

Hinweis: Es wird empfohlen, dass der Benutzer die Parameter **web_reg_save_param** nicht ändert, wenn diese automatisch generiert wurden. Alternativ können Sie manuell eine neue Funktion **web_reg_save_param** hinzufügen oder eine neue Korrelationsregel hinzufügen.

Tipps zum Aufzeichnen und Wiedergeben mit Oracle NCA

Befolgen Sie beim Aufzeichnen eines Oracle NCA-Vuser-Skripts die folgenden Richtlinien:

- Bevor Sie ein Skript aufzeichnen, sollten Sie Jinitiator installieren.
- Schließen Sie alle Browser, bevor Sie mit der Aufzeichnung beginnen.
- Zeichnen Sie das Anmeldeverfahren im Abschnitt **vuser_init** auf. Zeichnen Sie einen typischen Geschäftsprozess im Abschnitt **Actions** auf. Wenn Sie das Skript ausführen, können Sie mehrere Iterationen für einen bestimmten Geschäftsprozess festlegen. Weitere Informationen finden Sie unter "Lösungs-Explorer - Übersicht" auf Seite 57.
- VuGen unterstützt die Aufzeichnung von Oracle Forms-Anwendungen mit dem Forms Listener-Servlet im Mehrfachprotokollmodus. Der Anwendungsserver verwendet das **Forms Listener-Servlet**, um einen Laufzeitprozess für jeden Client zu erstellen. Der Laufzeitprozess **Forms Server Runtime** behält eine ständige Verbindung zum Client bei und sendet Informationen zum und vom Server.

Damit Forms 4.5 bei der Wiedergabe unterstützt wird, müssen Sie die Datei **mdrv_oracle_nca.dat** im Ordner **dat > mdrv** dem folgenden Beispiel entsprechend ändern:

```
[Oracle_NCA]
ExtPriorityType=protocol
WINNT_EXT_LIBS=ncarp110.dll
WIN95_EXT_LIBS=ncarp110.dll
LINUX_EXT_LIBS=liboranca.so
SOLARIS_EXT_LIBS=liboranca.so
HPUX_EXT_LIBS=liboranca.sl
AIX_EXT_LIBS=liboranca.so
LibCfgFunc=oracle_gui_configure
UtilityExt=lrun_api
```

Um die Forms-Unterstützung für Versionen nach 4.5 wiederherzustellen, stellen Sie die ursprünglichen Werte wieder her.

Pragma-Modus

Die Clientseite des Oracle NCA-Vusers kann so konfiguriert werden, dass eine zusätzliche Kopfzeile mit der Bezeichnung **Pragma** an den Server gesendet wird. Bei dieser Kopfzeile handelt es sich um einen Zähler, der sich folgendermaßen verhält: Die Anfangsmeldung des NCA-Handshake weist einen Wert von 1 auf.

Die Meldungen nach dem Handshake werden gezählt, dabei wird mit 3 begonnen. Der Zähler wird bei jeder vom Client gesendeten Meldung um 1 erhöht.

Wenn die vom Server empfangene Meldung den Typ `plain\text` aufweist und der Meldungstext mit `ifError:#/#00` beginnt, sendet der Client eine aus 0 Byte bestehende Meldung an den Server und der Pragma-Wert ändert sein Vorzeichen in ein Minus. Dieses Vorzeichen ändert sich wieder, wenn der Client die Informationen vom Server erfolgreich empfangen hat.

Das Aufzeichnen der Pragma-Kopfzeile wird nur im Modus mit mehreren Protokollen (Oracle NCA und Web) unterstützt. Sie können den Pragma-Modus in der Skriptdatei default.cfg ermitteln. Wenn Sie den Pragma-Modus verwenden, wird **UseServletMode** auf **2** gesetzt.

```
[HttpConnectMode]
UseHttpConnectMode=1
RelativeURL=<NCAJServSessionId>
UseServletMode=2
```

Weitere Informationen zu Pragma in Bezug auf Laufzeiteigenschaften finden Sie unter "Oracle NCA > Clientemulation (Knoten)" auf Seite 395.

Um den Pragma-Modus zu ermitteln, können Sie eine Aufzeichnung auf WinSocket-Ebene durchführen und den Pufferinhalt überprüfen. Im ersten Beispiel enthält der Puffer die Pragma-Werte als Zähler:

```
send buf108
    "POST /ss2servlet/oracle.forms.servlet.ListenerServlet?JServSessionIdss2ser"
    "vlet=gk5q79uqy1 HTTP/1.1\r\n"
    "Pragma: 1\r\n"
    ...
send buf110
    "POST /ss2servlet/oracle.forms.servlet.ListenerServlet?JServSessionIdss2ser"
    "vlet=gk5q79uqy1 HTTP/1.1\r\n"
    "Pragma: 3\r\n"
    ...
```

Im folgenden Beispiel enthält der Puffer die Pragma-Werte als Fehleranzeige:

```
recv buf129 281
    "HTTP/1.1 200 OK\r\n"
    "Date: Tue, 21 May 2002 00:03:48 GMT\r\n"
    "Server: Oracle HTTP Server Powered by Apache/1.3.19 (Unix) mod_fastcgi/2.2"
```

```
    ".10 mod_perl/1.25 mod_oprocmgr/1.0\r\n"
    "Content-Length: 13\r\n"
    "Content-Type: text/plain\r\n"
    "\r\n"
    "ifError:8/100"
send buf130
    "POST /ss2servlet/oracle.forms.servlet.ListenerServlet?JServSessionIdss2ser"
    "vlet=gk5q79uqy1 HTTP/1.1\r\n"
    "Pragma: -12\r\n"
    ...
```

Aktivieren der Aufzeichnung von Objekten nach Name

Wenn Sie ein Oracle NCA-Skript aufzeichnen, müssen Sie die Sitzung anhand von Objektnamen anstelle der standardmäßigen Objekt-ID aufzeichnen. Wenn das Skript anhand der Objekt-ID aufgezeichnet wird, tritt bei der Wiedergabe möglicherweise ein Fehler auf, weil die ID dynamisch vom Server erzeugt wird und zwischen Iterationen unterschiedlich sein kann. Sie können überprüfen, ob Ihr Skript mit Objektnamen aufgezeichnet wird, indem Sie die Funktion **nca_connect_server** untersuchen.

```
nca_connect_server("199.35.107.119","9002"/*version=11i*/,"module=/d1/oracle
        /visappl/fnd/11.5.0/forms/US/FNDSCSGN userid=APPLSYSPUB/PUB@VIS
        fndnam=apps record=names ");
```

Wenn das Argument **record=names** nicht in der Funktion **nca_connect_server** angezeigt wird, zeichnen Sie möglicherweise Objekt-IDs auf. Sie können festlegen, dass VuGen Objektnamen aufzeichnet, indem Sie eines der folgenden Elemente ändern:

Start-HTML-Datei

Wenn Sie Zugriff auf die Start-HTML-Datei haben, können Sie festlegen, dass VuGen Objektnamen anstelle der Objekt-ID aufzeichnet, indem Sie das Kennzeichen **record=names** in der Startdatei einstellen. Dabei handelt es sich um die Datei, die beim Starten der Oracle NCA-Anwendung geladen wird. In den folgenden Schritten wird beschrieben, wie Sie die Aufzeichnung von Objektnamen unter Verwendung der Start-HTML-Datei aktivieren.

1. Bearbeiten Sie die Startdatei, die beim Starten des Applet-Viewers aufgerufen wird, indem Sie die unten angezeigte Zeile ändern.

   ```
   <PARAM name="serverArgs ... fndnam=APPS">
   ```

2. Fügen Sie den Oracle-Schlüssel **record=names** wie unten gezeigt hinzu.

   ```
   <PARAM name="serverArgs ... fndnam=APPS record=names">
   ```

Forms-Konfigurationsdatei

Wenn die Anwendung über eine Start-HTML-Datei verfügt, die auf eine Forms Web CGI-Konfigurationsdatei **formsweb.cfg** verweist (allgemeiner Verweis), treten möglicherweise beim Hinzufügen von **record=names** zur Startdatei Probleme auf. In diesem Fall müssen Sie die Konfigurationsdatei ändern. In den folgenden Schritten wird beschrieben, wie Sie die Aufzeichnung

von Objektnamen unter Verwendung der Konfigurationsdatei aktivieren.

1. Gehen Sie zur Forms Web CGI-Konfigurationsdatei.
2. Definieren Sie einen neuen Parameter in dieser Datei (diese Änderung finden Sie im nachstehenden Beispiel der Web CGI-Konfigurationsdatei).

```
serverApp=forecast
serverPort=9001
serverHost=easgdev1.dats.ml.com
connectMode=socket
archive=f60web.jar
archive_ie=f60all.cab
xrecord=names
```

3. Öffnen Sie die Start-HTML-Datei und suchen Sie nach `PARAM NAME="serverArgs"`.
4. Fügen Sie den Variablennamen als Argument zum Parameter **ServerArgs** hinzu (siehe unten). Beispiel: **record=%xrecord%**.

```
<PARAM NAME="serverArgs" VALUE="module=%form% userid=%userid% %otherParams% r
ecord=%xrecord%">
```

5. Alternativ können Sie die Datei **basejini.htm** im Installationsordner von Oracle Forms bearbeiten. Dabei handelt es sich um die Standard-HTML-Datei zum Ausführen eines Formulars im Web anhand von JInitiator-style-Tags, um das Formular-Applet einzuschließen. Fügen Sie in der Datei **basejini.htm** folgende Zeile zu den Parameterdefinitionen hinzu:

```
<PARAM NAME="recordFileName" VALUE="%recordFileName%">
```

Fügen Sie im Tag <EMBED> folgende Zeile hinzu:

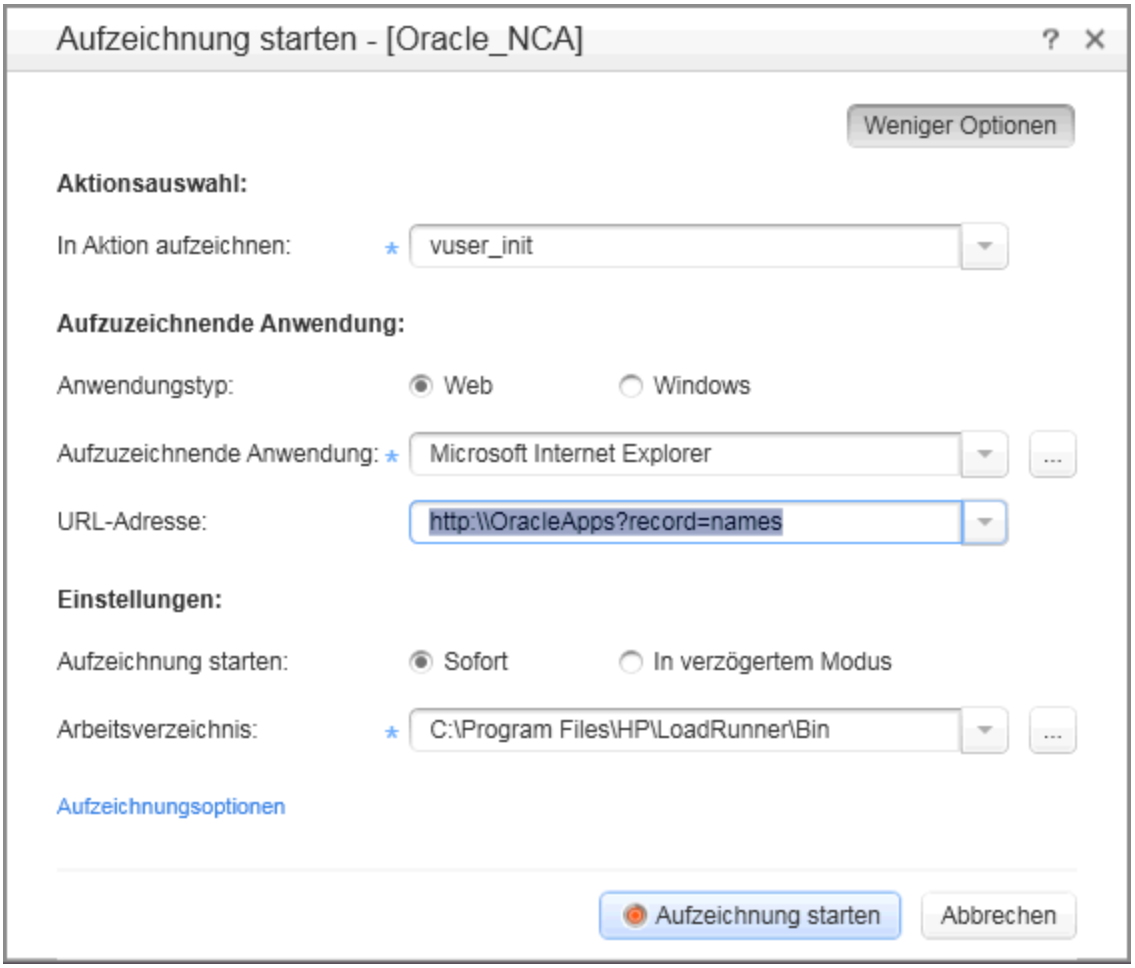
```
serverApp="%serverApp%"
logo="%logo%"
imageBase="%imageBase%"
formsMessageListener="%formsMessageListener%"
recordFileName="%recordFileName%"
```

Der Nachteil bei der Bearbeitung dieser Datei anstelle der Servlet-Konfigurationsdatei **formsweb.cfg** liegt darin, dass diese Datei bei der erneuten Installation von Oracle Forms ersetzt wird. Wenn Sie dies vermeiden möchten, können Sie eine Kopie der Datei **basejini.htm** erstellen und an einem anderen Speicherort speichern. Bearbeiten Sie in der Servlet-Konfigurationsdatei den Parameter **baseHTMLJinitiator**, um auf die neue Datei zu verweisen.

URL für Aufzeichnung

Wenn Sie keinen Zugriff auf die Start-HTML-Datei haben, können Sie dennoch festlegen, dass Oracle NCA Objektnamen anstelle der Objekt-ID aufzeichnet, indem Sie den URL für die Aufzeichnung ändern. Folgende Lösung funktioniert nur, wenn die Start-HTML-Datei beim Laden nicht auf eine andere Datei verweist.

Für diese Lösung fügen Sie im Dialogfeld **Start Recording** hinter dem URL-Namen für die Aufzeichnung **?record=names** hinzu.



Starten von Oracle-Anwendungen über die persönliche Startseite

Wenn Sie Oracle Forms-Anwendungen (Version 6i und höher) starten, indem Sie sich über die **persönliche Startseite** anmelden, müssen Sie mehrere Systemprofiloptionen auf der Benutzerebene festlegen. Es ist wünschenswert, solche Variablen auf der Benutzerebene, nicht auf der Site-Ebene weiterzugeben, wo alle Benutzer betroffen sind. In den folgenden Schritten wird die Vorgehensweise für die Konfiguration des Profils "ICX: Forms Launcher" beschrieben.

1. Melden Sie sich bei der Anwendung an und wählen Sie die Zuständigkeit **System Administrator** aus.
2. Wählen Sie im Menü **Navigator** den Befehl **Profile/System** aus.
3. Im Formular **Find System Profile Values**:

a. Wählen Sie die Option **Display > Site** aus.

b. **Users** = <Ihre Benutzeranmeldung> (z. B. **operations**, **mfg** usw.).

c. **Enter Profile** =%ICX%Launch%.

d. Klicken Sie auf **Find**.

4. Aktualisieren Sie den Benutzerwert für das Profil **ICX:Forms Launcher**:

   - Wenn kein Parameter an den URL übergeben wurde, hängen Sie die folgende Zeichenfolge am Ende des URL des Benutzerwertes an: **?**`play==;record=names`
   - Wenn ein Parameter an den URL übergeben wurde, hängen Sie die folgende Zeichenfolge am Ende des URL des Benutzerwertes an: **=;**`play==;record=names`

5. Speichern Sie die Transaktion.

6. Melden Sie sich von der Oracle Forms-Sitzung ab.

7. Melden Sie sich von der Sitzung der persönlichen Startseite ab.

8. Melden Sie sich über die **persönliche Startseite** mit Ihrem Benutzernamen erneut an.

Wenn Sie die Profiloption **ICX: Forms Launcher** nicht auf der Benutzerebene aktualisieren konnten, öffnen Sie die Zuständigkeit **Application Developer** und wählen Sie die Option **Updatable** für das Profil **ICX_FORMS_LAUNCHER** aus.

Der erste an den URL übergebene Parameter muss mit einem Fragezeichen (?) beginnen. Alle folgenden Parameter übergeben Sie mit einem kaufmännischen Und (=;). In den meisten Fällen enthält der URL bereits Parameter, die Sie ermitteln können, indem Sie nach einem Fragezeichen suchen.

Oracle NCA - Fehlerbehebung und Einschränkungen

In diesem Abschnitt werden die Fehlerbehebung und die für Oracle NCA-Protokolle geltenden Einschränkungen beschrieben.

Testen von gesicherten Oracle NCA-Anwendungen

- Löschen Sie im Knoten **Portzuordnung** des Dialogfelds **Aufzeichnungsoptionen** alle bestehenden Einträge für Port 443 und erstellen Sie einen neuen Eintrag für den Oracle-Servernamen:

Dienst-ID: HTTP
Zielserver: IP-Adresse des Oracle Forms-Servers oder langer Hostname
Port: 443
Verbindungstyp: SSL
SSL Version: Aktive SSL-Version. Wählen Sie im Zweifel **SSL 2/3** aus.

Beachten Sie, dass nicht **NCA**, sondern **HTTP** als **Dienst-ID** angegeben ist.

Weitere Informationen finden Sie unter "Netzwerk > Portzuordnung (Knoten)" auf Seite 224.

- Wenn bei der Wiedergabe eines NCA HTTPS-Skripts mit dem Befehl **nca_connect_server** Probleme auftreten, fügen Sie am Anfang des Skripts folgende Funktion ein.

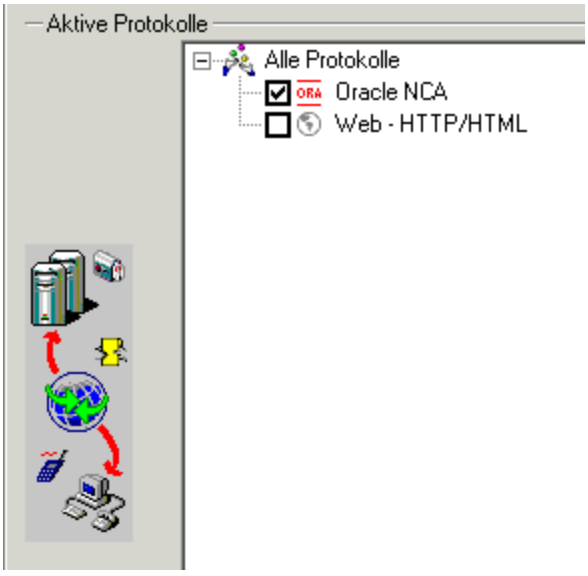
```
web_set_sockets_option("SSL_VERSION","3");
```

Testen von Servlets und anderen Oracle NCA-Anwendungen

Bestimmte NCA-Sitzungen verwenden Servlets:

- das Forms Listener-Servlet
- Anwendungen oder Module, die sowohl NCA- als auch HTTP-Kommunikation verwenden, beispielsweise Oracle Configurator
- die Initialisierung der NCA-Anwendung (Herunterladen der Applet-, JAR- und GIF-Dateien)

Beim Aufzeichnen von Servlets müssen Sie sowohl die Oracle NCA- als auch die Webfunktionen aufzeichnen. Verwenden Sie hierzu das Protokoll Oracle Apps 11i oder erstellen Sie ein Oracle NCA-Mehrfachprotokollskript. Wenn Sie bereits ein Skript mit einem Einzelprotokoll für Oracle NCA erstellt haben, können Sie alternativ in den Aufzeichnungsoptionen den Knoten **Allgemein > Protokolle** öffnen und das Webprotokoll aktivieren. Anschließend können Sie mit der Aufzeichnung beginnen.



Wenn Sie sich nicht sicher sind, ob Ihre Anwendung Servlets verwendet, überprüfen Sie nach der Aufzeichnung eines Skripts die Datei **default.cfg** im Skriptordner. Suchen Sie den Eintrag **"UseServletMode="**.

Wenn der Wert **1** oder **2** lautet, werden Servlets verwendet und Sie müssen zusätzlich zu Oracle NCA die HTTP-Aufzeichnung aktivieren.

Wenn Sie bereits ein Skript aufgezeichnet haben, können Sie den Code automatisch erneut erzeugen, um die Webfunktionen ohne erneute Aufzeichnung einzuschließen. Wählen Sie **Aufzeichnen > Skript neu generieren** und anschließend im Abschnitt **Protokolle** das Webprotokoll aus.

Bestimmen des Aufzeichnungsmodus

Wenn Sie Oracle NCA-Skripts aufzeichnen: VuGen bestimmt automatisch den richtigen Verbindungsmodus: HTTP- oder Socket-Modus. Generell müssen Sie keine Aufzeichnungseinstellungen ändern, da VuGen die Systemkonfiguration automatisch erkennt (es sei denn, Sie arbeiten mit Forms Server 4.5). In Systemen, in denen die standardmäßige Port-Zuordnung von anderen Anwendungen reserviert ist, müssen Sie die Einstellungen für die Port-Zuordnung möglicherweise abhängig vom Aufzeichnungsmodus ändern.

Zum Bestimmen des Aufzeichnungsmodus setzen Sie eine der folgenden Methoden ein:

- Wenn Sie die NCA-Anwendung verwenden, öffnen Sie die Java-Konsole.

```
proxyHost=null
proxyPort=0
connectMode=HTTP
Forms Applet version is: 60812
```

Mit dem Eintrag **connectMode** wird **HTTP**, **HTTPS** oder **socket** angegeben.

- Nach dem Aufzeichnen einer NCA-Sitzung öffnen Sie die Datei **default.cfg** im Vuser-Ordner und überprüfen den Wert des Eintrags **UseHttpConnectMode**.

```
[HttpConnectMode]
UseHttpConnectMode= 2
// 0 = socket 1 = http 2 = https
```

Wenn Sie im Dialogfeld **Servereintrag** eine neue Portzuordnung definieren, verwenden Sie im HTTP- oder HTTPS-Modus eine HTTP-**Dienst-ID**. Im Socket-Modus verwenden Sie eine NCA-**Dienst-ID**.

Weitere Informationen zu den Einstellungen für die Portzuordnung finden Sie unter "Netzwerk > Portzuordnung (Knoten)" auf Seite 224.

Aufzeichnen von Trace-Informationen für Oracle DB

Zum Debuggen des Skripts können Sie die Aufschlüsselungsdiagramme von Oracle DB verwenden. Um Daten für dieses Diagramm zu sammeln, aktivieren Sie vor dem Ausführen des Skripts den Trace-Mechanismus.

Um den Trace-Mechanismus manuell zu aktivieren, verwenden Sie die Funktion **nca_set_custom_dbtrace**. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

RDP-Protokoll

RDP-Protokoll – Übersicht

Hinweis: Dieses Thema bezieht sich nur auf RDP-Vuser-Skripts.

Das Microsoft RDP-Protokoll (Remote Desktop Protocol) ermöglicht einem Computer [Client] das Herstellen einer Verbindung zu einem anderen Computer [Server] über eine Netzwerkverbindung. So können Sie mit dem RDP-Protokoll beispielsweise eine Verbindung zu einem zentralen, leistungsstarken Server herstellen, um mit besonderen Geschäftsanwendungen oder Grafikterminals zu arbeiten. Bei der Verwendung dieses Protokolls profitieren Sie von der gleichen Darstellung und dem gleichen Verhalten wie beim Arbeiten an einem Standalone-PC. Der Clientcomputer verwendet für diesen Zweck RDP-Clientsoftware, während auf dem anderen Computer RDP-Serversoftware ausgeführt werden muss. Die Clientsoftware wird als **Remotedesktopverbindung** bezeichnet. Die Serversoftware wird als **Remotedesktopdienste** bezeichnet.

Weitere Informationen zu den von VuGen unterstützten RDP-Versionen finden Sie in der Matrix der Produktverfügbarkeit von LoadRunner.

Hinweis: Die RDP-Versionen ab 5.1 enthalten die Registerkarte **Experience**, auf der Sie verschiedene Optionen einstellen können. Diese Registerkarte wird beim Aufzeichnen mit VuGen allerdings nicht unterstützt. Alle Optionen befinden sich in der Position **ON**.

RDP-Aufzeichnungstipps

Hinweis: Dieses Thema bezieht sich nur auf RDP-Vuser-Skripts.

Gehen Sie beim Aufzeichnen eines RDP-Vuser-Skripts wie folgt vor, um ein möglichst wirksames Skript zu erhalten.

Einzel- und Mehrfachprotokoll-Skripts

Wenn Sie ein neues Skript erstellen, können Sie zwischen einem Einzel- und einem Mehrfachprotokoll wählen. Wenn Sie beispielsweise RDP-Verkehr und Web-Antworten aufzeichnen möchten, erstellen Sie ein Mehrfachprotokoll-Skript für RDP und Web, um das Aufzeichnen beider Protokolle zu ermöglichen.

Aufzeichnung in den richtigen Abschnitten

Zeichnen Sie den Verbindungsprozess im Abschnitt **vuser_init** und den Trennungsprozess im Abschnitt **vuser_end** auf. Dadurch werden Iterationen beim Herstellen und Trennen der Verbindung zu den Prozessen verhindert. Weitere Informationen zum Aufzeichnen in Abschnitten finden Sie unter "Lösungs-Explorer - Übersicht" auf Seite 57.

Eine leere Sitzung ausführen

Vergewissern Sie sich beim Aufzeichnen einer Sitzung, dass Sie den gesamten Geschäftsprozess von Herstellung der Verbindung bis zur Bereinigung ausführen. Beenden Sie Ihre Sitzung immer an

einem Punkt, an dem Sie den gesamten Prozess wieder von Beginn an starten könnten. Lassen Sie keine Client- oder Anwendungsfenster geöffnet.

Sie sollten Ihren Terminalserver so konfigurieren, dass Sitzungen mit einer getrennten Verbindung beendet werden. Wählen Sie **Administrative Tools > Terminal Services Configuration > Connection Properties > Sessions > Override User Settings** aus und stellen Sie den Server so ein, dass getrennte Sitzungen beendet werden.

Explizite Mausklicks

Wenn Sie auf die Optionen eines erweiterten Menüs zugreifen möchten, klicken Sie explizit auf jede einzelne Option und verlassen sich nicht auf das erweiterte Menü. Vergewissern Sie sich beispielsweise bei der Auswahl von **Start > Programme > Microsoft Word**, dass Sie explizit auf **Programme** klicken.

Verwenden von Windows-Logo-Tastenkombinationen

Hinweis: Dieser Tipp gilt nur für Windows 8-Installationen auf Remotecomputern.

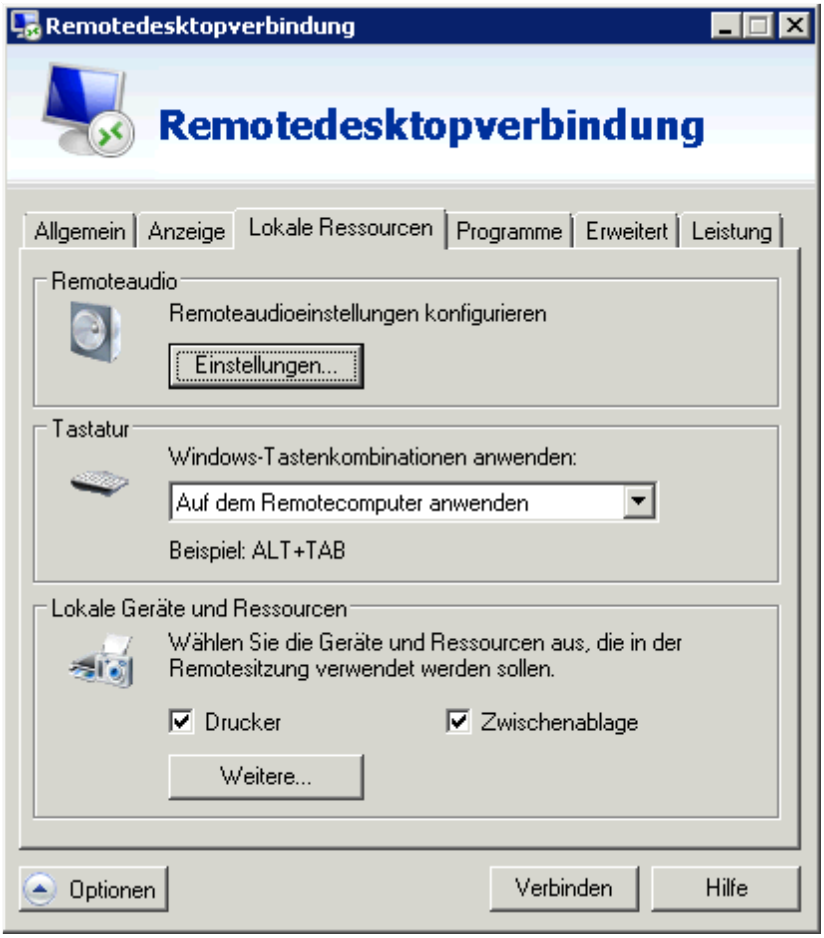
Da die Unterstützung für Mausbewegungen bei RDP-Vusern Leistungsprobleme verursachen kann, ist diese Unterstützung standardmäßig deaktiviert. Wenn Sie also ein RDP-Vuser-Skript aufzeichnen, empfiehlt es sich daher, die **Windows-Logo**-Tastenkombinationen zum Anzeigen des Startbildschirms [Windows-Logo-Taste], zum Anzeigen des Desktops [Windows-Logo-Taste + D] und zum Öffnen der Charms-Leiste [Windows-Logo-Taste + C] zu verwenden.

Wenn Sie ein RDP-Vuser-Skript ausführen, können die Windows-Tastenkombinationen entweder auf dem Host- oder auf dem Remotecomputer angewendet werden. Wenn Sie beim Aufzeichnen der Verbindung zum Remotecomputer sicherstellen möchten, dass die Windows-Tastenkombinationen auf dem Remotecomputer angewendet werden, müssen Sie dies entsprechend festlegen.

Anwenden der Windows-Tastenkombinationen auf dem Remotecomputer

1. Öffnen Sie das Dialogfeld **Remotedesktopverbindung**.
2. Klicken Sie auf **Optionen**, um das Dialogfeld zu erweitern.
3. Klicken Sie auf die Registerkarte **Lokale Ressourcen**.
4. Wählen Sie unter **Tastatur** aus der Liste **Windows-Tastenkombinationen anwenden** die

Option **Auf dem Remotecomputer anwenden**.



Synchronisieren mit Windows 8-Anwendungen

Hinweis: Dieser Tipp gilt nur für Windows 8-Installationen auf Remotecomputern.

Da viele Windows 8-Anwendungen dynamische Benutzeroberflächen aufweisen, sollten Sie diese Anwendungen nicht für Image-basierte Synchronisierungen verwenden.

Arbeiten mit Daten in der Zwischenablage (RDP-Protokoll)

Hinweis: Dieses Thema bezieht sich nur auf RDP-Vuser-Skripts.

Mit VuGen können Sie den Text in der Zwischenablage während einer RDP-Sitzung kopieren und einfügen. Sie können die Inhalte lokal kopieren und über eine Remote-Verbindung einfügen oder umgekehrt die Inhalte auf einem Remote-Computer kopieren und lokal einfügen. Unterstützt wird das Kopieren von Text in den Formaten TEXT, LOCALE und UNICODE.

VuGen generiert separate Funktionen, sobald Daten aus der Zwischenablage kopiert oder gespeichert werden.

Codebeispiel 1

Das folgende Beispiel beschreibt das Kopieren auf einem lokalen Computer und das Einfügen auf einem Remote-Computer:

```
//Notifies the Remote Desktop that new data is available in the Local machine's
//clipboard.The data can be provided in three formats: TEXT, UNICODE and LOCALE
rdp_notify_new_clipboard_data(
"StepDescription=Send local clipboard formats 1",
"Snapshot=snapshot1.inf",
"FormatsList=RDP_CF_TEXT|RDP_CF_UNICODE|RDP_CF_LOCALE",
RDP_LAST );
rdp_key(
"StepDescription=Key Press 2",
"Snapshot=snapshot_9.inf",
"KeyValue=V",
"KeyModifier=CONTROL_KEY",
RDP_LAST );
//Provides clipboard data to the Remote Desktop when it requests the data.
rdp_send_clipboard_data(
"StepDescription=Set Remote Desktop clipboard 1",
"Snapshot=snapshot1.inf",
"Timeout=20",
REQUEST_RESPONSE, "Format=RDP_CF_UNICODE", "Text=text for clipboard",
RDP_LAST);
```

Codebeispiel 2

Das folgende Beispiel beschreibt das Kopieren auf einem Remote-Computer und das Einfügen auf einem lokalen Computer:

```
rdp_key(
"StepDescription=Key Press 2",
"Snapshot=snapshot_9.inf",
"KeyValue=C",
"KeyModifier=CONTROL_KEY",
RDP_LAST);
// The function requests the Remote Desktop UNICODE text and saves it to a //par
ameter
rdp_receive_clipboard_data(
"StepDescription=Get Remote Desktop clipboard 1",
"Snapshot=snapshot1.inf",
"ClipboardDataFormat=RDP_CF_UNICODE",
```

```
"ParamToSaveData=MyParam",
RDP_LAST);
```

Normalerweise werden die Daten aus der Zwischenablage des Remote-Desktop-Computers im Format UNICODE gespeichert. Wenn der Remote-Desktop-Computer Daten im Format TEXT oder LOCALE abfragt, konvertiert die Funktion **rdp_send_clipboard_data** den Inhalt von MyParam aus UNICODE automatisch in das erforderliche Format und sendet es an den Remote-Desktop-Computer. Das Wiedergabeprotokoll zeigt diese Konvertierung mit einer entsprechenden Meldung an. Wenn die Konvertierung nicht möglich ist, scheitert der Schritt.

Weitere Informationen zu den rdp-Funktionen finden Sie unter Function Reference (**Hilfe > Funktionsreferenz**).

Korrelieren von Zwischenablageparametern

Wenn der Client dem Server während einer Aufzeichnungssitzung die gleichen Daten sendet, die er empfangen hat, ersetzt VuGen die gesendeten Daten bei der Codegenerierung durch einen Parameter. VuGen führt diese Korrelation nur durch, wenn die empfangenen und gesendeten Datenformate übereinstimmen.

Codebeispiel 3

Die folgenden Beispiele zeigen, wie der Parameter **MyParam** zum Empfangen und Senden von Daten verwendet wird.

```
// Receive the data from the server
rdp_receive_clipboard_data("StepDescription=Get Remote Desktop clipboard 1",
"Snapshot=snapshot_9.inf",
"Timeout=0",
"ClipboardDataFormat=RDP_CF_UNICODETEXT",
"ParamToSaveData=MyParam",
RDP_LAST);
...
// Send the data to the server
rdp_send_clipboard_data("StepDescription=Get Remote Desktop clipboard 1",
"Snapshot=snapshot_9.inf",
"Timeout=10",
REQUEST_RESPONSE, "Format=RDP_CF_UNICODETEXT", "Text={MyParam}",
RDP_LAST);
```

RDP-Snapshots - Übersicht

Hinweis: Dieses Thema bezieht sich nur auf RDP-Vuser-Skripts.

Auf dem RDP-Protokoll basierende Vuser-Skripts verwenden den Snapshot-Bereich von VuGen.

- Eine Einführung zum Snapshot-Bereich finden Sie unter "Snapshot-Bereich - Übersicht" auf Seite 62.
- Details zum Arbeiten mit dem Snapshot-Bereich finden Sie unter "Arbeiten mit Snapshots" auf Seite 65.
- Details zur Benutzeroberfläche des Snapshot-Bereichs finden Sie unter "Snapshot-Bereich" auf Seite 109.

Wenn Sie ein RDP-Vuser-Skript erstellen, stehen die Standardfunktionen des Snapshot-Bereichs von VuGen zur Verfügung. Im Snapshot-Bereich werden Snapshots der Remote-Anzeige angezeigt, die während der Aufzeichnung und Wiedergabe des Vuser-Skripts gespeichert wurden. Diese Snapshots werden normalerweise verwendet, um die Wiedergabe des Vuser-Skripts zu synchronisieren.

Neben dem Ausführen von Standardfunktionen ermöglicht der Snapshot-Bereich für RDP-Vuser-Skripts auch das Anzeigen von Snapshots in einer der folgenden Ansichten:

- **Bild.** Zeigt nur das Bild des Snapshots an und ist ideal geeignet, um zwei Bilder visuell zu vergleichen. Diese Ansicht zeigt den Snapshot schneller an und erfordert weniger Speicherplatz als die vollständige Ansicht. Sie können zwei Snapshots im Snapshot-Bereich synchronisieren, wenn beide Snapshots in der Bildansicht angezeigt werden. Die Bildansicht führt keinen automatischen Bildlauf durch, um den Bereich eines Snapshots anzuzeigen, der für die Synchronisierung verwendet wird.
- **Vollständig.** Führt einen Bildlauf durch und zeigt den Bereich an, der für die Synchronisierung verwendet wird. Diese Ansicht zeigt den Snapshot langsamer an und erfordert mehr Speicherplatz als die Bildansicht. Sie können zwei im Snapshot-Bereich angezeigte Snapshots synchronisieren, wenn einer der Snapshots in der vollständigen Ansicht angezeigt wird. Standardmäßig werden Snapshots in der vollständigen Ansicht angezeigt.

Um einen bestimmten Synchronisierungs-Snapshot im Snapshot-Bereich anzuzeigen, führen Sie eine der folgenden Aktionen aus:

- Wählen Sie im Editor den Schritt aus, der eine Referenz zu dem Snapshot enthält.
- Doppelklicken Sie im Schrittenavigator auf den Schritt, der eine Referenz zu dem Snapshot enthält.

Wenn Sie mit RDP-Vuser-Skripts arbeiten, können Sie im Snapshot-Bereich einen Snapshot in die Zwischenablage kopieren und einen Snapshot des letzten Wiedergabefehlers anzeigen. Weitere Informationen zur Verwendung des Snapshot-Bereichs finden Sie unter "Arbeiten mit Snapshots" auf Seite 65.

Übersicht über die Bildsynchronisierung

Hinweis: Dieses Thema bezieht sich nur auf RDP-Vuser-Skripts.

Eine RDP-Sitzung wird remote ausgeführt. Alle Tastatur- und Mausaktivitäten erfolgen über den Server, und der Server reagiert auf die entsprechenden Befehle. Wenn Sie beispielsweise eine Anwendung mit einem Doppelklick auf dem Desktop aktivieren, dann registriert der Server diesen Doppelklick, lädt die Anwendung herunter und zeigt sie an.

Wenn ein RDP-Client eine Verbindung zum Server herstellt, geschieht zweierlei:

- Der Client sendet die Aktionskoordinaten an den Server. Ein Beispiel: 'Linker Mausklick auf die Bildschirmkoordinaten (100, 100)'.
- Der Client empfängt Bilder von dem Server, die den aktuellen Status auf dem Bildschirm anzeigen, nachdem die Aktion stattgefunden hat.

Der RDP-Client (und dementsprechend LoadRunner) weiß nicht, dass der Bildschirm Fenster, Schaltflächen, Symbole und andere Objekte enthält. Er weiß nur, dass der Bildschirm eine Grafik enthält und er kennt die Koordinaten, an denen der Benutzer die Aktion durchgeführt hat. Wenn der Server die Aktionen korrekt interpretieren soll, müssen Sie innerhalb des Skripts Synchronisierungspunkte festlegen. Diese Punkte weisen das Skript an, mit dem Fortfahren zu warten, bis der Bildschirm auf dem Server mit dem gespeicherten Bildschirm übereinstimmt.

Wenn Sie das Skript das nächste Mal wiedergeben, wartet es, bis das vom Server zurückgegebene Bild mit dem ausgewählten Bild übereinstimmt.

Tipps zur Bildsynchronisierung (RDP-Protokoll)

Hinweis: Dieses Thema bezieht sich nur auf RDP-Vuser-Skripts.

Befolgen Sie die nachstehenden Anweisungen, um eine wirksame Bildsynchronisierung zu gewährleisten:

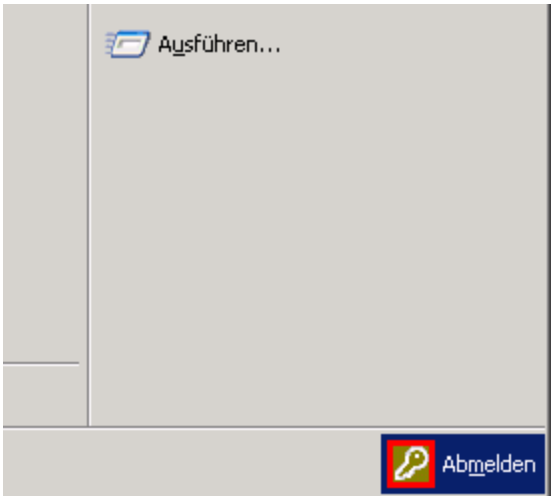
Synchronisieren des kleinsten signifikanten Bereichs

Versuchen Sie beim Synchronisieren eines Bilds nur den tatsächlich relevanten Ausschnitt zu markieren. Einzelne Bilddetails werden bei der Wiedergabe u. U. nicht reproduziert und können zu Synchronisierungsfehlern führen.

Wenn Sie beispielsweise das Bild einer Schaltfläche synchronisieren, wählen Sie nur den Text und nicht die gepunkteten Linien um den Text, da diese bei der Wiedergabe möglicherweise nicht angezeigt werden.

Beim Synchronisieren eines unterlegten Bereichs müssen Sie versuchen, nur den Bildausschnitt zu erfassen, der nicht unterlegt ist. Im folgenden Beispiel führen Sie eine Synchronisierung des Log

Off-Symbols durch, wobei allerdings nicht die gesamte Schaltfläche berücksichtigt wird, da der unterlegte Bereich bei der Wiedergabe u. U. nicht angezeigt wird und die Farbe aufgrund unterschiedlicher Farbschemata variieren könnte.



Synchronisieren vor jeder Benutzeraktion

Es empfiehlt sich, vor jeder Mausoperation eine Synchronisierung durchzuführen. Sie müssen die Synchronisierung zudem vor der ersten Operation des Typs **rdp_key** oder **rdp_type** durchführen, die auf eine Mausoperation folgt.

Bildsynchronisierung - Geänderte Koordinaten (RDP-Protokoll)

Hinweis: Dieses Thema bezieht sich nur auf RDP-Vuser-Skripts.

Bei der Wiedergabe eines Skripts kann ein aufgezeichnetes Objekt an verschiedenen Koordinaten auf dem Bildschirm erscheinen. Das Objekt ist immer das gleiche, nur seine Position wurde geändert. So kann ein Fenster beim Aufzeichnen beispielsweise an den Koordinaten (100, 100) und bei der Wiedergabe an den Koordinaten (200, 250) geöffnet werden.

In diesem Fall findet der Synchronisierungspunkt die neuen Koordinaten automatisch und ohne Eingreifen des Benutzers. Der Punkt registriert automatisch eine Abweichung von 100 Pixeln auf der horizontalen Achse und 150 Pixeln auf der vertikalen Achse.

Alle nachfolgenden Mausoperationen verwenden die geänderten Koordinaten, so dass ein an der Position (130, 130) aufgezeichneter Mausklick an der Position (230, 280) = (130 + 100, 130 + 150) wiedergegeben wird.

Die Änderung von Koordinaten wird über den Parameter **AddOffsetToInput** im Schritt **rdp_sync_on_image** gesteuert. Sie können diesen Parameter überschreiben, um die Abweichung der Position bei der Wiedergabe in den aufgezeichneten Koordinaten für eine spätere Verwendung hinzuzufügen. Wenn Sie diesen Parameter nicht überschreiben, übernimmt VuGen diesen Wert aus den Standardeinstellungen für die Laufzeit.

Der entsprechende Parameter in den Operationen (z. B. **rdp_mouse_click** oder **rdp_mouse_drag**) lautet **Origin**. Dieser Parameter entscheidet darüber, ob die Operation die Koordinaten aus den sauber aufgezeichneten Werten übernehmen oder die Abweichungen berücksichtigen soll, die von dem letzten Synchronisierungspunkt hinzugefügt wurden. Ohne ausdrückliche Vorgaben übernimmt VuGen den Wert für diesen Parameter aus den Laufzeiteinstellungen.

Festlegen der Sicherheitsebenen in RDP-Vuser-Skripts

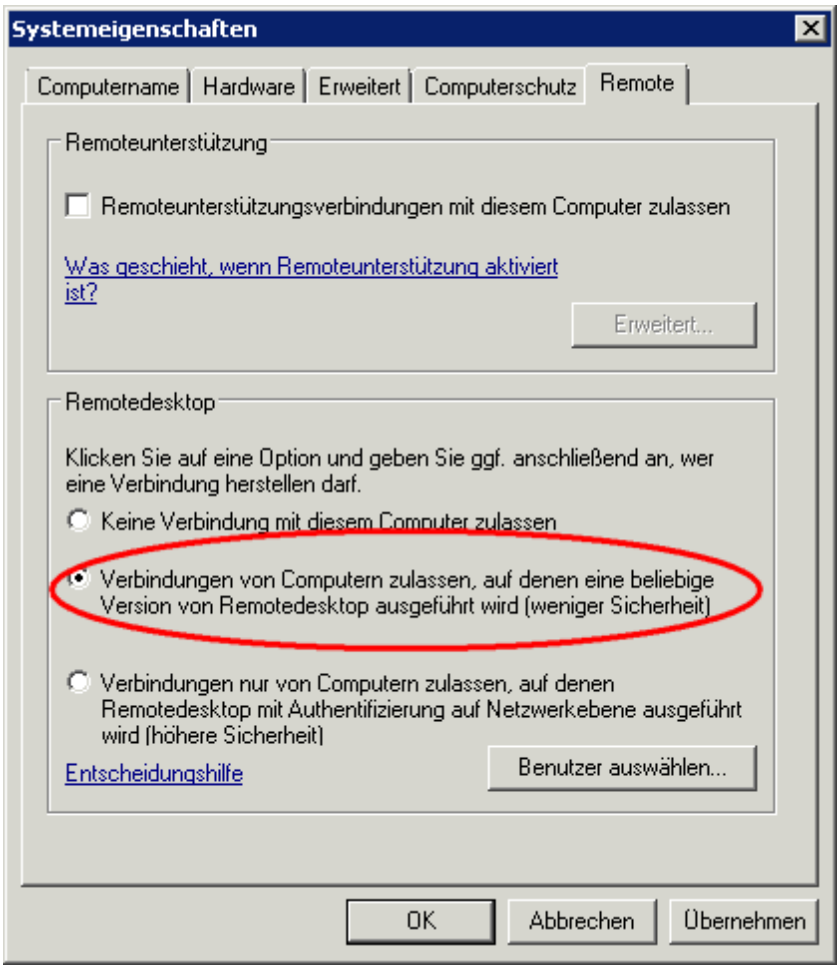
Hinweis: Dieses Thema bezieht sich nur auf RDP-Vuser-Skripts.

Mit dem RDP-Protokoll (Remote Desktop Protocol) kann ein Clientcomputer eine Verbindung zu einem Server herstellen. Für die Verbindung stehen verschiedene Sicherheitsoptionen zur Verfügung, die von den jeweiligen Windows-Betriebssystemen abhängen, die auf den Client- und Servercomputern installiert sind. Mit den Sicherheitsoptionen werden sicherheitsbezogene Themen definiert, beispielsweise die Authentifizierung und Verschlüsselung, die für die Verbindung verwendet werden.

Die Liste der für ein Vuser-Skript verfügbaren Sicherheitsoptionen ist unterschiedlich, wenn Sie ein Vuser-Skript aufzeichnen und wenn Sie das Skript wiedergeben.

Sicherheitsebenen beim Aufzeichnen eines RDP-Vuser-Skripts

Die standardmäßige RDP-Sicherheit ist die einzige Sicherheitsvariante, die Sie beim Aufzeichnen eines RDP-Vuser-Skripts verwenden können. Bevor Sie ein RDP-Vuser-Skript aufzeichnen, stellen Sie sicher, dass der Server so konfiguriert ist, dass Verbindungen von Computern zulässig sind, auf denen eine beliebige Version von Remote Desktop ausgeführt wird, und nicht nur von Computern, auf denen Remote Desktop mit Authentifizierung auf Netzwerkebene ausgeführt wird. Im Dialogfeld **Systemeigenschaften** auf dem Server können Sie auf der Registerkarte **Remote** die zum Herstellen der Verbindung erforderliche Sicherheitsebene festlegen.



Sicherheitsebenen beim Wiedergeben eines RDP-Vuser-Skripts

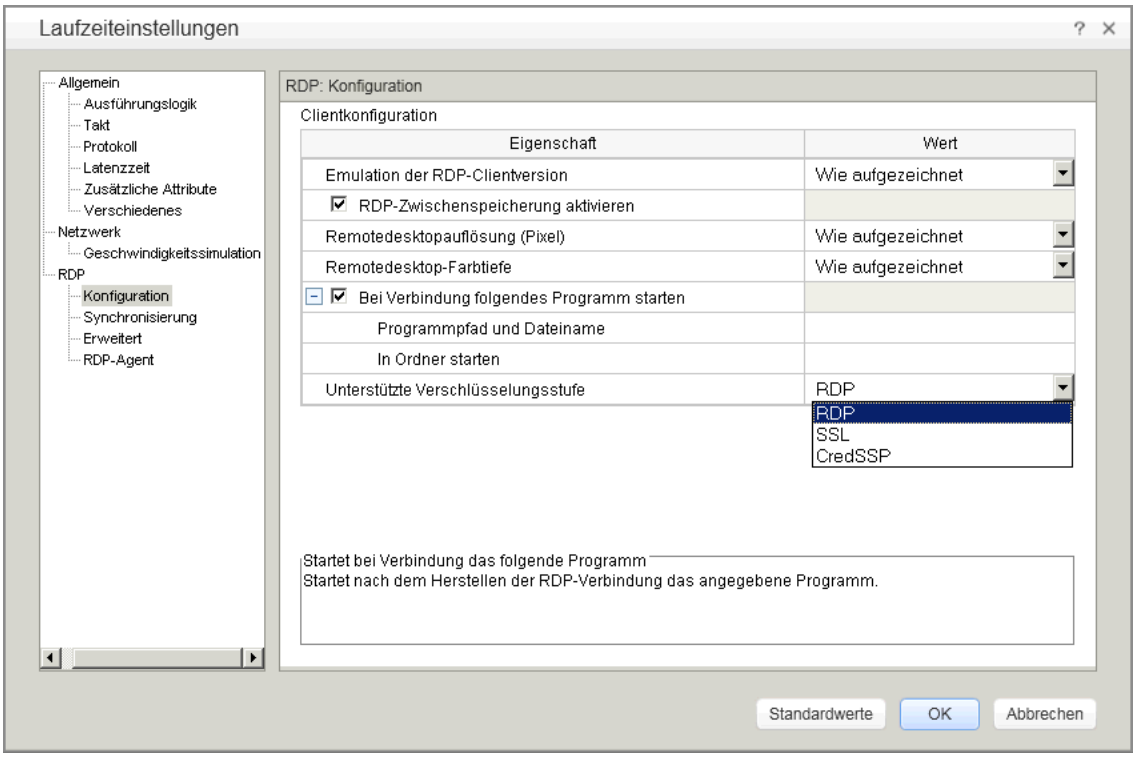
Sie können die Laufzeiteinstellungen des Vuser-Skripts verwenden, um die Sicherheitseinstellungen für die Verbindung beim Ausführen des Vuser-Skripts vorzunehmen. Folgende Sicherheitsebenen sind verfügbar:

- **RDP:** Stellt eine Verbindung mit der standardmäßigen RDP-Sicherheit her. RDP stellt die am wenigsten sichere Verbindung bereit.
- **SSL:** Stellt eine Verbindung mit SSL als externem Sicherheitsprotokoll her, um die standardmäßige RDP-Sicherheit zu verbessern. SSL bietet eine mittlere Sicherheitsebene.
- **CredSSP:** Stellt eine Verbindung mit dem CredSSP-Protokoll (Credential Security Support Provider) her. CredSSP bietet die sicherste Verbindung.

> **Hinweis:** Wenn Sie die CredSSP-Authentifizierung festlegen, müssen Sie bei jeder Neugenerierung des Vuser-Skripts bestimmte Änderungen am Skript vornehmen. Weitere Details finden Sie nachstehend unter Ändern eines Skripts für die Unterstützung der CredSSP-Authentifizierung.

Die Sicherheitsebene, die Sie in den Laufzeiteinstellungen festlegen, gibt dem Server Hinweise auf die maximale, vom Client unterstützte Sicherheitsebene. Die tatsächlich für die Verbindung verwendete Sicherheit wird jedoch durch die Servereinstellungen definiert. Wenn Sie beispielsweise CredSSP als Verschlüsselungsebene in den Laufzeiteinstellungen festlegen, informiert beim Ausführen des Vusers dieser den Server, dass der Vuser die CredSSP-, SSL- und RDP-Sicherheit unterstützt. Wenn der Server nur die RDP-Sicherheit unterstützt (z. B. weil das Betriebssystem Windows 2003 ist), wird die Verbindung mit RDP vorgenommen.

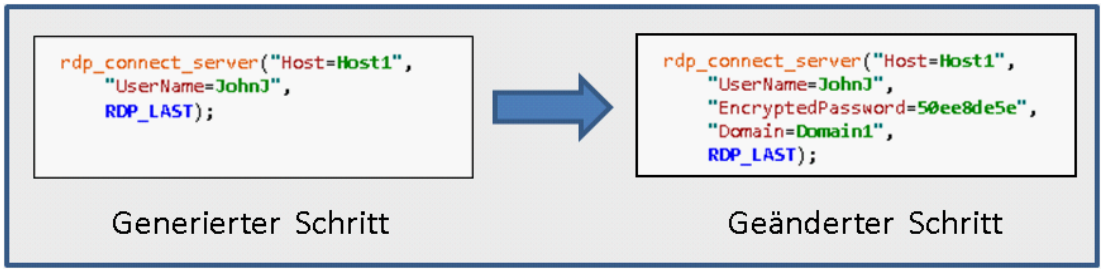
Um die RDP-Sicherheitsebene für das Vuser-Skript festzulegen, klicken Sie auf **Wiedergeben > Laufzeiteinstellungen > RDP > Konfiguration** und wählen dann die gewünschte Ebene aus der Liste **Unterstützte Verschlüsselungsstufe** aus.



Ändern eines Skripts für die Unterstützung der CredSSP-Authentifizierung

Wenn Sie die CredSSP-Authentifizierung in den Laufzeiteinstellungen des Vuser-Skripts festlegen, müssen Sie bei jeder Neugenerierung des Skripts die folgenden Aufgaben ausführen:

1. Ändern Sie im Schritt **rdp_connect_server** den Schritt so, dass der Benutzername, das Kennwort und die Domäne angegeben werden, die für den Zugriff auf den Server erforderlich sind. Weitere Informationen zum Schritt **rdp_connect_server** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).



2. Entfernen Sie den Codeblock, der die anmeldebezogenen Maus-, Tastatur- und Bildsynchronisierungsschritte enthält, wie nachstehend beschrieben aus dem generierten Skript.

   a. Suchen Sie im Vuser-Skript nach dem Schritt **rdp_connect_server**.

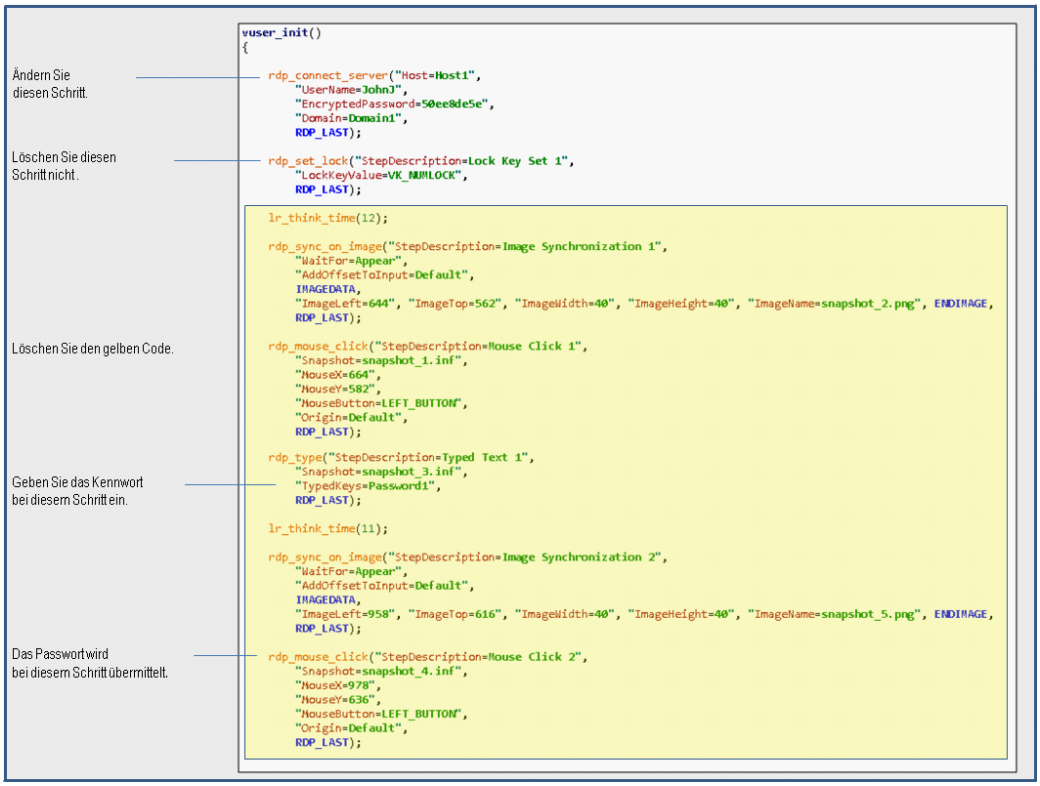
      Der Schritt nach dem Schritt **rdp_connect_server** ist der erste zu löschende Schritt im Codeblock.

   b. Suchen Sie nach dem Schritt **rdp_mouse_click** oder nach dem Schritt **rdp_key**, mit dem das Kennwort an den Server gesendet wird.

      Dies ist der letzte zu löschende Schritt im Codeblock.

      Hinweis: Wenn ein Schritt **rdp_set_lock** unmittelbar nach dem Schritt **rdp_connect_server** vorhanden ist, löschen Sie den Schritt **rdp_set_lock** nicht.

c. Löschen Sie alle Schritte im oben definierten Codeblock.



RDP-Agent (Agent für Microsoft Terminal Server) – Übersicht

Hinweis: Dieses Thema bezieht sich nur auf RDP-Vuser-Skripts.

Der Agent für Microsoft Terminal Server ist ein optionales Dienstprogramm zur Installation auf dem RDP-Server. Dieses optimiert die reguläre RDP-Funktionalität. Es befindet sich auf der Installations-DVD von LoadRunner und kann auf jedem beliebigen RDP-Server installiert werden. Dieser Agent bietet intuitive und lesbare Skripts, integrierte Synchronisierung, detaillierte Informationen über relevante Objekte. Hinweis: Wenn Sie RDP-Vuser mit einem installierten Agenten ausführen, führt jeder Vuser seinen eigenen **lrrpagent.exe**-Prozess aus. Dies führt zu einer leichten Verringerung der Anzahl der Vuser, die auf dem Servercomputer ausgeführt werden können.

Tipps zum Verwenden des Agenten für Microsoft Terminal Server

- Wenn Sie das Menü einer Anwendung (z. B. **Datei**, **Bearbeiten**…) mit der Maus öffnen, schlagen die Synchronisierungsschritte manchmal fehl. Um dieses Problem zu vermeiden, wählen Sie beim Aufzeichnen die Menüelemente über die Tastatur aus.

- Wenn Sie einen Schritt des Typs **sync_on_object_mouse_click** manuell hinzufügen, geben die Koordinaten die absolute Position (mit Bezug auf den gesamten Bildschirm) an. Um den Synchronisierungspunkt zu erstellen, müssen Sie im Fenster (relative Koordinaten) die Abweichung von der gewünschten Klickposition berechnen und die absoluten Koordinaten entsprechend ändern, um eine erfolgreiche Wiedergabe der Synchronisierung zu gewährleisten.

- Wenn sich ein Synchronisierungsobjekt bei der Wiedergabe zum rechten Zeitpunkt an der richtigen Position befindet, jedoch von einem anderen Fenster (wie z. B. ein Popup) überdeckt wird, wird der Synchronisierungsschritt übersprungen und ein Mausklick in dem Fenster ausgeführt, das den Synchronisierungspunkt abdeckt und den Skriptfluss somit behindert.

- Wenn Sie das Anwendungsfenster während der Aufzeichnung in den Vordergrund stellen möchten, klicken Sie entweder auf die Titelleiste oder verwenden Sie die Tastatur (ALT+TAB). Hinweis: Wenn Sie in das Anwendungsfenster klicken, um es wieder in den Vordergrund zu bringen, wird die RDP-Sitzung möglicherweise abrupt beendet.

Der Agent für Microsoft Terminal Server bietet die folgenden Verbesserungen gegenüber der RDP-Standardfunktionalität:

Detaillierte Aufzeichnung von Objekten

Wenn der Agent für Microsoft Terminal Server installiert ist, zeichnet VuGen anstelle von allgemeinen Information zur Aktion spezifische Informationen zum aktiven Objekt auf. VuGen erzeugt beispielsweise die Schritte **sync_object mouse_click** und **sync_object_mouse_double_click** anstelle der mit dem Agenten erstellten Schritte **mouse_click** und **mouse_double_click**.

Das folgende Beispiel demonstriert die Aufzeichnung einer Doppelmausklickaktion mit und ohne Agenteninstallation. Beachten Sie, dass VuGen mit dem Agenten für alle Mausaktionen Funktionen des Typs **sync_object** generiert.

```
rdp_sync_object_mouse_double_click("StepDescription=Mouse Double Click on Synchr
onized Object 1",
        "Snapshot=snapshot_12.inf",
        "WindowTitle=RDP2",
        "Attribute=TEXT",
        "Value=button1",
        "MouseX=100",
        "MouseY=71",
        "MouseButton=LEFT_BUTTON",
        RDP_LAST);
rdp_mouse_double_click("StepDescription=Mouse Double Click 1",
        "Snapshot=snapshot_2.inf",
        "MouseX=268",
        "MouseY=592",
        "MouseButton=LEFT_BUTTON",
        "Origin=Default",
        RDP_LAST);
```

Erweitertes Kontextmenü

Wenn Sie in einen Snapshot klicken, können Sie über das Kontextmenü verschiedene Funktionen

in das Skript einfügen. Wenn der Agent nicht aktiv ist, können Sie nur die Schritte **mouse_click**, **mouse_double_click** und **sync_on_Image** einfügen. Wenn der Agent installiert ist, können Sie alle möglichen Schritte einfügen, die einen RDP-Agenten einbeziehen:

- **get_object_info** und **Sync_on_object_info**. Stellt Informationen über den Status des Objekts zur Verfügung und synchronisiert eine spezifische Objekteigenschaft, wie z. B.: ENABLED, FOCUSED, CONTROL_ID, ITEM_TEXT, TEXT, CHECKED und LINES.
- **insert_sync_on_text** und **get_text**. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Codebeispiel

Im folgenden Beispiel bewirkt die Funktion **rdp_sync_on_object_info** eine Synchronisierung, da sie so lange abwartet, bis sich der Fokus auf dem Dialogfeld **Internetoptionen** befindet.

```
rdp_sync_on_object_info("StepDescription=Sync on Object Info 0",
        "Snapshot=snapshot_30.inf",
        "WindowTitle=Internet Options",
        "ObjectX=172",
        "ObjectY=155",
        "Attribute=FOCUSED",
        "Value={valueParam}",
        "Timeout=10",
        "FailStepIfNotFound=No",
        RDP_LAST);
```

Installieren/Deinstallieren des RDP-Agenten

Hinweis: Dieses Thema bezieht sich nur auf RDP-Vuser-Skripts.

Die Installationsdatei für den Microsoft Terminal Server-Agenten befindet sich auf dem LoadRunner-Installationsdatenträger im Ordner **Additional Components\Agent for Microsoft Terminal Server**.

Hinweis: Der Agent darf nur auf RDP-Servern, nicht auf Load Generator-Computern installiert werden.

Wenn Sie ein Upgrade des Agenten durchführen, vergewissern Sie sich vor der Installation, ob die Vorversion entfernt wurde (siehe nachfolgende Anweisungen zur Deinstallation).

Installieren des RDP-Agenten

1. Wenn auf Ihrem Server für die Installation von Software Administratorberechtigungen erforderlich sind, melden Sie sich als Administrator an.
2. Suchen Sie die Installationsdatei **Setup.exe** auf der LoadRunner-DVD im Ordner **Additional**

Components\Agent for Microsoft Terminal Server.

3. Folgen Sie den Anweisungen des Installations-Assistenten.

Sie müssen die Aufzeichnungsoptionen festlegen, bevor Sie den Agenten verwenden und ein Vuser-Skript aufzeichnen können. Klicken Sie im Dialogfeld **Aufzeichnung starten** auf **Optionen**. Aktivieren Sie im Knoten **Erweiterte Codegenerierung** die Option **RDP-Agent verwenden**.

Deinstallieren des RDP-Agenten

1. Wenn zum Deinstallieren von Software auf dem Server Administratorberechtigungen erforderlich sind, melden Sie sich als Administrator an.

2. Wählen Sie **Systemsteuerung > Software > HP Software Agent for Microsoft Terminal Server** aus, und klicken Sie dann auf **Ändern/Entfernen**.

Hinzufügen von Bildsynchronisierungspunkten zu einem Skript

Hinweis: Dieses Thema bezieht sich nur auf RDP-Vuser-Skripts.

1. Wählen Sie eine Operation aus, der Sie einen Synchronisierungspunkt im Skript hinzufügen möchten.

2. Klicken Sie mit der rechten Maustaste auf den Bild-Snapshot und wählen Sie im Menü **Synchronisierung bei Bild-Schritt einfügen** aus. Der Cursor wandelt sich in ein Fadenkreuz.

3. Markieren Sie den Bereich auf dem Bildschirm, für den Sie eine Synchronisierung durchführen möchten mit einem Linksklick und ziehen Sie einen Rahmen um den jeweiligen Bereich. Sobald Sie die Maustaste loslassen, wird das Dialogfeld **Sync on Image** geöffnet.

4. Klicken Sie auf **OK**. VuGen fügt einen neuen **Sync on Image**-Schritt vor dem ausgewählten Schritt hinzu. Wenn Sie diesen Schritt auswählen, zeigt VuGen einen Snapshot mit einem rosafarbenen Rahmen um den Bereich an, den Sie für die Synchronisierung ausgewählt haben.

Wenn Sie das Skript das nächste Mal wiedergeben, wartet es, bis das vom Server zurückgegebene Bild mit dem ausgewählten Bild übereinstimmt.

Dialogfeld "Fehler bei Bildsynchronisierung" (RDP-Protokoll)

Dieses Dialogfeld wird geöffnet, wenn eine Bildsynchronisierung während der Wiedergabe eines Skripts fehlschlägt. Sie können das Skript beenden oder die Wiedergabe trotz des Fehlers fortsetzen.

Zugriff	Wird automatisch bei einem Bildsynchronisierungsfehler geöffnet.
Wichtige Informationen	**Der Inhalt dieses Dialogfeldes ist je nach dem jeweiligen Grund für den Synchronisierungsfehler unterschiedlich:** • **Snapshot anhängen.** Das Dialogfeld **Fehler bei der Bildsynchronisierung - Snapshot anhängen** wird geöffnet, wenn das wiedergegebene Bild so stark von dem aufgezeichneten Bild abweicht, dass eine Änderung der Toleranzstufe nicht ausreicht. • **Toleranz erhöhen.** Kann das Skript das abgefragte Bild bei der Wiedergabe nicht finden, wird das Dialogfeld **Fehler bei der Bildsynchronisierung - Toleranz erhöhen** geöffnet. Wenn die Toleranzstufe beim Synchronisieren von Bildern angehoben wird, findet das Skript das gesuchte Bild auch, wenn es nicht ganz genau mit dem abgefragten Bild übereinstimmt. • **Toleranz verringern.** Wenn die Bedingung **NotAppear** bzw. **Change** bei der Wiedergabe des Skripts nicht erfüllt wird, wird das Dialogfeld **Fehler bei der Bildsynchronisierung - Toleranz verringern** geöffnet. VuGen erkennt eine Bildübereinstimmung dort, wo Sie nicht damit rechnen. Wenn die Toleranzstufe gesenkt würde, würden aufgezeichnetes und wiedergegebenes Bild nicht übereinstimmen. Die Bedingung **NotAppear** bzw. **Change** wird erfüllt und bewirkt eine erfolgreiche Wiedergabe. • **Ohne nähere Bestimmung.** Wenn keine der Synchronisierungsbedingungen, z. B. **NotAppear** oder **Change**, bei der Wiedergabe des Skripts erfüllt wird, wird das Dialogfeld **Fehler bei Bildsynchronisierung** ohne weitere Angaben geöffnet. VuGen hat an den ursprünglichen Koordinaten kein anderes Bild gefunden, das an das Skript angehängt werden könnte.
Siehe auch	Bildsynchronisierung - Übersicht [RDP-Protokoll]

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Anhalten	Die Abweichung wird als Fehler behandelt. Dieser Fehler wird wie alle anderen Fehler behandelt, sodass die Ausführung des Skripts angehalten wird.

Forts.

Weiter	Je nach Art des geöffneten Dialogfelds führt diese Schaltfläche unterschiedliche Aktionen aus: • **Snapshot anhängen.** Falsche Zuordnung annehmen. VuGen hängt den Wiedergabe-Snapshot als "Aufzeichnungs"-Snapshot für diesen Schritt an. Bei der zukünftigen Wiedergabe des Schrittes verwendet VuGen alle vorhandenen Aufzeichnungs-Snapshots und den angehängten Snapshot als Grundlage für den Vergleich von Bildschirmen. Wenn beim Wiedergeben einer der Aufzeichnungs-Snapshots zurückgegeben wird, wird kein Vuser-Fehler ausgegeben. Sie können den ursprünglichen und die angehängten Snapshots für einen Schritt anzeigen, indem Sie in der Symbolleiste des Snapshot-Bereichs auf die Navigationspfeile klicken. • **Toleranz verringern.** Falsche Zuordnung annehmen und Toleranzstufe senken, sodass VuGen bei der Wiedergabe eine geringere Abweichung zwischen den aufgezeichneten Bildern und den angezeigten Bildern toleriert. • **Toleranz erhöhen.** Falsche Zuordnung annehmen und Toleranzstufe erhöhen, sodass VuGen bei der Wiedergabe eine größere Abweichung zwischen den aufgezeichneten Bildern und den angezeigten Bildern toleriert. • **Ohne nähere Bestimmung.** Falsche Zuordnung annahmen und keine Änderungen am Skript vornehmen. Skriptausführung trotz der falschen Zuordnung fortsetzen. **Hinweis:** Durch das Anheben oder Absenken der Toleranzstufe in diesem Dialogfeld ändert sich nur die Stufe für den aktuellen Schritt. Wenn Sie die Toleranzstufe für das gesamte Skript ändern möchten, müssen Sie die Einstellung für die **Standardtoleranz für Bildsynchronisierung** unter **Laufzeiteinstellungen > RDP > Synchronisierung** ändern.

RDP - Fehlerbehebung und Einschränkungen

Hinweis: Dieses Thema bezieht sich nur auf RDP-Vuser-Skripts.

Dieser Abschnitt enthält Informationen zur Fehlerbehebung von RDP-Skripts, für die der Agent für Microsoft Terminal Server verwendet wurde.

Wiedergabefehler bei "rdp_sync_object_mouse_click-/double_click"-Schritten

Für Wiedergabefehler bei den Schritten **rdp_sync_object_mouse_click** bzw. **rdp_sync_object_mouse_double_click** sind Problemumgehungen vorhanden. Es wird empfohlen, die Problemumgehungen in der angegebenen Reihenfolge auszuprobieren.

Lösung: Bearbeiten der Datei "RDPAgentCodeGen.cfg"

Über die Datei **RDPAgentCodeGen.cfg** können Sie VuGen so konfigurieren, dass bei der

nächsten Skriptgenerierung für jeden **rdp_sync_object_mouse_click/double_click**-Schritt automatisch ein **rdp_sync_on_image**- und ein **rdp_mouse_click**-Schritt erstellt wird. Hierzu geben Sie den Namen des Fensters an, ändern den Wert einer Variable, die die Anzahl der Fenster speichert, für die dieser Prozess ausgeführt wird, und generieren das Skript neu.

Bearbeiten der Datei "RDPAgentCodeGen.cfg"

1. Öffnen Sie die Datei **RDPAgentCodeGen.cfg** im Ordner für die Skriptdaten.
2. Öffnen Sie den **Schrittenavigator**, und doppelklicken Sie dann auf den fehlgeschlagenen Schritt.
3. Kopieren Sie den Namen des Fensters.
4. Erhöhen Sie in der Datei **RDPAgentCodeGen.cfg** den Wert für **NumberOfTitles** um 1.
5. Fügen Sie eine Zeile in folgendem Format hinzu:

   ```
   WindowTitleX=<Name des Fensters>
   ```

   wobei **X** der neue Wert für **NumberOfTitles** ist.
6. Generieren Sie das Skript neu.

> **Hinweis:** Die Datei **RDPAgentCodeGen.cfg** kann eingesetzt werden, um **rdp_sync_on_image**- und **rdp_mouse_click** -Schritte auf ähnliche Weise automatisch für **rdp_sync_object_mouse_click/double_click** -Schritte zu produzieren, die ebenfalls auf unterschiedliche Weise angegeben werden. Die Schritte können auf Grundlage des Klassenattributs des Steuerelements spezifiziert werden. Wenn Sie weitere Informationen wünschen, setzen Sie sich bitte mit dem HP Software Support in Verbindung.

Lösung: Manuelles Einfügen eines neuen Schritts

Sie können die Problemumgehung auch manuell vornehmen. Fügen Sie einfach einen **rdp_sync_on_image**- und **rdp_mouse_click** -Schritt für jeden fehlgeschlagenen Schritt ein. Diese Vorgehensweise ist jedoch nicht empfehlenswert, da die hinzugefügten Schritte bei Neugenerierung des Skripts verloren gehen.

- Das gemeinsame Verwenden der Zwischenablage wird nur für kurze, einfache Texte unterstützt.
- Beim Aufzeichnen mit dem RDP-Agenten werden Anwendungen, die mit CBuilder entwickelt wurden, möglicherweise nicht fehlerfrei aufgezeichnet.
- RDP unterstützt keine 32-Bit-Farbtiefe. Wenn beim Aufzeichnen diese Farbtiefe verwendet wird, wechselt VuGen automatisch zu einer niedrigeren Farbtiefe, und im Aufzeichnungsfenster wird ein Warnungsprotokollelement angezeigt, das sinngemäß lautet: "[RDP Analyzer-Warnung (790: 418)] 32-Bit-Farbtiefe wird nicht unterstützt, Wechsel zu einer anderen".

RTE-Protokoll

Übersicht über das RTE-Protokoll

Ein RTE-Vuser gibt Zeichen in einen Terminalemulator ein, übermittelt die Daten an den Server und wartet dann auf die Antwort des Servers. Stellen wir uns als Beispiel einen Server vor, auf dem die Kundeninformationen eines Unternehmens verwaltet werden, das technische Wartungen ausführt. Jedes Mal, wenn ein Außendienstmitarbeiter eine Reparatur ausführt, greift er per Modem auf die Serverdatenbank zu und verwendet dabei einen Terminalemulator. Dabei ruft er Informationen über den Kunden ab und speichert die Details der ausgeführten Reparatur.

Sie könnten einen solchen Fall mithilfe von RTE-Vusern simulieren. Ein RTE-Vuser würde beispielsweise die folgenden Aktionen ausführen:

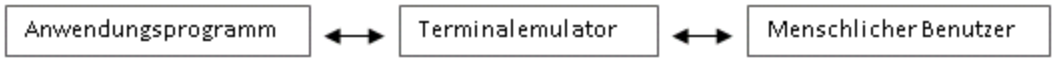
1. Eingeben von **60** an der Befehlszeile, um ein Anwendungsprogramm zu öffnen.
2. Eingeben von **F296**, der Kennung des Außendienstmitarbeiters.
3. Eingeben von **NY270**, der Kundennummer.
4. Abwarten, bis das Wort "Details" auf dem Bildschirm angezeigt wird. "Details" gibt an, dass alle Kundendetails auf dem Bildschirm angezeigt werden.
5. Eingeben der Reparaturdetails, hier z. B. **Austausch von Dichtung P249, Große Inspektion wurde ausgeführt**.
6. Eingeben von **Q**, um das Anwendungsprogramm zu schließen.

Sie können VuGen zum Entwickeln von RTE-Vuser-Skripts verwenden. Der Skriptgenerator zeichnet die Aktionen auf, die ein menschlicher Benutzer im Terminalemulator ausführt. Er zeichnet die Tastatureingabe im Terminalfenster auf, generiert entsprechende Anweisungen und fügt diese in das Vuser-Skript ein. Während der Aufzeichnung fügt der Skriptgenerator automatisch Synchronisierungsfunktionen in das Skript ein. Weitere Informationen finden Sie unter "Übersicht über die RTE-Synchronisierung" auf Seite 840.

Die Funktionen, die zum Emulieren der Kommunikation zwischen einem Terminal und einem Server entwickelt wurden, werden TE-Vuser-Funktionen genannt. Alle TE-Vuser-Funktionen beginnen mit dem Präfix **TE**. VuGen zeichnet die meisten in diesem Abschnitt aufgeführten TE-Funktionen im Rahmen einer RTE-Aufzeichnungssitzung automatisch auf. Sie können alle Funktionen in Ihrem Skript allerdings auch manuell programmieren**.**

Weitere Informationen zur Syntax und Beispiele von TE-Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Ein RTE-Vuser emuliert die Aktionen eines echten Benutzers. Menschliche Benutzer nutzen Terminals bzw. Terminalemulatoren zu Bedienung von Anwendungsprogrammen.



In der RTE-Vuser-Umgebung wird der menschliche Benutzer durch einen Vuser ersetzt. Der Vuser bedient PowerTerm, einen Terminalemulator.

PowerTerm arbeitet wie ein standardmäßiger Terminalemulator und unterstützt gängige Protokolle, wie z. B. IBM 3270 und 5250, VT100, VT220 und VT420-7.

Arbeiten mit der Ericom-Terminalemulation

VuGen unterstützt die Aufzeichnung und die Wiedergabe mit Ericom-Terminalemulatoren.

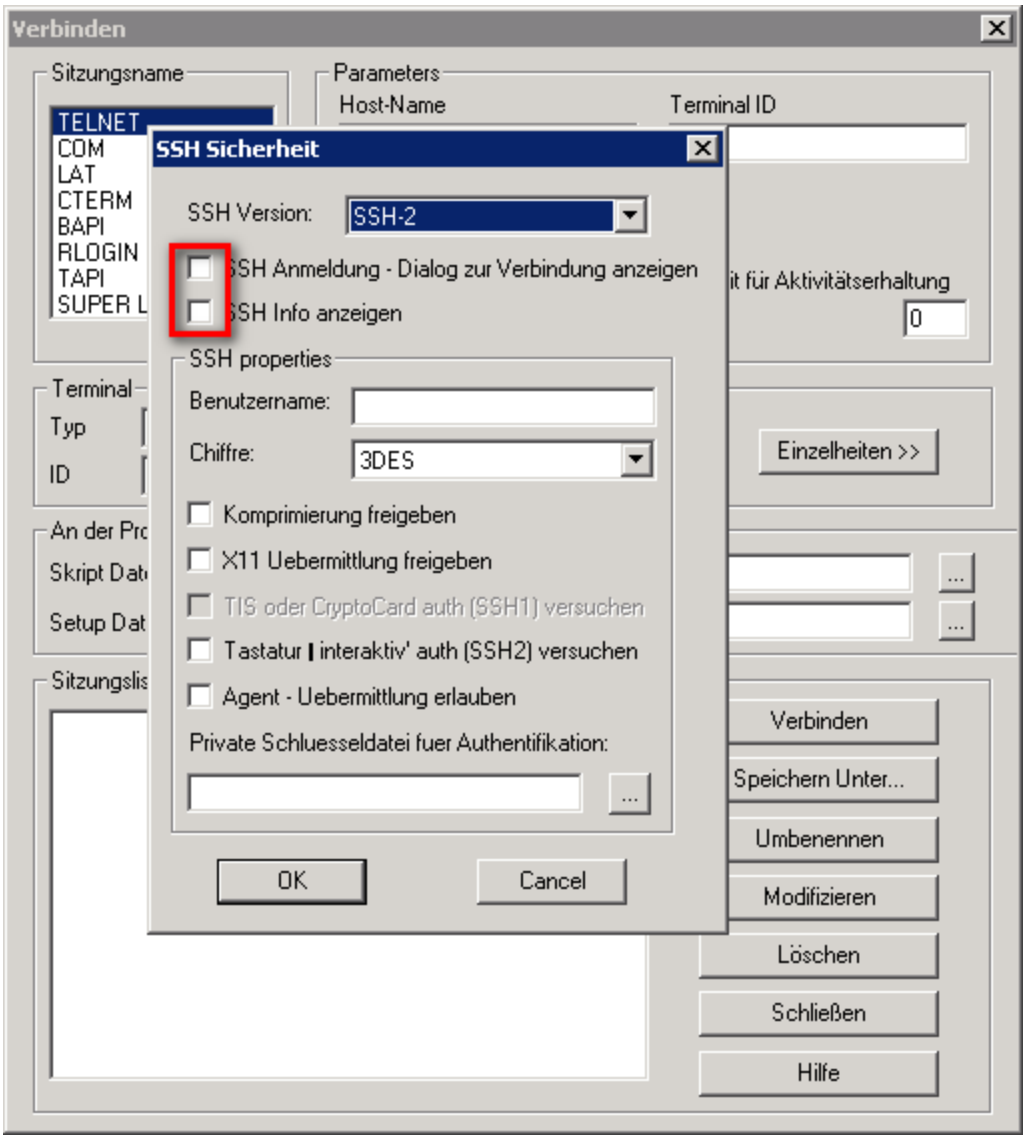
Durch die Unterstützung von Ericom können Escapesequenzen beim Aufzeichnen und Wiedergeben verarbeitet werden. Mithilfe von Ericom PowerTerm können Sie der PC-Tastatur benutzerdefinierte Escapesequenzen zuordnen. Informationen zur Tastaturzuordnung finden Sie in der PowerTerm-Hilfe.

Sobald ein Benutzer beim Aufzeichnen einer Ericom VT-Sitzung eine zugeordnete Taste drückt, generiert VuGen **TE_send_text**- anstatt der standardmäßigen **TE_type**-Funktionen. Hierdurch kann das Skript benutzerdefinierte Escapesequenzen in einem einzelnen Schritt verarbeiten. Weitere Informationen zur **TE_send_text**-Funktion finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

SSL- und SSH-Unterstützung für Ericom

VuGen unterstützt ebenfalls Aufzeichnung und Wiedergabe von SSL/SSH für die RTE-Ericom-Bibliothek. Um mit SSL oder SSH zu arbeiten, wählen Sie im Dialogfeld **Connect** im Abschnitt **Security** den entsprechenden Typ aus.

Wenn Sie mit SSH Security arbeiten, wird standardmäßig ein Popupfenster geöffnet, das Sie zur Eingabe weiterer Informationen auffordert. Es wird empfohlen, die Optionen unter **Anzeigen** zu deaktivieren, um die Anzeige dieser Popupfenster zu verhindern. Wenn diese Popups aktiviert sind, könnte dies die Wiedergabe beeinträchtigen. Sie können über die Schaltfläche **Details** auf die erweiterten Sicherheitsoptionen zugreifen.



Zeicheneingabe in einem Terminalemulator

Zwei TE-Vuser-Funktionen ermöglichen es Vusern, Zeichen in den PowerTerm-Terminalemulator "einzugeben":

- **TE_type** sendet Zeichen an den Terminalemulator. Bei der Aufzeichnung generiert VuGen automatisch **TE_type**-Funktionen für Tastatureingaben im Terminalfenster. Details finden Sie unten.
- **TE_typing_style** legt die Eingabegeschwindigkeit des Vusers fest. Sie können den Eingabestil durch Einfügen der Funktion **TE_typing_style** in das Vuser-Skript manuell festlegen. Alternativ können Sie den Eingabestil auch in den Laufzeiteinstellungen festlegen. Weitere Informationen finden Sie unter "Laufzeiteinstellungen" auf Seite 330.

Hinweis: Vermeiden Sie es während der Aufzeichnung von RTE-Vuser-Skripts, den Cursor im Terminalemulatorfenster mithilfe der Maus zu positionieren. Diese Cursorbewegungen werden von VuGen nicht aufgezeichnet.

Verwenden der TE_type-Funktion

Beim Aufzeichnen eines Skripts zeichnet VuGen alle Tastatureingaben auf und generiert die entsprechenden **TE_type**-Funktionen. Während der Ausführung senden die **TE_type**-Funktionen formatierte Zeichenfolgen an den Terminalemulator.

Tastatureingaben werden als herkömmliche Textzeichenfolgen (einschließlich Leerzeichen) definiert. Beispiel:

```
TE_type ("hello, world");
```

Tastennamen, die mehr als ein Zeichen lang sind, werden durch Voranstellung des Buchstabens k gekennzeichnet und in eckige Klammern (< >) gefasst.

Die Funktion unten stellt beispielsweise die Tastenreihenfolge **Eingabetaste**, **Strg-Taste**, **y-Taste** dar.

```
TE_type("<kReturn><kControl-y>");
```

Weitere Beispiele sind: <kF1>, <kUp>, <kF10>, <kHelp>, <kTab>.

Um die Bezeichnung einer Taste zu ermitteln, brauchen Sie den entsprechenden Tastaturanschlag nur aufzuzeichnen und die Bezeichnung in der generierten Anweisung zu überprüfen.

Hinweis: Wenn Sie eine **TE_type**-Anweisung programmieren (anstatt sie aufzuzeichnen), verwenden Sie die Tastendefinitionen in der Function Reference (**Hilfe > Funktionsreferenz**).

Festlegen des Zeitüberschreitungswerts für "TE_type"

Wenn ein Vuser versucht, eine **TE_type**-Anweisung zu übermitteln, während sich das System im X SYSTEM-Modus (d. h. im Eingabesperrmodus) befindet, wird die Eingabe bis zur Beendigung des Modus verzögert. Wenn der X SYSTEM-Modus länger als der in TE_XSYSTEM_TIMEOUT angegebene Wert in Millisekunden andauert, gibt die Funktion **TE_type** einen TE_TIMEOUT-Fehler zurück.

Sie können den Wert für TE_XSYSTEM_TIMEOUT mithilfe von **TE_setvar** festlegen. Der Standardwert für TE_XSYSTEM_TIMEOUT beträgt 30 Sekunden.

Zulassen von Vorauseingaben

Unter bestimmten Umständen kann es wünschenswert sein, dass Vuser Tastatureingaben übermitteln können, obwohl das System sich im X SYSTEM-Modus (d. h. im Eingabesperrmodus) befindet. So könnte es beispielsweise von Vorteil sein, dass der Vuser die `Untbr`-Taste betätigen kann. Über die Variable TE_ALLOW_TYPEAHEAD können Sie festlegen, dass der Vuser Tastenanschläge übermitteln kann, obwohl sich das System im X SYSTEM-Modus befindet.

Wenn Sie TE_ALLOW_TYPEAHEAD auf Null setzen, werden Vorauseingaben verhindert. Bei allen anderen Werten sind Vorauseingaben möglich. Der Wert von TE_ALLOW_TYPEAHEAD wird mithilfe der Funktion **TE_setvar** festgelegt. In der Standardeinstellung (Null) wird verhindert, dass Tastaturanschläge im X SYSTEM-Modus übermittelt werden.

Weitere Informationen zur Funktion **TE_type** und den zugehörigen Konventionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Festlegen des Eingabestils

Es können zwei unterschiedliche Eingabestile für RTE-Vuser festgelegt werden: FAST und HUMAN. Bei FAST führt der Vuser die Tastatureingabe im Terminalemulator so schnell wie möglich aus. Der Stil HUMAN sieht eine kurze Pause nach jedem Tastaturanschlag vor. Auf diese Weise kann der Vuser die Tastatureingabe durch einen menschlichen Benutzer genauer simulieren.

Sie können den Eingabestil mithilfe der Funktion **TE_typing_style** festlegen. Die Syntax von **TE_typing_style** sieht folgendermaßen aus:

int TE_typing_style (char *style);

wobei style entweder FAST oder HUMAN sein kann. Der Standardeingabestil ist HUMAN. Das Format für den Eingabestil HUMAN sieht folgendermaßen aus:

HUMAN, delay [,first_delay]

Hierbei gibt delay den Intervall in Millisekunden zwischen zwei Tastaturanschlägen an. Der optionale Parameter first_delay gibt die Verzögerungszeit in Millisekunden bis zur Eingabe des ersten Zeichens in der Zeichenfolge an. Beispiel:

```
TE_typing_style ("HUMAN, 100, 500");
TE_type ("ABC");
```

Dies bedeutet, dass der Vuser 0,5 Sekunden wartet, bis er den Buchstaben A eingibt und dann vor der Eingabe der Buchstaben B und C jeweils 0,1 Sekunden wartet.

Weitere Informationen zur Funktion **TE_typing_style** und den zugehörigen Konventionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Alternativ zur Funktion **TE_typing_style** können Sie den Eingabestil auch in den Laufzeiteinstellungen festlegen. Weitere Informationen finden Sie unter "Laufzeiteinstellungen" auf Seite 330.

Generieren von eindeutigen Gerätenamen

Bei einigen Protokollen, z. B. bei APPC, ist für jedes angemeldete Terminal ein eindeutiger Gerätename erforderlich. In den Laufzeiteinstellungen können Sie festlegen, dass die Funktion TE_connect für jeden Vuser einen eindeutigen Gerätenamen aus 8 Zeichen generiert, der zum Herstellen der Verbindung dient. Bei einigen Systemen muss zusäzlich eine weitere Anforderung erfüllt werden: Der Gerätename muss ein bestimmtes Format aufweisen. Weitere Informationen finden Sie unter "Laufzeiteinstellungen" auf Seite 330.

Erweitern Sie das Vuser-Skript durch die Funktion **RteGenerateDeviceName**, um das Format des Gerätenamens festzulegen, den die Funktion TE_connect zum Anmelden von Vusern im System

verwendet. Die Funktion hat den folgenden Prototyp:

```
void RteGenerateDeviceName(char buf[32])
```

Der Gerätename sollte in **buf** eingegeben werden.

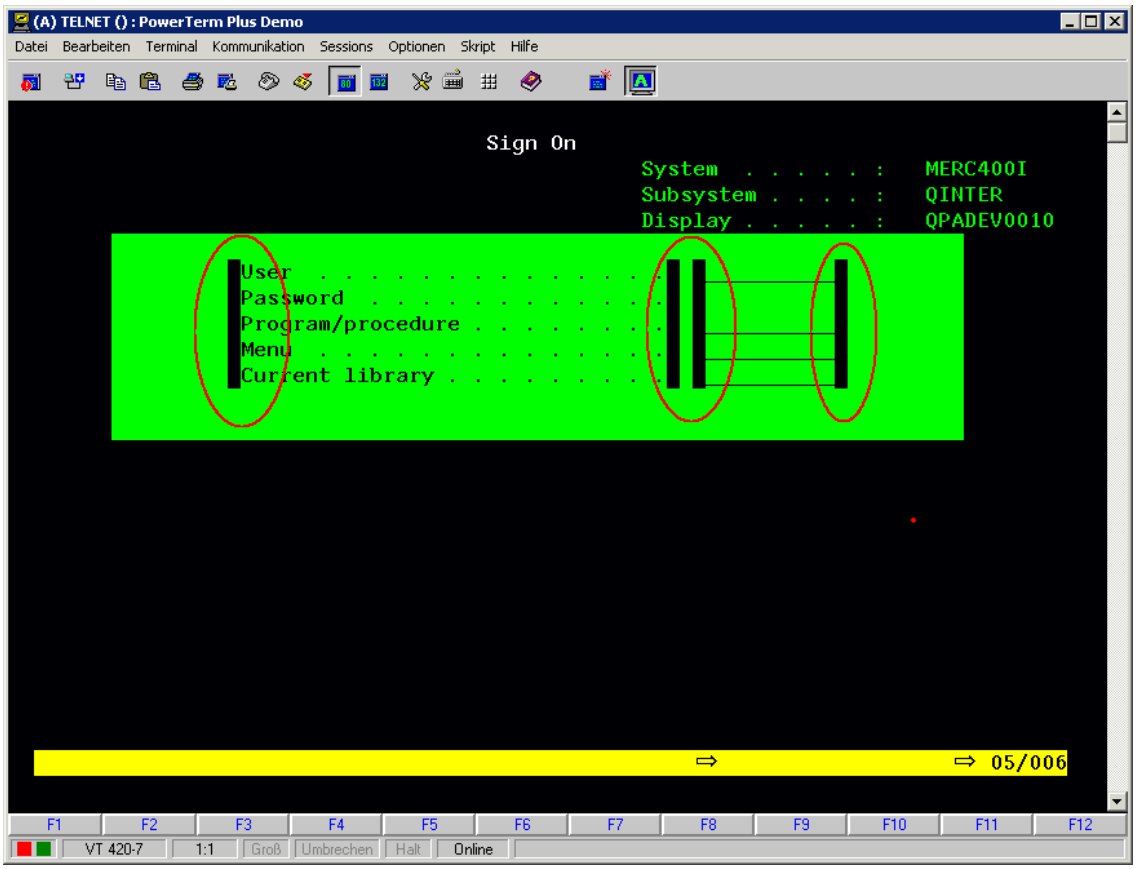
Wenn die Funktion **RteGenerateDeviceName** in einem Vuser-Skript vorhanden ist, wird sie jedes Mal vom Vuser aufgerufen, wenn ein neuer Gerätename benötigt wird. Ist die Funktion **RteGenerateDeviceName** nicht im Skript definiert, und es sind eindeutige Gerätenamen erforderlich, übernimmt die Funktion **TE_connect** die Generierung der Namen.

Im folgenden Beispiel generiert die Funktion **RteGenerateDeviceName** eindeutige Gerätenamen im Format **TERMx**. Der erste Name lautet **TERM0** gefolgt von **TERM1**, **TERM2** usw.

```
RteGenerateDeviceName(char buf[32])
{
    static int n=0;
    sprintf(buf, "TERM%d", n);
    n=n+1;
}
```

Festlegen der Feldbegrenzungszeichen

Bei einigen Terminalemulatoren werden Begrenzungszeichen verwendet, die Anfang und Ende eines Feldes markieren. Diese Begrenzungszeichen sind eigentlich nicht sichtbar – sie werden am Bildschirm als Leerzeichen dargestellt. In dem unten gezeigten Terminalemulator wurden die Farben im Abschnitt in der Mitte invertiert, um die Feldbegrenzungszeichen hervorzuheben. Die Stellen sind rot eingekreist.



Die Funktionen **TE_wait_text**, **TE_get_text** und **TE_find_text** identifizieren die Zeichen in einem bestimmten Bereich des Bildschirms. Ein Feldbegrenzungszeichen, das sich in dem angegebenen Bereich befindet, kann entweder als Leerzeichen oder als ASCII-Zeichen identifiziert werden. Mithilfe der Systemvariable TE_FIELD_CHARS können Sie die Identifizierungsmethode angeben. Mögliche Werte für TE_FIELD_CHARS sind 0 oder1:

- 0 bewirkt, dass das Zeichen an der Position des Feldbegrenzungszeichens als Leerzeichen zurückgegeben wird.
- 1 bewirkt, dass das Zeichen an der Position des Feldbegrenzungszeichens als ASCII-Zeichen (ASCII-0 oder ASCII-1) zurückgegeben wird.

Die Standardeinstellung von TE_FIELD_CHARS ist 0.

Sie können den Wert von TE_FIELD_CHARS mithilfe der Funktionen **TE_getvar** und **TE_setvar** abrufen bzw. festlegen.

Abrufen von Text vom Terminalbildschirm

Es gibt eine Reihe von Vuser-Funktionen, die von RTE-Vusern für den Abruf von Text vom Terminalbildschirm genutzt werden können. Sie können die Funktionen **TE_find_text** und **TE_get_text_line** verwenden, um zu prüfen, ob der Terminalemulator korrekt reagiert oder um die Logik des Skripts zu erweitern.

Nach der Aufzeichnung können Sie die Anweisungen **TE_find_text** und **TE_get_text_line** direkt in Ihre RTE-Vuser-Skripts einfügen.

Suchen nach Text auf dem Bildschirm

Die Funktion **TE_find_text** sucht nach einer Textzeile auf dem Bildschirm. Die Syntax dieser Funktion lautet:

```
int TE_find_text (char *pattern, int col1, int row1, int col2, int row2,
        int *retcol, int *retrow, char *match);
```

Diese Funktion sucht in dem durch col1, row1, col2, row2 definierten Rechteck nach Text, der mit dem unter pattern angegebenen Muster übereinstimmt. Der mit dem Muster übereinstimmende Text wird an match zurückgegeben, die tatsächliche Zeilen- und Spaltenposition an retcol und retrow. Die Suche beginnt in der oberen linken Ecke des Bildschirms. Wenn mehr als eine Zeichenfolge mit pattern übereinstimmt, wird diejenige zurückgegeben, die der oberen linken Ecke am nächsten ist.

Das **pattern** kann einen regulären Ausdruck enthalten. Weitere Informationen zur Verwendung regulärer Ausdrücke finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

TE_find_text-Anweisungen müssen manuell in das Vuser-Skript eingegeben werden. Details zur Syntax der Funktion **TE_find_text** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Abrufen von Text vom Terminalbildschirm

Die Funktion **TE_get_text_line** ruft eine Textzeile aus einem von Ihnen definierten Bildschirmbereich ab. Die Syntax dieser Funktion lautet:

```
char *TE_get_text_line (int col, int row, int width, char *text);
```

Diese Funktion kopiert eine Textzeile vom Terminalbildschirm in den Puffer text. Das erste Zeichen in der Zeile wird durch col, row definiert. Die Spaltenkoordinate des letzten Zeichens in der Zeile wird durch width vorgegeben. Der Text auf dem Bildschirm wird an den Puffer text übergeben. Wenn die Zeile Tabulator- oder Leerzeichen enthält, wird die entsprechende Anzahl von Leerzeichen zurückgegeben.

Darüber hinaus kann die Funktion **TE_get_cursor_position** auch dazu verwendet werden, die aktuelle Position des Cursors auf dem Terminalbildschirm abzurufen. Die Funktion **TE_get_line_attribute** gibt die Zeichenformatierung (z. B. fett oder unterstrichen) einer Textzeile zurück.

TE_get_text_line-Anweisungen müssen manuell in das Vuser-Skript eingegeben werden. Details zur Syntax der Funktion **TE_get_text_line** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Übersicht über die RTE-Synchronisierung

Je nachdem, welches System Sie testen, kann es erforderlich sein, die an einen Terminalemulator übermittelten Vuser-Eingaben mit den Antworten des Servers zu synchronisieren. Die Synchronisierung der Eingabe bedeutet, den Vuser anzuweisen, die Skriptausführung anzuhalten, bis ein Systemsignal empfangen wird und dann erst mit der nächsten Aktion fortzufahren.

Angenommen, ein menschlicher Benutzer übermittelt die folgende Abfolge von Eingaben an eine Bankanwendung:

1. Eingabe von **1** zur Auswahl des Menüpunkts **Finanzdaten**.
2. Eingabe von **3**, sobald die Meldung **Gewünschte Informationen?** angezeigt wird, um den Menüpunkt **Dow Jones-Index** auszuwählen.
3. Nach vollständiger Ausgabe des Berichts am Bildschirm, Eingabe von **5**, um die Bankanwendung zu beenden.

In diesem Beispiel sind die Eingaben in die Bankanwendung synchronisiert, da der menschliche Benutzer vor jeder neuen Eingabe auf ein visuelles Signal vom System wartet.
Dieses Signal kann entweder eine bestimmte Meldung sein, die auf dem Bildschirm angezeigt wird, oder die Tatsache, dass sich die angeforderten Informationen nicht mehr verändern, d. h. vollständig sind.

Die Eingaben von Vusern lassen sich auf ähnliche Weise mithilfe der TE-Funktionen **TE_wait_sync**, **TE_wait_text**, **TE_wait_silent** und **TE_wait_cursor** synchronisieren. Diese Funktionen emulieren auf effektive Weise einen menschlichen Benutzer, der Eingaben in ein Terminalfenster macht und dann die Serverrückgabe abwartet, bevor er den nächsten Befehl übermittelt.

Die Funktion **TE_wait_sync** dient ausschließlich der Synchronisierung von Block-Mode-Terminals (IBM). Die übrigen TE-Synchronisierungsfunktionen dienen der Synchronisierung von Character-Mode-Terminals (VT).

Wenn Sie ein RTE-Vuser-Skript aufzeichnen, generiert VuGen automatisch **TE_wait_sync**-, **TE_wait_text**- und **TE_wait_cursor** -Anweisungen und fügt diese in das Skript ein. In den Aufzeichnungsoptionen von VuGen können Sie angeben, welche Synchronisierungsfunktionen eingefügt werden sollen.

Hinweis: Sie sollten keine Synchronisierungsanweisungen in den Abschnitt Vuser_end eines Vuser-Skripts einfügen. Da eine Vuser-Sitzung jederzeit abgebrochen werden kann, können Sie nicht voraussagen, wann die Aktionen im Abschnitt Vuser_end ausgeführt werden.

Synchronisieren von Block-Mode-Terminals (IBM)

Die Funktion **TE_wait_sync** dient zur Synchronisierung von RTE-Vusern, die Block-Mode-Terminals (IBM) bedienen. Block-Mode-Terminals zeigen die Meldung **X SYSTEM** an, um zu signalisieren, dass sich das System im Eingabesperrmodus befindet. In diesem Modus werden keine Tastatureingaben entgegengenommen, da der Terminalemulator auf eine Datenübertragung vom Server wartet.

Wenn Sie ein Skript für ein Block-Mode-Terminal aufzeichnen, fügt VuGen automatisch eine **TE_wait_sync** -Funktion in das Skript ein, wenn die Meldung **X SYSTEM** angezeigt wird. In den Aufzeichnungsoptionen von VuGen können Sie festlegen, ob VuGen automatisch **TE_wait_sync**-Funktionen einfügen soll.

Beim Ausführen eines Vuser-Skripts überprüft die Funktion **TE_wait_sync**, ob sich das System im X SYSTEM-Modus befindet. Wenn sich das System im X SYSTEM-Modus befindet, hält die

Funktion **TE_wait_sync** die Skriptausführung an. Sobald die Meldung **X SYSTEM** nicht mehr angezeigt wird, wird die Skriptausführung fortgesetzt.

> **Hinweis:** Sie können die Funktion **TE_wait_sync** nur bei Emulatoren für Block-Mode-Terminals von IBM (5250 und 3270) verwenden.

Im Allgemeinen wird durch die Funktion **TE_wait_sync** eine angemessene Synchronisierung für alle Block-Mode-Terminalemulatoren gewährleistet. Sollte sich die Funktion **TE_wait_sync** unter bestimmten Umständen jedoch als ineffektiv erweisen, können Sie die Synchronisierung durch die Funktion **TE_wait_text** erweitern. Weitere Informationen zur Funktion **TE_wait_text** finden Sie unter "Synchronisieren von Character-Mode-Terminals (VT)" auf Seite 844 und in der Function Reference (**Hilfe > Funktionsreferenz**).

In dem folgenden Skriptausschnitt meldet sich der Vuser mit dem Benutzernamen **QUSER** und dem Kennwort **HPLAB** an. Der Vuser drückt dann die `EINGABETASTE`, um die Anmeldeinformationen an den Server zu übermitteln. Während das System auf die Serverrückmeldung wartet, zeigt der Terminalemulator die Meldung **X SYSTEM** an.

Der Vuser wird durch **TE_wait_sync** angewiesen, zu warten, bis der Server auf die Anmeldeanfrage reagiert hat, d. h. die Meldung **X SYSTEM** vom Bildschirm entfernt wurde, bevor die nächste Skriptzeile ausgeführt wird.

```
TE_type("QUSER");
lr_think_time(2);
TE_type("<kTab>HPLAB");
lr_think_time(3);
TE_type("<kEnter>");
TE_wait_sync();
....
```

Wenn die Funktion **TE_wait_sync** die Skriptausführung anhält, während die Meldung **X SYSTEM** angezeigt wird, überwacht der Vuser weiterhin das System und wartet darauf, dass die Meldung **X SYSTEM** entfernt wird. Wenn die Meldung **X SYSTEM** auch nach Ablauf des Zeitüberschreitungswert für die Synchronisierung noch angezeigt wird, gibt die Funktion **TE_wait_sync** einen Fehlercode zurück. Der Standard-Zeitüberschreitungswert beträgt 60 Sekunden.

Festlegen des Zeitüberschreitungswerts TE_wait_sync

1. Wählen Sie **Vuser** > **Run-Time Settings** aus. Das Dialogfeld **Run-Time Settings** wird geöffnet.

2. Wählen Sie in der Struktur **Run-Time settings** den Knoten **RTE:RTE** aus.

3. Geben Sie unter **X SYSTEM Synchronization** im Feld **Timeout** einen Wert (in Sekunden) ein.

4. Klicken Sie auf **OK**, um das Dialogfeld **Laufzeiteinstellungen** zu schließen.

   Nach der Ausführung der Funktion **TE_wait_sync** wartet der Vuser, bis sich das Terminal nicht mehr im X SYSTEM-Modus befindet. Wenn das Terminal den X SYSTEM-Modus verlässt, überwacht der Vuser das System noch für eine kurze Zeit, um sicherzustellen, dass das Terminal vollständig stabil ist, d. h. nicht wieder zum X SYSTEM-Modus zurückkehrt. Erst dann

wird die Funktion **TE_wait_sync** beendet und der Vuser kann mit der Skriptausführung fortfahren. Der Zeitraum, während dessen der Vuser das System nach Verlassen des X SYSTEM-Modus überwacht, wird als Stabilisierungszeit bezeichnet. Die Standardeinstellung für **Stable time** lautet 1000 Millisekunden.

Wenn das System folgendes Verhalten zeigt, müssen Sie diesen Wert möglicherweise erhöhen:

Bei der Rückkehr des Systems aus dem X SYSTEM-Modus "flackert" das System und wechselt kurzfristig aus dem X SYSTEM-Modus hin- und zurück, bevor es sich endgültig stabilisiert. Wenn das System den X SYSTEM-Modus für mehr als eine Sekunde verlässt und dann wieder in den X SYSTEM-Modus zurückkehrt, nimmt die Funktion **TE_wait_sync** an, dass das System stabil ist. Wenn ein Vuser dann eine Eingabe tätigt, wechselt das System in den Tastatursperrmodus.

Wenn das System andererseits beim Verlassen des X SYSTEM-Modus kein Flackern zeigt, können Sie die Stabilisierungszeit auf weniger als den Standardwert von einer Sekunde verringern.

Ändern der Stabilisierungszeit für die TE_wait_sync-Funktionen

1. Wählen Sie **Vuser** > **Run-Time Settings** aus. Das Dialogfeld **Run-Time Settings** wird geöffnet.
2. Wählen Sie den Knoten **RTE:RTE** aus.
3. Geben Sie unter **X SYSTEM Synchronization** im Feld **Stable time** einen Wert (in Millisekunden) ein.
4. Klicken Sie auf **OK**, um das Dialogfeld **Laufzeiteinstellungen** zu schließen.

Weitere Informationen zur Funktion **TE_wait_sync** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

VuGen kann bei jedem Wechsel in den X SYSTEM-Modus die Zeit messen, die sich das System in diesem Modus befindet. Hierzu fügt VuGen wie im Skriptsegment unten dargestellt die Funktion **TE_wait_sync_transaction** nach jeder **TE_wait_sync**-Funktion ein:

```
TE_wait_sync();
TE_wait_sync_transaction("syncTrans1");
```

Jede **TE_wait_sync_transaction**-Funktion erstellt eine Transaktion mit dem Namen **default**. Sie ermöglicht Ihnen zu analysieren, wie lange der Terminalemulator während einer Szenarioausführung auf Rückmeldungen vom Server wartet. In den Aufzeichnungsoptionen können Sie festlegen, ob VuGen **TE_wait_sync_transaction**-Anweisungen generieren und einfügen soll.

VuGen anweisen, **TE_wait_sync_transaction**-Anweisungen einzufügen

1. Wählen Sie **Vuser** > **Recording Options** aus. Das Dialogfeld **Recording Settings** wird geöffnet.

2. Aktivieren Sie die Option **Generate Automatic X SYSTEM transactions** und klicken Sie dann auf **OK**.

Synchronisieren von Character-Mode-Terminals (VT)

Es gibt drei Synchronisierungstypen für Character-Mode-Terminals (VT). Der zu wählende Synchronisierungstyp hängt von den folgenden Faktoren ab:

- Der Aufbau der Anwendung, auf die über den Terminalemulator zugegriffen wird.
- Die spezifische Aktion, die synchronisiert werden soll.

Warten auf die Anzeige des Cursors an einer angegebenen Position

Die empfohlene Synchronisierungsmethode für VT-Terminals ist die Cursorsynchronisierung. Im Gegensatz zu Anwendungen mit Bildlaufleistenfunktion oder TTY-Anwendungen empfiehlt sich die Cursorsynchronisierung besonders bei Vollbild- oder Formularanwendungen.

Für die Cursorsynchronisierung wird die Funktion **TE_wait_cursor** verwendet. Beim Ausführen eines RTE-Vuser-Skripts weist die Funktion **TE_wait_cursor** einen Vuser an, die Skriptausführung so lange anzuhalten, bis der Cursor an einer bestimmten Position auf dem Bildschirm eingeblendet wird. Das Anzeigen des Cursors an der angegebenen Position bedeutet, dass die Anwendung für die nächste Eingabe durch den Terminalemulator bereit ist.

Die Funktion **TE_wait_cursor** hat folgende Syntax:

int TE_wait_cursor (int col, int row, int stable, int timeout);

Während der Skriptausführung wartet die Funktion **TE_wait_cursor** ab, bis der Cursor die durch col, row spezifizierte Position erreicht hat.

Der Parameter **stable** gibt die Zeit (in Millisekunden) an, die der Cursor an der angegebenen Position verbleiben muss. Standardmäßig ist **stable** auf 100 Millisekunden eingestellt. Wenn sich die Clientanwendung innerhalb des durch den Parameter **timeout** vorgegebenen Zeitraums nicht stabilisiert, gibt die Funktion den Fehler TIMEOUT zurück. Wenn Sie ein Skript mithilfe von VuGen aufzeichnen, ist **timeout** standardmäßig auf den Wert von TIMEOUT gesetzt, also 90 Sekunden. Sie können den Wert der Parameter **stable** und **timeout** direkt in dem aufgezeichneten Skript ändern.

In der folgenden Anweisung wird 3 Sekunden auf die Stabilisierung des Cursors gewartet. Wenn der Cursor sich nicht innerhalb von 10 Sekunden stabilisiert, gibt die Funktion eine Zeitüberschreitung zurück.

```
TE_wait_cursor (10, 24, 3000, 10);
```

Weitere Informationen zur Funktion **TE_wait_cursor** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Sie können VuGen auch anweisen, **TE_wait_cursor**-Anweisungen während der Aufzeichnung automatisch zu generieren und in das Skript einzufügen. Das Beispiel unten zeigt eine automatisch generierte **TE_wait_cursor**-Anweisung:

```
TE_wait_cursor (7, 20, 100, 90);
```

VuGen anweisen, TE_wait_cursor-Anweisungen während der Aufzeichnung automatisch zu generieren und in das Skript einzufügen

1. Wählen Sie **Vuser** > **Recording Options** aus. Das Dialogfeld **Recording Settings** wird geöffnet.
2. Aktivieren Sie unter **Generate Automatic Synchronization Commands** das Kontrollkästchen **Cursor** und klicken Sie dann auf **OK**.

Warten auf die Anzeige von Text

Sie können die Textsynchronisierung verwenden, um einen RTE-Vuser zu simulieren, der auf einem VT-Terminalemulator ausgeführt wird. Für die Textsynchronisierung wird die Funktion **TE_wait_text** verwendet. Die Funktion **TE_wait_text** hält das Skript an, bis eine bestimmte Zeichenfolge im Terminalfenster angezeigt wird, und fährt dann mit der Skriptausführung fort. Die Textsynchronisierungen empfiehlt sich bei Anwendungen, bei denen der Cursor sich nicht dauerhaft in einem vordefinierten Bildschirmbereich befindet.

Hinweis: Obwohl die Textsynchronisierung eigentlich für Character-Mode-Terminals (VT) vorgesehen ist, kann sie auch bei IBM-Block-Mode-Terminals eingesetzt werden. Die automatische Textsynchronisierung sollte nicht bei Block-Mode-Terminals verwendet werden.

Die Funktion **TE_wait_text** hat folgende Syntax:

```
int TE_wait_text (char *pattern, int timeout, int col1, int row1, int col2, int row2,
        int *retcol, int *retrow, char *match);
```

Diese Funktion wartet darauf, dass Text, der mit dem unter pattern angegebenen Muster übereinstimmt, in dem durch col1, row1, col2, row2 definierten Rechteck angezeigt wird. Der mit dem Muster übereinstimmende Text wird an **match** zurückgegeben, die tatsächliche Zeilen- und Spaltenposition an **retcol** und **retrow**. Wenn vor Ablauf des unter **timeout** angegebenen Zeitraums keine Übereinstimmung mit **pattern** festgestellt wird, gibt die Funktion einen Fehlercode zurück. Das **pattern** kann einen regulären Ausdruck enthalten. Informationen zur Verwendung von regulären Ausdrücken finden Sie in der Function Reference. Alle anderen Parameter außer **pattern** und **timeout** sind optional.

Wenn **pattern** als leere Zeichenfolge übergeben wird, wartet die Funktion den Zeitüberschreitungswert ab, falls innerhalb des Rechtecks überhaupt Text gefunden wird. Wenn kein Text gefunden wird, erfolgt umgehend eine Rückgabe.

Wenn das unter pattern spezifizierte Muster gefunden wird, wartet die Funktion innerhalb des durch die Systemvariablen **TE_SILENT_SEC** und **TE_SILENT_MILLI** vorgegebenen Intervalls ab, bis sich der Emulator stabilisiert hat (Bildschirmaktualisierung beendet, keine Anzeige weiterer Zeichen). Dies ermöglicht es dem Terminal, sich zu stabilisieren, und emuliert einen menschlichen Benutzer.

Wenn sich das Terminal nicht innerhalb des durch **TE_SILENT_TIMEOUT** vorgegebenen Intervalls stabilisiert, wird die Skriptausführung fortgesetzt. Die Funktion gibt bei Erfolg 0 zurück, setzt

jedoch die Variable **TE_errno**, um zu signalisieren, dass das Terminal nach der Anzeige des Texts noch aktiv war.

Sie können die Werte der **TE_SILENT**-Systemvariablen mithilfe der Funktionen TE_getvar und TE_setvar abrufen bzw. modifizieren. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Im Beispiel unten gibt der Vuser seinen Namen ein und wartet auf die Rückmeldung durch die Anwendung.

```
/* Declare variables for TE_wait_text */
int ret_row;
int ret_col;
char ret_text [80];
/* Type in user name. */
TE_type ("John");
/* Wait for teller to respond. */
TE_wait_text ("Enter secret code:", 30, 29, 13, 1, 13, =;ret_col, =;ret_row,
            ret_text);
```

Sie können VuGen auch anweisen, **TE_wait_text**-Anweisungen während der Aufzeichnung automatisch zu generieren und in das Skript einzufügen.

VuGen anweisen, TE_wait_text-Anweisungen während der Aufzeichnung automatisch zu generieren und in das Skript einzufügen

1. Wählen Sie **Vuser** > **Recording Options** aus. Das Dialogfeld **Recording Settings** wird geöffnet.

2. Aktivieren Sie unter **Generate Automatic Synchronization Commands** das Kontrollkästchen **Prompt** und klicken Sie dann auf **OK**.

   Das Beispiel unten zeigt eine automatisch generierte **TE_wait_text**-Anweisung: Die Funktion wartet bis zu 20 Sekunden darauf, dass die Zeichenfolge **keys** an einer beliebigen Stelle auf dem Bildschirm angezeigt wird. Beachten Sie, dass VuGen bei der automatischen Generierung einer **TE_wait_text**-Funktion alle optionalen Parameter auslässt.

   ```
   TE_wait_text("keys", 20);
   ```

Warten auf die Inaktivität des Terminals

In Fällen, in denen weder die Cursor- noch die Textsynchronisierung wirksam ist, können Sie zur Synchronisierung des Skripts die "stille Synchronisierung" verwenden. Bei der "stillen Synchronisierung" wartet der Vuser darauf, dass der Terminalemulator eine bestimmte Zeit lang inaktiv ist. Der Emulator gilt als inaktiv, wenn eine bestimmte Zeit lang keine Eingabe vom Server empfangen wird.

Hinweis: Verwenden Sie die stille Synchronisierung, wenn weder die Cursor- noch die Textsynchronisierung wirksam ist.

Mithilfe der Funktion **TE_wait_silent** können Sie das Skript anweisen, auf die Inaktivität des Terminals zu warten. Sie können das Intervall festlegen, während dessen das Terminal inaktiv sein muss. Wenn das Terminal in dem angegebenen Intervall inaktiv ist, nimmt die Funktion **TE_wait_silent** an, dass die Anwendung keinen Text mehr an den Terminalbildschirm ausgibt und sich der Bildschirm stabilisiert hat.

Die Syntax dieser Funktion lautet:

```
int TE_wait_silent (int sec, int milli, int timeout);
```

Die Funktion **TE_wait_silent** wartet in dem durch sec (Sekunden) und milli (Millisekunden) festgelegten Zeitraum ab, ob der Terminalemulator inaktiv ist. Der Emulator gilt als inaktiv, wenn keine Eingaben vom Server empfangen werden. Wenn der Emulator während des in der Variable timeout festgelegten Zeitraums aktiv bleibt (d. h. weiterhin Zeichen empfängt), wird ein Fehler zurückgegeben.

In der folgenden Anweisung wird 3 Sekunden auf die Stabilisierung des Bildschirms gewartet. Wenn der Bildschirm sich nach zehn Sekunden nicht stabilisiert hat, gibt die Funktion einen Fehler zurück.

```
TE_wait_silent (3, 0, 10);
```

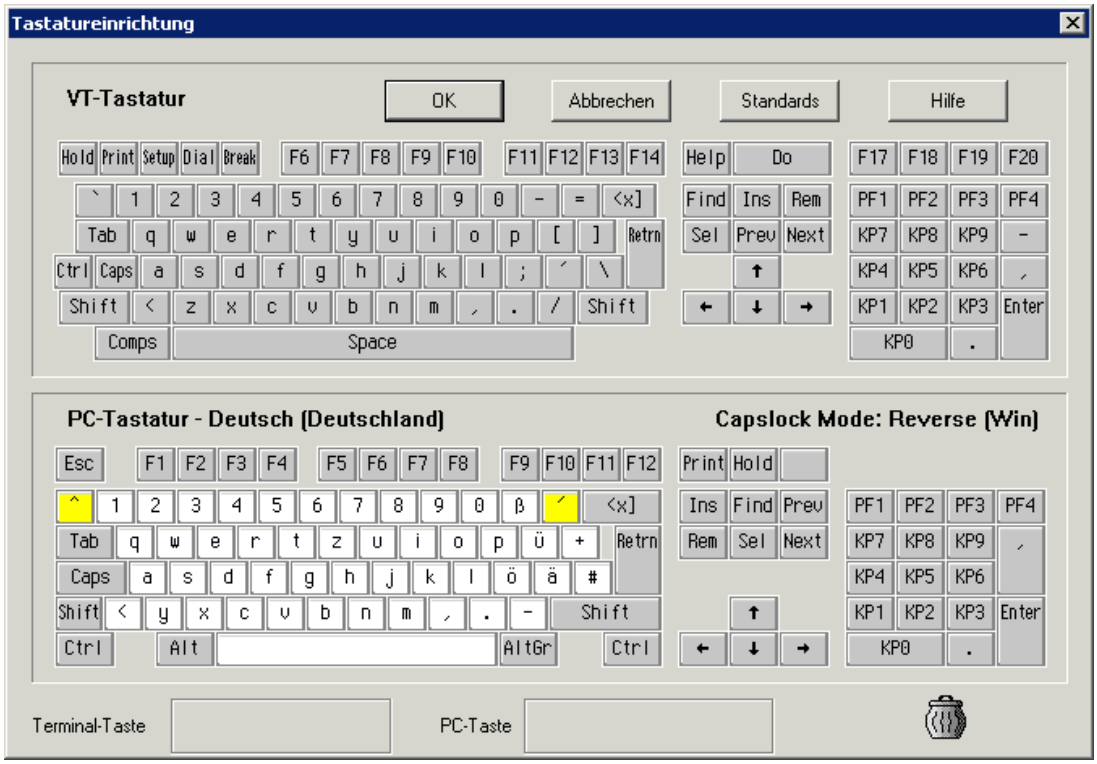
Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Zuordnen von Terminal- und PC-Tastatur

Da Sie mit einem Terminalemulator arbeiten, verwenden Sie anstatt einer Terminal- eine PC-Tastatur. Viele Tasten der Terminaltastatur stehen auf einer PC-Tastatur nicht zur Verfügung. Beispiele für solche Tasten sind HELP, AUTHOR und PUSH, die sich auf der IBM 5250-Tastatur befinden. Um mit dem Terminalemulator und den entsprechenden Anwendungsprogrammen arbeiten zu können, ist es u. U. erforderlich, bestimmte Tasten auf der Terminaltastatur Tasten auf der PC-Tastatur zuzuordnen.

Zuordnen einer Terminaltaste zu einer Taste auf der PC-Tastatur

1. Wählen Sie im Terminalemulator **Optionen** > **Tastaturzuordnung** aus oder klicken Sie auf die Schaltfläche **Tastaturzuordnung** . Das Dialogfeld **Keyboard Mapping** wird geöffnet.



2. Klicken Sie in der Symbolleiste auf die Schaltfläche **Mapping**. Um eine Taste auf der Terminaltastatur einer PC-Taste zuzuordnen, ziehen Sie die gewünschte Taste aus der Terminaltastatur oben und legen Sie sie auf einer Taste auf der PC-Tastatur unten ab.

   Wenn Sie auf der oberen Tastatur auf die Umschalt- und/oder Strg-Taste klicken, werden zusätzliche Tastenfunktionen angezeigt, die vorher nicht verfügbar waren. Sie können dann die gewünschte Taste von der oberen Terminaltastatur auf eine Taste auf der PC-Tastatur unten ziehen.

   Wenn Sie eine Belegung wieder aufheben möchten, ziehen Sie die neu zugeordnete PC-Taste in den Papierkorb. Hierdurch wird die Standardfunktion der PC-Taste wiederhergestellt.

   Klicken Sie zum Wiederherstellen der Standardbelegung auf **Use Defaults**.

Aufzeichnen von RTE-Vuser-Skripts

Sie können VuGen zum Aufzeichnen von RTE-Vuser-Skripts verwenden. VuGen verwendet PowerTerm zur Emulation einer großen Vielzahl unterschiedlicher Terminaltypen.

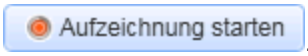
In dieser Aufgabe wird beschrieben, wie RTE-Vuser-Skripts aufgezeichnet werden. Dieses Verfahren unterscheidet sich von dem allgemeinen Aufzeichnungsverfahren in "Aufzeichnen eines Vuser-Skripts" auf Seite 162.

1. **Aufzeichnen der Terminaleinrichtung und -verbindung**

a. Öffnen Sie ein vorhandenes RTE-Vuser-Skript oder erstellen Sie ein neues.

b. Wählen Sie im Feld **Abschnitt** den Abschnitt **vuser_init** aus, der in die aufgezeichneten Anweisungen eingefügt werden soll.

c. Platzieren Sie den Cursor an der Stelle, an der Sie mit der Aufzeichnung beginnen möchten.

d. Klicken Sie auf die Schaltfläche **Aufzeichnen** 

. Das Hauptfenster von PowerTerm wird geöffnet.

e. Wählen Sie in der Menüleiste von PowerTerm **Terminal** > **Setup** aus, um das Dialogfeld **Terminal Setup** zu öffnen.

f. Wählen Sie den Emulierungstyp unter **VT Terminal** oder **IBM Terminal** aus und klicken Sie dann auf **OK**.

> **Hinweis:** Wählen Sie einen IBM-Terminaltyp aus, wenn Sie eine Verbindung zu einem AS/400-System oder einem IBM-Großrechner herstellen möchten. Wählen Sie einen VT-Terminaltyp aus, wenn Sie eine Verbindung zu einer Linux-Workstation herstellen möchten.

g. Wählen Sie **Kommunikation** > **Verbinden** aus, um das Dialogfeld **Verbinden** zu öffnen.

h. Wählen Sie unter **Session Type** den zu verwendenden Verbindungstyp aus.

i. Geben Sie unter **Parameters** die erforderlichen Optionen an. Welche Parameter zur Verfügung stehen, hängt von dem ausgewählten Sitzungstyp ab. Details zu den Parametern erhalten Sie, indem Sie auf **Help** klicken.

> **Tipp:** Klicken Sie auf **Save As**, um die konfigurierten Parametersätze für die Wiederverwendung zu speichern. Gespeicherte Parametersätze werden im Feld **Sessions List** angezeigt.

j. Klicken Sie auf **Verbindung**. PowerTerm stellt eine Verbindung zu dem angegebenen System her. VuGen fügt eine **TE_connect**-Funktion am Einfügepunkt ein. Die Anweisung **TE_connect** ist folgendermaßen aufgebaut:

```
/* *** The terminal type is VT 100. */
TE_connect(
    "comm-type = telnet;"
    "host-name = alfa;"
    "telnet-port = 992;"
    "terminal-id = ;"
    "set-window-size = true;"
```

```
    "security-type = ssl;"
    "ssl-type = tls1;"
    "terminal-type = vt100;"
    "terminal-model = vt100;"
    "login-command-file = ;"
    "terminal-setup-file = ;"
    , 60000);
if (TE_errno != TE_SUCCESS)
    return -1;
```

Auf die **TE_connect**-Anweisung folgt ein **if**-Anweisung, die bei der Wiedergabe überprüft, ob die **TE_connect**-Funktion erfolgreich ausgeführt wurde.

> **Hinweis:** Vermeiden Sie es, mehr als eine Serververbindung **(TE_connect)** in einem Vuser-Skript aufzuzeichnen.

2. ## Aufzeichnen der typischen Benutzeraktionen

Nach der Aufzeichnung des Einrichtungsvorgangs können Sie nun typische Benutzeraktionen oder Geschäftsvorgänge aufzeichnen. Zur Aufzeichnung dieser Art von Vorgängen ist im Vuser-Skript der Abschnitt **Actions** vorgesehen. Nur dieser Abschnitt wird wiederholt, wenn Sie mehrere Iterationen eines des Skripts ausführen.

Beim Aufzeichnen einer Sitzung zeichnet VuGen die Tastaturanschläge und nicht den Text selbst auf. Aus diesem Grund wird nicht empfohlen, Befehle zu kopieren und in das PowerTerm-Fenster einzufügen. Geben Sie die Befehle direkt in das Fenster ein.

a. Wählen Sie den Abschnitt **Actions** im Feld **Abschnitt** aus.

b. Führen Sie nun typische Benutzeraktionen im Terminalemulator aus. VuGen generiert die entsprechenden Anweisungen und fügt diese parallel zu Ihrer Eingabe in das Vuser-Skript ein. Falls erforderlich, können Sie die Anweisungen noch während der Aufzeichnung bearbeiten.

3. ## Aufzeichnen des Abmeldevorgangs

a. Vergewissern Sie sich, dass Sie die typischen Benutzeraktionen wie im vorherigen Abschnitt beschrieben aufgezeichnet haben.

b. Klicken Sie im VuGen-Hauptfenster im Feld **Section** auf **vuser_end**.

c. Führen Sie den Abmeldevorgang aus. VuGen zeichnet die Prozedur im Abschnitt **vuser_ end** auf.

d. Klicken Sie in der Symbolleiste **Aufzeichnen** auf **Aufzeichnung beenden** . Im VuGen-Hauptfenster werden alle aufgezeichneten Anweisungen angezeigt.

e. Klicken Sie auf **Speichern**, um die aufgezeichnete Sitzung zu speichern. Nachdem Sie ein Skript aufgezeichnet haben, können Sie es manuell im Hauptfenster von VuGen

bearbeiten.

Implementieren der Funktion "Bei Fehler fortfahren"

So konfigurieren Sie die Funktion **Bei Fehler fortfahren** in RTE-Skripts:

- Fügen Sie folgende Funktion ein, damit das Skript bei Fehlern weiterhin ausgeführt wird:

TE_setvar(TE_IGNORE_ERRORS, 1)

- Fügen Sie folgende Funktion ein, um das Standardverhalten des Skripts (Fehlschlagen bei Fehler) wiederherzustellen:

TE_setvar(TE_IGNORE_ERRORS, 0)

RTE - Terminalemulation - Fehlerbehebung und Einschränkungen

In diesem Abschnitt werden die Fehlerbehebung und die für RTE-Vuser geltenden Einschränkungen beschrieben.

- IP-Spoofing wird für RTE-Vuser nicht unterstützt.
- Das Ausführen der RTE-Skripts im Controller wird unter Windows 8 oder Server 2012 nicht unterstützt.

SAP-Protokolle

Auswählen eines SAP-Protokolltyps

- Wenn Sie einen SAP GUI-Benutzer testen möchten, der nur auf den Client zugreift, verwenden Sie den Vuser-Typ **SAP GUI**.
- Wenn Sie einen SAP GUI-Benutzer testen möchten, der auch auf einen Webbrowser zugreift, verwenden Sie das SAP (Click & Script)- oder SAP-Web-Protokoll.

Wenn Sie eine SAP GUI-Sitzung aufzeichnen möchten, in der Browsersteuerelemente verwendet werden, erstellen Sie ein Mehrfachprotokoll-Vuser-Skript mit dem SAP GUI- und SAP-Web-Protokoll. So kann VuGen beim Antreffen der Browsersteuerelemente webspezifische Funktionen aufzeichnen. Dies funktioniert nicht, wenn Sie versuchen, das SAP GUI- und das Web-Protokoll zu kombinieren.

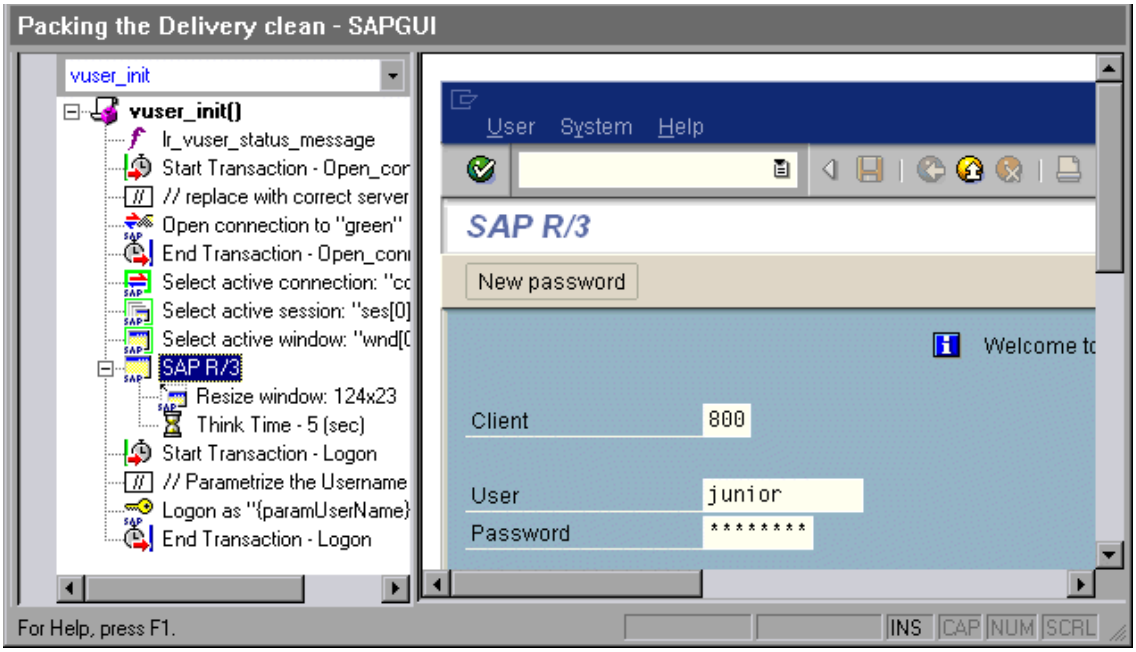
Vergewissern Sie sich vor der Aufzeichnung einer Sitzung, dass VuGen Ihre Module und Clientschnittstellen unterstützt. In der folgenden Tabelle werden die SAP-Clientmodule für SAP-Unternehmensanwendungen sowie die entsprechenden Tools aufgeführt:

SAP-Modul	VuGen-Unterstützung
SAP Web Client oder mySAP.com	Verwenden Sie das SAP-Web-Protokoll.
SAP GUI for Windows	Verwenden Sie das SAP GUI-Protokoll. Er unterstützt auch die Aufzeichnung im Zusammenhang mit dem APO-Modul (erfordert Patchlevel 24 für APO 3.0 für SAP 6.20).
SAP GUI for Windows und Webbrowser	Verwenden Sie das SAP (Click & Script)-Protokoll.
SAP GUI for Java	Dieser Client wird nicht unterstützt.

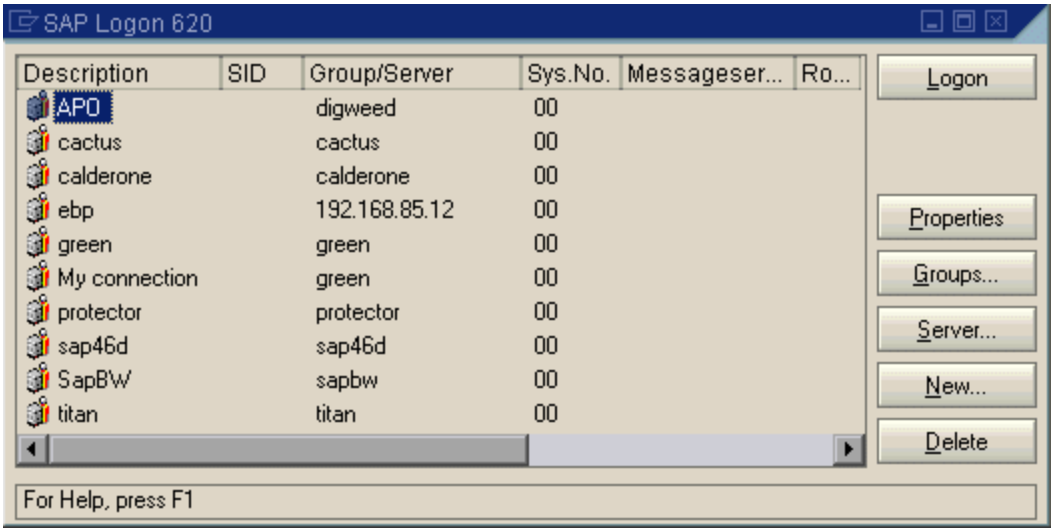
SAP GUI-Protokoll

Ein SAP GUI-Vuser-Skript setzt sich in der Regel aus mehreren SAP-Transaktionen zusammen, die einen Geschäftsprozess ergeben. Ein Geschäftsprozess besteht aus einer Reihe von Funktionen, die Benutzeraktionen emulieren. Öffnen Sie den **Schrittenavigator**, um die einzelnen Benutzeraktionen als Schritt im Vuser-Skript anzuzeigen.

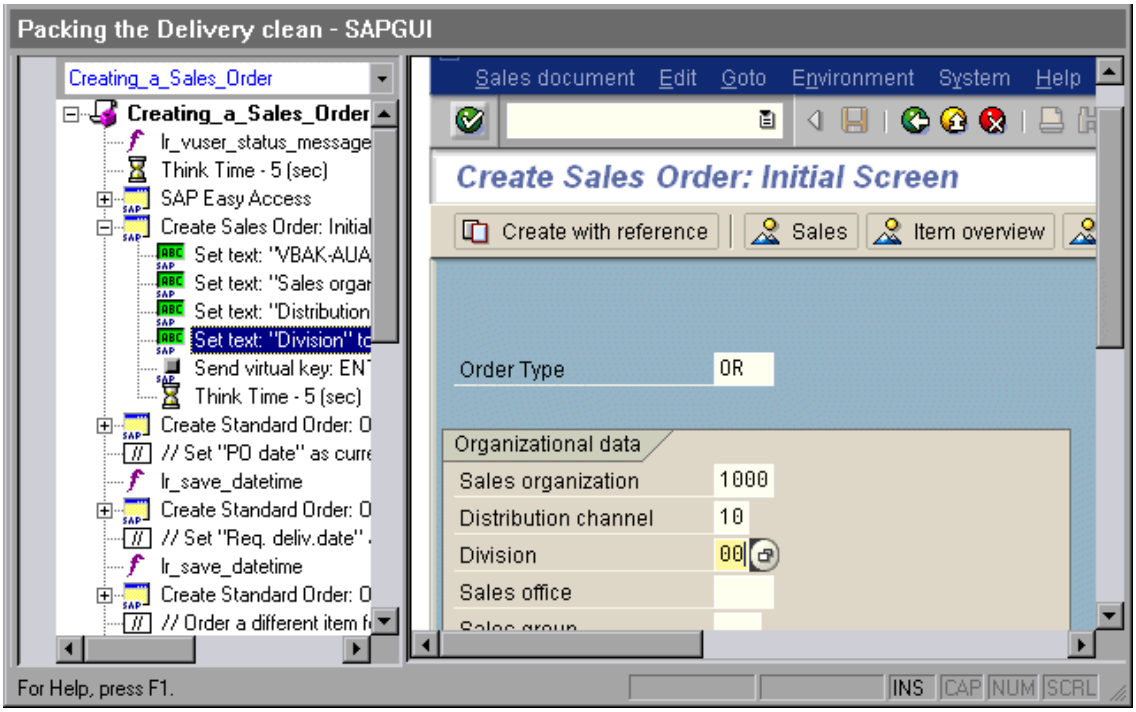
Das folgende Beispiel zeigt die typische Aufzeichnung eines SAP GUI-Clients. Der erste Abschnitt **vuser_init** enthält die Herstellung der Verbindung und die Anmeldung.



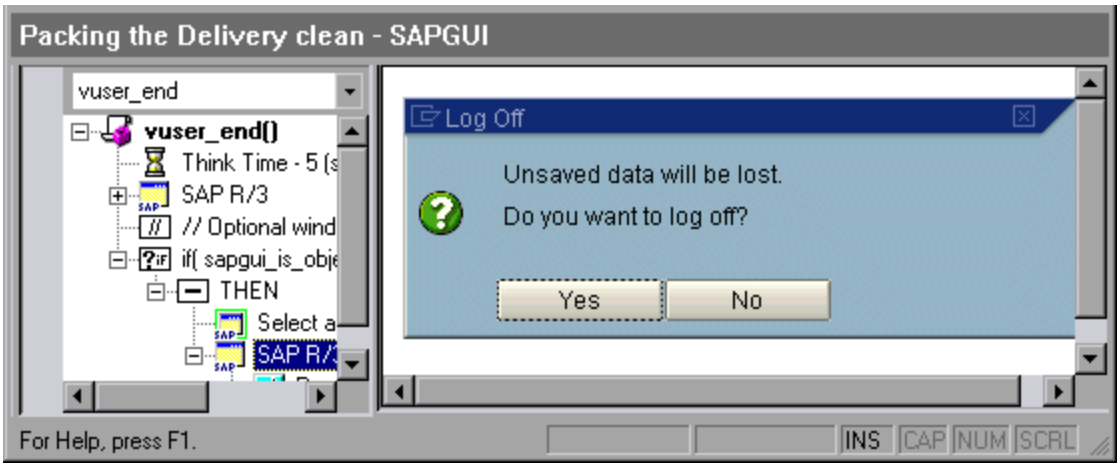
Hinweis: Der **Open Connection**-Schritt verwendet einen der Verbindungsnamen aus der Liste **SAP Logon Descriptions**. Befindet sich der angegebene Verbindungsname nicht in der Liste, sucht der Vuser nach einem Server mit diesem Namen.



Im folgenden Abschnitt werden durch die Funktionen typische Benutzeroperationen emuliert, z. B. die Auswahl von Menüs und die Aktivierung von Kontrollkästchen.



Im letzten Abschnitt **vuser_end** wird das Abmeldeverfahren dargestellt.



Bei der Aufzeichnung eines Mehrfachprotokoll-Skripts für SAP GUI und Web generiert VuGen Schritte für beide Protokolle. In der Skriptansicht sehen Sie sowohl die **sapgui**- als auch die **web**-Funktionen.

Im folgenden Beispiel wird eine Mehrfachprotokoll-Aufzeichnung dargestellt, in der ein SAP GUI-Client ein Websteuerelement öffnet. Beachten Sie den Wechsel von **sapgui**- zu **web**-Funktionen.

```
sapgui_tree_double_click_item("Use as general WWW browser, REPTITLE",
    "shellcont/shell",
    "000732",
    "REPTITLE",
    BEGIN_OPTIONAL,
        "AdditionalInfo=sapgui1020",
    END_OPTIONAL);
...
sapgui_set_text("",
    "http:\\\\yahoo.com",
    "usr/txtEDURL",
    BEGIN_OPTIONAL,
        "AdditionalInfo=sapgui1021",
    END_OPTIONAL);
...
web_add_cookie("B=7pt5cisv1p3m2=;b=2; DOMAIN=www.yahoo.com");
web_url("yahoo.com",
        "URL=http://yahoo.com/",
         "Resource=0",
         "RecContentType=text/html",
         "Referer=",
         "Snapshot=t1.inf",
         "Mode=HTML",
        EXTRARES,
"URL=http://srd.yahoo.com/hpt1/ni=17/ct=lan/sss=1043752588/t1=1043752575385/d1=1
251
          /d2=1312/d3=1642/d4=4757/0.4097009487287739/*1",
          "Referer=http://www.yahoo.com/", ENDITEM,
        LAST);
```

SAP-Web-Protokoll

Ein SAP-Web-Vuser-Skript setzt sich in der Regel aus mehreren SAP-Transaktionen zusammen, die einen Geschäftsprozess ergeben. Ein Geschäftsprozess besteht aus einer Reihe von Funktionen, die Benutzeraktionen emulieren. Ausführliche Informationen zu diesen Funktionen finden Sie unter "Webfunktionen" in der Function Reference (**Hilfe > Funktionsreferenz**).

Beispiel:

Das folgende Beispiel zeigt die typische Aufzeichnung eines SAP Portal-Clients:

```
vuser_init()
{
    web_reg_find("Text=SAP Portals Enterprise Portal 5.0",
        LAST);
    web_set_user("junior{UserNumber}",
        lr_decrypt("3ed4cfe457afe04e"),
        "sonata.hplab.com:80");
    web_url("sapportal",
        "URL=http://sonata.hplab.com/sapportal",
         "Resource=0",
         "RecContentType=text/html",
         "Snapshot=t1.inf",
         "Mode=HTML",
        EXTRARES,
        "Url=/SAPPortal/IE/Media/sap_mango_polarwind/images/header/branding_imag
e.jpg",
           "Referer=http://sonata.hplab.com/hrnp$30001/sonata.hplab.coml:80/Acti
on/26011[header]"
           , ENDITEM,
        "Url=/SAPPortal/IE/Media/sap_mango_polarwind/images/header/logo.gif",
           "Referer=http://sonata.hplab.com/hrnp$30001/sonata.hplab.com:80/Actio
n/26011[header]",
             ENDITEM,
...
        LAST);
```

Im folgenden Abschnitt wird eine Mehrfachprotokoll-Aufzeichnung für SAP-Web und SAP GUI dargestellt, in der der Portal-Client ein SAP-Websteuerelement öffnet. Wie Sie sehen, werden jetzt **sapgui_xxx**-Funktionen anstelle von **web_xxx**-Funktionen verwendet.

```
web_url("dummy",
        "URL=http://sonata.hplab.com:1000/hrnp$30000/sonata.hplab.com:
          1000/Action/dummy?PASS_PARAMS=YES=;dummyComp=dummy=;
Tcode=VA01=;draggable=0=;CompFName=VA01=;Style=sap_mango_polarwind",
         "Resource=0",
         "RecContentType=text/html",
        "Referer=http://sonata.hplab.com/sapportal",
        "Snapshot=t9.inf",
         "Mode=HTML",
```

```
        LAST);
    sapgui_open_connection_ex(" /H/Protector/S/3200 /WP",
        "",
        "con[0]");
    sapgui_select_active_connection("con[0]");
    sapgui_select_active_session("ses[0]");
    /*Before running script, enter password in place of asterisks in logon funct
ion*/
    sapgui_logon("JUNIOR{UserNumber}",
        "ides",
        "800",
        "EN",
        BEGIN_OPTIONAL,
            "AdditionalInfo=sapgui102",
        END_OPTIONAL);
```

SAP (Click & Script)-Protokoll

Mit VuGen lassen sich Vuser-Skripts für SAP Enterprise portal7- und SAP ITS 6.20/6.40-Umgebungen mithilfe von Testobjekten und -methoden erstellen, die speziell für SAP angepasst wurden. Bei diesen Objekte handelt es sich um APIs auf der Grundlage der HP QuickTest oder Unified Functional Testing-Unterstützung für SAP.

Die Aufzeichnung eines Tests oder einer Komponente in Ihrer SAP-Anwendung besteht darin, dass VuGen die von Ihnen ausgeführten Operationen aufzeichnet. Dabei erkennt VuGen spezielle SAP Windows-Objekte, z. B. Frames, Tabellensteuerungen, iViews und Portale.

Einen Überblick über die Click and Script-Protokolle finden Sie unter "Click & Script-Protokolle – Überblick" auf Seite 656.

In VuGen wird die Aufzeichnung folgender SAP-Steuerelemente unterstützt: Schaltflächen, Kontrollkästchen, Dropdown-Menüs, Bearbeitungsfelder, iViews, Listen, Menüs, Navigationsleisten, OK-Codes, Portale, Gruppen mit Optionsschaltflächen, Statusleisten, Registerkarten, Tabellen und Strukturansichten.

VuGen nutzt zur Simulation einer Operation mit einem GUI-Steuerelement die Steuerhandlerschicht. Wenn VuGen während der Aufzeichnung auf eines der unterstützten SAP-Steuerelemente stößt, wird eine Funktion mit einem Präfix im Format **sap_xxx** generiert.

Beispiel:

Im folgenden Beispiel hat ein Benutzer die Registerkarte **User Profile** ausgewählt. VuGen hat eine Funktion **sap_portal** generiert.

```
web_browser("Close_2",
        "Snapshot=t7.inf",
        DESCRIPTION,
        "Ordinal=2",
        ACTION,
        "UserAction=Close",
        LAST);
```

```
lr_think_time(7);
web_text_link("Personalize",
        "Snapshot=t8.inf",
        DESCRIPTION,
        "Text=Personalize",
        ACTION,
        "UserAction=Click",
        LAST);
    lr_think_time(6);
sap_portal("Sap Portal_2",
        "Snapshot=t9.inf",
        DESCRIPTION,
        "BrowserOrdinal=2",
        ACTION,
        "DetailedNavigation=User Profile",
        LAST);
```

Hinweis: Wenn Sie eine SAP (Click & Script)-Sitzung aufzeichnen, generiert VuGen für Objekte, die nicht SAP-spezifisch sind, Web (Click & Script)-Standardfunktionen. In diesem Fall müssen Sie das Web-Protokoll nicht explizit angeben. Im obigen Beispiel hat VuGen beim Klicken auf **Personalize** eine **web_text_link**-Funktion generiert.

Wiedergabe von optionalen Fenstern mit SAP GUI-Skripts

Beim Arbeiten mit SAP GUI-Vuser-Skripts treffen Sie bisweilen im SAP GUI-Client auf optionale Fenster, die während der Aufzeichnung vorhanden waren, nicht aber während der Wiedergabe. Wenn Sie das Skript wie aufgezeichnet wiedergeben, tritt bei dem Versuch, die fehlenden Fenster zu finden, ein Fehler auf.

VuGen ist aber so beschaffen, dass Aktionen im Zusammenhang mit Fenstern erst ausgeführt werden, wenn gesichert ist, dass das Fenster wirklich vorhanden ist. Dementsprechend überprüft der Vuser, ob das im Schritt **Select active window** angegebene Fenster existiert. Wird das Fenster bei der Wiedergabe gefunden, werden die im Skript aufgezeichneten Aktionen durchgeführt. Andernfalls ignoriert der Vuser bis zum nächsten **Select active window**-Schritt alle Aktionen im Zusammenhang mit diesem Fenster. Hinweis: Nur SAP GUI-Schritte (mit dem Präfix **sapgui** versehen) werden ignoriert.

Wählen Sie zur Verwendung dieser Funktion in der Strukturansicht den entsprechenden **Select Active Window**-Schritt aus und klicken Sie anschließend im Kontextmenü auf **Schritte nur für das Fenster ausführen, wenn es vorhanden** .

Wenn Sie diese Funktion deaktivieren möchten, damit die Schritte immer und unabhängig davon ausgeführt werden sollen, ob der Vuser das Fenster findet, klicken Sie im Kontextmenü auf **Immer Schritte für dieses Fenster ausführen**.

Konfigurieren der SAP-Umgebung

In dieser Aufgabe wird beschrieben, wie Sie die SAP-Umgebung für die Verwendung mit VuGen konfigurieren und überprüfen.

Die Unterstützung des SAP GUI for Windows-Clients durch VuGen basiert auf der Scripting-API von SAP. Mithilfe dieser API können Vuser mit dem SAP GUI-Client interagieren, Benachrichtigungen empfangen und Operationen ausführen.

Die Scripting-API ist nur in den aktuellen Versionen des SAP-Kernels verfügbar. In Kernelversionen, die Scripting unterstützen, ist die entsprechende Option standardmäßig deaktiviert. Vergewissern Sie sich vor der Verwendung von VuGen zunächst, dass die SAP-Server die Scripting-API unterstützen, und aktivieren Sie sie auf dem Server und den Clients. Weitere Informationen, unter anderem zum Herunterladen von Patches, finden Sie im SAP Online Services System (OSS) unter dem Hinweis 480149.

Überprüft, ob die SAPGUI Scripting API aktiviert ist.

Führen Sie die Datei **VerifyScripting.exe** aus dem Verzeichnis **Additional Components\SAP_Tools\VerifySAPGUI** aus. Weitere Informationen finden Sie unter "Weitere Komponenten" auf Seite 1108. Weitere Informationen finden Sie in der Datei **VerifyScripting.htm**, die im Lieferumfang des Dienstprogramms enthalten ist.

Überprüfen des Patchlevels des SAP GUI for Windows-Clients

Den jeweiligen Patchlevel Ihres SAP GUI for Windows-Clients können Sie dem Feld **Info** entnehmen. Die Mindestanforderung ist Version 6.20 Patchlevel 32.

Überprüfen des Patchlevels

1. Öffnen Sie das SAP GUI-Anmeldefenster. Klicken Sie in die linke obere Ecke des SAP-Anmeldedialogfelds und wählen Sie aus dem Menü die Option **About SAP Logon** aus.
2. Das Dialogfeld mit Informationen zur SAP-Version wird geöffnet. Vergewissern Sie sich, dass der Patchlevel 32 oder höher ist.

Überprüfen des Kernel-Patchlevels

1. Melden Sie sich am SAP-System an
2. Wählen Sie **System > Status** aus
3. Klicken Sie auf die Schaltfläche **Sonstige Kernelinformationen** .
4. Überprüfen Sie im Bereich **Kernel Information** den Wert **Sup. Pkg. lvl.**

   Je nach verwendeter SAP-Version muss der Level größer sein als der Level, der in der folgenden Tabelle aufgeführt wird.

Softwarekomponente	SAP-Version	Kernel-Patchlevel

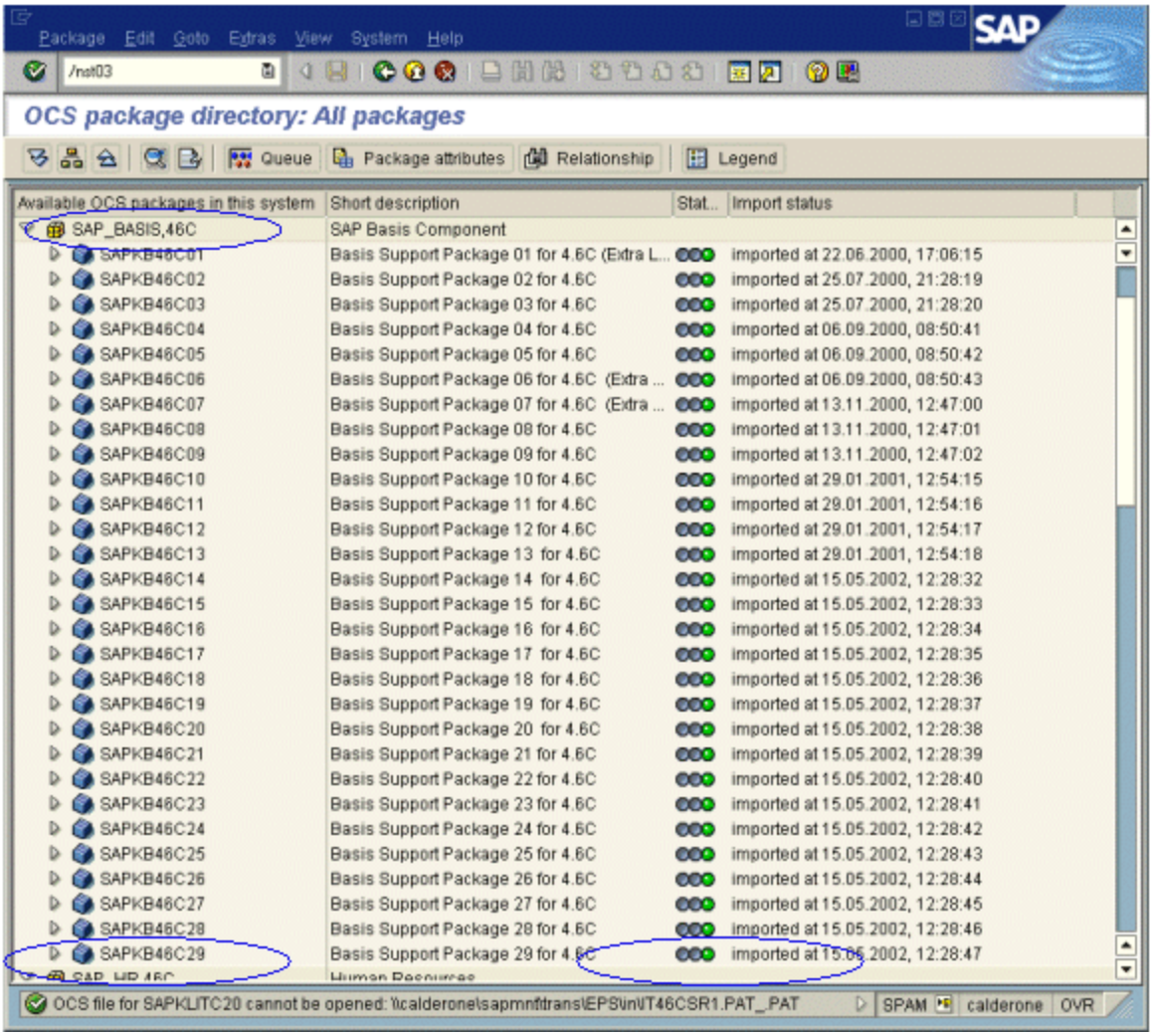
SAP_APPL	31I	Kernel 3.1I Level 650
SAP_APPL	40B	Kernel 4,0B Level 903
SAP_APPL	45B	Kernel 4.5B Level 753
SAP_BASIS	46B	Kernel 4.6D Level 948
SAP_BASIS	46C	Kernel 4.6D Level 948
SAP_BASIS	46D	Kernel 4.6D Level 948
SAP_BASIS	610	Kernel 6.10 Level 360

Überprüfen der Support Packages für R/3

1. Melden Sie sich am SAP-System an und führen Sie die SPAM-Transaktion aus.
2. Wählen Sie im Bereich **Directory** die Option **All Support Packages** aus und klicken sie auf **Display**.
3. Überprüfen Sie anhand der Tabelle unten, ob das geeignete Paket für Ihre SAP-Version installiert ist.

Softwarekomponente	Version	Paketname
SAP_APPL	31I	SAPKH31I96
SAP_APPL	40B	SAPKH40B71
SAP_APPL	45B	SAPKH45B49
SAP_BASIS	46B	SAPKB46B37
SAP_BASIS	46C	SAPKB46C29
SAP_BASIS	46D	SAPKB46D17
SAP_BASIS	610	SAPKB61012

Falls die korrekte Version installiert ist, wird in der Spalte **Status** ein grüner Kreis angezeigt.

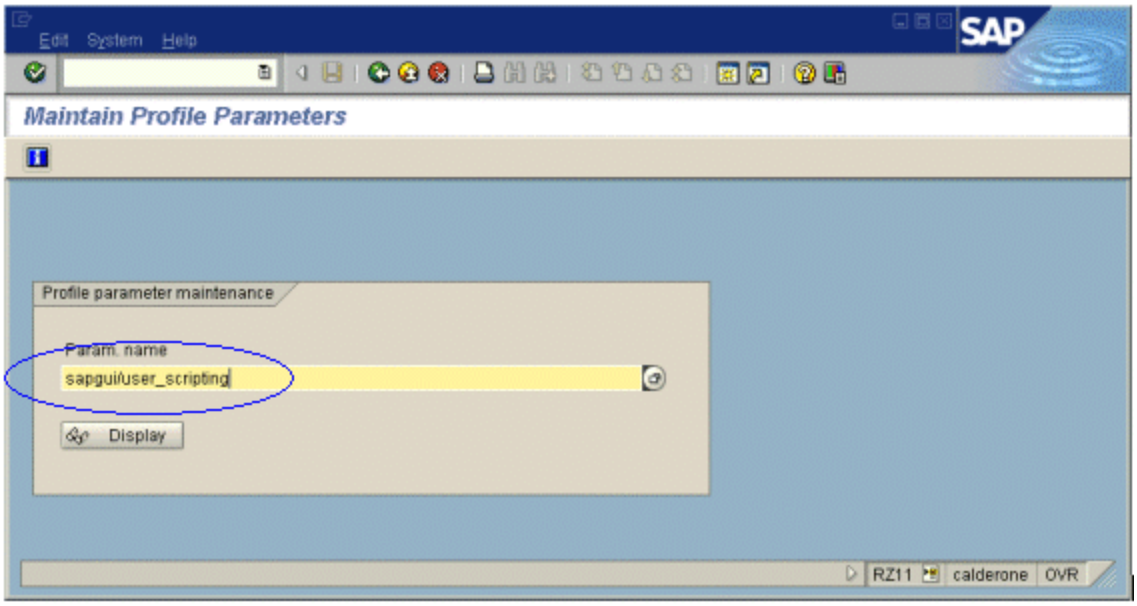


Wenn das OCS-Paket nicht installiert ist, laden Sie es von der SAP-Website (www.sap.com) herunter. Weitere Informationen finden Sie unter dem Hinweis 480149 im SAP OSS.

Aktivieren der Skripterstellung auf dem SAP-Anwendungsserver

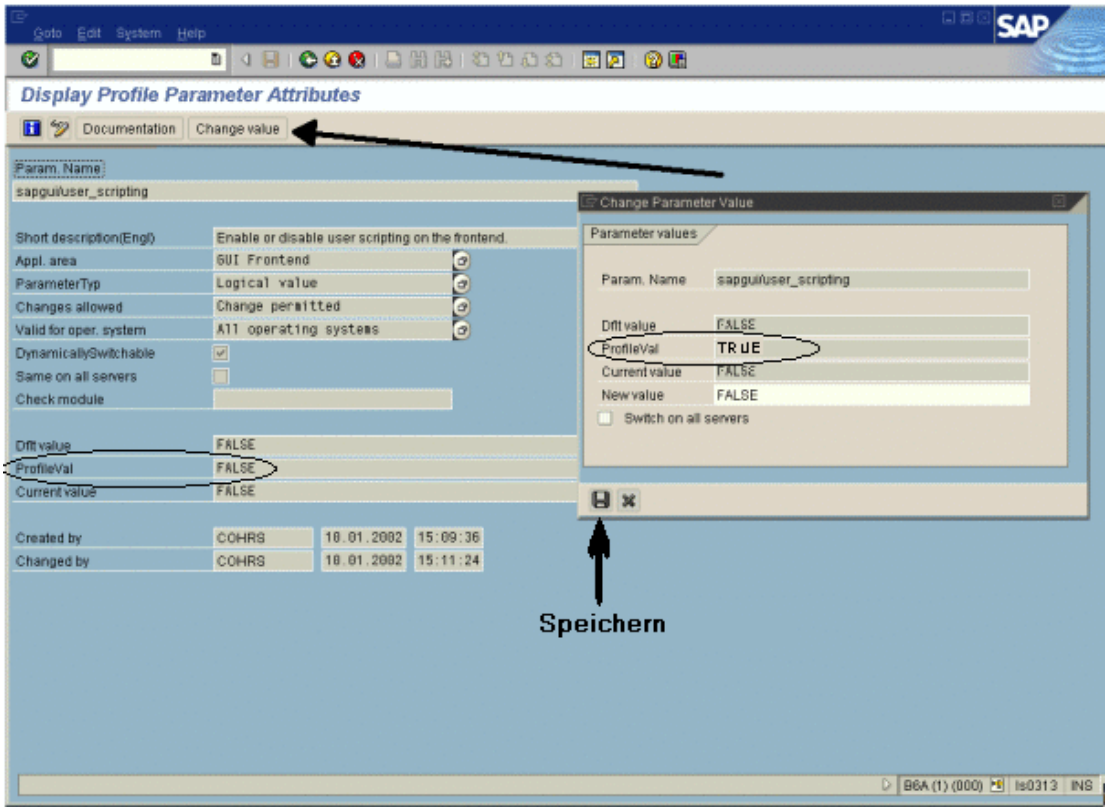
Für Benutzer mit Administratorrechten: Stellen Sie den Profilparameter **sapgui/user_scripting** auf dem Anwendungsserver auf **TRUE** ein. Für alle anderen Benutzer: Stellen Sie diesen Parameter auf allen Anwendungsservern ein. Wenn Sie Scripting für bestimmte Benutzergruppen aktivieren möchten, stellen Sie den entsprechenden Parameter nur auf Anwendungsservern mit den gewünschten Zugriffsbeschränkungen ein. In den folgenden Schritten wird erklärt, wie Sie den Profilparameter ändern.

1. Öffnen Sie die Transaktion **rz11.** Geben Sie den Parameternamen **sapgui/user_scripting** an und klicken Sie auf **Anzeigen**. Das Fenster **Attribute des Profilparameters anzeigen** wird geöffnet.



Wenn in der Statusleiste die Meldung **Parameter name is unknown** angezeigt wird, fehlt das aktuelle Support Package. Importieren Sie das Support Package für die entsprechende SAP BASIS- und Kernelversion des Anwendungsservers wie in den vorangehenden Schritten beschrieben.

2. Ändern Sie den Wert **Profile Val**, falls er auf FALSE eingestellt ist. Klicken Sie hierzu in der Symbolleiste auf **Change value**. Das Fenster **Change Parameter** wird angezeigt. Geben Sie in das Feld **ProfileVal** den Wert TRUE ein und klicken Sie auf **Save**.



Nach dem Speichern der Änderung wird das Fenster geschlossen und der Wert **ProfileVal** wird auf TRUE gesetzt.

3. Starten Sie den Anwendungsserver neu und melden Sie sich am System an, damit die Änderungen wirksam werden.

Wenn der Wert **ProfileVal** trotz Neustarts nicht aktualisiert wurde, ist der Kernel des Anwendungsservers veraltet. Importieren Sie in diesem Fall den erforderlichen Kernelpatch wie in den vorangehenden Schritten beschrieben.

Hinweis: Bei den folgenden Kernelversionen wird der Profilwert mithilfe von Transaktion rz11 möglicherweise dynamisch ohne Neustart des Anwendungsservers aktiviert.

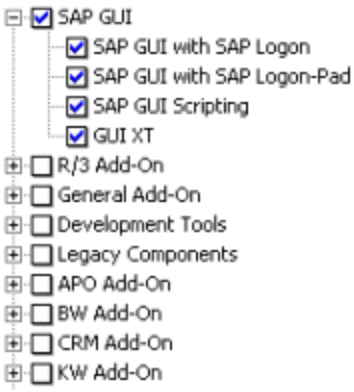
Version	Kernelversion	Patchlevel
4.6B, 4.6C, 4.6D	4.6D	972
6.10	6.10	391
6.20	Alle Versionen	Alle Level

Aktivieren der Skripterstellung auf dem SAP GUI 6.20-Client

Um Skripts in VuGen ausführen zu können, müssen Sie die Skripterstellung auf dem SAP GUI-Client aktivieren. Sie sollten die Clientkonfiguration auch dahingehend ändern, dass bestimmte Nachrichten unterdrückt werden, etwa beim Herstellen einer Verbindung oder beim Anhängen eines

Skripts an den GUI-Prozess. In den folgenden Schritten die Konfiguration des SAP GUI-Clients beschrieben, mit dem Sie arbeiten möchten.

1. **Während der Installation.** Aktivieren Sie bei der Installation des SAP GUI-Clients die Option **SAP GUI-Skripterstellung**.



2. **Nach der Installation.** Unterdrücken Sie Warnmeldungen. Öffnen Sie im SAP GUI-Client das Dialogfeld **Optionen**. Deaktivieren Sie auf der Registerkarte **Scripting** folgende Optionen:

   - **Notify when a script attaches to a running GUI**
   - **Notify when a script opens a connection**

   Alternativ können Sie diese Nachrichten auch unterdrücken, indem Sie die Werte **WarnOnAttach** und **WarnOnConnection** folgender Registrierungsschlüssel auf 0 setzen:

   `HKCU\SOFTWARE\SAP\SAPGUI Front\SAP Frontend Server\Security.`

Aufzeichnen von SAP GUI-Skripts

Die folgenden Schritte beschreiben einige der Voraussetzungen für das Aufzeichnen eines SAP GUI-Skripts.

Konfigurieren des Anwendungsservers für die Skripterstellung

Aus Sicherheitsgründen ist die Skripterstellung standardmäßig deaktiviert. Zum Aufzeichnen müssen Sie die Skripterstellung auf dem Anwendungsserver aktivieren. Legen Sie in der RZ11-Transaktion folgende Profilparameter so fest:

- sapgui/user_scripting TRUE
- sapgui/user_scripting_force_notification FALSE
- sapgui/user_scripting_set_readonly FALSE
- sapgui/user_scripting_disable_recording FALSE

Schließen der SAPLogon-Anwendung bei Aufzeichnung eines Mehrfachprotokoll-Skripts

Wenn Sie ein Mehrfachprotokoll-Skript aufzeichnen und der SAP GUI-Client Websteuerelemente enthält, schließen Sie die SAPLogon-Anwendung vor der Aufzeichnung.

Verwenden von modalen Dialogfeldern für die F1-Hilfe

Gehen Sie folgendermaßen vor, um den SAP GUI-Client anzuweisen, die F1-Hilfe in einem modalen Dialogfeld zu öffnen:

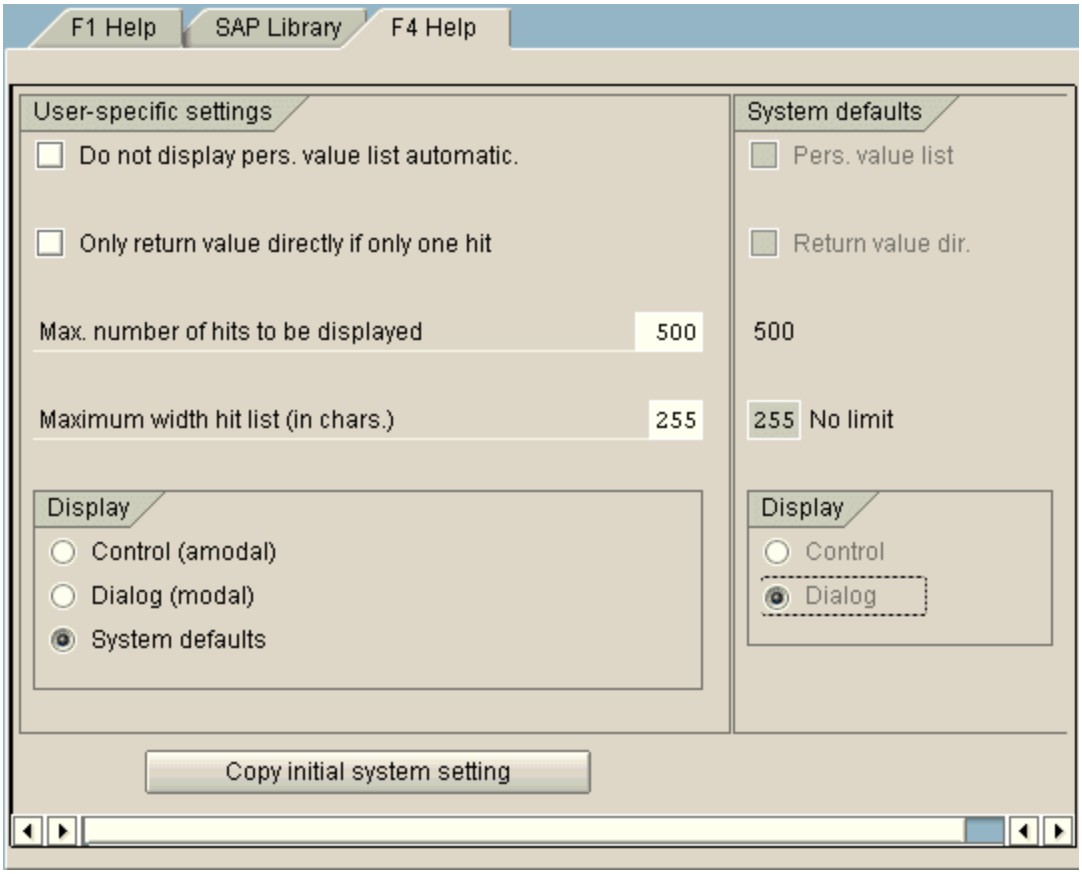
1. Wählen Sie **Help > Settings** aus.
2. Klicken Sie auf die Registerkarte **F1 Help**.
3. Wählen Sie im Abschnitt **Display** die Option **in modal dialog box** aus.

Verwenden von modalen Dialogfeldern für die F4-Hilfe

Hinweis: Diese Vorgehensweise kann nur vom Administrator ausgeführt werden.

Gehen Sie folgendermaßen vor, um den SAP GUI-Client anzuweisen, die F4-Hilfe in einem modalen Dialogfeld zu öffnen:

1. Vergewissern Sie sich, dass alle Benutzer vom Server abgemeldet sind.
2. Wählen Sie **Help > Settings** aus. Klicken Sie auf die Registerkarte **F4 Help**.



3. Klicken Sie im Bereich **Display** auf **System defaults**.
4. Klicken Sie unter **System defaults** im Teilbereich **Display** auf **Dialog**.
5. Speichern Sie die Änderungen. Klicken Sie hierzu auf **Copy initial system setting**.
6. Vergewissern Sie sich, dass in der Statusleiste die Meldung **Data was saved** angezeigt wird.
7. Schließen Sie die Sitzung und starten Sie den Dienst über die SAP-Verwaltungskonsole neu.

Wiedergeben von SAP GUI-Skripts

Die folgenden Schritte beschreiben die Voraussetzungen für die Wiedergabe von SAP GUI-Skripts.

Ersetzen des verschlüsselten Kennworts

Ersetzen Sie das verschlüsselte Kennwort in der Funktion **sapgui_logon**, das während der Aufzeichnung erstellt wurde, durch das Klartextkennwort. Hierbei handelt es sich um das zweite Argument der Funktion, das hinter dem folgenden Benutzernamen steht:

```
sapgui_logon("user", "pswd", "800", "EN");
```

Um zusätzliche Sicherheit zu erreichen, können Sie das Kennwort im Code verschlüsseln. Wählen Sie den Text des Kennworts (nicht *****) aus und klicken Sie im Kontextmenü auf **Encrypt string**. VuGen fügt an die Stelle des Kennworts eine **lr_decrypt**-Funktion ein:

```
sapgui_logon("user", lr_decrypt("3ea037b758"), "800", "EN");
```

Anzeigen der SAP GUI während der Wiedergabe (optional)

Wenn Sie ein Skript erstmals ausführen, sollten Sie VuGen so konfigurieren, dass die SAP GUI-Benutzeroberfläche bei der Wiedergabe angezeigt wird, damit die ausgeführten Operationen sichtbar werden. Wählen Sie **Wiedergeben > Laufzeiteinstellungen > SAP GUI > Allgemein** aus und aktivieren Sie dann **SAP-Client während der Wiedergabe anzeigen**. In einem Belastungsszenario empfiehlt es sich, diese Option zu deaktivieren, weil die Darstellung der Benutzeroberfläche für mehrere Vuser sehr ressourcenintensiv ist.

Ausführen von SAP GUI-Skripts in einem Szenario

Die folgenden Schritte enthalten Tipps zur Ausführung von SAP GUI-Skripts in einem Szenario.

LoadRunner-Controller-Einstellungen

Wenn Sie mit einem LoadRunner-Szenario arbeiten, legen Sie folgende Werte fest, wenn Sie Ihr Skript in einer Lasttestkonfiguration ausführen:

- **Ramp Up.** In der Planung: **One by one** (für eine korrekte Anmeldung).
- **Think Time.** In den Laufzeiteinstellungen: **Random think time**.
- **Users per load generator.** Im Dialogfeld **Load Generators**: **50 Vusers for machine with 512 MB of memory**.

Sicherstellen, dass der Agent im Prozessmodus ausgeführt wird

Vergewissern Sie sich, dass der Remote-Agent von LoadRunner (bzw. Performance Center-) im Prozessmodus ausgeführt wird. Der Dienstmodus wird nicht unterstützt.

Platzieren Sie hierzu die Maus in der Windows-Taskleiste über dem Symbol für den Agenten, woraufhin dessen Beschreibung angezeigt wird. **LoadRunner Agent Service** bedeutet, dass das Programm als Dienst ausgeführt wird.

In den folgenden Schritten wird beschrieben, wie Sie den Agenten als Prozess neu starten.

1. Beenden Sie den Agenten. Klicken Sie hierzu mit der rechten Maustaste auf das Symbol für den LoadRunner-Agenten und klicken Sie auf **Schließen**.
2. Führen Sie die Datei **magentproc.exe** im Ordner **launch_service\bin** auf dem LoadRunner-Installationsdatenträger aus.

3. Vergewissern Sie sich, dass beim nächsten Systemstart der richtige Agent gestartet wird. Ändern Sie hierzu den Starttyp des Agentendienstes von automatisch in manuell. Fügen Sie anschließend dem Windows-Startmenü eine Verknüpfung zu **magentproc.exe** hinzu.

   Terminal Sessions. Auf Computern, die SAP GUI-Vuser ausführen, kann die Anzahl der ausführbaren Vuser aufgrund der verfügbaren Grafikressourcen beschränkt sein. Zur Erhöhung der Vuser-Anzahl pro Computer können Sie zusätzliche Terminalserversitzungen auf den Lastgeneratorcomputern öffnen. Wählen Sie hierzu unter **Start > Programme > <Produktname> > Erweiterte Einstellungen** die Option **Agentenkonfiguration** aus und klicken Sie anschließend auf **Terminaldienste aktivieren**. Geben Sie die Anzahl der Terminalsitzungen in den Eigenschaften des Lastgeneratorcomputers an. Weitere Informationen finden Sie unter "Konfigurieren von Terminaldiensten".

-

  Hinweis: Wenn der LoadRunner-Agent in einer Terminalsitzung mit minimiertem Fenster ausgeführt wird, werden im Falle von Fehlern keine Snapshots erfasst.

Erweitern von SAP GUI-Skripts

Die folgenden Schritte beziehen sich auf zusätzliche Optionen, mit denen sich SAP GUI-Skripts erweitern lassen.

Aufzeichnen an der Cursorposition

VuGen ermöglicht die Aufzeichnung von Aktionen in einem bereits vorhandenen Skript. Hierzu können Sie entweder neue Aktionen einfügen oder vorhandene Aktionen ersetzen. Für diese Art der Aufzeichnung kann es folgende Gründe geben:

- Ihnen ist in den aufgezeichneten Aktionen ein Fehler unterlaufen.
- Die durchgeführten Aktionen waren korrekt, Sie benötigen aber zusätzliche Daten, z. B. zum Umgang mit Popup-Fenstern. Der SAP-Server gibt möglicherweise eine Warnung zum Bestand aus, die während der Aufzeichnungssitzung nicht erfolgt ist.

In den folgenden Schritten wird die Aufzeichnung an der Cursorposition beschrieben.

1. Öffnen Sie das Vuser-Skript im **Editor** und klicken Sie am linken Rand neben eine vorhandene Funktion.
2. Klicken Sie auf **Am Cursor aufzeichnen**. VuGen fordert Sie dazu auf, eine Auswahl vorzunehmen.
3. Wählen Sie die Option **Schritte in die Aktion einfügen** oder **Den Rest des Skripts überschreiben** aus.

   a. **Schritte in die Aktion einfügen** fügt die neu aufgezeichneten Schritte an der Cursorposition ein, ohne vorhandene Schritte zu überschreiben. Das neue Segment wird dabei zur Definition des Anfangs und des Endes des hinzugefügten Abschnitts in

Kommentare gesetzt. Diese Option empfiehlt sich für die Verarbeitung gelegentlich angezeigter Popup-Fenster, die bei der Aufzeichnung nicht vorhanden waren.

b. **Den Rest des Skripts überschreiben** ersetzt alle Schritte ab der Cursorposition bis zum Ende des Skripts. Wenn Sie diese Option auswählen, wird der Rest der aktuellen Aktion überschrieben und alle folgenden Aktionen werden gelöscht. Die Abschnitte **vuser_init** und **vuser_end** bleiben davon unberührt.

4. Klicken Sie auf **OK**. VuGen gibt das Skript bis zur Cursorposition wieder.

5. Warten Sie, bis die freischwebende Aufzeichnungssymbolleiste geöffnet wird.

6. Führen Sie die erforderlichen Aktionen im SAP GUI-Client durch und wechseln Sie dabei nach Bedarf zwischen Abschnitten und Aktionen.

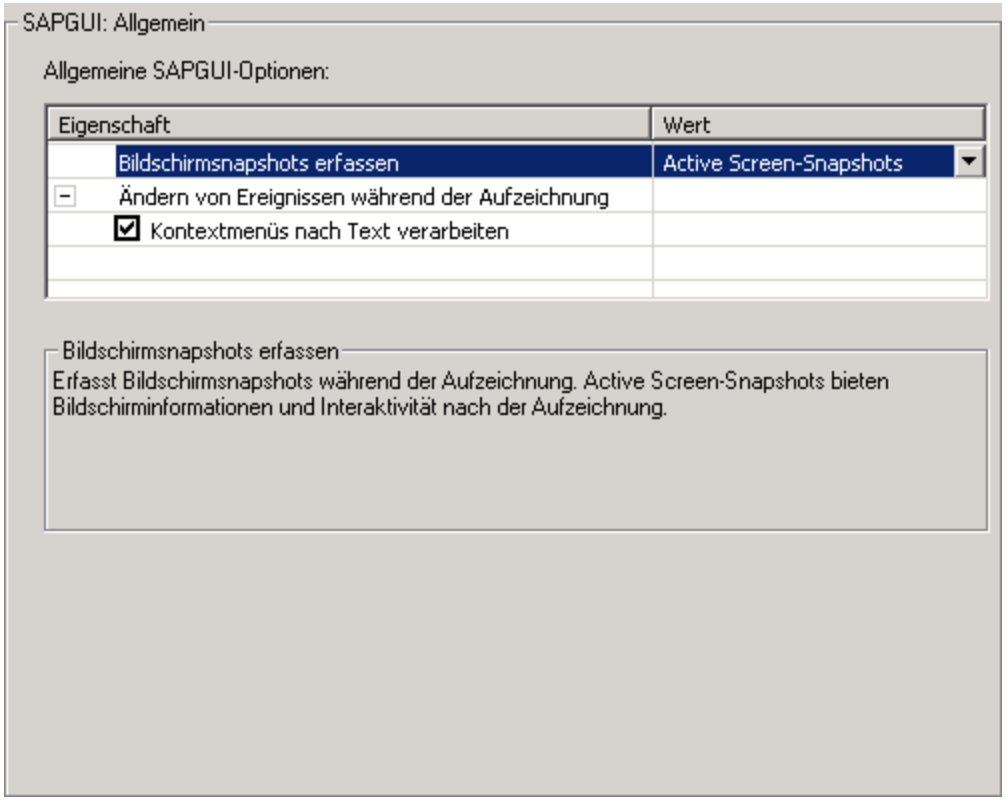
7. Zum Beenden der Aufzeichnungssitzung klicken Sie in der freischwebenden Symbolleiste auf **Anhalten** .

Interaktives Einfügen von Schritten in ein SAP GUI-Skript

Nach der Aufzeichnung können Sie entweder im **Editor** oder im **Schrittenavigator** manuell Schritte zum Skript hinzufügen. Zusätzlich zum manuellen Hinzufügen von neuen Funktionen können Sie auch direkt aus dem Snapshot heraus interaktiv neue Schritte für SAP GUI-Vuser hinzufügen. Über das Kontextmenü können Sie objektbezogene Schritte hinzufügen.

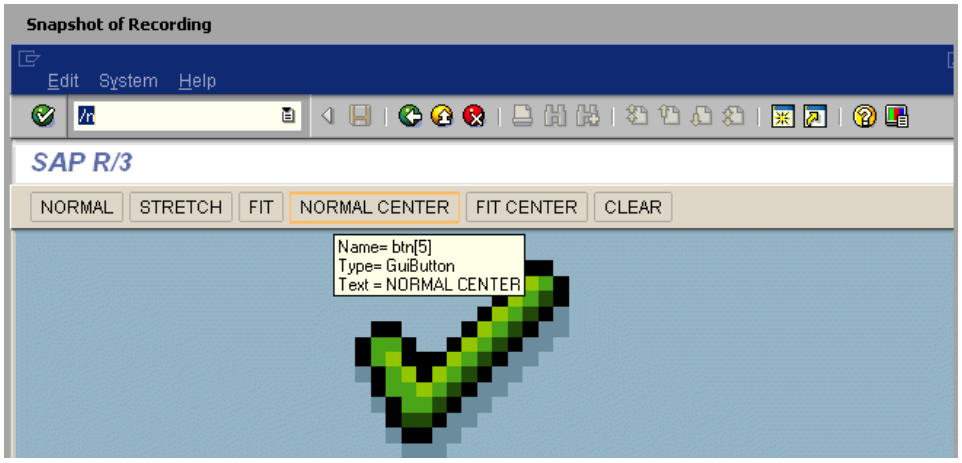
Wenn Sie einen Schritt aus einem Snapshot heraus hinzufügen, bestimmt VuGen die ID der einzelnen Objekte im SAP GUI-Clientfenster mithilfe der Active-Screen-Funktion (vorausgesetzt, Active-Screen-Snapshots wurden unter ""SAPGUI > Knoten "Allgemein"" auf Seite 242" aktiviert). In den folgenden Schritten wird das interaktive Einfügen eines Schritts für ein bestimmtes Objekt beschrieben.

1. Überprüfen Sie, ob bei der Aufzeichnung des Skripts in den Aufzeichnungsoptionen unter dem Strukturknoten **SAPGUI: Allgemein** die Option für die Aktivierung der Active-Screen-Snapshots ausgewählt war (standardmäßig aktiviert).



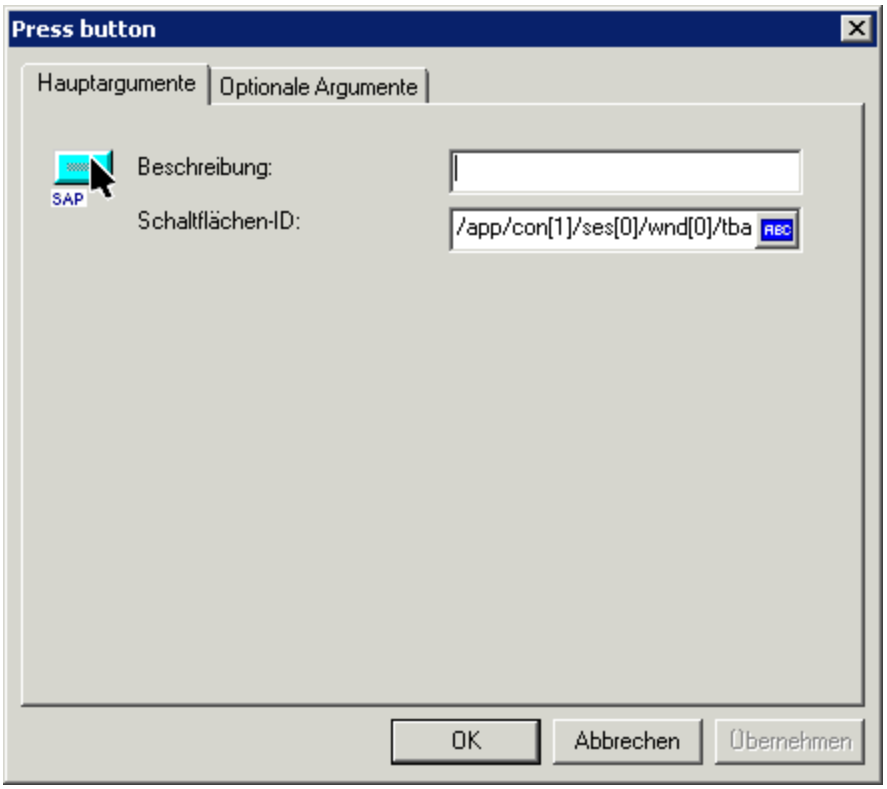
2. Klicken Sie in den Snapshot-Bereich.

3. Platzieren Sie die Maus über dem Objekt, für das Sie eine Funktion hinzufügen möchten. Vergewissern Sie sich, dass das Objekt von VuGen erkannt und umrandet wird.



4. Klicken Sie mit der rechten Maustaste auf das Objekt, klicken Sie auf **Neuen Schritt einfügen** und wählen Sie einen Schritt aus der Liste mit den für das Objekt verfügbaren Schritten aus.

Das Dialogfeld **Eigenschaften** des Schritts wird geöffnet; es enthält gegebenenfalls die Steuerelement-ID des Objekts. Wenn Sie beispielsweise für die Schaltfläche für die normale Mitte wie oben dargestellt einen Schritt **Auf Schaltfläche klicken** hinzufügen, wird im Feld **Eigenschaften** die folgende ID angezeigt:



5. Geben Sie im Feld **Description** den Namen des Objekts an. Klicken Sie auf **OK**. VuGen fügt den neuen Schritt hinter dem ausgewählten Schritt ein.

Hinweis: Sie können die Steuerelement-ID des Objekts abrufen, um sie an einem bestimmten Speicherort einzufügen. Wählen Sie hierzu im Kontextmenü **Steuerelement-ID kopieren** aus. Anschließend können Sie sie in ein Eigenschaftenfeld oder in der Skriptansicht direkt in den Code einfügen.

Hinzufügen von Verifizierungsfunktionen

Beim Arbeiten mit optionalen oder dynamischen Fenstern bzw. Frames können Sie mithilfe von Verifizierungsfunktionen bestimmen, ob das Fenster oder Objekt verfügbar ist. Ein optionales Fenster ist während einer SAP-Sitzung nicht permanent geöffnet. Mit dieser Funktion kann das Vuser-Skript auch dann weiterhin ausgeführt werden, wenn es auf ein optionales Fenster oder eine Ausnahme trifft.

Im ersten Beispiel wird geprüft, ob ein Fenster verfügbar ist. Falls ja, schließt der Vuser das Fenster, bevor er fortfährt.

```
if (!sapgui_is_object_available("wnd[1]"))
        sapgui_call_method("{ButtonID}",
            "press",
            LAST,
            AdditionalInfo=info1011");
sapgui_press_button(.....)
```

Das nächste Beispiel zeigt ein dynamisches Objekt in der Transaktion ME51N. Die Darstellung des Frames mit der Dokumentübersicht ist optional; der Frame kann über die Schaltfläche **Document overview on/off** ein- bzw. ausgeblendet werden.

Der Code liest die Beschriftung der Schaltfläche **Document overview** aus. Zeigt die Schaltfläche den Text **Document overview on** an, wird darauf geklickt und der Frame mit der Dokumentübersicht wird geschlossen.

```
    if(sapgui_is_object_available("tbar[1]/btn[9]"))
    {
        sapgui_get_text("Document overview on/off button",
            "tbar[1]/btn[9]",
            "paramButtonText",
            LAST);
        if(0 == strcmp("Document overview off", lr_eval_string("{paramButtonTex
t}")))
            sapgui_press_button("Document overview off",
                "tbar[1]/btn[9]",
                BEGIN_OPTIONAL,
                    "AdditionalInfo=sapgui1013",
                END_OPTIONAL);
    }
```

Abrufen von Informationen

Beim Arbeiten mit SAP GUI-Vusern können Sie den aktuellen Wert eines SAP GUI-Objekts mithilfe der **sapgui_get_<xxx>**-Funktionen abrufen. Sie können diesen Wert dann als Eingabe für andere Geschäftsprozesse verwenden oder ihn im Ausgabeprotokoll anzeigen.

Das folgende Beispiel zeigt, wie sich zum Abrufen der Auftragsnummer ein Teil der Statusmeldung speichern lässt.

1. Navigieren Sie zu der gewünschten Stelle im Text der Statusleiste und wählen Sie **Neuen Schritt** einfügen. Wählen Sie die Funktion **sapgui_status_bar_get_type** aus. Damit wird überprüft, ob der Vuser erfolgreich Text aus der Statusleiste abrufen kann.

2. Fügen Sie eine **if**-Anweisung ein, mit der die erfolgreiche Ausführung der vorherigen Anweisung überprüft wird. Speichern Sie bei erfolgreicher Ausführung den Wert des Arguments mithilfe der Funktion **sapgui_status_bar_get_param**.

   Mit der Funktion **sapgui_status_bar_get_param** wird die Auftragsnummer in einem benutzerdefinierten Parameter gespeichert. Im vorliegenden Beispiel stellt der zweite Index der Zeichenfolge in der Statusleiste die Auftragsnummer dar.

```
sapgui_press_button("Save (Ctrl+S)",
    "tbar[0]/btn[11]",
    BEGIN_OPTIONAL,
        "AdditionalInfo=sapgui1038",
    END_OPTIONAL);
sapgui_status_bar_get_type("Status");
if(0==strcmp(lr_eval_string("{Status}"),"Success"))
  sapgui_status_bar_get_param("2", " Order_Number ");
```

Während der Ausführung des Tests werden der Wert und der Parametername in das Ausführungsprotokoll geschrieben:

```
Action.c(240): Pressed button " Save (Ctrl+S)"
Action.c(248): The type of the status bar is "Success"
Action.c(251): The value of parameter 2 in the status bar is "33232"
```

Speichern von Datumsinformationen

Wenn Sie ein Skript erstellen, das Datumsangaben enthält, kann es passieren, dass es nicht ordnungsgemäß ausgeführt wird. Ein Skript, das Sie am 2. Juni aufzeichnen und am 3. Juni wiedergeben, enthält beispielsweise fehlerhafte Datumsfelder. Daher müssen Sie das Datum während der Ausführung als Parameter speichern und diesen gespeicherten Wert als Eingabe für andere Datumsfelder verwenden. Verwenden Sie zum Speichern des aktuellen Datums oder der aktuellen Uhrzeit während der Skriptausführung die Funktion **lr_save_datetime**. Fügen Sie diese Funktion vor der Funktion ein, für die die Datumsinformationen erforderlich sind. Hinweis: Das Datumsformat ist abhängig von Ihrem jeweiligen Gebietsschema. Verwenden Sie in der Funktion **lr_save_datetime** das Datumsformat für Ihr Land. Geben Sie beispielsweise zur Darstellung von **Tag.Monat.Jahr** das Format "`%d.%m.%Y`" an.

Im folgenden Beispiel speichert die Funktion **lr_save_datetime** das aktuelle Datum. Dieser Wert wird später von der Funktion **sapgui_set_text** verwendet, um das Lieferdatum auf zwei Tage später festzulegen.

```
lr_save_datetime("%d.%m.%Y", DATE_NOW + (2 * ONE_DAY),              "paramDateTo
dayPlus2");
sapgui_set_text("Req. deliv.date",
        "{paramDateTodayPlus2}",
        "usr/ctxtRV45A-KETDAT",
        BEGIN_OPTIONAL,
            "AdditionalInfo=sapgui1025",
        END_OPTIONAL);
```

Weitere SAP-Ressourcen

Weitere Informationen finden Sie auf der Website von SAP unter www.sap.com oder an folgenden Stellen:

- **SAP-Hinweise** – https://websmp103.sap-ag.de/notes

Hinweis 480149: New profile parameter for user scripting on the front end

Hinweis 587202: Drag & Drop is a known limitation of the SAP GUI interface

- **SAP-Patches** – https://websmp104.sap-ag.de/patches

SAP GUI for Windows – SAP GUI 6.20 Patch 32 (oder höher)

SAP GUI, SAP Web und SAP (Click & Script) - Fehlerbehebung und Einschränkungen

In diesem Abschnitt werden die Fehlerbehebung und die für SAP GUI-, SAP Web- und SAP (Click and Script)-Protokolle geltenden Einschränkungen beschrieben.

SAP (Click & Script) – Allgemeine Einschränkungen

- Sie können keine Transaktionen definieren, um die Zeit zu messen, die für ausgewählte Schritte beansprucht wurde, die in einem modalen Dialogfeld ausgeführt wurden.
- Wenn Sie während der Aufzeichnung in einer Strukturansicht auf eine Zelle außerhalb des Textes doppelklicken, zeichnet VuGen das Ereignis "Select" anstelle des Ereignisses "Activate" auf. Dadurch fehlt bei der Wiedergabe das POST-Ereignis.
- Enthält die Liste der abgerufenen Werte eine Bildlaufleiste, kann der Benutzer keinen Eintrag auswählen, für den er die Bildlaufleiste benötigt, um den Eintrag vom Server abzurufen.
- Die Skalierbarkeit ist abhängig von der Größe und der Funktionalität der Geschäftsprozesse geringer als beim SAP-Webprotokoll.
- Keine Unterstützung der Laufzeiteinstellung **Durch alternative Navigation ersetzen**.
- In bestimmten SAP-Umgebungen (wie zum Beispiel 6.20 und 6.40) treten beim Wiedergeben von Tests, in denen die Funktion **web_element** zur Auswahl eines Elements mit dem Tagnamen "TD" aufgerufen wird, Fehler auf.
- Wenn Sie während der Aufzeichnung anstatt eines Steuerelements der Benutzeroberfläche die Tastatur verwenden (wenn Sie beispielsweise auf die Eingabetaste drücken, anstatt auf die Schaltfläche **Anmelden** zu klicken), wird dieser Schritt möglicherweise nicht aufgezeichnet. Es wird allgemein für die Aufzeichnung von Skripts empfohlen, anstatt der Tastatur die Steuerelemente der Benutzeroberfläche zu nutzen.

Allgemeine SAPGUI-Einschränkungen

- Wenn der Geschäftsprozess eines Kunden in SAP GUI 7.30 die Auswahl eine Elements aus einer Kombinationsliste umfasst, wird der Geschäftsprozess in LoadRunner, Version 9.52 und höher, möglicherweise nicht korrekt wiedergegeben.

Problemumgehung: Fügen Sie den Pfad zum SAP GUI-Installationsordner der Windows-Umgebungsvariablen PATH hinzu.

Frage 1: Ein Skript wurde erfolgreich aufgezeichnet. Warum tritt bei der Wiedergabe ein Fehler auf?

Antwort: Vergewissern Sie sich in LoadRunner, dass der LoadRunner-Remote-Agent im Prozessmodus ausgeführt wird. Der Dienstmodus wird nicht unterstützt. Weitere Informationen finden Sie unter "Wiedergeben von SAP GUI-Skripts" auf Seite 865.

Frage 2: Warum wurden bestimmte SAP GUI-Steuerelemente nicht aufgezeichnet?

Antwort: Einige SAP GUI-Steuerelemente werden nur im Kontext ihrer jeweiligen Menüs oder Symbolleisten unterstützt. Versuchen Sie, die fragliche Aktion alternativ auszuführen – über eine Menüoption, das Kontextmenü, die Symbolleiste o. ä.

Frage 3: Warum können Skripts in VuGen nicht aufgezeichnet oder wiedergegeben werden?

Antwort:

1. Überprüfen Sie, ob der aktuelle Patch von SAP GUI 6.20 installiert ist. Die Mindestanforderung ist Patchlevel 32.

2. Vergewissern Sie sich, dass Scripting aktiviert ist. Informationen hierzu finden Sie unter "Konfigurieren der SAP-Umgebung" auf Seite 858.

3. Überprüfen Sie, ob im SAP GUI for Windows-Client Benachrichtigungen deaktiviert sind. Klicken Sie hierzu auf **Customizing of Local Layout** oder drücken Sie Alt+F12. Klicken Sie auf **Options** und anschließend auf die Registerkarte **Scripting**. Deaktivieren Sie beide Optionen unter **Notify**.

Frage 4: Was bedeuten die Popup-Fehlermeldungen, die bei dem Versuch, das Skript auszuführen, angezeigt werden?

Antwort: In einigen SAP-Anwendungen wird das letzte Layout der Benutzer gespeichert, z. B. die Information, ob Frames sichtbar oder verborgen sind. Wenn das gespeicherte Layout seit der Aufzeichnung des Skripts geändert wurde, können Probleme bei der Wiedergabe auftreten. In der ME52N-Transaktion wird beispielsweise durch die Schaltfläche **Document overview Off/On** die Anzahl der sichtbaren Frames geändert.

Gehen Sie in diesem Fall folgendermaßen vor:

1. Stellen Sie vor der Wiedergabe des Skripts das Layout der Transaktion vor der Aufzeichnung wieder her. Welches Layout die Transaktion vor der Aufzeichnung hatte, können Sie dem Snapshot-Viewer entnehmen.

2. Stellen Sie das für die Wiedergabe gewünschte Layout her, indem Sie dem Skript Anweisungen hinzufügen. Wenn beispielsweise die Wiedergabe durch einen offenen Frame

verhindert wird, fügen Sie eine Verifizierungsfunktion ein, mit der Sie überprüfen, ob der Frame offen ist. Klicken Sie in diesem Fall auf eine Schaltfläche, um ihn zu schließen. Beispiele zur Verifizierung finden Sie unter "Erweitern von SAP GUI-Skripts" auf Seite 867.

Frage 5: Kann die Einzelanmeldung verwendet werden, wenn das Skript auf einem Remotecomputer ausgeführt wird?

Antwort: Nein. VuGen unterstützt keine Verbindungen über Einzelanmeldung. Öffnen Sie die erweiterten Optionen in Ihrem SAP GUI-Client und deaktivieren Sie die Funktion **Sichere Netzwerkkommunikation aktivieren**. Hinweis: Nach dem Ändern der Verbindungseinstellungen müssen Sie Ihr Skript erneut aufzeichnen.

Frage 6: Kann VuGen alle SAP-Objekte aufzeichnen?

Antwort: Von der SAP GUI-Skripterstellung nicht unterstützte Objekte können nicht aufgezeichnet werden. Weitere Informationen zu solchen Objekten finden Sie im Aufzeichnungsprotokoll.

Frage 7: Werden alle Geschäftsprozesse unterstützt?

Antwort: VuGen unterstützt keine Geschäftsprozesse, durch die Steuerelemente von Microsoft Office-Programmen aufgerufen werden oder die GuiXT erfordern. Sie können **GuiXT** im Menü **Optionen** des SAP GUI for Windows-Clients deaktivieren.

Frage 8: Warum ist die Liste mit Servernamen in den Aufzeichnungsoptionen unter "Automatisch anmelden" leer?

Antwort: Dies ist ein bekannter Fehler von SAP GUI Client 7.20. Um das Problem zu beheben, kopieren Sie die Datei **saplogon.ini** im Ordner **%APPDATA%\SAP\Common**, wobei **%APPDATA%** die Umgebungsvariable für den Anwendungsdatenordner ist, der sich direkt unterhalb des Ordners mit den Benutzerprofilen befindet. Fügen Sie die Datei in den Ordner **%WINDIR%** (C:\Windows) ein.

Siebel-Webprotokoll

Übersicht über das Siebel-Webprotokoll

Das Siebel-Webprotokoll ähnelt dem standardmäßigen Web-Vuser-Protokoll, wobei es durch einige Änderungen an den Standardeinstellungen für die Zusammenarbeit mit der Siebel CRM-Anwendung (Customer Relationship Management) angepasst wurde.

Sie zeichnen die typischen Aktivitäten während einer Siebel-Sitzung auf. VuGen zeichnet die Aktionen auf und generiert Funktionen mit dem Präfix **web_**, welche diese Aktionen emulieren.

Siebel-Webaufzeichnungsoptionen und -Laufzeiteinstellungen

Vor dem Aufzeichnen eines Siebel-Web-Vuser-Skripts legen Sie die folgenden Aufzeichnungsoptionen fest:

- Knoten **Aufzeichnen**: **HTML-basiertes Skript**

  HTML erweitert - Skripttyp: **Ein Skript, das nur explizite URLs enthält**

  HTML erweitert - Nicht-HTML-generierte Elemente: **Nicht aufzeichnen**

- Knoten **Erweitert**: Deaktivieren Sie das Kontrollkästchen **Kontext für jede Aktion zurücksetzen**.

  Vor dem Ausführen eines Siebel-Web-Vuser-Skripts legen Sie die folgende Laufzeiteinstellung fest:

  Deaktivieren Sie in den Laufzeiteinstellungen im Knoten **Browseremulation** das Kontrollkästchen **Bei jeder Iteration neuen Benutzer simulieren**.

Aufzeichnen von Transaktionsaufschlüsselungsdaten

VuGen stellt ein Diagnosetool für die Analyse der Transaktionskomponenten in Ihren Tests zur Verfügung - die **Transaktionsaufschlüsselung**. Mithilfe der Transaktionsaufschlüsselung können Sie Engpässe und andere Probleme innerhalb Ihres Skripts identifizieren.

Es wird empfohlen, beim Vorbereiten des Skripts für die Transaktionsanalyse am Ende jeder Transaktion einen Latenzzeitschritt (je eine Sekunde pro Stunde Testzeit) hinzuzufügen. Weitere Informationen zum Hinzufügen von Latenzzeitschritten finden Sie unter "Einfügen von Schritten in ein Skript" auf Seite 434.

Um die Transaktionsanalysedaten aufzeichnen zu können, müssen Sie die Parametrierungsfunktionen innerhalb des Skripts modifizieren.

Vorbereiten des Skripts auf die Transaktionsaufschlüsselung

1. Identifizieren Sie den bei der Parametrierung im Skript ersetzten Wert für die Sitzungs-ID.

```
/* Registering parameter(s) from source task id 15
   // {Siebel_sn_body4} = "28eMu9uzkn.YGFFevN1FdrCfCCOc8c_"
   // */
   web_reg_save_param("Siebel_sn_body4",
       "LB/IC=_sn=",
       "RB/IC==;",
       "Ord=1",
       "Search=Body",
       "RelFrameId=1",
       LAST);
```

2. Kennzeichnen Sie die nächste **web_submit_data**-Funktion als Transaktion, indem Sie sie durch die Funktionen **lr_start_transaction** und **lr_end_transaction** einfassen.

3. Fügen Sie vor dem Ende der Transaktion einen Aufruf der Funktion **lr_transaction_instance_**

add_info ein, wobei der erste Parameter (**0**) erforderlich ist und die Sitzungs-ID das Präfix **SSQLBD** aufweist.

```
lr_start_transaction("LoginSQLSync");
    web_submit_data("start.swe_2",
        "Action=http://design/callcenter_enu/start.swe",
        "Method=POST",
         "RecContentType=text/html",
        "Referer=http://design/callcenter_enu/start.swe",
        "Snapshot=t2.inf",
         "Mode=HTML",
        ITEMDATA,
        "Name=SWEUserName", "Value=wrun", ENDITEM,
        "Name=SWEPassword", "Value=wrun", ENDITEM,
        "Name=SWERememberUser", "Value=Yes", ENDITEM,
        "Name=SWENeedContext", "Value=false", ENDITEM,
        "Name=SWEFo", "Value=SWEEntryForm", ENDITEM,
        "Name=SWETS", "Value={SiebelTimeStamp}", ENDITEM,
        "Name=SWECmd", "Value=ExecuteLogin", ENDITEM,
        "Name=SWEBID", "Value=-1", ENDITEM,
        "Name=SWEC", "Value=0", ENDITEM,
        LAST);
lr_transaction_instance_add_info(0,lr_eval_string("SSQLBD:{Siebel_sn_body4}")
);
lr_end_transaction("LoginSQLSync", LR_AUTO);
```

Hinweis: Um Sitzungs-ID-Konflikte zu vermeiden, sollten Sie sicherstellen, dass sich die Vuser am Ende jeder Sitzung von der Datenbank abmelden.

Siebel-Web - Fehlerbeseitigung und Beschränkungen

In diesem Abschnitt werden die Fehlerbehebung sowie die für Siebel-Web-Vuser-Skripts geltenden Einschränkungen beschrieben.

Fehler beim Klicken auf "Zurück" oder "Aktualisieren"

Eine Fehlermeldung in Zusammenhang mit dem Klicken auf die Schaltflächen **Zurück** oder **Aktualisieren** hat normalerweise den folgenden Inhalt:

Ihre Anfrage konnte nicht bearbeitet werden. Sie haben möglicherweise die Schaltflächen **Zurück** oder **Aktualisieren** des Browsers verwendet, um zu dieser Stelle zu gelangen.

Ursache: Dieses Problem kann die folgenden Ursachen haben:

- Der SWEC-Parameter wurde für die aktuelle Anfrage nicht korrekt korreliert.
- Der SWETS-Parameter wurde für die aktuelle Anfrage nicht korrekt korreliert.

- Die Anfrage wurde zweimal an den Siebel-Server übermittelt, ohne den SWEC-Parameter zu aktualisieren.
- Bei einer vorherigen Anfrage hätte für den Download durch den Browser ein Frame geöffnet werden müssen. Dieser Frame wurde auf dem Server nicht erstellt, wahrscheinlich weil die SWEMethod seit der Aufzeichnung aktualisiert wurde.

Identische Werte

Normalerweise erfolgt bei einem Fehler aufgrund identischer Werte folgende Rückmeldung von der Webseite:

@0`0`3`3``0`UC`1`Status`Error`SWEC`10`0`1`Errors`0`2`0`Level0`0`ErrMsg`The same values for 'Name' already exist. If you would like to enter a new record, please make sure that the field values are unique.`ErrCode`28591`

Ursache: Möglicherweise dupliziert einer der Werte in der Anfrage (im Beispiel oben der Wert im Feld **Name**) einen Wert in einer anderen Zeile der Datenbanktabelle. Dieser Wert muss durch einen eindeutigen Wert ersetzt werden, der für jede Iteration pro Benutzer verwendet wird. Die empfohlene Lösung besteht darin, die Zeilen-ID durch den zugehörigen Parameter zu ersetzen, anstatt sicherzustellen, dass diese eindeutig ist.

HTTP-Antwort "No Content"

Normalerweise erfolgt bei einem Fehler vom Typ **No Content HTTP Response** die folgende HTTP-Antwort :

HTTP/1.1 204 No Content

Server: Microsoft-IIS/5.0

Date: Fri, 31 Jan 2003 21:52:30 GMT

Content-Language: en

Cache-Control: no-cache

Ursache: Möglicherweise wurde die Zeilen-ID entweder überhaupt nicht oder inkorrekt korreliert.

Wiederherstellen des Kontexts

Normalerweise erfolgt bei einem Fehler vom Typ **Restoring the Context** folgende Rückmeldung von der Webseite:

@0`0`3`3``0`UC`1`Status`Error`SWEC`9`0`1`Errors`0`2`0`Level0`0`ErrMsg`An error happened during restoring the context for requested location`ErrCode`27631`

Ursache: Möglicherweise wurde die Zeilen-ID entweder überhaupt nicht oder inkorrekt korreliert.

Datensatz nicht vorhanden

Normalerweise erfolgt bei einem Fehler vom Typ **Cannot locate record** folgende Rückmeldung von der Webseite:

@0`0`3`3``0`UC`1`Status`Error`SWEC`23`0`2`Errors`0`2`0`Level0`0`ErrMsg`Cannot locate record within view: Contact Detail - Opportunities View applet: Opportunity List Applet.`ErrCode`27573`

Ursache: Möglicherweise enthält der Eingabename **SWERowId** keine Zeilen-ID für einen Datensatz innerhalb der Webseite. Dieser Eingabename hätte parametriert werden müssen. Der Speicherort des Quellwerts für den Parameter wurde möglicherweise geändert.

Ende der Datei

Normalerweise erfolgt bei einem Fehler vom Typ **End of File** folgende Rückmeldung von der Webseite:

@0`0`3`3``0`UC`1`Status`Error`SWEC`28`0`1`Errors`0`2`0`Level0`0`ErrMsg`An end of file error has occurred. Please continue or ask your systems administrator to check your application configuration if the problem persists.`ErrCode`28601`

Ursache: Möglicherweise enthält der Eingabename **SWERowId** keine Zeilen-ID für einen Datensatz innerhalb der Webseite. Dieser Eingabename hätte parametriert werden müssen. Der Speicherort des Quellwerts für den Parameter wurde möglicherweise geändert.

Suchkategorien können nicht abgerufen werden

Normalerweise erfolgt bei einem Fehler vom Typ **Unable to Retrieve Search Categories** folgende Rückmeldung von der Webseite:

Ursache: Möglicherweise wurde der Frame für die Suche nicht vom Server heruntergeladen. Dieser Fehler tritt auf, wenn in der vorherigen Anfrage die Anweisung an den Server, den Suchframe zu erstellen, nicht korrekt formuliert war.

Silverlight-Protokoll

Silverlight-Protokoll - Übersicht

Microsoft Silverlight ist ein Webanwendungsframework, das Grafiken, Animationen und Interaktivität unterstützt. Das Silverlight-Protokoll ermöglicht das Aufzeichnen von Anwendungen, die mit Microsoft Silverlight erstellt wurden. Es beinhaltet das Protokoll Web (HTTP/HTML) als Untermenge sowie zahlreiche zusätzliche Funktionen, Aufzeichnungsoptionen und Laufzeiteinstellungen.

Zum Aufzeichnen von abstrakten Vuser-Skripts können Sie in den Aufzeichnungsoptionen WSDL-Dateien importieren, die von Ihrer Anwendung verwendet werden.

Importieren von WSDL-Dateien

In den folgenden Schritten wird beschrieben, wie WSDL-Dateien manuell oder automatisch in ein Silverlight-Vuser-Skript importiert werden. Alternativ können Sie WSDL-Dateien deaktivieren und SOAP-Anforderung generieren. Alle diese Optionen werden im Knoten **Silverlight > Dienste** des Dialogfelds **Aufzeichnungsoptionen** durchgeführt. Informationen zur Benutzeroberfläche finden Sie unter "Knoten "Silverlight > Dienste"" auf Seite 243.

Automatisches Suchen von WSDL-Dateien

Um VuGen für das automatische Erkennen und Suchen der von Ihrem Skript verwendeten WSDL-Dateien zu konfigurieren, wählen Sie **In Skript enthaltene WSDL-Dateien verwenden** und **WSDL-Dateien automatisch erkennen und Dienste während der Codegenerierung importieren** aus. Wird eine WSDL-Datei erkannt, die nicht importiert werden kann, werden Sie im Feld **Benachrichtigungen für die Codegenerierung** benachrichtigt.

Manuelles Suchen von WSDL-Dateien

Für die manuelle Suche nach WSDL-Dateien stehen im Dialogfeld **Dienst hinzufügen** zahlreiche Möglichkeiten zur Verfügung. Um eine WSDL-Datei zu suchen, deren URL bekannt ist, verwenden Sie die Option **URL**. Befindet sich die WSDL-Datei auf Ihrem lokalen Computer, verwenden Sie die Option **Datei**. Um die WSDL-Historie (Liste der zuvor importierten WSDL-Dateien) nach der WSDL-Datei zu durchsuchen, wählen Sie **Zuvor importiert** aus und klicken zum Öffnen der Liste auf **...**.

Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Dienst hinzufügen/bearbeiten"" auf Seite 244.

Deaktivieren von WSDL-Dateien

Sie können WSDL-Dateien deaktivieren und stattdessen SOAP-Anforderungen generieren. Dies führt zu einem Skript mit einer tieferen Ebene, steigert jedoch im Allgemeinen die Skriptleistung. Wählen Sie zum Deaktivieren von WSDL-Dateien **Keine WSDL-Dateien verwenden** aus.

Erweiterte Sicherheitseinstellungen

Sicherheits- und Kennworteinstellungen können Sie im Dialogfeld **Daten für das Szenario zu Protokoll und Sicherheit** ändern. Weitere Informationen finden Sie unter "Dialogfeld "Daten für das Szenario zu Protokoll und Sicherheit"" auf Seite 245.

Silverlight - Fehlerbehebung und Einschränkungen

In diesem Abschnitt werden die Fehlerbehebung und die für das Silverlight-Protokoll geltenden Einschränkungen beschrieben.

- Während der Aufzeichnung einer in Silverlight entwickelten Website wird der Schritt **Silverlight installieren** auch dann aufgezeichnet, wenn der Aufzeichnungsprozess die Installation des Silverlight-Plug-ins nicht einschloss.

  Lösung

  Konfigurieren Sie die Laufzeiteinstellungen so, dass die folgende Adresse ausgeschlossen wird: http://go.microsoft.com/fwlink/?LinkId=108181

  a. Wählen Sie **Laufzeiteinstellungen > Internetprotokoll > Download-Filter** aus.

  b. Wählen Sie das Optionsfeld **In der Liste enthaltene Adressen ausschließen** aus.

  c. Klicken Sie auf **Hinzufügen**, und fügen Sie "http://go.microsoft.com/fwlink/?LinkId=108181" zur Liste hinzu.

Hinweis: Der Teil **?LinkId=108181** der URL-Adresse ist möglicherweise nicht statisch und muss u.U. aktualisiert werden.

- REST-Dienste generieren keine Silverlight-Dienstaufrufe. Sie können jedoch aufgezeichnet und wiedergegeben werden.
- Der WSDL-Speicherort kann im Dialogfeld **Szenario zu Protokoll und Sicherheit** nicht bearbeitet werden.
- Die Schaltfläche "Aktualisieren" im Silverlight Service-Knoten im Dialogfeld "Aufzeichnungsoptionen" aktualisiert den Service, wenn der WSDL-Speicherort sich nicht geändert hat.
- Wenn sich der WSDL-Speicherort geändert hat, wird der Service erneut importiert (Service löschen und Service erneut importieren).
- Duplexbindungen (Polling) für WCF-Webdienste werden nicht unterstützt.
- Der Silverlight 4- und Silverlight 5-Client werden zwar unterstützt, Anwendungen, die unter Verwendung der neuen Kommunikationsfunktionen, z. B. net.tcp-Bindung, entwickelt wurden, jedoch nicht.
- Silverlight-Steuerelemente werden von VuGen-Snapshot-Viewer nicht unterstützt.
- Anwendungen, in denen Japanisch, Koreanisch, vereinfachtes oder traditionelles Chinesisch verwendet wird, werden vom Silverlight nicht unterstützt.

Webprotokoll (HTTP/HTML)

Es gehört zu den *Web-Vuser*-Protokollen von LoadRunner. Dieser Abschnitt enthält Informationen speziell zum Web (HTTP/HTML)-Vuser-Protokoll. Informationen, die allgemein für alle Web-Vuser-Protokolle gelten, finden Sie unter "Webprotokolle (Allgemein)" auf Seite 898.

Webprotokoll (HTTP/HTML) – Übersicht

Hinweis: Dieser Abschnitt bezieht sich nur auf Web (HTTP/HTML)-Vuser-Skripts.

Das Vuser-Webprotokoll (HTTP/HTML) emuliert die Kommunikation zwischen Browser und Webserver auf HTTP- oder HTML-Ebene.

Hinweis: Es gehört zu den *Web-Vuser*-Protokollen von LoadRunner. Eine vollständige Liste der Web-Vuser-Protokolle finden Sie unter "Web-Vuser-Typen" auf Seite 900.

Dieses Thema liefert einen Überblick über die verschiedenen Themen zu Web (HTTP/HTML)-Vuser-Skripts.

Wann sollten Sie das Web (HTTP/HTML)-Vuser-Protokoll einsetzen?

Verwenden Sie das Web (HTTP/HTML)-Vuser-Protokoll für Browseranwendungen mit Applets und VB Script oder für Anwendungen, die nicht in einem Browser ausgeführt werden.

Verwenden Sie das Web (HTTP/HTML)-Vuser-Protokoll, wenn die Client-Server-Kommunikation über http/s-Verbindungen abläuft und die Kommunikation nicht so komplex ist, als dass Inhaltsänderungen erforderlich sind. Falls Inhaltsänderungen erforderlich sind, empfiehlt sich die Verwendung des TruClient-Protokolls. Weitere Informationen zum TruClient-Protokoll finden Sie unter "TruClient-Protokoll " auf Seite 537.

Weitere Informationen zur Auswahl eines Vuser-Protokolls finden Sie unter "Übersicht über den Protocol Advisor" auf Seite 133.

Web (HTTP/HTML)-Vuser-Technologie

Sie können VuGen zum Entwickeln von Web (HTTP/HTML)-Vuser-Skripts verwenden. Zum Aufzeichnen eines Web (HTTP/HTML)-Vuser-Skripts (HTTP/HTML) navigieren Sie durch eine Website und führen dabei typische Benutzeraktivitäten aus. VuGen zeichnet die Aktionen auf und generiert ein Web (HTTP/HTML)-Vuser-Skript. Das Skript enthält detaillierte Informationen über den aufgezeichneten Verkehr. Wenn Sie das Skript ausführen, emuliert der daraus resultierende Vuser einen Benutzer, der auf das Internet zugreift.

Weitere Informationen finden Sie unter "Web Vuser-Technologie" auf Seite 899.

Die nachfolgende Tabelle enthält eine Liste der LoadRunner-Dokumentation in Bezug auf die Entwicklung eines Web (HTTP/HTML)-Vuser-Skripts.

Thema	Beschreibung
Erstellen eines Web (HTTP/HTML)-Vuser-Skripts	Allgemeine VuGen-Dokumentation zum Erstellen von Vuser-Skripts ["Erstellen von über Vuser-Skripts – Übersicht" auf Seite 148].

Thema	Beschreibung
Aufzeichnen	Zusätzlich zur allgemeinen Dokumentation zum Aufzeichnen von Vuser-Skripts ["Aufzeichnen – Übersicht" auf Seite 162] siehe: • "Übersicht über Aufzeichnungsebenen" auf Seite 256 **Aufzeichnungsoptionen** Sie können die folgenden Aufzeichnungsoptionen für Ihr Web (HTTP/HTML)-Vuser-Skript konfigurieren: • "Allgemein > Skript (Knoten)" auf Seite 196 • "Allgemein > Protokolle (Knoten)" auf Seite 193 • "Allgemein > Aufzeichnen (Knoten)" auf Seite 194 • "Netzwerk > Portzuordnung (Knoten)" auf Seite 224 • "HTTP > Erweitert (Knoten)" auf Seite 202 • HTTP > Correlation Node • "Datenformaterweiterung > Kettenkonfiguration (Knoten)" auf Seite 184 • "Datenformaterweiterung > Codegenerierung (Knoten)" auf Seite 188
Korrelieren	Zusätzlich zur VuGen-Dokumentation zum Korrelieren von Vuser-Skripts ["Korrelieren – Übersicht" auf Seite 281] siehe: • "Korrelieren von Skripts - Web (Manuell)" auf Seite 294 • "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912 • "Verwenden des JavaScript-Moduls von LoadRunner" auf Seite 885

Thema	Beschreibung
Wiedergeben	Zusätzlich zur VuGen-Dokumentation zum Wiedergeben von Vuser-Skripts ["Übersicht über die Wiedergabe" auf Seite 329] siehe: • "Browseremulation - Übersicht" auf Seite 905 • "Arbeiten mit Daten im Cache" auf Seite 910 **Laufzeiteinstellungen** Sie können die folgenden Laufzeiteinstellungen für Ihr Web (HTTP/HTML)-Vuser-Skript konfigurieren: • Allgemein – Ausführungslogik, Takt, Protokoll, Latenzzeit, Zusätzliche Attribute, Verschiedenes • "Netzwerk > Geschwindigkeitssimulation (Knoten)" auf Seite 394 • "Browser > Browseremulation (Knoten)" auf Seite 338 • "Internetprotokoll > Proxy (Knoten)" auf Seite 385 • "Internetprotokoll > Einstellungen (Knoten)" auf Seite 371 • "Internetprotokoll > Download-Filter (Knoten)" auf Seite 370 • "Internetprotokoll > ContentCheck (Knoten)" auf Seite 369
Debuggen	Informationen zum Debuggen von Vuser-Skripts finden Sie in der allgemeinen VuGen-Dokumentation["Debuggen – Übersicht" auf Seite 413].
Parametrierung	Zusätzlich zur VuGen-Dokumentation zum Parametrieren von Vuser-Skripts ["Parametrierung – Übersicht" auf Seite 436] siehe: • "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912 • "Verwenden des JavaScript-Moduls von LoadRunner" auf der nächsten Seite
Hinzufügen von Lasttestfunktionen	Zusätzlich zur VuGen-Dokumentation zum Hinzufügen von Lasttestfunktionen ["Erweitern eines Skripts für Lasttests – Übersicht" auf Seite 422] siehe: • "Überprüfen von Text und Bildern (Web-Vuser-Skripts) – Übersicht" auf Seite 901
Anzeigen von Testergebnissen	Siehe die allgemeine VuGen-Dokumentation zum Anzeigen von Testergebnissen ["Anzeigen von Wiedergabeergebnissen – Übersicht" auf Seite 513].

Thema	Beschreibung
Verschiedenes	Die folgenden gemischten Themen beziehen sich auf Web (HTTP/HTML)-Vuser-Skripts. • "Konvertieren eines TruClient-Skripts in ein HTTP/HTML-Skript" auf Seite 616 • "Konvertieren eines Web (HTTP/HTML)-Vuser-Skripts in ein Java-Vuser-Skript" auf Seite 892 • "Web-Snapshots - Übersicht" auf Seite 904 • "Aufzeichnen des SPDY-Protokolls" auf Seite 895 • "Anwendungen, die mit Smooth Streaming arbeiten" auf Seite 895

Verwenden des JavaScript-Moduls von LoadRunner

Hinweis: Dieser Abschnitt bezieht sich nur auf Web (HTTP/HTML)-Vuser-Skripts.

Was ist das JavaScript-Modul von LoadRunner?

Normalerweise enthalten Web (HTTP/HTML)-Vuser-Skripts C-Code. Das JavaScript-Modul von LoadRunner ermöglicht das Einfügen von JavaScript-Codefragmenten in den C-Code.

Was kann ich mit JavaScript in einem Vuser-Skript tun?

Sie können JavaScript-Code in ein Web (HTTP/HTML)-Vuser-Skript einfügen, um Textzeichenfolgen zu manipulieren, die in den Anforderungs- und Antwortnachrichten enthalten sind, die zwischen Client und Server gesendet werden. Die Zeichenfolgenmanipulation ist häufig für Korrelations- und Parametrierungszwecke sinnvoll. Typische Zeichenfolgenmanipulationen umfassen die Konvertierung von Dezimal- in Hexadezimalzahlen, die Base64-Codierung und -Decodierung, die URL-Codierung und -Decodierung sowie den Zugriff auf Objektwerte in JSON-formatierten Daten.

Hinweis: Sie können viele dieser Verfahren zur Zeichenfolgenmanipulation mithilfe der integrierten Datenformaterweiterungen (DFEs) von LoadRunner durchführen. Weitere Informationen finden Sie unter "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912.

Das Einfügen von JavaScript-Code in ein Vuser-Skript kann auch sinnvoll sein, wenn clientseitige Logik in JavaScript implementiert ist. Das Einfügen von Codefragmenten des ursprünglichen clientseitigen JavaScript-Codes in das Vuser-Skript sorgt dafür, dass Sie die JavaScript-Logik nicht mehr erneut in den C-Code schreiben müssen, damit sie in das Vuser-Skript aufgenommen wird.

Sie können JavaScript-Code in einem Vuser-Skript verwenden, um eine **XMLHttpRequest** auszuführen. Auf diese Weise können Sie HTTP- oder HTTPS-Anforderungen mit

standardmäßigen JavaScript-APIs generieren und senden. Solche APIs umfassen z. B. das Senden asynchroner Anforderungen, das Zuweisen von Callbacks zur Bearbeitung von Antworten, das Lesen von Antworten im XML-Format. Eine auf diese Weise eingesetzte **XMLHTTPRequest** kann einen Aufruf eines Aktionsschrittes wie **web_url** oder **web_custom_request** ersetzen.

Warum sollten Sie JavaScript-Codefragmente verwenden?

Auch wenn die erforderliche Funktionalität möglicherweise auch allein durch die Verwendung des C-Codes erreicht werden kann, ist es aus folgenden Gründen unter Umständen vorteilhaft, JavaScript in ein Vuser-Skript einzuschließen:

- JavaScript bietet häufig eine intuitivere, leichter zu implementierende Lösung als C.
- Die JavaScript-Bibliothek für reguläre Ausdrücke vereinfacht das anspruchsvolle Arbeiten mit regulären Ausdrücken.
- Es gibt zahlreiche JavaScript-Bibliotheken, die Sie bei der Zeichenfolgenmanipulation unterstützen.
- Clientseitige Logik wird häufig in JavaScript implementiert. Beim Einfügen von Fragmenten des ursprünglichen JavaScript-Codes fällt die Anforderung weg, die JavaScript-Clientlogik in C-Code zu übersetzen.

Kann ich das JavaScript-Modul in Vuser-Skripts aller Protokolle verwenden?

Nein, das JavaScript-Modul ermöglicht Ihnen nur das Einfügen von JavaScript in Web (HTTP/HTML)-Vuser-Skripts.

In welchen Szenarios kann das JavaScript-Modul beispielsweise hilfreich sein?

Das Einschließen von JavaScript-Code in ein Vuser-Skript kann in folgenden Szenarios hilfreich sein:

Szenario 1: Konvertieren einer Dezimalzahl in die zugehörige Hexadezimaldarstellung

In diesem Szenario muss die Antwort, die ein Vuser an den Server sendet, einen 13-stelligen Zeitstempel im Hexadezimalformat enthalten. Beispielsweise muss der Wert für Datum/Zeitstempel "1234567891234" vom Vuser in einen Hexadezimalwert konvertiert und als "11F71FB0922" gesendet werden. LoadRunner beinhaltet keine Standardfunktionen zum Durchführen dieser Konvertierung, und die Entwicklung des erforderlichen C-Codes ist nicht ganz einfach. Dieses Problem kann behoben werden, indem Sie den folgenden JavaScript-Code in das Vuser-Skript einfügen:

```
web_js_run(
    "Code=getHexTimestamp();",
    "ResultParam=HexTS",
SOURCES,
    "Code=getHexTimestamp=function(){return new Date().getTime().toString(16).toUpperCase();}", ENDITEM,
LAST);

lr_output_message("[%s]",lr_eval_string("{HexTS}"));
```

Szenario 2: Base64-Codierung und -Decodierung

Die Anforderungs- und Antwortnachrichten, die zwischen Client und Server gesendet werden,

enthalten Daten, die mit einem Base64-Codierungsschema codiert sind. Da die Daten codiert sind, ist es häufig schwierig oder unmöglich, die Rohdaten zu parametrieren oder zu korrelieren. Die Daten müssen vor der Parametrierung oder Korrelierung decodiert und anschließend vor der Übermittlung an den Server erneut codiert werden. Durch die Verwendung von JavaScript-Codefragmenten in einem Vuser-Skript können Sie auf externe JavaScript-Bibliotheken zugreifen, die die erforderliche Funktionalität zur Base64-Decodierung und -Codierung implementieren.

Hinweis: Sie können die Base64-Decodierung und -Codierung mit der integrierten Base64-Datenformaterweiterung (DFE) von LoadRunner implementieren. Weitere Informationen finden Sie unter "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912.

Szenario 3: URL-Codierung und -Decodierung

In diesem Szenario enthalten die Anforderungs- und Antwortnachrichten, die zwischen Client und Server gesendet werden, URLs, die mit der JavaScript URL-Codierung codiert wurden. Wenn Sie ein JavaScript-Codefragment in ein Vuser-Skript einfügen, können Sie auf die JavaScript-Funktionen **encodeURI()** und **decodeURI()** zugreifen, um die gewünschten Decodierungs- und Codierungsverfahren durchzuführen.

Hinweis: Sie können die URL-Decodierung und -Codierung mit der integrierten Datenformaterweiterung (DFE) für die URL-Codierung von LoadRunner implementieren. Weitere Informationen finden Sie unter "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912.

Szenario 4: Zugriff auf Objekte in JSON-formatierten Daten

In diesem Szenario enthalten die Anforderungs- und Antwortnachrichten, die zwischen Client und Server gesendet werden, Daten im JSON-Format. Für den Zugriff auf Objekte innerhalb der JSON-formatierten Daten können Sie ein JavaScript-Codefragment in das Vuser-Skript einschließen, um auf die JavaScript-Funktion **eval()** zuzugreifen.

Hinweis: Mit der integrierten JSON-To-XML-DFE (Datenformaterweiterung für JSON-in-XML) von LoadRunner können Sie auf Objekte in JSON-formatierten Daten zugreifen. Weitere Informationen finden Sie unter "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912.

Szenario 5: Verwenden von "XMLHTTPRequest"

Mit JavaScript-Code in einem Vuser-Skript können Sie eine **XMLHttpRequest** für den Download von Aktienkursen von einer bestimmten Website ausführen. Ein Beispiel für die Ausführung einer **XMLHTTPRequest** finden Sie unter "JavaScript-Modul: Beispiel für XMLHTTPRequest" auf Seite 891.

Szenario 6: Bereits vorhandener clientseitiger JavaScript-Code

Für den Zugriff auf eine bestimmte Website muss der Vuser einen Benutzernamen und ein verschlüsseltes Kennwort übermitteln. Der Server sendet ein Server-Hash an den Browser, damit der Browser das erforderliche verschlüsselte Kennwort generieren kann. Der Code für die Hash-Generierung ist kompliziert und liegt in JavaScript vor. Dieser JavaScript-Code kann in das Vuser-Skript eingefügt werden. Dadurch müssen Sie die JavaScript-Logik nicht erneut in den C-Code schreiben.

Welche LoadRunner-API-Funktionen kann ich in einem Vuser-Skript verwenden, um JavaScript-Code auszuführen?

Folgende LoadRunner-API-Funktionen stehen zum Einfügen von JavaScript in ein Vuser-Skript zur Verfügung:

1. **web_js_run:** Führt den angegebenen JavaScript-Code aus.
2. **web_js_reset:** Löscht den JavaScript-Kontext.

Sie verwenden die Funktion **web_js_run**, um JavaScript-Code in ein Vuser-Skript einzufügen. Mit der Funktion **web_js_run** können Sie entweder den gewünschten JavaScript-Code in das Vuser-Skript direkt einfügen oder Sie können auf eine Datei verweisen, die den gewünschten JavaScript-Code enthält.

Beispiel für eingefügten JavaScript-Code

Im Folgenden finden Sie ein Beispiel, wie JavaScript-Code direkt aus dem Vuser-Skript eingefügt wird:

```
web_js_run(
       "Code=xor((LR.getParam('buffer'), 0xFFFF));",
       "ResultParam=param",
LAST);
```

Beispiel für eine referenzierte Datei, die JavaScript-Code enthält

Im Folgenden finden Sie ein Beispiel, wie Sie JavaScript-Code einfügen können, indem Sie eine Datei aufrufen, die den JavaScript-Code enthält:

```
web_js_run(
       "File=XMLHTTPRequest_sync_sample.js;",
       "ResultParam=param",
LAST);
```

Weitere Informationen zu den oben genannten Funktionen und Beispiele für ihre Verwendung finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Kann ich JavaScript für den Zugriff auf sämtliche "internen" LoadRunner-API-Funktionen verwenden?

Mit JavaScript in einem Vuser-Skript erhalten Sie Zugriff auf eine Reihe "interner" LoadRunner-API-Funktionen, die direkt aus einer Funktion **web_js_run** im JavaScript-Code aufgerufen werden können. Diese Funktionen werden hauptsächlich für die Verwaltung von Parametern verwendet, ermöglichen Ihnen jedoch auch, bestimmte Meldungen zu protokollieren, Daten aufzuzeichnen und eine **XMLHTTPRequest** auszuführen.

LoadRunner-API-Funktionen

API-Funktion	Beschreibung	Argumente
LR.advanceParam (Parameter)	Legt für den angegebenen Parameter den nächsten Wert in der Datei fest.	**Parameter.** Der Name des weiterzusetzenden Parameters. Dies muss ein Parameter des Typs Datei oder eindeutige Zahl sein.
LR.setParam (Name, Wert)	Speichert eine Zeichenfolge in einem Parameter und erstellt den Parameter, wenn dieser nicht existiert.	**Name.** Der Name des Parameters, in dem der Wert gespeichert werden soll. **Wert.** Der Wert.
LR.freeParam (name)	Löscht einen dynamischen Parameter während der Laufzeit und gibt dadurch den zugehörigen Puffer frei.	**Name.** Der Parametername.
LR.getParam (Name)	Gibt den Wert des angegebenen Parameters zurück.	**Name.** Der Parametername.
LR.log(Text, Ebene)	Protokolliert eine Meldung.	**Text.** Die Meldung. **Ebene.** Eine der Folgenden: • "Fehler" • "Warnung" • "Standard" • "Erweitert" • "Status" **Beispiel:** LR.log("Text", "Fehler");
LR.userDataPoint (Name, Wert)	Zeichnet einen benutzerdefinierten Datenpunkt für die Analyse auf.	**Name.** Der Name des Datenpunkts. Ein Datenpunktname darf nicht mit einer dieser Zeichenfolgen beginnen: HTTP, NON_ HTTP, RETRY, mic_, stream_, mms_ **Wert.** Der numerische Wert.

Wie kann ich das JavaScript-Modul von LoadRunner aktivieren?

Damit Sie JavaScript aus einem Vuser-Skript ausführen können, muss das JavaScript-Modul für das Vuser-Skript aktiviert sein. Mithilfe der Laufzeiteinstellungen aktivieren Sie das JavaScript-Modul.

Klicken Sie auf **Wiedergeben > Laufzeiteinstellungen > Internetprotokoll > Einstellungen > Erweiterte Optionen festlegen - Optionen > Web JavaScript > Ausführen von JavaScript-Code ermöglichen** und wählen Sie **Ja** aus.

Hinweis: Das Aktivieren dieser Option bewirkt das Erstellen eines JavaScript-Laufzeitmoduls, auch wenn das Skript keine JavaScript-Schritte enthält.

Wie kann ich das JavaScript-Modul von LoadRunner konfigurieren?

Verwenden Sie zum Konfigurieren des JavaScript-Moduls von LoadRunner die Laufzeiteinstellungen des Vuser-Skripts.

Für den Zugriff auf die JSE-Laufzeiteinstellungen wählen Sie **Wiedergeben > Laufzeiteinstellungen > Internetprotokoll > Einstellungen > Erweiterte Optionen festlegen - Optionen** aus, und führen Sie einen Bildlauf nach unten zur Überschrift **Web JavaScript** durch.

- **Größe des JavaScript-Laufzeitmoduls:** Gibt die Größe des zugeordneten JavaScript-Laufzeitmodulspeichers in Kilobyte an. Dieser Wert muss möglicherweise erhöht werden, wenn eine große Anzahl an Vusern ausgeführt wird.
- **Stapelgröße pro Thread für JavaScript-Modul:** Gibt die Größe jedes Vuser-Threads im JavaScript-Laufzeitmodulspeicher in Kilobyte an. Dieser Wert muss möglicherweise für große Objekte oder umfangreiche Aufruflisten erhöht werden.

Weitere Informationen über die Laufzeiteinstellungen des JavaScript-Moduls finden Sie unter "Dialogfeld "Erweiterte Einstellungen"" auf Seite 373.

Welche Verbindung besteht zwischen dem JavaScript-Modul und dem JavaScript-Protokoll von LoadRunner?

Es gibt keine Verbindung zwischen dem LoadRunner-JavaScript-Modul und dem LoadRunner-JavaScript-Protokoll.

Fehlerbehebung

Wenn bei der Implementierung der Unterstützung für das JavaScript-Modul Probleme auftreten, finden Sie möglicherweise in den nachstehenden Punkten eine Lösung.

1. Stellen Sie sicher, dass das JavaScript-Modul von LoadRunner aktiviert ist. Weitere Informationen finden Sie unter Wie kann ich das JavaScript-Modul von LoadRunner aktivieren?.

2. Speicherprobleme

   Wenn die Laufzeiteinstellung **Bei jeder Iteration neuen Benutzer simulieren > Cache bei jeder Iteration leeren** aktiviert ist, wird **web_js_reset** beim Start jeder Iteration automatisch aufgerufen. Wenn **Bei jeder Iteration neuen Benutzer simulieren > Cache bei jeder Iteration leeren** nicht aktiviert ist, vermeiden Sie eine übermäßige Speicherbelegung, indem Sie an den Stellen, wo der gespeicherte Kontext nicht länger benötigt wird, **web_js_reset**-Aufrufe in Ihr Vuser-Skript einfügen. Weitere Informationen zur Funktion **web_js_reset** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**). Weitere Informationen zu den Laufzeiteinstellungen finden Sie unter "Browser > Browseremulation (Knoten)" auf Seite 338.

3. Leistungsprobleme

Wenn Leistungsprobleme auftreten, können Sie die Laufzeiteinstellungen für Web JavaScript ändern. Weitere Informationen finden Sie unter "Dialogfeld "Erweiterte Einstellungen"" auf Seite 373.

JavaScript-Modul: Beispiel für XMLHTTPRequest

Hinweis: Dieser Abschnitt bezieht sich nur auf Web (HTTP/HTML)-Vuser-Skripts.

Mit dem LoadRunner-JavaScript-Modul können Sie JavaScript-Code in ein Vuser-Skript einschließen. Weitere Informationen zum JavaScript-Modul finden Sie unter "Verwenden des JavaScript-Moduls von LoadRunner" auf Seite 885.

Das nachstehende Beispiel zeigt, wie Sie ein JavaScript-Objekt **XMLHTTPRequest** in einem Web (HTTP/HTML)-Vuser-Skript verwenden können. In diesem Beispiel kann der Vuser mit dem Objekt **XMLHTTPRequest** einen Aktienkurs von "finance.example.com" herunterladen und den Wert anschließend für eine spätere Verwendung in einem Parameter speichern.

Der nachstehende Skriptabschnitt zeigt eine Funktion **web_js_run**, die in ein Vuser-Skript eingefügt wurde. Die Funktion **web_js_run** enthält einen Verweis auf eine Datei mit der Bezeichnung **XMLHTTPRequest_sync_sample.js**. Diese Datei enthält den JavaScript-Code zum Ausführen der Funktion **XMLHTTPRequest**.

```
web_js_run(
  "Code=getQuotes(LR.getParam('symbol'));",
  "ResultParam=param",
SOURCES,
"File=XMLHTTPRequest_sync_sample.js", ENDITEM,
LAST);
```

Der Inhalt der Datei **XMLHTTPRequest_sync_sample.js** ist im Folgenden dargestellt.

```
var req2;

function getQuotes(mySymbol)
{
var myURL="http://download.finance.example.com/d/quotes.csv?s="+mySymbol+"&f=l1&e=.csv";

   req2 = false;
   // branch for native XMLHttpRequest object
   try {
        req2 = new XMLHttpRequest();
     } catch(e) {
        req2 = false;
     }
   if(req2) {
     req2.open("GET", myURL, false);
     req2.send("");

   }
   return 1*req2.responseText;
}
```

- Weitere Beispiele für Code, der mit dem JavaScript-Modul verwendet wird, finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).
- Weitere Informationen zum Objekt **XMLHTTPRequest** finden Sie unter http://www.w3schools.com/ajax/ajax_xmlhttprequest_send.asp.

Konvertieren eines Web (HTTP/HTML)-Vuser-Skripts in ein Java-Vuser-Skript

Hinweis: Dieser Abschnitt bezieht sich nur auf Web (HTTP/HTML)- und Java-Vuser-Skripts.

VuGen enthält ein Dienstprogramm, mit dem Sie ein Web (HTTP/HTML)-Vuser-Skript in ein Java-Vuser-Skript konvertieren können. Auf diese Weise können Sie ein hybrides Vuser-Skript für Web und Java erstellen.

Konvertieren eines Web (HTTP/HTML)-Vuser-Skripts in ein Java-Vuser-Skript

1. Erstellen Sie ein leeres **Java-Vuser**-Skript, und speichern Sie es.
2. Erstellen Sie ein leeres **Web (HTML/HTTP)**-Vuser-Skript, und speichern Sie es.
3. Zeichnen Sie eine Sitzung im Web (HTML/HTTP)-Vuser-Skript auf.
4. Geben Sie das Web (HTML/HTTP)-Vuser-Skript wieder. Wenn das Skript fehlerfrei wiedergegeben wird, schneiden Sie das gesamte Skript aus, fügen Sie es in einen Text-Editor ein, und speichern Sie es als Textdatei (TXT).
   Ändern Sie in der Textdatei alle Parameterklammern vom Typ Web (`"{ }"`) in den Typ Java

("< >").

5. Öffnen Sie ein DOS-Befehlsfenster, und navigieren Sie zum Ordner <LoadRunner Installation> / **dat**.

6. Geben Sie den folgenden Befehl ein:

```
<LoadRunner_Installation Folder>\bin\sed -f web_to_java.sed filename > ou
tputfilename
```

wobei **filename** dem vollständigen Pfad und Dateinamen der zuvor gespeicherten Textdatei und **outputfilename** dem vollständigen Pfad und Dateinamen der Ausgabedatei entspricht.

7. Öffnen Sie die Ausgabedatei und kopieren Sie den Inhalt an die gewünschte Stelle im Aktionsabschnitt des Java-Vuser-Skripts.
Wenn Sie die Inhalte in eine leere, angepasste Java-Vorlage (Java-Vuser-Type) einfügen, müssen Sie die Zeile mit `public int action()` wie folgt ändern:

```
public int action() throws Throwable
```

Für aufgezeichnete Java-Benutzer (RMI und CORBA) erfolgt diese Änderung automatisch.

8. Parametrieren und korrelieren Sie das Vuser-Skript, wie Sie es in einem herkömmlichen Java-Skript tun würden, und führen Sie das Skript aus.

Erstellen eines Skripts für eine REST-API

In den letzten Jahren hat sich Representational State Transfer (REST) als häufig verwendetes Modell für die Softwarearchitektur etabliert, insbesondere für webbasierte Anwendungen. Mittlerweile verwenden die meisten großen Websites, etwa Twitter, Google, Flickr usw., REST-APIs. Mithilfe der LoadRunner-Funktion **web_custom_request** können Sie ein Lasttestskript für eine REST-API erstellen.

Ein REST-API-Aufruf besteht aus folgenden Komponenten:

- **Uniform Resource Identifier (URI).** Eine Zeichenfolge aus Host, Pfad der Funktionskomponente und Abfragezeichenfolge mit Schlüssel-Wert-Paaren. Beispiel: `http://www.shopping.hp.com/en_US/home-office/-/products/Tablets/Tablets?SearchParameter=ElitePad`:

- **Aktion.** Die auszuführende Aktion, z. B. GET, POST, PUT, DELETE und PATCH.

- **Daten.** Die an den Server zu sendenden Daten, in der Regel im JSON-Format.

Alle diese Komponenten sind als Parameter einer Funktion vom Typ **web_cutom_request** enthalten.

In dieser Aufgabe wird beschrieben, wie Sie die Skriptschritte für ein Web (HTTP/HTML)-Skript erstellen, das eine REST-API aufruft.

1. Erstellen Sie ein Web (HTTP/HTML)-Vuser-Skript. Weitere Informationen finden Sie unter "Erstellen von über Vuser-Skripts – Übersicht" auf Seite 148.

2. Zeichnen Sie die REST-API-Anwendung auf. Führen Sie typische Geschäftsprozesse in Ihrer Anwendung aus.

3. Fügen Sie nach dem Beenden der Aufzeichnung neue Aufrufe vom Typ **web_custom_request** in folgendem Format hinzu:

```
web_custom_request("<Step_Name>",
 "URL=<URI of the REST API>",
 "Method=<Action>",
 "Resource=0",
 "EncType=application/json",
 "Mode=HTTP",
 "Body=<Body text in JSON format>,
 LAST);
```

4. Führen Sie folgende Schritte aus, wenn Sie den eigentlichen Text nicht eingeben und stattdessen auf eine Datei mit den JSON-Daten verweisen möchten:
   a. Klicken Sie im Lösungs-Explorer mit der rechten Maustaste auf den Knoten **Zusätzliche Dateien**, und wählen Sie **Dateien zum Skript hinzufügen** aus, um die JSON-Datendatei hinzuzufügen.

   b. Ersetzen Sie das Argument **Body** durch **BodyFilePath=<file_name.json>**.

   c. Lassen Sie die Verwendung von JSON-Dateien zu. Wählen Sie **Extras > Optionen > Skripterstellung > Skriptverwaltung** aus, und fügen Sie ".json" der Liste der zulässigen Erweiterungen hinzu.

Beispiele

Das folgende Beispiel zeigt eine REST-API-Funktion, die Werte mithilfe einer PUT-Aktion aktualisiert:

```
web_custom_request("Update My App",
 "URL=http://localhost:3000/MyRESTAPI/3",
 "Method=PUT",
 "Resource=0",
 "EncType=application/json",
 "Mode=HTTP",
 "Body={\"id\":3,
  \"isFinished\":false,\"MyText\":\"Updated name\"}",
 LAST);
```

Das nächste Beispiel zeigt eine REST-API-Funktion, die Werte mithilfe einer POST-Aktion unter Verwendung von Daten aus einer Datei aktualisiert:

```
web_custom_request("Update My App",
 "URL=http://localhost:3000/MyRESTAPI/3",
 "Method=POST",
 "Resource=0",
 "EncType=application/json",
 "Mode=HTTP",
 "BodyFilePath=data.json",
 LAST);
```

Aufzeichnen des SPDY-Protokolls

Bei SPDY handelt es sich um ein offenes Netzwerkprotokoll für den Transport von Webinhalten. Eines der Hauptziel von SPDY ist es, die Ladezeiten von Webseiten zu reduzieren. Der Transport erfolgt über SSL (HTTPS). SPDY soll HTTP nicht ersetzen, es wird lediglich die Art und Weise geändert, wie HTTP-Anforderungen und -Antworten gesendet werden.

In dieser Aufgabe wird beschrieben, wie Sie ein Web (HTTP/HTML)-Skript aufzeichnen, das das SPDY-Protokoll verwendet.

1. Erstellen Sie ein Web (HTTP/HTML)-Vuser-Skript. Weitere Informationen finden Sie unter "Erstellen von über Vuser-Skripts – Übersicht" auf Seite 148.

2. Wählen Sie **Aufzeichnungsoptionen > Netzwerk > Portzuordnung > Optionen > TLS NPN**, um die Aufzeichnung des SPDY-Protokolls zu ermöglichen.

3. Klicken Sie auf die Schaltfläche **Aufzeichnen**. Wählen Sie im Dialogfeld "Aufzeichnung starten" einen Browser aus. **Hinweis:** Wenn Sie in Chrome aufzeichnen, stellen Sie sicher, dass alle anderen Instanzen von Chrome geschlossen sind, bevor Sie mit der Aufzeichnung beginnen.

4. Starten Sie die Aufzeichnung und führen Sie die typischen Geschäftsprozesse in Ihrer Anwendung aus.

Anwendungen, die mit Smooth Streaming arbeiten

In dieser Aufgabe wird erläutert, wie Sie Anwendungen aufzeichnen, die Smooth Streaming verwenden.

Was ist Smooth Streaming?

Smooth Streaming ist eine Internet Information Services (IIS)Media Services-Erweiterung, die hochqualitatives Videostreaming über HTTP ermöglicht. Smooth Streaming überwacht die lokale Bandbreite und Videowiedergabeleistung des Client und passt die Streamrate und -qualität

entsprechend an. Bei herkömmlichen Streamingverfahren werden die Inhalte mit fest vorgegebener Streamingrate und -qualität übertragen.

Vorbereiten eines Skripts für Lasttests für Anwendungen, die Smooth Streaming verwenden:

1. Erstellen Sie ein Web (HTTP/HTML)-Vuser-Skript. Weitere Informationen finden Sie unter "Erstellen von über Vuser-Skripts – Übersicht" auf Seite 148.

2. Suchen Sie nach der "Manifest"-Anforderung am Anfang der Streamingverbindung:

```
web_custom_request("Manifest",
        "URL=http://mediadl.microsoft.com/mediadl/iisnet/smoothmedia/E
xperience/BigBuckBunny_720p.ism/Manifest",
        "Method=GET",
        "Resource=0",
        "RecContentType=text/xml",
        "Referer=",
        "Snapshot=t11.inf",
        "Mode=HTTP",
        LAST);
```

3. Auf die "Manifest" -Anforderung sollten eine Reihe von Streaminganforderungen folgen:

```
    web_custom_request("Fragments(video=0)",
        "URL=http://mediadl.microsoft.com/mediadl/iisnet/smoothmedia/E
xperience/BigBuckBunny_720p.ism/QualityLevels(350000)/Fragments(video=
0)",
        "Method=GET",
        "Resource=1",
        "RecContentType=video/mp4",
        "Referer=",
        "Snapshot=t12.inf",
        LAST);

    web_custom_request("Fragments(audio=0)",
        "URL=http://mediadl.microsoft.com/mediadl/iisnet/smoothmedia/E
xperience/BigBuckBunny_720p.ism/QualityLevels(64000)/Fragments(audio=0
)",
        "Method=GET",
        "Resource=1",
        "RecContentType=video/mp4",
        "Referer=",
        "Snapshot=t13.inf",
        LAST);
```

   Erstellen und konfigurieren Sie Parameter zur Emulation anderer als der aufgezeichneten Bandbreiten.

   Beispiel:

a. Ersetzen Sie in der Streaminganfrage "QualityLevel" durch den Parameter 'qualityLevel'.

```
web_custom_request("Fragments(video=0)",
        "URL=http://mediadl.microsoft.com/mediadl/iisnet/smoothmedia/E
xperience/BigBuckBunny_720p.ism/QualityLevels({qualityLevel})/Fragment
s(video=0)",
        "Method=GET",
        "Resource=1",
        "RecContentType=video/mp4",
        "Referer=",
        "Snapshot=t12.inf",
        LAST);
```

b. Konfigurieren Sie die 'qualitylevel'-Werte, die bei jeder Iteration des Lasttests verwendet werden, im "Dialogfeld "Parametereigenschaften"" auf Seite 456.

4. Geben Sie das Skript wieder und vergewissern Sie sich, dass die Antwortgröße für jede Anfrage dem Parameterwert entspricht, der übermittelt wurde.

Konvertieren eines TruClient-Skript in ein Web HTTP/HTML-Skript

Profitieren Sie gleichzeitig von einer Skriptentwicklung und der Ausführung einer Vielzahl von Vusern, indem Sie ein TruClient-Skript in ein HTTP/HTML umwandeln.

Hinweis: Sie müssen über Kenntnisse des Web-HTTP/HTML-Protokolls verfügen, um das Skript erfolgreich wiedergeben zu können.

1. Erstellen Sie ein TruClient-Skript. Weitere Informationen finden Sie unter "Entwickeln von TruClient-Skripts" auf Seite 549.

2. Speichern Sie das Skript und schließen Sie die TruClient-Seitenleiste.

3. Klicken Sie auf der VuGen-Symbolleiste auf die Schaltfläche **Konvertieren**, um das Dialogfeld zum Konvertieren eines TruClient-Skripts in ein Web HTTP/HTML-Skript zu öffnen.

4. Geben Sie einen Skriptnamen und einen Speicherort ein, oder akzeptieren Sie die Standardwerte. Klicken Sie auf **Convert** (Konvertieren).

5. Nachdem das Skript generiert wurde, überprüfen Sie es. Denken Sie dabei daran, dass das Web-HTTP/HTML-Protokoll auf der Transportebene aufgezeichnet wird. So könnte es beispielsweise sein, dass Sie Probleme mit der Korrelation oder mit Parametern in dem konvertierten Skript beheben müssen. Weitere Informationen finden Sie unter "Webprotokoll (HTTP/HTML) – Übersicht" auf Seite 881.

Einschränkungen

Bei der Konvertierung von TruClient-Skripts in Web-HTTP/HTML-Skripts werden die Konvertierungsschritte, die die Adresse 127.0.0.1 (localhost) aufrufen, nicht unterstützt.

Fehlerbehebung und Einschränkungen – Webprotokoll (HTTP/HTML)

In diesem Abschnitt werden die Fehlerbehebung und die für das Webprotokoll (HTTP/HTML) geltenden Einschränkungen beschrieben.

- Die Konfiguration von Port-Zuordnungen wird im Modus **Proxy-Aufzeichnung** nicht unterstützt.
- VuGen kann während des Aufzeichnens einer Sitzung kein Clientzertifikat von Internet Explorer 10 abrufen.
  Lösung: Geben Sie ein Clientzertifikat in den Einstellungen für die Port-Zuordnung an.
- Wenn verstärkte Sicherheit für den privaten Schlüssel für ein Zertifikat festgelegt ist und die Wiedergabe im WinInet-Modus erfolgt, müssen Sie bei Wiedergabe des Skripts Authentifizierungsdetails möglicherweise manuell eingeben.
- In früheren Versionen von LoadRunner wurde der C-Datentyp "char" als Datentyp "char" mit Vorzeichen betrachtet. Ab LoadRunner 11.50 wird er als "char" ohne Vorzeichen betrachtet. Wenn Sie "char" verwendet haben, ohne anzugeben, ob es ein Vorzeichen aufweist oder nicht, und arithmetische Operationen mit dieser Variablen durchgeführt haben, dann können die aktuellen Ergebnisse von denen aus früheren LoadRunner-Versionen abweichen.

Webprotokolle (Allgemein)

Der Abschnitt **Webprotokolle** enthält Informationen, die allgemein für alle Web-Vuser-Protokolle gelten. Informationen zu einem bestimmten Web-Vuser-Protokoll finden Sie in der Dokumentation zu dem jeweiligen Protokoll.

Webprotokolle – Übersicht

Hinweis: Dieses Thema bezieht sich nur auf Web-Vuser-Protokolle. Eine Liste der Web-Vuser-Protokolle finden Sie unter "Web-Vuser-Typen" auf Seite 900.

Sie können VuGen zum Entwickeln von Web-Vuser-Skripts verwenden. Während Sie in einer Website typische Benutzeraktivitäten ausführen, zeichnet VuGen Ihre Aktionen auf und generiert ein Vuser-Skript. Wenn Sie das Skript ausführen, emuliert der daraus resultierende Vuser einen Benutzer, der auf das Internet zugreift.

Angenommen, Sie verfügen über eine Website, auf der Informationen über die Produkte Ihres Unternehmens erscheinen. Die Website wird von Kunden und potenziellen Kunden besucht. Sie möchten sicherstellen, dass die Antwortzeit für Kundenanfragen unter einem vorgegebenen Wert liegt (beispielsweise 20 Sekunden), auch wenn eine große Anzahl von Benutzern (z. B. 200) gleichzeitig auf die Website zugreift. Verwenden Sie Vuser zum Emulieren dieser Situation, bei der

alle Webserverdienste gleichzeitig verschiedene Informationen anfragen. Jeder Vuser kann Folgendes tun:

- Laden der Startseite
- Navigieren zu der Seite mit den Produktinformationen
- Übermitteln einer Abfrage
- Warten auf eine Antwort vom Server

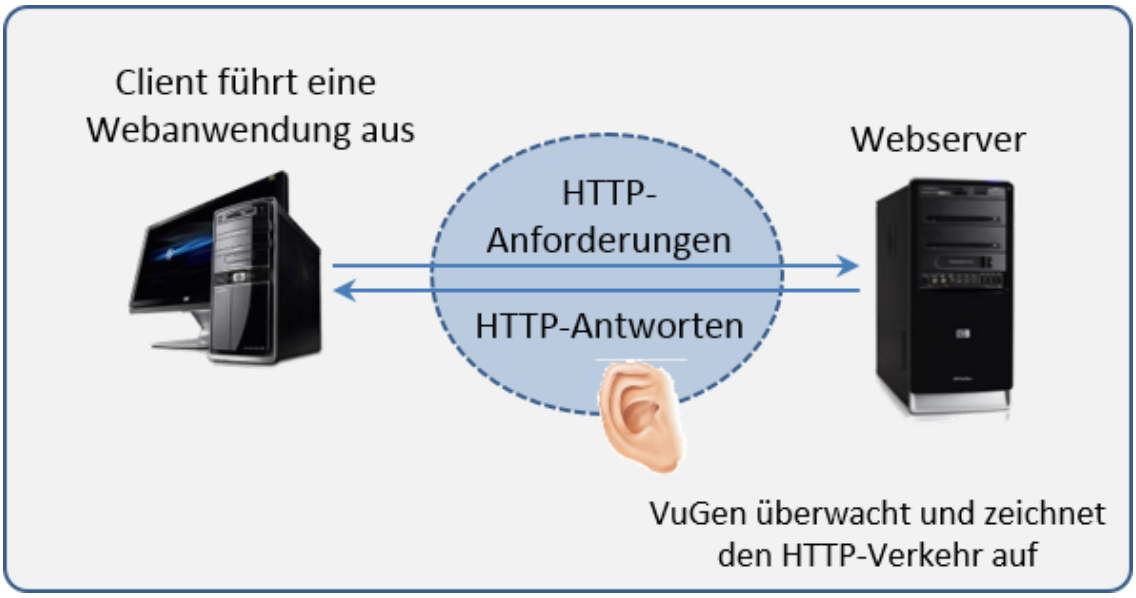
Sie können mehrere hundert Vuser auf die verfügbaren Testcomputer verteilen, wobei jeder Vuser über eine API für den Zugriff auf den Server verfügt. Auf diese Weise können Sie die Leistung des Servers unter der Last einer großen Anzahl von Benutzern messen.

Weitere Informationen zu Web-Vuser-Skripts finden Sie unter "Web Vuser-Technologie" unten.

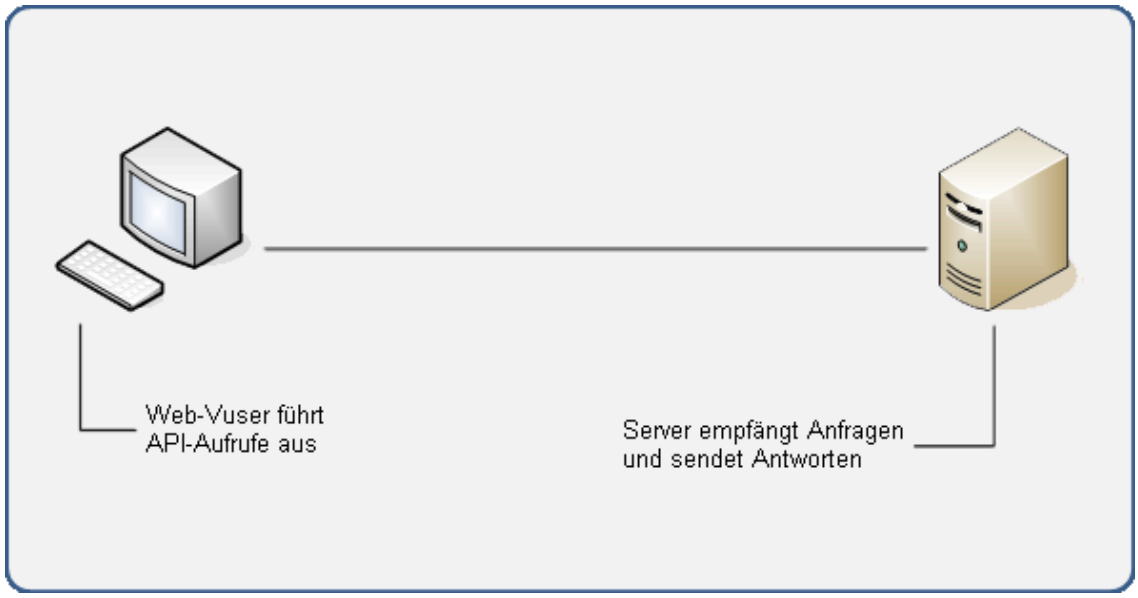
Web Vuser-Technologie

Hinweis: Dieses Thema bezieht sich nur auf Web-Vuser-Protokolle. Eine Liste der Web-Vuser-Protokolle finden Sie unter "Web-Vuser-Typen" auf der nächsten Seite.

VuGen erstellt Web-Vuser-Skripts durch Überwachen und Aufzeichnen des Webdatenverkehrs zwischen einem Webbrowser und einem Webserver, während Sie typische Geschäftsprozesse ausführen. Der Webdatenverkehr enthält HTTP-Anforderungen, die vom Browser an den Server gesendet werden, sowie die vom Server zurückgegebenen HTTP-Antworten.



Wenn Sie ein Web-Vuser-Skript ausführen, kommuniziert der sich daraus ergebende Vuser direkt mit dem Webserver, ohne einen Browser oder Clientsoftware hinzuzuziehen. Damit die Kommunikation ausgeführt werden kann, sendet das Vuser-Skript Web-API-Funktionen direkt an den Webserver.



Web-Vuser-Typen

Mithilfe von LoadRunner können Sie webbasierte Vuser-Skripts unter Verwendung verschiedenster Web-Vuser-Protokolle erstellen und ausführen. In der folgenden Liste sind die Web-Vuser-Protokolle von LoadRunner aufgeführt:

- Ajax (Click & Script)
- Flex
- Java over HTTP
- Oracle Web Applications 11i
- SAP (Click & Script)
- SAP – Web
- Siebel – Web
- Silverlight
- TruClient - Ajax
- Web (HTTP/HTML)

Überprüfen von Text und Bildern (Web-Vuser-Skripts) – Übersicht

Hinweis: Dieses Thema bezieht sich nur auf Web-Vuser-Protokolle. Eine Liste der Web-Vuser-Protokolle finden Sie unter "Web-Vuser-Typen" auf der vorherigen Seite.

VuGen ermöglicht Ihnen, Prüfungen zu Ihren Web-Vuser-Skripts hinzuzufügen. Bei einer Prüfung wird festgestellt, ob ein bestimmtes Objekt auf einer Webseite vorhanden ist. Bei dem Objekt kann es sich um eine Textzeichenfolge oder ein Bild handeln.

Durch Prüfungen können Sie feststellen, ob Ihre Website beim Zugriff durch zahlreiche Vuser ordnungsgemäß ausgeführt wird, d. h. ob der Server die richtigen Webseiten zurückgibt. Dies ist besonders in den Zeiten von Bedeutung, in denen die Website von zahlreichen Benutzern frequentiert wird, weil die Wahrscheinlichkeit, dass der Server falsche Seiten zurückgibt, steigt.

Angenommen, Ihre Website zeigt Informationen zur Temperatur in großen Städten weltweit an. Unter Verwendung von VuGen erstellen Sie ein Vuser-Skript, dass auf Ihre Website zugreift.

Der Vuser greift auf die Site zu und führt eine Textprüfung auf dieser Webseite aus. Wenn beispielsweise das Word **Temperature** auf der Seite angezeigt wird, wurde die Prüfung bestanden. Wenn das Wort **Temperature** nicht angezeigt wird, weil der Server beispielsweise nicht die richtige Seite zurückgegeben hat, wurde die Prüfung nicht bestanden. Beachten Sie, dass der Textprüfungsschritt vor dem URL-Schritt angezeigt wird. Dies ist darauf zurückzuführen, dass VuGen die für den nächsten Schritt relevanten Suchinformationen registriert oder vorab vorbereitet. Wenn Sie das Vuser-Skript ausführen, führt VuGen die Prüfung auf der folgenden Webseite durch.

Zwar zeigt der Server möglicherweise die richtige Seite an, wenn Sie das Skript aufzeichnen und wenn ein einzelner Vuser das Skript ausführt, aber es ist möglich, dass nicht die richtige Seite zurückgegeben wird, wenn der Server sich unter der Last zahlreicher Vuser befindet. Der Server kann überlastet sein und daher bedeutungslosen oder falschen HTML-Code zurückgeben. Ist ein Server überlastet, kann er in einigen Fällen auch die Seite **500 Server Error** zurückgeben. In beiden Fällen können Sie eine Prüfung einfügen, um festzustellen, ob die richtige Seite vom Server zurückgegeben wird.

Hinweis: Prüfungen erhöhen den Datenverarbeitungsaufwand für einen Vuser. Daher empfiehlt es sich möglicherweise, weniger Vuser pro Lastgenerator auszuführen. Sie sollten Prüfungen nur verwenden, wenn ein Server erfahrungsgemäß mitunter eine falsche Seite zurückgibt.

- Weitere Informationen finden Sie unter "Grundlegendes zu den Funktionen für die Webtextprüfung" auf der nächsten Seite.

- Ausführliche Informationen zum Hinzufügen einer Text- oder Bildprüfung finden Sie unter "Hinzufügen von Text- und Bildprüfungen (Web-Vuser-Protokolle)" auf der nächsten Seite.

Grundlegendes zu den Funktionen für die Webtextprüfung

Hinweis: Dieses Thema bezieht sich nur auf Web-Vuser-Protokolle. Eine Liste der Web-Vuser-Protokolle finden Sie unter "Web-Vuser-Typen" auf Seite 900.

Wenn Sie einem Web-Vuser-Skript eine Textprüfung hinzufügen, fügt VuGen dem Skript die Funktion **web_reg_find** hinzu. Diese Funktion registriert eine Suche nach einer Textzeichenfolge auf einer HTML-Seite. Registrierung bedeutet, dass der Vuser die Suche nicht sofort ausführt, sondern nur nach Ausführung der nächsten Aktionsfunktion, z. B. **web_url**.

Hinweis: Wenn Sie mit einer Gruppe gleichzeitiger Funktionen arbeiten, wird die Funktion **web_reg_find** nur am Ende der Gruppe ausgeführt.

Im folgenden Beispiel sucht die Funktion **web_reg_find** nach der Textzeichenfolge "Welcome". Wird die Zeichenfolge nicht gefunden, schlägt die nächste Aktionsfunktion fehl und die Skriptausführung wird angehalten.

```
web_reg_find("Text=Welcome", "Fail=Found", LAST);
web_url("Step", "URL=...", LAST);
```

Neben der Funktion **web_reg_find** können Sie andere Funktionen zum Suchen nach Text innerhalb einer HTML-Seite verwenden:

Einige zusätzliche Funktionen können für die Suche nach Text verwendet werden:

- web_find
- web_global_verification

Die Funktion **web_find**, die hauptsächlich aus Gründen der Abwärtskompatibilität verwendet wird, unterscheidet sich von der Funktion **web_reg_find** dahingehend, dass **web_find** auf HTML-basierte Skripts begrenzt ist (siehe Registerkarte **Aufzeichnungsoptionen > Aufzeichnen**). Sie besitzt auch weniger Attribute, z. B. das Attribut instance, mit dem Sie feststellen können, wie oft der Text angezeigt wird. Bei Durchführung einer standardmäßigen Textsuche ist **web_reg_find** die bevorzugte Funktion.

Die Funktion **web_global_verification** ermöglicht die Suche nach Daten eines gesamten Geschäftsprozesses. Im Gegensatz zu der Funktion **web_reg_find**, die nur auf die nächste Aktionsfunktion angewendet wird, wird diese Funktion auf **alle** nachfolgenden Aktionsfunktionen, wie z. B. **web_url**, angewendet. Standardmäßig ist der Umfang der Suche NORESOURCE, wobei nur der HTML-Text ohne Kopfzeilen und Ressourcen durchsucht wird.

Die Funktion **web_global_verification** eignet sich optimal zum Erkennen von Fehlern auf Anwendungsebene, die nicht in den HTTP-Statuscodes enthalten sind. Diese Funktion ist nicht auf ein HTML-basiertes Skript begrenzt (siehe Registerkarte **Recording Options > Recording**).

Hinzufügen von Text- und Bildprüfungen (Web-Vuser-Protokolle)

Hinweis: Dieses Thema bezieht sich nur auf Web-Vuser-Protokolle. Eine Liste der Web-Vuser-Protokolle finden Sie unter "Web-Vuser-Typen" auf Seite 900.

Es gibt eine Reihe unterschiedlicher Prüfungsarten, die Sie Ihren Web-Vuser-Skripts hinzufügen können. Hintergrundinformationen finden Sie unter "Überprüfen von Text und Bildern (Web-Vuser-Skripts) – Übersicht" auf Seite 901.

Hinzufügen einer Textprüfung während der Aufzeichnung

1. Markieren Sie im Webbrowser den gewünschten Text.
2. Klicken Sie in der VuGen-Aufzeichnungssymbolleiste auf die Schaltfläche **Textprüfung einfügen** . VuGen fügt die Funktion **web_reg_find** zum Skript hinzu.

Hinzufügen einer Textprüfung nach der Aufzeichnung

1. Zeigen Sie im Snapshot-Bereich einen Snapshot an, der den zu verifizierenden Text enthält.
2. Klicken Sie in der Symbolleiste des Snapshot-Bereichs auf **HTTP-Daten**, um die HTTP-Datenansicht des Snapshots anzuzeigen.
3. Wählen Sie im Snapshot den zu überprüfenden Text aus. Der Text muss sich in einem Antwortabschnitt des Snapshots befinden und nicht in einem Anforderungsabschnitt.
4. Klicken Sie mit der rechten Maustaste und wählen Sie im Menü **Schritt zur Textüberprüfung hinzufügen** aus.
5. Ändern Sie die Optionen im Dialogfeld **Text suchen**. Details zu den Optionen des Dialogfelds erhalten Sie, indem Sie auf F1 drücken, um die Function Reference zu öffnen.
6. Klicken Sie auf **OK**, um die Funktion in das Vuser-Skript einzufügen.

Hinzufügen anderer Textprüfungen nach der Aufzeichnung

1. Navigieren Sie im Skript-Editor zu der Position, an der Sie die Prüfung einfügen möchten.
2. Wählen Sie **Ansicht > Toolbox** aus, um die Toolbox zu öffnen.
   a. Um eine **web_reg_find**-Funktion einzufügen, wählen Sie in der Toolbox unter **Dienste** die Funktion **web_reg_find** aus.

b. Um eine **web_global_verification**-Funktion hinzuzufügen, wählen Sie in der Toolbox unter **Dienste** die erforderliche Funktion **web_global_verification** aus.

3. Ziehen Sie die ausgewählte Funktion an die gewünschte Position im Editor.

4. Geben Sie im geöffneten Dialogfeld die erforderlichen Informationen ein. Details zu den Optionen des Dialogfelds erhalten Sie, indem Sie auf F1 drücken, um die Function Reference zu öffnen.

5. Klicken Sie auf **OK**, um die Funktion in das Vuser-Skript einzufügen.

Hinzufügen einer Bildprüfung nach der Aufzeichnung

1. Navigieren Sie im Editor zu der Position, an der Sie die Prüfung einfügen möchten.

2. Wählen Sie **Ansicht > Toolbox** aus, um die Toolbox zu öffnen.

3. Wählen Sie in der Toolbox unter **Webprüfungen** die Funktion **web_image_check** aus.

4. Ziehen Sie die ausgewählte Funktion **web_image_check** an die gewünschte Position im Editor.

5. Geben Sie im Dialogfeld **Eigenschaften von Bildprüfung** die erforderlichen Informationen ein. Details zu den Optionen des Dialogfelds erhalten Sie, indem Sie auf F1 drücken, um die Function Reference zu öffnen.

6. Klicken Sie auf **OK**, um die Funktion in das Vuser-Skript einzufügen.

Web-Snapshots - Übersicht

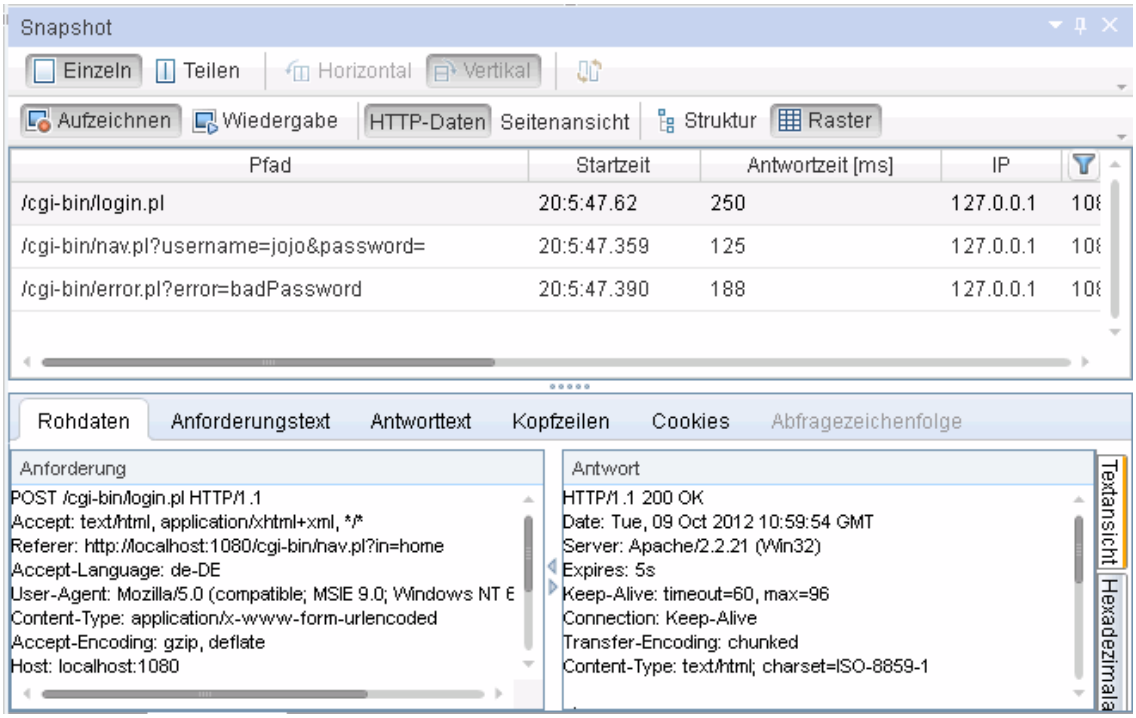
Hinweis: Dieses Thema bezieht sich nur auf Web-Vuser-Protokolle. Eine Liste der Web-Vuser-Protokolle finden Sie unter "Web-Vuser-Typen" auf Seite 900.

Web-Vuser-Skripts verwenden den Snapshot-Bereich von VuGen.

- Eine Einführung zum Snapshot-Bereich finden Sie unter "Snapshot-Bereich - Übersicht" auf Seite 62.
- Details zum Arbeiten mit dem Snapshot-Bereich finden Sie unter "Arbeiten mit Snapshots" auf Seite 65.
- Details zur Benutzeroberfläche des Snapshot-Bereichs finden Sie unter "Snapshot-Bereich" auf Seite 109.

Die Snapshots enthalten detaillierte Informationen zu einigen Schritten im Vuser-Skript. Snapshots können in der Seitenansicht oder in der detaillierteren HTTP-Daten-Ansicht angezeigt werden.

Die HTTP-Daten-Ansicht gibt jede HTTP-Transaktion in einer Struktur- oder Tabellenansicht wieder. Die Transaktionsdaten werden in Antwort- und Anforderungsdaten, Kopfzeilen, Cookies und Abfragezeichenfolgen aufgeschlüsselt.



Daten in den Snapshots können in den folgenden Formaten angezeigt werden: Datenansicht, Textansicht und Hexadezimalansicht.

Sie können den Snapshot-Bereich aufteilen und zwei Snapshots gleichzeitig anzeigen - in der Regel der Aufzeichnungs-Snapshot mit dem zugehörigen Wiedergabe-Snapshot. Wenn beide Snapshots in der HTTP-Daten-Ansicht angezeigt werden, können Sie auf die Schaltfläche **Synchronisieren** in der Symbolleiste des Snapshot-Bereichs klicken, um die in den beiden Snapshots angezeigten Daten zu synchronisieren. Weitere Informationen finden Sie unter "Arbeiten mit Snapshots" auf Seite 65.

Die Antwortdaten können durch einen Rechtsklick auf den ausgewählten Text korreliert und parametriert werden.

Für Daten, die schwierig zu bearbeiten sind (z. B. Binärdaten), bietet VuGen eine Vielzahl von Datenformaterweiterungen, die bestimmte Datentypen in besser lesbare Formate umwandeln. Daten, die von einer Datenformaterweiterung formatiert wurden, können entweder im Originalformat oder dem konvertierten Format angezeigt werden. Weitere Informationen finden Sie unter "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912.

Browseremulation - Übersicht

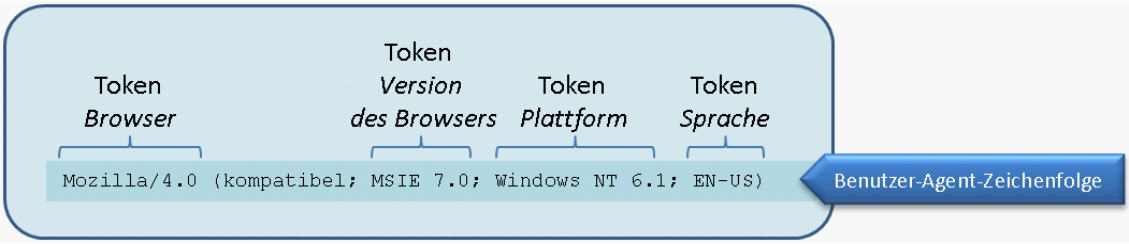
Hinweis: Dieses Thema bezieht sich nur auf Web-Vuser-Protokolle. Eine Liste der Web-

Vuser-Protokolle finden Sie unter "Web-Vuser-Typen" auf Seite 900.

Beim Ausführen eines Web-Vusers, der auf eine Website zugreift, verwendet der Vuser für den Zugriff auf die Website keinen wirklichen Browser. Stattdessen *emuliert* der Vuser einen Browserzugriff auf die Website. Um die Emulation zu aktivieren, verwendet der Vuser eine Benutzer-Agent-Zeichenfolge.

Was ist eine Benutzer-Agent-Zeichenfolge?

Wenn ein Browser eine Anforderung an einen Server sendet, sendet der Browser eine **Benutzer-Agent-Zeichenfolge**, die sich selbst gegenüber dem Server identifiziert. Die charakteristischen Details in der Benutzer-Agent-Zeichenfolge sind in verschiedenen **Token** eingeschlossen. Diese Token bieten verschiedene Details beispielsweise darüber, welcher Browser verwendet wird, welche Version der Browser aufweist und unter welchem Betriebssystem der Browser ausgeführt wird.



Die Benutzer-Agent-Zeichenfolge ist Bestandteil einer **Benutzer-Agent**-Kopfzeile, die wiederum in jeder vom Browser an den Server gesendeten Anforderung enthalten ist. Server können die Informationen, die in der Benutzer-Agent-Zeichenfolge enthalten sind, verwenden, um auf den jeweiligen Browser zugeschnittene Inhalte bereitzustellen.

Im Folgenden finden Sie ein Beispiel für eine Benutzer-Agent-Kopfzeile, die eine Benutzer-Agent-Zeichenfolge enthält, die an einen Server gesendet wird. In diesem Beispiel erkennt die Benutzer-Agent-Zeichenfolge den Browser als Internet Explorer 7.0, der unter Windows 7 ausgeführt wird.

Benutzer-Agent: Mozilla/4.0 (kompatibel; MSIE 7.0; Windows NT 6.1)

Emulieren eines bestimmten Browsers

Beim Ausführen eines Web-Vusers, der auf eine Website zugreift, *emuliert* der Vuser einen Browser, der auf die Website zugreift. Um die Emulation zu aktivieren, erstellt der Vuser eine Benutzer-Agent-Zeichenfolge, die verschiedene Token enthält.

- Mit den Token *Browser* und *Version* werden der zu emulierende Browser und die Version des emulierten Browsers definiert.
- Mit dem Token *Plattform* wird das Betriebssystem definiert, unter dem der emulierte Browser ausgeführt wird.
- Mit dem Token *Sprache* wird die Sprache definiert, für die der emulierte Browser lokalisiert wurde.

Die Benutzer-Agent-Zeichenfolge ist Bestandteil einer **Benutzer-Agent**-Kopfzeile, die in jeder Anforderung enthalten ist, die der Vuser an den Server sendet, auf dem sich die Website befindet.

Welche Informationen sind in der Benutzer-Agent-Zeichenfolge des Web-Vusers enthalten?

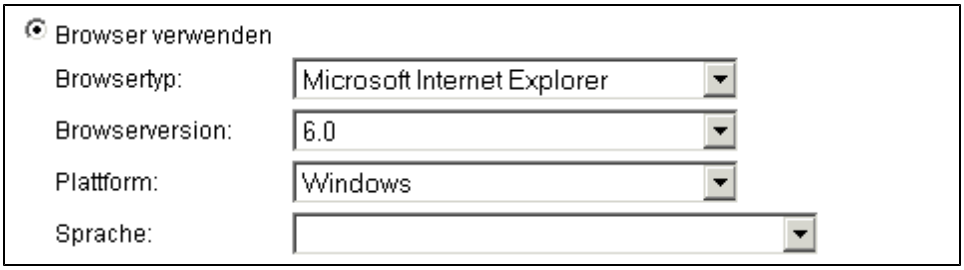
1. Wenn ein Vuser-Skript zum ersten Mal erstellt wird, ist als Browser in der Benutzer-Agent-Zeichenfolge Internet Explorer 6 und als Betriebssystem Windows angegeben.
2. Nachdem der Code in einem Vuser-Skript generiert wurde, wird der Browser in der Benutzer-Agent-Zeichenfolge in den Browser geändert, der beim Aufzeichnen des Skripts verwendet wurde. Auch das Betriebssystem wird in das bei der Aufzeichnung des Skripts verwendete Betriebssystem geändert.

Nachdem der Code in einem Vuser-Skript generiert wurde, können Sie die Laufzeiteinstellungen des Vuser-Skripts verwenden, um einen emulierten Browser festzulegen, der sich von dem bei der Skriptaufzeichnung verwendeten Browser unterscheidet. Es gibt Szenarios, bei denen Sie auf diese Weise die reale Situation genauer emulieren können. Wenn Sie den emulierten Browser festlegen, können Sie Typ und Version des Browsers, die Plattform, auf der der Vuser ausgeführt wird, und die Sprache für den Browser angeben. VuGen erstellt eine Benutzer-Agent-Zeichenfolge, die die von Ihnen angegebenen Details enthält. Falls erforderlich, können Sie die Benutzer-Agent-Zeichenfolge zum Erstellen einer benutzerdefinierten Benutzer-Agent-Zeichenfolge bearbeiten.

Angeben eines bestimmten Browsers für die Emulation

So legen Sie einen emulierten Browser fest, der sich von dem bei der Skriptaufzeichnung verwendeten Browser unterscheidet:

1. Öffnen Sie das Vuser-Skript.
2. Klicken Sie auf **Wiedergeben > Laufzeiteinstellungen**.
3. Klicken Sie im Dialogfeld **Laufzeiteinstellungen** auf **Browser > Browseremulation**, und wählen Sie dann **Browser verwenden**.
4. Wählen Sie eine der verfügbaren Optionen aus der entsprechenden Liste aus, um den zu emulierenden Browser anzugeben.



Hinweis: Die angezeigte **Benutzer-Agent-Zeichenfolge** wird nach jeder Auswahl aktualisiert.

Anpassen der Benutzer-Agent-Zeichenfolge

Sie können die Benutzer-Agent-Zeichenfolge wie folgt anpassen:

1. Öffnen Sie das Vuser-Skript.
2. Klicken Sie auf **Wiedergeben > Laufzeiteinstellungen**.
3. Klicken Sie im Dialogfeld **Laufzeiteinstellungen** auf **Browser > Browseremulation**, und wählen Sie dann **Benutzerdefinierten Wert verwenden**.
4. Ändern Sie die Benutzer-Agent-Zeichenfolge wie gewünscht.

Maximale Anzahl gleichzeitiger Verbindungen

Wenn ein Browser auf eine Website zugreift, verwaltet der Browser mehrere gleichzeitige Verbindungen mit dem Webserver, der als Host für die Website dient. Wenn Sie also auf eine Webseite zugreifen, die viele verschiedene Objekte wie Bilder, JavaScript-Dateien, Frames, Data Feeds etc. enthält, versucht der Browser möglicherweise, die Leistung durch das gleichzeitige Herunterladen mehrerer Objekte zu verbessern. Die maximale Anzahl gleichzeitiger Verbindungen ist abhängig vom Browsertyp und der Version des Browsers. Zum Beispiel beschränkt Internet Explorer 6 die Anzahl gleichzeitiger Verbindungen auf 2; Internet Explorer 8 und Firefox 3+ begrenzen die Anzahl gleichzeitiger Verbindungen auf 6.

Definieren der maximal zulässigen Anzahl gleichzeitiger Verbindungen für einen Vuser

Wenn Sie einen Web-Vuser ausführen, wird die maximale Anzahl gleichzeitiger Verbindungen, die der Vuser mit einem Webserver verwalten kann, standardmäßig mit dem Browser definiert, der in der Benutzer-Agent-Zeichenfolge des Vuser-Skripts angegeben ist. Sie können jedoch die Option **MAX_CONNECTIONS_PER_HOST** in der Funktion **web_set_sockets_option** verwenden, um den Standardwert zu überschreiben. Weitere Informationen finden Sie in der Function Reference **(Hilfe > Funktionsreferenz)**.

Im folgenden Beispiel wird die maximale Anzahl gleichzeitiger Verbindungen auf 10 festgelegt:

```
web_set_sockets_option("MAX_CONNECTIONS_PER_HOST", "10");
```

Hinweis: Beim Ausführen eines Web-Vuser-Skripts wird die maximale Anzahl gleichzeitiger Verbindungen, die der Vuser mit dem Server verwalten kann, im Wiedergabeprotokoll angezeigt.

Durchführen von Lasttests mit nCipher-HSM

In dieser Aufgabe wird beschrieben, wie Sie ein Skript einrichten, mit dem Sie eine Umgebung mit einem nCipher-HSM (Hardwaresicherheitsmodul) Lasttests unterziehen können.

1. **Voraussetzung**

   Generieren Sie die Clientzertifikatsdatei (client_cert.pem) mit dem privaten Schlüssel (client_key.pem) mit einem Verweis auf die im HSM gespeicherten privaten Schlüssel. Stellen Sie sicher, dass Sie mit einer generierten Datei der Zertifizierungsstelle (ca-certs.pem) eine Verbindung mit dem Webserver (A.com) herstellen. Der erfolgreiche Openssl-Befehl muss folgendes Format aufweisen:

   ```
   openssl s_client –connect A.com:443 –CAfile ca-certs.pem –cert client_cert.pem –certform PEM –key client_key.pem – keyform PEM –engine CHIL
   ```

2. **Einrichten der Umgebungsvariablen PATH**

   Fügen Sie der Umgebungsvariablen PATH die Datei **nfhwcrhk.dll** hinzu, die sich normalerweise unter "c:\Programme (x86)\nCipher\nfast\toolkits\hwcrhk" befindet. Starten Sie VuGen neu, damit die Änderung in Kraft tritt.

3. **Aktivieren des Abrufens des nCipher-Schlüssels**

   Legen Sie in den erweiterten Laufzeiteinstellungen die Option **Schlüsselabruf aus nCipher-HSM aktivieren** auf "Ja" fest. Weitere Informationen finden Sie unter "Dialogfeld "Erweiterte Einstellungen"" auf Seite 373.

4. **Bearbeiten des Vuser-Skripts**

   Fügen Sie dem Vuser-Skript im Abschnitt **vuser_init** folgenden Inhalt hinzu.

   ```
   web_set_sockets_option(SSL_VERSION, "TLS");

   web_set_sockets_option(DEFAULT_VERIFY_PATH, <full_path>\ca-certs.pem);

   web_set_certificate_ex (

   "CertFilePath=<full_path_to_client_cert_file>/client_cert.pem",

   "CertFormat=PEM",
   ```

```
"KeyFilePath= <full_path_to_client_private_key_file> /client_key.pem",

"KeyFormat=PEM",

LAST);
```

Arbeiten mit Daten im Cache

Hinweis: Dieses Thema bezieht sich nur auf Web-Vuser-Protokolle. Eine Liste der Web-Vuser-Protokolle finden Sie unter "Web-Vuser-Typen" auf Seite 900.

Webbrowser verwalten einen Zwischenspeicher mit Objekten, die vom Browser heruntergeladen wurden. Wenn beim Zugreifen auf eine Website eines dieser Objekte benötigt wird, kann der Browser die Objekte direkt aus dem Zwischenspeicher verwenden, d. h., sie müssen nicht erneut heruntergeladen werden. Auf diese Weise können die angeforderten Seiten schneller geladen werden. Wenn Sie ein Vuser-Skript ausführen, startet der Vuser standardmäßig mit einem leeren Zwischenspeicher. Dies bedeutet, dass der Vuser Objekte, die das erste Mal angefordert werden, herunterladen muss. Für eine bessere Emulation einer realen Situation können Sie für den Vuser in jeder Phase seiner Ausführung einen vordefinierten Zwischenspeicher bereitstellen. Der Vuser kann dann auf die erforderlichen Objekte direkt im Zwischenspeicher zugreifen, ohne dass sie heruntergeladen werden müssen. Um für einen Vuser einen vordefinierten Zwischenspeicher bereitzustellen, erstellen Sie zunächst eine Datei, in der der Zwischenspeicher enthalten ist, und laden Sie den Zwischenspeicher dann in den Vuser. Zum Erstellen der Zwischenspeicherdatei verwenden Sie die Funktion **web_dump_cache**; zum Laden der Zwischenspeicherdatei in den Vuser die Funktion **web_load_cache**.

- Ausführliche Informationen zu den Zwischenspeicherfunktionen finden Sie in unter *Function Reference* **(Hilfe > Funktionsreferenz)**.
- Weitere Informationen zum Implementieren der Zwischenspeicherfunktionen finden Sie unter "Einfügen von Zwischenspeicherfunktionen" auf Seite 912.

Erstellen der Zwischenspeicherdatei

Zum Erstellen einer Zwischenspeicherdatei verwenden Sie die Funktion **web_dump_cache**. Mit der Funktion **web_dump_cache** wird eine Datei erstellt, in der alle Objekte enthalten sind, die im Vuser-Zwischenspeicher vorhanden sind, wenn die Funktion **web_dump_cache** ausgeführt wird. Fügen Sie die Funktion **web_dump_cache** in ein Vuser-Skript ein, normalerweise am Ende des Abschnitts *Action* des Skripts. So wird sichergestellt, dass im Vuser-Zwischenspeicher die erforderlichen Objekte enthalten sind, wenn die Funktion **web_dump_cache** zum Erstellen der Zwischenspeicherdatei ausgeführt wird. Nach dem Einfügen der Funktion **web_dump_cache** führen Sie das Skript aus, um den Vuser-Zwischenspeicher und die Zwischenspeicherdatei zu erstellen.

Hinweis: Sie müssen die Funktion **web_dump_cache** nur einmal ausführen, um die

erforderliche Zwischenspeicherdatei zu generieren. In der Regel gehört diese Ausführung nicht zu einem Szenario. Wenn die Zwischenspeicherdatei erstellt wurde und Sie das Vuser-Skript als Teil eines Szenarios ausführen, muss die Funktion **web_dump_cache** nicht ausgeführt werden. Daher sollten Sie die Funktion **web_dump_cache** im Vuser-Skript auskommentieren.

Verwenden Sie das Argument **FileName** in der Funktion **web_dump_cache**, um den Namen und den Speicherort der zu erstellenden Datei anzugeben. Der Pfad **FileName** kann absolut sein (z. B. "FileName=c:\\MyDir\\User1.cache") oder relativ zum aktuellen Vuser-Verzeichnis (z. B. "FileName=Iteration1.cache").

- **Absolute Pfadnamen:** Verwenden Sie absolute Pfadnamen, wenn die Zwischenspeicherdatei nicht mit dem Skript verknüpft sein soll. Verwenden Sie beispielsweise einen absoluten Pfad, wenn Sie auf jedem Host einen anderen Zwischenspeicher verwenden möchten.
- **Relative Pfadnamen:** Wenn Sie einen relativen Pfadnamen verwenden, wird die Zwischenspeicherdatei im Vuser-Verzeichnis erstellt. Wenn Sie das Vuser-Skript kopieren, an einem neuen Speicherort ablegen oder in einen Lastgeneratorhost laden, wird auch die Zwischenspeicherdatei kopiert. Relative Pfade sind unabhängig von Laufwerkzuordnungen und Netzwerkspeicherorten.

Die Dateierweiterung der Zwischenspeicherdatei lautet immer ".cache". Die Erweiterung ".cache" wird hinzugefügt, wenn sie im Argument **FileName** nicht angegeben wird. Wenn Sie beispielsweise "FileName=Iteration2.txt" angeben, lautet der Name der Zwischenspeicherdatei "FileName=Iteration2.txt.cache".

Dateinamen im Argument **FileName** können parametriert werden, sodass von unterschiedlichen Vusern oder Iterationen verschiedene Zwischenspeicherdateien verwendet werden können. Beispiel: "FileName=Iteration{param}.cache"

Im folgenden Beispiel erstellt die Funktion **web_dump_cache** für jeden Parameter des Typs "VuserName", der das Skript ausführt, eine Zwischenspeicherdatei. Die Zwischenspeicherdateien befinden sich im Verzeichnis "c:\temp".

```
web_dump_cache("paycheckcache","FileName=c:\\temp\\{Vuser
         Name}paycheck", "Replace=yes", LAST)
```

Wenn Sie einen einzelnen Vuser-Benutzer zehn Mal ausführen, erstellt VuGen zehn Zwischenspeicherdateien im folgenden Format, wobei das Präfix "Kunnn" dem Wert für VuserName entspricht:

```
Ku001paycheck.cache
Ku002paycheck.cache
Ku003paycheck.cache
...
```

Laden der Zwischenspeicherdatei in einen Vuser

Die Funktion **web_load_cache** lädt die Zwischenspeicherdatei in einen Vuser. Das Argument **FileName** in der Funktion **web_load_cache** gibt den Namen und den Speicherort der zu ladenden Zwischenspeicherdatei an. Die angegebene Zwischenspeicherdatei muss vorhanden sein, bevor

die Funktion **web_load_cache** ausgeführt wird. Daher können Sie die Funktion **web_load_cache** erst ausführen, nachdem Sie die Funktion **web_dump_cache** ausgeführt haben, um die Zwischenspeicherdatei zu erstellen.

Im folgenden Beispiel lädt die Funktion **web_load_cache** die Zwischenspeicherdateien des Typs **{VuserName}paycheck** aus dem Verzeichnis **c:\temp**.

```
web_load_cache("ActionLoad","FileName=c:\\temp\\{VuserName}paycheck",LAST)
```

Einfügen von Zwischenspeicherfunktionen

Hinweis: Dieses Thema bezieht sich nur auf Web-Vuser-Protokolle. Eine Liste der Web-Vuser-Protokolle finden Sie unter "Web-Vuser-Typen" auf Seite 900.

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie Zwischenspeicherfunktionen verwenden. Mithilfe von Zwischenspeicherfunktionen können Sie eine Zwischenspeicherdatei erstellen, die den Vuser-Cache enthält, und die Daten aus der Zwischenspeicherdatei dann in einen Vuser laden. Weitere Informationen finden Sie unter "Arbeiten mit Daten im Cache" auf Seite 910. In den folgenden Schritten wird erklärt, wie Sie die Zwischenspeicherfunktionen nutzen.

1. Fügen Sie die Funktion **web_dump_cache** in das Vuser-Skript ein.
2. Führen Sie das Skript aus, um die Zwischenspeicherdatei zu erstellen.
3. Fügen Sie die Funktion **web_load_cache** im Skript vor den Aktionen des Typs Vuser ein.
4. Kommentieren Sie die Funktion **web_dump_cache** aus.
5. Führen Sie das Skript aus und speichern Sie es.

Ausführliche Informationen zu den Zwischenspeicherfunktionen finden Sie in der *Function Reference* (**Hilfe > Funktionsreferenz**).

Datenformaterweiterungen (DFEs) - Übersicht

Hinweis: Dieser Abschnitt gilt für Web (HTTP/HTML)- und Silverlight-Vuser-Skripts sowie für Web (HTTP/HTML)-Schritte in Flex-Vuser-Skripts.

Definition: Die Unterstützung der Datenformaterweiterung (kurz DFE-Unterstützung genannt) ermöglicht durch die Bereitstellung von Funktionen zum Decodieren und Codieren formatierter Daten für den Austausch zwischen Client und Server eine einfachere Skripterstellung für Webanwendungen. Dadurch werden auch die Korrelation und die Parametrierung der generierten Vuser-Skripts vereinfacht.

Weitere Informationen zum Implementieren von DFEs in Ihre Vuser-Skripts finden Sie unter "Implementieren der Unterstützung für Datenformaterweiterungen (DFEs)" auf Seite 916.

In welchen Situationen sind DFEs sinnvoll?

Beim Aufzeichnen eines Vuser-Skripts zeichnet VuGen die HTTP-Anforderungen und -Antworten auf, die zwischen Client und Webserver ausgetauscht werden. Die Daten in diesen HTTP-Anforderungen und -Antworten sind häufig codiert. Beispielsweise können einige der Daten im binären Format sein. Die codierten Daten können sich in der HTTP-Abfragezeichenfolge, in Kopfzeilen, Text oder Cookies befinden. Wenn die codierten Daten in einem Vuser-Skript enthalten sind, kann das sich ergebende Skript Daten enthalten, die schwer zu entschlüsseln sind. Das macht es schwierig, Textzeichenfolgen zu erkennen, die zur Parametrierung und Korrelierung verwendet werden können.

Das nachstehende Skriptsegment zeigt einen Abschnitt eines Vuser-Skripts, das beim Aufzeichnen von Geschäftsprozessen für eine GWT-basierte Anwendung generiert wurde. Beachten Sie, dass einige Abschnitte des Skripts codierte Daten enthalten und somit schwer zu entschlüsseln sind.

LoadRunner verwendet Datenformaterweiterungen (DFEs) zur Behebung der Schwierigkeiten, die sich aus codierten Daten in Vuser-Skripts ergeben. Die DFE-Unterstützung ermöglicht eine einfachere Erstellung von Vuser-Skripts durch die Möglichkeit zum Decodieren der codierten Daten, die zwischen dem Client und dem Server ausgetauscht werden. Durch die Bereitstellung des decodierten Formats der Daten werden die Informationen im Vuser-Skript in einem lesbaren Format dargestellt, sodass Sie das Skript nach Bedarf korrelieren und parametrieren können. Wenn das Skript wiedergegeben wird, codiert die DFE-Unterstützung das geänderte Vuser-Skript wieder und ermöglicht es dem Vuser so, die korrekt codierte Anforderung an den Server zu senden.

LoadRunner enthält eine Reihe vordefinierter DFEs. Mit jeder DFE können Sie einen bestimmten Typ von Daten decodieren und codieren. Mit der GWT-DFE werden beispielsweise GWT-Daten beim Generieren eines Skripts ins XML-Format decodiert. Vor der Wiedergabe des Skripts werden die XML-formatierten Daten dann wieder in GWT-formatierte Daten codiert. Eine vollständige Liste der vordefinierten DFEs finden Sie unter "Liste der Datenformaterweiterungen" auf Seite 922.

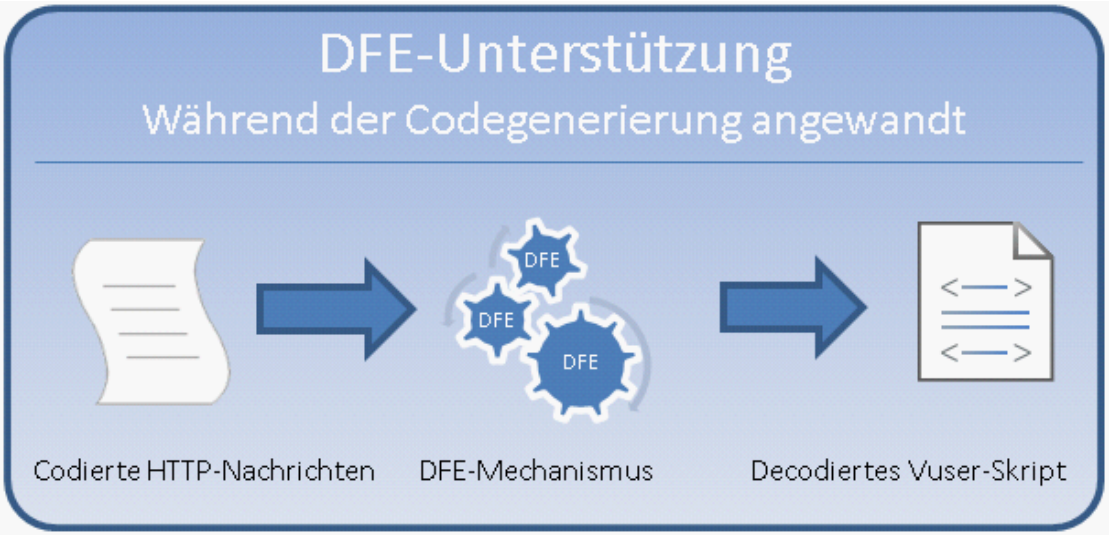
Wenn eine DFE auf ein Vuser-Skript angewendet wird und das Skript dann neu generiert wird, ändert die DFE das Skript und ersetzt die codierten Daten durch die decodierten Daten. Weitere Informationen zum Ändern des Skripts finden Sie unter "So ändern DFEs ein Vuser-Skript" auf Seite 921.

Wann wird die DFE-Funktionalität angewendet?

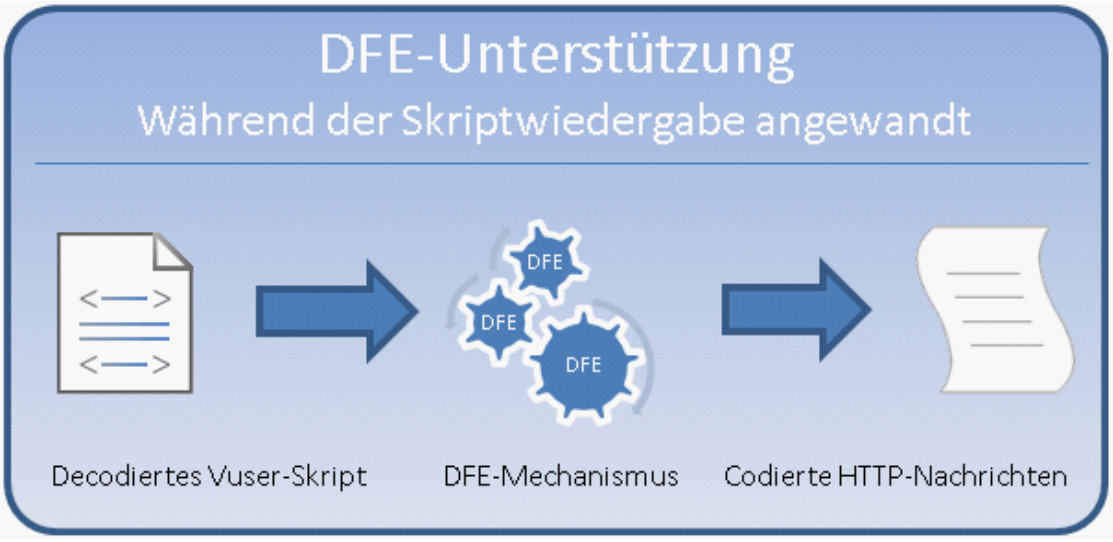
Sie aktivieren die DFE-Unterstützung für jedes Vuser-Skript, für das codierte Daten decodiert werden müssen. Wenn die DFE-Unterstützung aktiviert ist, wird sie in folgenden Situationen

angewendet:

- Bei jeder Generierung (nach der Aufzeichnung) oder Neugenerierung des Skripts. Die DFEs werden zum Decodieren der codierten Daten angewendet, um ein einfacher zu entschlüsselndes Skript zu erzeugen.



- Bei jeder Ausführung des Skripts. Die DFEs werden angewendet, um die decodierten Daten erneut zu codieren, damit HTTP-Nachrichten mit codierten Daten in einem vom Server erwarteten Format erzeugt werden.

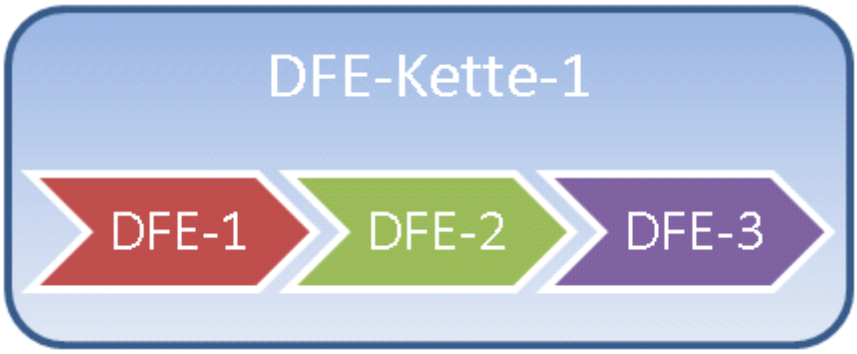


Hinweis: Zusätzlich zum Anwenden von DFEs bei der Generierung oder Wiedergabe eines Skripts ist es möglich, eine DFE auf eine ausgewählte Zeichenfolge in einem Vuser-Skript anzuwenden. Weitere Informationen finden Sie unter "Anwenden von DFEs auf eine Zeichenfolge" auf Seite 923.

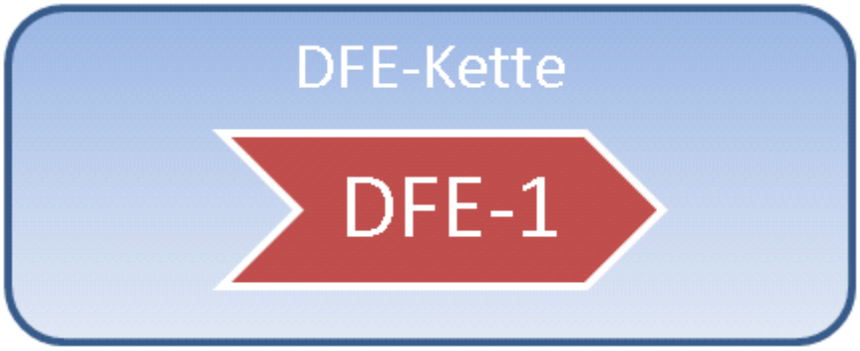
In bestimmten Szenarios muss die Decodierung codierter Daten in mehreren Stufen durchgeführt werden, bis vollständig decodierte Daten erzeugt werden. Jede Stufe im Konvertierungsprozess wird durch Anwenden einer bestimmten DFE ausgeführt. Beispielsweise können codierte Daten aus einer Antwort durch Anwenden von drei DFEs decodiert werden: zuerst DFE-1, dann DFE-2 und schließlich DFE-3. Auf jeder Stufe wird die Ausgabe von einer DFE zur Eingabe für die nächste DFE, bis die vollständig decodierten Daten bereitgestellt werden.

DFE-Ketten

Die Reihe der DFEs, die zum Decodieren codierter Daten erforderlich sind, ist in einer **Kette** definiert. Sie können zum Beispiel eine Kette mit der Bezeichnung **DFE-Kette-1** erstellen, die drei DFEs umfasst: DFE-1, DFE-2 und DFE-3. Die Abfolge der DFEs innerhalb einer Kette ist signifikant - sie gibt die Reihenfolge an, in der die DFEs auf die codierten Daten angewendet werden.



Hinweis: Wenn nur eine einzige DFE zum Decodieren codierter Daten erforderlich ist, muss die DFE dennoch Bestandteil einer Kette sein.



Zuweisen von DFE-Ketten

HTTP-Nachrichten können in mehrere Abschnitte unterteilt werden, beispielsweise in Text, Kopfzeilen, Cookies und eine Abfragezeichenfolge. Nachdem Sie die DFE-Ketten definiert haben, die zum Decodieren und Codieren eines Vuser-Skripts angewendet werden, müssen Sie angeben,

auf welche Abschnitte der HTTP-Nachrichten die DFE-Ketten angewendet werden sollen. Da jede HTTP-Meldung nur einen Abschnitt **Text** und einen Abschnitt **Abfragezeichenfolge** aufweist, können Sie für jeden dieser Abschnitte nur eine einzelne DFE-Kette angeben. Im Gegensatz dazu kann jede HTTP-Nachricht mehrere Kopfzeilen und Cookies enthalten. Dementsprechend können Sie eine bestimmte DFE-Kette festlegen, die für jede Kopfzeile und jedes Cookie gelten soll. Weitere Informationen finden Sie unter "Anwenden von DFE-Ketten auf Abschnitte der HTTP-Nachricht" auf Seite 920.

Wiedergeben von Vuser-Skripts, die DFE-Funktionen enthalten

Wenn Sie ein Vuser-Skript mit DFE-Funktionalität wiedergeben, werden verschiedene Meldungen zum Wiedergabeprotokoll im VuGen-Ausgabebereich hinzugefügt. Achten Sie darauf, diese Meldungen zu überprüfen, um sicherzustellen, dass die DFE-Funktionalität korrekt implementiert ist. Details finden Sie unter "Fehlerbehebung - Datenformaterweiterung (DFE) " auf Seite 930.

Benutzerdefinierte DFEs

Fortgeschrittene LoadRunner-Benutzer können benutzerdefinierte DFEs erstellen. Weitere Details finden Sie im *HP LoadRunner Data Format Extensions Developer Guide*. Darüber hinaus können Sie das LoadRunner-JavaScript-Modul zum Codieren und Decodieren von Daten mit allgemeinen JavaScript-Bibliotheken verwenden. Weitere Informationen finden Sie unter "Verwenden des JavaScript-Moduls von LoadRunner" auf Seite 885.

Implementieren der Unterstützung für Datenformaterweiterungen (DFEs)

Hinweis: Dieser Abschnitt gilt für Web (HTTP/HTML)- und Silverlight-Vuser-Skripts sowie für Web (HTTP/HTML)-Schritte in Flex-Vuser-Skripts.

Um die DFE-Unterstützung für ein Vuser-Skript zu implementieren, müssen Sie die in der nachfolgenden Abbildung dargestellten Schritte ausführen:

Weitere Informationen zu DFEs finden Sie unter "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912.

Definieren einer DFE-Kette

Hinweis: Dieser Abschnitt gilt für Web (HTTP/HTML)- und Silverlight-Vuser-Skripts sowie für

Web (HTTP/HTML)-Schritte in Flex-Vuser-Skripts.

Um die DFE-Unterstützung für ein Vuser-Skript zu implementieren, müssen Sie zunächst die DFE-Ketten definieren, die vom Vuser angewendet werden. Das Definieren der DFE-Ketten ist der erste Schritt bei der Implementierung der DFE-Unterstützung.

Das Definieren der DFE-Ketten umfasst die folgenden Aufgaben:

1. **Hinzufügen einer DFE-Kette**

   Öffnen Sie in VuGen das Vuser-Skript.

   Klicken Sie auf **Aufzeichnen** > **Aufzeichnungsoptionen** > **Datenformaterweiterung** > **Kettenkonfiguration**.

   Weitere Informationen zu den Optionen des Dialogfelds finden Sie unter "Datenformaterweiterung > Kettenkonfiguration (Knoten)" auf Seite 184.

   Klicken Sie unter **Ketten** auf die Schaltfläche **Neue Kette** , und erstellen Sie eine neue Kette. Die neue Kette ist im Bereich **Ketten** aufgeführt.

2. **Hinzufügen von DFEs zur neuen DFE-Kette**

   Wählen Sie im Bereich **Ketten** die Kette aus, zu der Sie DFEs hinzufügen möchten.

   Klicken Sie im Bereich **Kette: <Kettenname>** auf die Schaltfläche **DFE hinzufügen** .

   Wählen Sie im Dialogfeld **Datenformaterweiterung hinzufügen** die DFE aus, die Sie zur Kette hinzufügen möchten, und klicken Sie anschließend auf **OK**.

   Wenn Sie die **GWT**-DFE oder die **Präfix/Postfix**-DFE zu einer Kette hinzufügen, müssen Sie zusätzliche Konfigurationsdetails zur DFE angeben. Weitere Informationen finden Sie in der Dokumentation zu der jeweiligen DFE.

   Informationen zum Bereich **Kette: <Kettenname>** finden Sie unter "Datenformaterweiterung > Kettenkonfiguration (Knoten)" auf Seite 184.

   Nachdem Sie der Kette eine DFE hinzugefügt haben, wählen Sie die geeignete Option in der Liste **Verarbeitung fortsetzen** aus. Weitere Informationen zur Option **Verarbeitung fortsetzen** finden Sie unter "Datenformaterweiterung > Kettenkonfiguration (Knoten)" auf Seite 184.

Fügen Sie nach Bedarf zusätzliche DFEs zur Kette hinzu.

- Nachdem Sie die erforderlichen DFE-Ketten definiert haben, müssen Sie die DFE-Unterstützung für das Vuser-Skript aktivieren. Weitere Informationen hierzu finden Sie unter "Aktivieren der DFE-Unterstützung" unten.
- Eine Übersicht über den Implementierungsprozess für die DFE-Unterstützung finden Sie unter "Implementieren der Unterstützung für Datenformaterweiterungen (DFEs)" auf Seite 916.
- Weitere Informationen zu DFEs finden Sie unter "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912.

Aktivieren der DFE-Unterstützung

Hinweis: Dieser Abschnitt gilt für Web (HTTP/HTML)- und Silverlight-Vuser-Skripts sowie für Web (HTTP/HTML)-Schritte in Flex-Vuser-Skripts.

Nachdem Sie die für ein Vuser-Skript verfügbaren DFE-Ketten definiert haben, müssen Sie die DFE-Unterstützung wie nachfolgend beschrieben aktivieren. Das Aktivieren der DFE-Unterstützung ist der zweite Schritt bei der Implementierung der DFE-Unterstützung.

1. Öffnen Sie in VuGen das Vuser-Skript.
2. Klicken Sie auf **Aufzeichnen** > **Aufzeichnungsoptionen** > **Datenformaterweiterung** > **Codegenerierung**.

   Weitere Informationen zu den Optionen des Dialogfelds finden Sie unter "Datenformaterweiterung > Kettenkonfiguration (Knoten)" auf Seite 184.
3. Aktivieren Sie das Kontrollkästchen **Datenformaterweiterung aktivieren**.

- Nach dem Aktivieren der DFE-Unterstützung können Sie diese wie unter "Konfigurieren der DFE-Unterstützung" auf der nächsten Seite beschrieben konfigurieren.
- Weitere Informationen zum Definieren von DFE-Ketten finden Sie unter "Definieren einer DFE-Kette" auf Seite 916.

- Eine Übersicht über den Implementierungsprozess für die DFE-Unterstützung finden Sie unter "Implementieren der Unterstützung für Datenformaterweiterungen (DFEs)" auf Seite 916.
- Weitere Informationen zu DFEs finden Sie unter "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912.

Konfigurieren der DFE-Unterstützung

Hinweis: Dieser Abschnitt gilt für Web (HTTP/HTML)- und Silverlight-Vuser-Skripts sowie für Web (HTTP/HTML)-Schritte in Flex-Vuser-Skripts.

Nachdem Sie die DFE-Unterstützung für ein Vuser-Skript aktiviert haben, können Sie die DFE-Unterstützung wie nachstehend beschrieben konfigurieren. Das Konfigurieren der DFE-Unterstützung ist der dritte Schritt bei der Implementierung der DFE-Unterstützung.

1. Öffnen Sie das Vuser-Skript in VuGen.
2. Klicken Sie auf **Aufzeichnen** > **Aufzeichnungsoptionen** > **Datenformaterweiterung** > **Codegenerierung**.

   Weitere Informationen zu den Optionen des Dialogfelds finden Sie unter "Datenformaterweiterung > Codegenerierung (Knoten)" auf Seite 188.
3. Wählen Sie unter **Konfiguration** aus der Liste **Format** die Teile des Vuser-Skripts aus, auf die die DFEs angewendet werden sollen.
   - **Code und Snapshots**. Wendet DFEs zum Konvertieren des Vuser-Skriptcodes und der Snapshotdaten an.
   - **Snapshots**. Wendet DFEs nur zum Konvertieren der Snapshotdaten an - nicht zum Konvertieren des Vuser-Skriptcodes.
4. Aktivieren Sie das Kontrollkästchen **Formatierte Daten überprüfen**, um die Ergebnisse der Datenkonvertierung zu überprüfen, indem Sie die konvertierten Daten zurück in den Originalzustand konvertieren und anschließend feststellen, ob diese mit den ursprünglichen Daten übereinstimmen.

- Weitere Informationen zu den Optionen des Dialogfelds finden Sie unter "Datenformaterweiterung > Codegenerierung (Knoten)" auf Seite 188.

- Nach dem Konfigurieren der DFE-Unterstützung können Sie die DFE-Ketten bestimmten Abschnitten der HTTP-Nachrichten zuordnen. Weitere Informationen finden Sie unter "Anwenden von DFE-Ketten auf Abschnitte der HTTP-Nachricht" unten.
- Eine Übersicht über den Implementierungsprozess für die DFE-Unterstützung finden Sie unter "Implementieren der Unterstützung für Datenformaterweiterungen (DFEs)" auf Seite 916.
- Weitere Informationen zu DFEs finden Sie unter "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912.

Anwenden von DFE-Ketten auf Abschnitte der HTTP-Nachricht

Hinweis: Dieser Abschnitt gilt für Web (HTTP/HTML)- und Silverlight-Vuser-Skripts sowie für Web (HTTP/HTML)-Schritte in Flex-Vuser-Skripts.

Nachdem Sie die DFE-Ketten definiert haben, die für ein Vuser-Skript zur Verfügung stehen, und die DFE-Unterstützung aktiviert und konfiguriert haben, müssen Sie festlegen, auf welche Abschnitte der HTTP-Nachricht die DFE-Ketten angewendet werden sollen. Dieser Vorgang wird nachfolgend beschrieben. Das Anwenden von DFE-Ketten auf Nachrichtenabschnitte ist der letzte Schritt bei der Implementierung der DFE-Unterstützung.

1. Öffnen Sie das Vuser-Skript in VuGen.
2. Klicken Sie auf **Aufzeichnen** > **Aufzeichnungsoptionen** > **Datenformaterweiterung** > **Codegenerierung**.

   Weitere Informationen zu den Optionen des Dialogfelds finden Sie unter "Datenformaterweiterung > Kettenkonfiguration (Knoten)" auf Seite 184.
3. [Optional] Klicken Sie im Bereich **Nachrichtenabschnitt** auf **Text**, und wählen Sie dann die Kette aus, die auf den Nachrichtentext angewendet werden soll.
4. [Optional] Klicken Sie im Bereich **Nachrichtenabschnitt** auf **Kopfzeilen**, und wählen Sie dann die Kette aus, die auf die Nachrichtenkopfzeilen angewendet werden soll.

Hinweis: Damit VuGen die Kette korrekt einer bestimmten Kopfzeile zuweisen kann, muss der Eintrag in der Spalte **Name** genau mit dem Namen in der Kopfzeile der Nachricht übereinstimmen.

5. [Optional] Klicken Sie im Bereich **Nachrichtenabschnitt** auf **Cookies**, und wählen Sie dann die Kette aus, die auf die Nachrichtencookies angewendet werden soll.

Hinweis: Damit VuGen die Kette korrekt einem bestimmten Cookie zuweisen kann, muss der Eintrag in der Spalte **Name** genau mit dem Namen des Cookies in der Nachricht übereinstimmen.

6. [Optional] Klicken Sie im Bereich **Nachrichtenabschnitt** auf **Abfragezeichenfolge**, und wählen Sie dann die Kette aus, die auf die Abfragezeichenfolge der Nachricht angewendet werden soll.

Hinweis: Für die Bereiche **Text** und **Abfragezeichenfolge** können Sie lediglich die Standardkette ändern; den Bereichen **Kopfzeilen** und **Cookies** hingegen können Sie mehrere Ketten hinzufügen.

- Nach dem Konfigurieren der DFE-Unterstützung können Sie die DFE-Ketten bestimmten Abschnitten der HTTP-Nachrichten zuordnen. Weitere Informationen finden Sie unter "Anwenden von DFE-Ketten auf Abschnitte der HTTP-Nachricht" auf der vorherigen Seite.
- Eine Übersicht über den Implementierungsprozess für die DFE-Unterstützung finden Sie unter "Implementieren der Unterstützung für Datenformaterweiterungen (DFEs)" auf Seite 916.
- Weitere Informationen zu DFEs finden Sie unter "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912.

So ändern DFEs ein Vuser-Skript

Hinweis: Dieser Abschnitt gilt für Web (HTTP/HTML)- und Silverlight-Vuser-Skripts sowie für Web (HTTP/HTML)-Schritte in Flex-Vuser-Skripts.

Wenn eine DFE auf ein Vuser-Skript angewendet und das Skript anschließend neu generiert wird, müssen aufgrund der DFE folgende Änderungen am Skript vorgenommen werden:

- VuGen ersetzt die ursprüngliche codierte Textzeichenfolge durch einen Parameter.
- VuGen fügt eine Funktion **web_convert_from_formatted** vor der Funktion ein, die den neuen Parameter enthält. Die Funktion **web_convert_from_formatted** enthält den decodierten Wert des ursprünglichen codierten Textes.

Der nachstehende Skriptabschnitt zeigt eine Funktion **web_custom_request**, die ohne DFE-Unterstützung generiert wurde. Das **Body**-Tag in der Funktion enthält eine Textzeichenfolge, die mit Base64 codiert ist, **Body=TW9uZGF5**. Da der Wert des **Body**-Tags codiert ist, kann der zugehörige Wert nicht so leicht geändert werden, selbst wenn dies für Korrelations- oder Parametrierungszwecke erforderlich ist.

```
web_custom_request("echo_post.asp",
    "URL=http://example.devlab.ad/cgi-bin/temp/echo_post.asp",
    "Method=POST",
    "Resource=0",
    "RecContentType=text/html",
    "Referer=",
    "Snapshot=t1.inf",
    "Mode=HTTP",
    "Body=TW9uZGF5",
    LAST);
```

Nach dem Anwenden der Base64-DFE-Unterstützung wird der Wert des **Body**-Tags im neu generierten Skript durch einen Parameter namens DFE_PARAM, "**Body={DFE_PARAM}** (siehe unten) ersetzt.

```
/*TODO: A Correlation scan needs to be performed.*/
    web_convert_from_formatted("FormattedData/EscapedBinary=<HP_EXTENSION name=\"Base64\">Monday</HP_EXTENSION>",
        "TargetParam=DFE_PARAM",
        LAST);

    web_custom_request("echo_post.asp",
        "URL=http://example.devlab.ad/cgi-bin/temp/echo_post.asp",
        "Method=POST",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=",
        "Snapshot=t1.inf",
        "Mode=HTTP",
        "Body={DFE_PARAM}",
        LAST);

    return 0;
```

Darüber hinaus enthält der geänderte Code auch eine Funktion **web_convert_from_formatted**. Die Funktion gibt an, dass der decodierte Wert der ursprünglich codierten Zeichenfolge **Montag** lautet. Jetzt können Sie ganz einfach den Wert von **Montag** in einen beliebigen anderen Tag ändern. Dazu ändern Sie einfach den decodierten Wert in der Funktion **web_convert_from_formatted**.

Liste der Datenformaterweiterungen

Hinweis: Dieser Abschnitt gilt für Web (HTTP/HTML)- und Silverlight-Vuser-Skripts sowie für Web (HTTP/HTML)-Schritte in Flex-Vuser-Skripts.

Die folgende Tabelle enthält eine Liste der vordefinierten LoadRunner-DFEs (Datenformaterweiterungen). Weitere Informationen zu DFEs finden Sie unter "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912.

Datenformaterweiterung	Beschreibung

Base64-Erweiterung	Decodiert Zeichenfolgen, die mit einem BASE64-Encoder codiert wurden.
Binär-zu-XML-Erweiterung	Wandelt binäre Microsoft WCF-XML-Daten in das XML-Format um.
GWT-Erweiterung	Wandelt GWT-Daten in das XML-Format um.
JSON-zu-XML-Erweiterung	Wandelt JSON-Daten in das XML-Format um.
Präfix/Postfix-Erweiterung	Damit können Sie Daten am Anfang und/oder Ende einer Zeichenfolge ausschneiden, die nicht decodiert werden sollen. Sie können so viele Präfix/Postfix-Erweiterungen hinzufügen und anpassen, wie erforderlich sind. Jede erstellte Präfix/Postfix-Erweiterung sollte einen eindeutigen Anzeige- und Tagnamen erhalten.
Remedy-zu-XML-Erweiterung	Wandelt Remedy-Anforderungsdaten in XML-Format um. Beachten Sie, dass diese Erweiterung keine Remedy-Antwortdaten umwandelt, da diese im JavaScript-Code sind.
Erweiterung für URL-Codierung	Zur Dekodierung von Zeichenfolgen, die mit einem URL-Format codiert wurden.
XML-Erweiterung	Überprüft Daten auf ihre Übereinstimmung mit dem XML-Syntax. Diese Prüfung ermöglicht es VuGen, Korrelationen auf Grundlage von XPath auszuführen und Snapshot-Daten in einer XML-Anzeige darzustellen.
XSS-Erweiterung	Ermöglicht es Ihnen, Sites zu testen, die XSS-Abwehrcode (Cross Site Scripting) verwenden.

Anwenden von DFEs auf eine Zeichenfolge

Hinweis: Dieser Abschnitt gilt für Web (HTTP/HTML)- und Silverlight-Vuser-Skripts sowie für Web (HTTP/HTML)-Schritte in Flex-Vuser-Skripts.

Sie können DFEs auf ein Vuser-Skript anwenden, um codierte Daten im Skript zu decodieren. Sie können die DFEs auf folgende Elemente anwenden:

- Auf angegebene Abschnitte des Vuser-Skripts, wenn das Skript generiert oder erneut generiert wird. Weitere Informationen finden Sie unter "Implementieren der Unterstützung für Datenformaterweiterungen (DFEs)" auf Seite 916.
- Auf eine einzelne codierte Zeichenfolge im Vuser-Skript, wie in diesem Abschnitt beschrieben.

Eine Übersicht über DFEs finden Sie unter "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912.

Hinweis: Nachdem Sie eine DFE auf eine Zeichenfolge in einem Vuser-Skript angewendet haben, fügt VuGen einen Eintrag auf der Registerkarte **Aufgaben** hinzu. Der hinzugefügte Eintrag gibt an, dass eine Korrelationsprüfung durchgeführt werden muss. Weitere Informationen zum Durchführen einer Korrelationsprüfung finden Sie unter "Correlation Studio" auf Seite 281.

So wenden Sie eine DFE auf eine codierte Zeichenfolge in einem Vuser-Skript an:

1. Öffnen Sie das Skript in VuGen, und wählen Sie die codierte Textzeichenfolge aus.
2. Klicken Sie mit der rechten Maustaste in die Auswahl, klicken Sie auf **Decodieren mit DFE**, und klicken Sie auf den Namen der Kette mit den DFEs, um die codierte Zeichenfolge zu decodieren. Weitere Informationen zum Definieren einer DFE-Kette finden Sie unter "Definieren einer DFE-Kette" auf Seite 916.

   VuGen ersetzt den ausgewählten Text durch einen Parameter und fügt eine Funktion **web_convert_from_formatted** hinzu, die die decodierte Entsprechung des ursprünglich ausgewählten Textes enthält.

   Hinweis: Um den Namen zu ändern, den VuGen neuen Parametern zuweist, klicken Sie mit der rechten Maustaste auf codierten Text im Vuser-Skript, klicken Sie dann auf **Decodieren mit DFE > Erweitert** und geben den Parameternamen im Feld **Zielparameter** an. VuGen fügt einen Zähler zum Parameternamen hinzu und erhöht den Zähler nach Bedarf.

Übersicht über GWT-DFE (Google Web Toolkit - Datenformaterweiterung)

Hinweis: Dieser Abschnitt gilt für Web (HTTP/HTML)- und Silverlight-Vuser-Skripts sowie für Web (HTTP/HTML)-Schritte in Flex-Vuser-Skripts.

Die GWT-DFE ist eine der standardmäßigen LoadRunner-DFEs. Mit der GWT-DFE-Unterstützung können Sie Vuser-Skripts für GWT-basierte Websites generieren, die den GWT-RPC-Mechanismus verwenden. Wenn Sie eine GWT-RPC-basierte Website ohne aktivierte GWT-DFE-Unterstützung aufzeichnen, kann das resultierende Vuser-Skript signifikante Datenmengen enthalten, die kryptisch und deshalb schwer zu entschlüsseln sind. Ein Beispiel finden Sie im nachstehenden Codesegment:

Die kryptische Formatierung der Daten macht es schwierig, Textzeichenfolgen zu erkennen, die für die Korrelation, Parametrierung und Überprüfung verwendet werden sollen.

Hinweis: Das Vorhandensein zahlreicher Pipes in den aufgezeichneten Daten weist darauf hin, dass die aufgezeichnete Website möglicherweise eine GWT-basierte Website ist, die den GWT-RPC-Mechanismus verwendet.

Wenn Sie GWT-DFE-Unterstützung aktivieren, kann VuGen einen Großteil der komplexen Daten in den HTTP-Antworten und -Anforderungen decodieren. Dadurch kann VuGen Vuser-Skripts generieren, die Daten im XML-Format enthalten. Darüber hinaus enthalten die ursprünglichen codierten Daten nur Werte ohne die zugehörigen Namen der Datenfelder. Nach dem Anwenden der GWT-DFE enthalten die resultierenden XML-formatierten Daten sowohl die Namen als auch die Werte der Datenfelder. So lassen sich die XML-formatierten Daten in den Skripts einfacher entschlüsseln. Dadurch können die Skripts leichter korreliert, parametriert und für Überprüfungszwecke verwendet werden. Die folgende Tabelle zeigt ein Codebeispiel, das vor und nach dem Aktivieren der GWT-DFE-Unterstützung generiert wurde.

Codebeispiel, das mit und ohne GWT-DFE-Unterstützung generiert wurde:

Ursprüngliches Skript - ohne GWT-DFE-Unterstützung
6|0|11|http://localhost:8081/MyTestApp/testapp/|624C899BB846618A2E7F49092 8212946|com.test.client.GreetingService|greetServeCompAns|com.test.client.ComplexObject/198661839 |GWT User|inside object|java.util.HashSet/1594477813|java.lang.String/2004016611|add string1| string2|1|2|3|4|1|5|5|1001|1999|6|5|321|1234|7|0|8|0|8|2|9|10|9|11|

Skript nach dem Anwenden der GWT-DFE-Unterstützung

```
<HP_EXTENSION name="GWT_DFE_1">
<com.hp.dfe.GWT__Request>
<moduleBaseURL>http://localhost:8081/MyTestApp/testapp/</moduleBaseURL>
<rpcRequest>
<flags>0</flags>
<Methode>
<class>com.test.client.GreetingService</class>
<name>greetServeCompAns</name>
<parameter-types>
<class>com.test.client.ComplexObject</class>
</parameter-types>
</method>
<Parameter>
<com.test.client.ComplexObject>
<anIntField>1001</anIntField>
<anotherIntField>1999</anotherIntField>
<name>GWT User</name>
<objectInComposingField>
<anIntField>321</anIntField>
<anotherIntField>1234</anotherIntField>
<name>inside object</name>
<stringsSet/>
</objectInComposingField>
<stringsSet>
<string>add string2</string>
<string>add string1</string>
</stringsSet>
</com.test.client.ComplexObject>
</parameters>
<serializationPolicy class="com.google.gwt.user.server.rpc.impl.StandardSerial
izationPolicy"/>
```

Skript nach dem Anwenden der GWT-DFE-Unterstützung
</rpcRequest> </com.hp.dfe.GWT__Request> </HP_EXTENSION>

Damit VuGen in der Lage ist, die komplexen Daten in der HTTP-Kommunikation zu decodieren, müssen Sie die *.war*-Datei angeben, die von der Webanwendung verwendet wird. Die *.war*-Datei enthält die Logik, die von GWT zum Codieren und Decodieren der Informationen in der HTTP-Kommunikation verwendet wird. VuGen benötigt Zugriff auf die *.war*-Datei, um ähnliche Codierungs- und Decodierungsverfahren durchführen zu können. Üblicherweise befinden sich diese *.war*-Dateien auf dem Anwendungsserver im Ordner für die Webanwendungen.

Hinweis: Stellen Sie sicher, dass die *.war*-Datei, die Sie mit dem Vuser-Skript verknüpfen, die aktuelle *.war*-Datei für Ihre Anwendung darstellt. Die **.war**-Datei wird jedes Mal geändert, wenn Sie Änderungen an der Webanwendung vornehmen. Die GWT-DFE-Unterstützung funktioniert nur ordnungsgemäß, wenn die aktuelle *.war*-Datei verfügbar ist.

- Eine Einführung zur Verwendung von DFEs in Vuser-Skripts finden Sie unter "Datenformaterweiterungen (DFEs) - Übersicht" auf Seite 912.
- Weitere Informationen zum Implementieren der DFE-Unterstützung finden Sie unter "Implementieren der Unterstützung für Datenformaterweiterungen (DFEs)" auf Seite 916.
- Eine vollständige Liste der standardmäßigen LoadRunner-DFEs finden Sie unter "Liste der Datenformaterweiterungen" auf Seite 922.

Hinweis: Die GWT-Datenformaterweiterung stellt eine automatische Lösung für GWT-spezifische Korrelationen (STRONG_NAME_HEADER) bereit.

- **Automatische Erkennung von GWT-RPCs (Remote Procedure Calls):**

  Wenn VuGen ein Vuser-Skript generiert oder erneut generiert, überprüft VuGen die HTTP-Kopfzeilen in den Anforderungen, die an den Server gesendet werden. Wenn VuGen sowohl eine Textzeichenfolge **x-gwt-module-base** als auch eine Textzeichenfolge **x-gwt-permutation** in einer dieser HTTP-Kopfzeilen erkennt, wird auf der Registerkarte für VuGen-Fehler eine Warnung angezeigt. Die Warnung empfiehlt, die GWT-DFE für das Vuser-Skript zu aktivieren.

Hinweis: VuGen gibt die obige Warnung weiterhin aus, und zwar bei jeder Generierung oder Neugenerierung des Skripts, bis die GWT-DFE aktiviert ist.

Implementieren der GWT-DFE-Unterstützung

Hinweis: Dieser Abschnitt gilt für Web (HTTP/HTML)- und Silverlight-Vuser-Skripts sowie für Web (HTTP/HTML)-Schritte in Flex-Vuser-Skripts.

Dieser Abschnitt enthält Informationen, die speziell für die Implementierung der GWT-DFE-Unterstützung gelten. Verwenden Sie diese Informationen zusätzlich zu den allgemeinen Informationen zur Implementierung der DFE-Unterstützung, wie unter "Implementieren der Unterstützung für Datenformaterweiterungen (DFEs)" auf Seite 916 beschrieben.

Voraussetzungen für die Implementierung der GWT-DFE-Unterstützung

LoadRunner enthält JRE 1.5. Viele Anwendungen wurden jedoch für JVM 1.6 und höher kompiliert. Wenn Ihre Anwendung mit JDK 1.6 oder höher kompiliert wurde, ersetzen Sie den Ordner **<LoadRunner>\jre** durch **<Programme>\Java\jre6**, bevor Sie eine GWT-Anwendung aufzeichnen.

Aufzeichnen von GWT-DFE-Kopfzeilen

Im Rahmen des Implementierungsprozesses der GWT-DFE-Unterstützung müssen Sie festlegen, dass VuGen **x-gwt-permutation**-Kopfzeilen beim Aufzeichnen von Geschäftsprozessen aufzeichnet. Dieses im Folgenden beschriebene Verfahren kann auf jeder Stufe des Implementierungsprozesses für die GWT-DFE-Unterstützung durchgeführt werden.

1. Erstellen Sie ein Skript mit dem Webprotokoll (HTTP/HTML).
2. Wählen Sie **Aufzeichnen > Aufzeichnungsoptionen > HTTP-Eigenschaften > Erweitert** und klicken Sie dann auf **Kopfzeilen**.
3. Wählen Sie im Dialogfeld **Kopfzeilen** den Eintrag **Kopfzeilen aus Liste aufzeichnen**.
4. Aktivieren Sie in der Liste **Kopfzeilen** das Kontrollkästchen **x-gwt-permutation header**.

Anwenden von GWT-DFE-Ketten

Wenn Sie eine beliebige DFE-Unterstützung implementieren, müssen Sie die DFE-Ketten auf bestimmte Abschnitte der HTTP-Kommunikation anwenden. Der grundlegende Prozess wird unter "Anwenden von DFE-Ketten auf Abschnitte der HTTP-Nachricht" auf Seite 920 beschrieben. Dieser Abschnitt enthält Informationen, die beim Zuordnen von Ketten während der Implementierung der GWT-DFE-Unterstützung erforderlich sind. Wenn Sie die Ketten während der Implementierung der GWT-DFE-Unterstützung zuordnen, müssen Sie die Klassenpfadeinträge angeben, die mit der Anwendung verknüpft sind, die von den Vusern betrieben wird. Damit Sie die Klassenpfadeinträge zuordnen können, benötigen Sie Zugriff auf den GWT-WAR-Ordner, der von Ihrem Entwicklerteam verwendet wird. Der WAR-Ordner enthält die folgenden Dateitypen:

- *.gwt.rpc-Dateien
- *.jar-Dateien
- *.class-Dateien

Festlegen der Klassenpfadeinträge

1. Wählen Sie **Aufzeichnen > Aufzeichnungsoptionen > Datenformaterweiterung > Kettenkonfiguration**.
2. Klicken Sie unter **Ketten** auf , um eine neue DFE-Kette zu erstellen und zu benennen.
3. Klicken Sie auf .
4. Wählen Sie im Dialogfeld **Datenformaterweiterung hinzufügen** den Eintrag **GWT-Erweiterung** und klicken Sie auf **OK**.
5. Legen Sie im Dialogfeld **GWT zu Kette hinzufügen** die Klassenpfadeinträge fest:
   a. Wenn die Klassenpfadeinträge in einer einzelnen *.war*-Datei enthalten sind, klicken Sie auf und legen anschließend den Speicherort der *.war*-Datei fest.

   b. Wenn die Klassenpfadeinträge nicht in einer einzelnen *.war*-Datei enthalten sind:

   - Klicken Sie auf , um den Ordner mit den .gwt.rpc-Dateien hinzuzufügen.

   - Klicken Sie auf , um den Anwendungsordner **classes** hinzuzufügen.

   - Klicken Sie auf , um die **JAR**-Anwendungsdateien aus dem Ordner **WEB-INF\lib** hinzuzufügen.

   Wenn die WAR-Datei auf dem Server extrahiert wird, ergibt sich die folgende Ordnerstruktur:

   <Serververzeichnis>\<Anwendungsverzeichnis>\<GWT-Anwendung>\<Verzeichnis mit .gwt.rpc-Datei>

   WEB-INF\classes

   WEB-INF\lib\gwt-servlet.jar

WEB-INF\lib\gwt-servlet-deps.jar

WEB-INF\lib\log4j.jar

WEB-INF\lib\<Weitere AUT-bezogene JAR-Datei>.jar

Hinweis: Wenn der Speicherort der Klassenpfadeinträge auf dem Computer, auf dem das Skript aufgezeichnet wurde, und auf dem Computer, auf dem das Skript wiedergegeben wird, nicht identisch ist, müssen Sie die Laufzeiteinstellungen für das Skript ändern. Wählen Sie **Wiedergeben > Laufzeiteinstellungen > Datenformaterweiterung > Kettenkonfiguration**, und legen Sie den Speicherort der Klassenpfadeinträge auf dem Computer fest, auf dem das Skript wiedergegeben wird.

6. Wählen Sie **Aufzeichnungsoptionen > Datenformaterweiterung > Codegenerierung** aus.
7. Aktivieren Sie das Kontrollkästchen **Datenformaterweiterung aktivieren**.
8. Wählen Sie unter **Konfiguration** die Option **Code und Snapshots** aus der Liste **Format**.
9. Wählen Sie unter **Kettenzuweisung** die Option **Text** aus und weisen Sie die Kette zu. Wählen Sie **Kopfzeilen** aus und weisen Sie die gleiche Kette zu.

Fehlerbehebung - Datenformaterweiterung (DFE)

In diesem Abschnitt werden Aufgaben zur Fehlerbehebung für Vuser-Skripts beschrieben, die DFE-Funktionen enthalten.

Replay log: Warning -27040: Data Format Extension: Convert: Empty string returned from extension {Extension name}

Wenn Sie ein Vuser-Skript mit DFE-Funktionalität wiedergeben, werden verschiedene Meldungen zum Wiedergabeprotokoll im VuGen-Ausgabebereich hinzugefügt. Mit der obigen Meldung wird angegeben, dass bei der Wiedergabe des Vuser-Skripts das Ergebnis des angegebenen Schrittes **web_convert_from_formatted** im Skript eine leere Zeichenfolge war. Bei bestimmten Datenformaterweiterungen ist die Rückgabe einer leeren Zeichenfolge aus einem Schritt **web_convert_from_formatted** das ordnungsgemäße Verhalten. Wenn jedoch das Vuser-Skript GWT-DFE-Funktionen enthält, kann die obige Meldung auf einen oder beide der folgenden Punkte hinweisen:

- Einige der erforderlichen Klassenpfaddateien sind nicht in den Laufzeiteinstellungen für das Vuser-Skript enthalten.
- Einige der erforderlichen Klassenpfaddateien sind nicht am angegebenen Speicherort auf dem Lastgenerator vorhanden.

Weitere Informationen zum Beheben dieser Probleme finden Sie unter "Implementieren der GWT-DFE-Unterstützung" auf Seite 928.

Wenn Sie Ihre eigene Version von Datenformaterweiterung implementiert haben, wurde die Definition von "class HTTPEntity" in der Datei **DfeDefinitions.h** in LoadRunner 11.50 aktualisiert. Es ist keine Codeänderung erforderlich, aber sämtliche Datenformaterweiterungen müssen erneut kompiliert werden.

Webdienste

Webdienste - Hinzufügen von Skriptinhalt

Übersicht über Webdiensttests

SOA-Systeme basieren auf Webdiensten, eigenständigen Anwendungen, die im Internet auf einer großen Bandbreite von Plattformen ausgeführt werden können. Diese Dienste werden in XML (Extensible Markup Language) entwickelt und nutzen das SOA-Protokoll (Simple Object Access Protocol). Sie sind modular aufgebaut und ermöglichen dadurch die schnelle Entwicklung und Bereitstellung von neuen Anwendungen.

Mit VuGen können Sie Vuser-Skripts zum Testen Ihrer SOA-Umgebung erstellen. Sie können entweder einen Assistenten für die automatische Generierung von Skripts verwenden oder die Skripts manuell erstellen.

Hinzufügen von Webdienst-Skriptinhalt - Übersicht

Mit Webdienstskripts können Sie Ihre Umgebung durch Emulierung von Webdienstclients testen.

Nach dem Erstellen eines leeren Webdienstskripts (wie unter "Dialogfeld "Neues Skript erstellen"" auf Seite 157 beschrieben) können Sie auf eine der folgenden Weisen Inhalt hinzufügen: Aufzeichnen, manuelles Einfügen von Webdienstaufrufen, Importieren von SOAP oder Analysieren von Serververkehr.

Aufzeichnen eines Webdienstskripts

Bei der Aufzeichnung einer Webdienstsitzung werden die Ereignisse eines typischen Geschäftsprozesses erfasst. Wenn Sie bereits einen Client für die Interaktion mit dem Webdienst erstellt haben, können Sie sämtliche Aktionen aufzeichnen, die dieser Client ausführt. Das dabei erstellte Skript emuliert die Operationen des Webdienstclients. Nach der Aufzeichnung können Sie zusätzlich Webdienstaufrufe hinzufügen und andere Erweiterungen für das Skript vornehmen.

Das Aufzeichnen einer Anwendung kann mit oder ohne Verwendung einer Webdienst-WSDL-Datei ausgeführt werden. Wenn Sie eine WSDL-Datei einbeziehen, können Sie in VuGen ein Skript erstellen, indem Sie die gewünschten Methoden auswählen und Werte für die zugehörigen

Argumente eingeben. VuGen erstellt ein beschreibendes Skript, das bei Änderungen in der WSDL aktualisiert werden kann.

Wenn Sie ein Skript aufzeichnen, ohne zuvor einen Dienst zu importieren (dies wird nicht empfohlen), erstellt VuGen anstatt von Schritten für Webdienstaufrufe einen Test mit SOAP-Anfragen. SOAP-Anforderungsargumente sind weniger intuitiv und schwerer zu verwalten.

Weitere Informationen finden Sie unter "Hinzufügen von Inhalt" auf Seite 941.

Hinzufügen neuer Webdienstaufrufe

Sie können ein Webdienstskript erstellen, indem Sie manuell Webdienstaufrufe hinzufügen. Sie entwerfen den Aufruf basierend auf einem Vorgang, dem Transport, Argumenten und anderen Eigenschaften.

Weitere Informationen finden Sie unter "Hinzufügen von Inhalt" auf Seite 941.

Importieren von SOAP-Anforderungen

VuGen ermöglicht Ihnen das Erstellen von Webdienstaufrufen auf der Grundlage von SOAP-Dateien. Sie können SOAP-Anfragedateien direkt in ein Skript laden. VuGen importiert die gesamte SOAP-Anforderung (mit Ausnahme der Sicherheitskopfzeilen) mit den Argumentwerten, so wie sie in den XML-Elementen definiert wurden. Durch Importieren der SOAP-Anfrage entfällt das manuelle Festlegen von Argumentenwerten, wie es bei standardmäßigen Webdienstaufrufen der Fall ist.

Angenommen, es liegt eine SOAP-Anforderung mit den folgenden Elementen vor:

```
- <soap:Body soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  - <q1:AddAddr xmlns:q1="http://tempuri.org/AddrBook/message/">
            <Addr href="#id1" />
    </q1:AddAddr>
   - <q2:Addr id="id1" xsi:type="q2:Addr" xmlns:q2="http://tempuri.org/AddrBook/
type/">
        <name xsi:type="xsd:string">Tom Smith</name>
             <street xsi:type="xsd:string">15 Elm Street</street>
             <city xsi:type="xsd:string">Pheonix</city>
            <state xsi:type="xsd:string">AZ</state>
            <zip-code xsi:type="xsd:string">97432</zip-code>
            <phone-numbers href="#id2" />
            <birthday xsi:type="xsd:date">1983-04-22</birthday>
     </q2:Addr>
...
```

Beim Import der SOAP-Anforderung importiert VuGen alle Werte in den Webdienstaufruf. Um die Werte im Schrittenavigator anzuzeigen, klicken Sie mit der rechten Maustaste auf den gewünschten Schritt und klicken Sie dann auf **Argumente anzeigen**.

Um einen neuen Webdienstaufruf auf Grundlage einer SOAP-Anfrage zu erstellen, müssen Sie zunächst eine WSDL-Datei importieren. Wenn keine WSDL-Datei verfügbar ist oder Sie den SOAP-Datenverkehr direkt übermitteln möchten, können Sie einen SOAP-Anfrageschritt erstellen. Sie geben hierzu den URL des Servers, die SOAP-Aktion und den Antwortparameter an.

Im Editor wird der SOAP-Anforderungsschritt als Funktion vom Typ **soap_request** angezeigt. Eine Beschreibung finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Weitere Informationen finden Sie unter "Hinzufügen von Inhalt" auf Seite 941.

Analysieren von Serververkehr

Das Hauptaugenmerk liegt beim Testen von Unternehmensumgebungen und anderen komplexen Systemen auf der Messung des Leistungsverhaltens auf der Clientseite. Normalerweise erfasst VuGen die Aktionen, die in Anwendungen oder Browsern ausgeführt werden, und generiert daraus ein Skript, das die Clientaktionen und Serveranfragen emuliert.

In bestimmten Testumgebungen ist es nicht möglich, die Aktionen der Clientanwendung aufzuzeichnen und daraus die Serveranfragen abzurufen. Dies kann daran liegen, dass der Server als Client fungiert, oder weil Sie keinen Zugriff auf die Clientanwendung haben. In diesen Fällen können Sie ein Skript mithilfe der VuGen-Funktion **Datenverkehr analysieren** erstellen.

Die Funktion **Datenverkehr analysieren** analysiert eine Sammlungsdatei für den Server-Netzwerkverkehr und erstellt ein Skript zur Emulation der Anfragen, die an bzw. von dem Server gesendet wurden.

Weitere Informationen finden Sie unter "Erstellen eines Skripts durch Analysieren des Datenverkehrs" auf Seite 945.

Skriptintegration

Mit dem fertig gestellten Skript können Sie Ihr System auf unterschiedliche Weise testen:

- **Funktionstests.** Führen Sie das Skript aus, um zu prüfen, ob die Webdienste ordnungsgemäß funktionieren. Sie können darüber hinaus prüfen, ob der Webdienst die erwarteten Werte generiert. Weitere Informationen finden Sie unter "Webdienste - Vorbereiten von Skripts für die Wiedergabe" auf Seite 967.
- **Lasttests.** Integrieren Sie das Skript in ein LoadRunner Controller-Szenario, um die Leistungsfähigkeit des Systems unter Last zu testen. Weitere Informationen finden Sie in der Dokumentation zu *HP Controller*.
- **Produktionstests.** Überprüfen Sie das Leistungsverhalten Ihres Webdiensts im zeitlichen Verlauf mithilfe einer Business Process Monitor-Konfiguration. Weitere Informationen finden Sie in der Dokumentation zu *HP Business Process Monitor*.

Webdienstaufruf - Anhänge

Wenn Sie binäre Dateien wie Bilder über SOAP übertragen, müssen die Daten in XML serialisiert sein. Die Serialisierung und Deserialisierung kann zu einer erheblichen Überlastung führen. Aus diesem Grund ist es üblich, große binäre Dateien in Form von Anhängen zu senden. Auf diese Weise bleiben die binären Daten intakt und die Belastung durch die Analyse wird vermindert.

Durch die Verwendung von Anhängen werden die ursprünglichen Daten außerhalb des SOAP-Umschlages gesendet, so dass die Serialisierung der Daten in XML überflüssig und die Datenübertragung effizienter ist.

Zum Weiterleiten einer SOAP-Meldung zusammen mit binären Daten werden die Spezifikationen des Formats MIME (Multipurpose Internet Mail Extensions) und das aktuellere und damit effizientere Format DIME (Direct Internet Message Encapsulation) verwendet. VuGen unterstützt DIME für alle Toolkits, MIME allerdings nur für das Toolkit Axis. Weitere Informationen zur Verwendung von MIME-Anhängen für das Toolkit .NET finden Sie unter "Beispiele für Benutzerhandler" auf Seite 981.

VuGen unterstützt das Senden und Empfangen von Anhängen mit SOAP-Meldungen. Sie können Anhänge mit Eingabeargumenten (Anfrage) senden oder Anhänge mit Ausgabeargumenten (Antwort) speichern. Weitere Informationen über die Aufgabe finden Sie unter "Hinzufügen von Inhalt" auf Seite 941.

Die Ausgabeanhänge dienen dazu, die Antwort als Anhang zu speichern. In diesem Fall haben Sie folgende Optionen: **Alle Anhänge speichern** oder **Anhänge nach Index speichern**.

Wenn Sie die Option **Alle Anhänge speichern** auswählen, erstellt VuGen drei Parameter für jeden Anhang, basierend auf dem von Ihnen angegebenen Parameternamen: Ein Parameter mit den Daten des Anhangs, der Inhaltstyp des Anhangs und eine eindeutige ID für den Anhang.

Wenn Sie beispielsweise im Feld **Inhalt** den Namen **MyParam** angeben, lauten die Parameternamen für den ersten Anhang wie folgt:

```
MyParam_1
MyParam_1_ContentType
MyParam_1_ContentID
```

Wenn Sie **Anhänge nach Index speichern** angeben, legen Sie die Indexnummer und den Namen des Parameters fest, in dem der Anhang gespeichert werden soll. Der Parametername, den Sie für den **Inhalt** angegeben haben, wird als Präfix für den Inhaltstyp und die Parameter der Inhalts-ID verwendet.

Spezielle Argumenttypen

VuGen verarbeitet spezielle Argumenttypen wie abgeleitete, rekursive, Choice- und optionale Elemente.

Abgeleitete Typen

VuGen unterstützt WSDLs mit abgeleiteten Typen. Wenn Sie die Eigenschaften für einen Webdienstaufruf festlegen, ermöglicht es Ihnen VuGen, als Argument den Basistyp oder den abgeleiteten Typ zu verwenden. Im Anschluss an die Auswahl eines Typs aktualisiert VuGen den Knoten der hierarchischen Argumentstruktur mit dem neuen Typen. Weitere Informationen finden Sie unter "Dialogfeld "Neuer Webdienstaufruf"" auf Seite 950.

Abstrakte Typen

Abstrakt ist ein vom Programmierer deklarierter Typ. Wenn ein Element oder ein Typ als **abstrakt** deklariert wird, ist eine Verwendung als Instanzdokument nicht möglich. Stattdessen muss ein Mitglied der im XML-Schema enthaltenen Substitutionsgruppe des Elements in dem Instanzdokument aufgeführt sein. In einem solchen Fall müssen alle Instanzen des Elements **xsi:type** zum Anzeigen eines abgeleiteten Typs verwenden, der nicht abstrakt ist.

Im Falle eines abstrakten Typs kann VuGen keine abstrakte Klasse erstellen und es tritt ein Fehler bei der Wiedergabe auf. VuGen zeigt dann eine Warnmeldung unterhalb des Felds **Typ** an und weist Sie an, den abstrakten Typen durch einen abgeleiteten Typen zu ersetzen.

Optionale Elemente

In den WSDL-Dateien werden die optionalen Parameter durch eines der nachstehenden Attribute definiert:

```
minoccurs='0'
```

```
nillable='true'
```

minoccurs = 0 verweist auf ein echtes, optionales Element, das ausgelassen werden kann. Nillable bedeutet, dass das Element ohne seinen gewöhnlichen Inhalt präsentiert werden kann, sofern für das Nillable-Attribut der Wert **true** oder **1** festgelegt wurde. Für die Attribute **minOccurs** und **maxOccurs** wird standardmäßig der Wert **1** festgelegt.

Im folgenden Beispiel ist **name** erforderlich, **age** optional und **phone** kann NULL sein.

```
<s:element minOccurs="1" name="name" type="s:string" />
<s:element minOccurs="0" name="age" type="s:int" />
<s:element minOccurs="1" name="phone" nillable="true" type="s:string" />
```

Die nachstehende Tabelle enthält die Verfügbarkeit der Optionen:

Parametertyp	Optionsschaltfläche für den Nullwert	Argument im Aufruf berücksichtigen
Erforderlich	Deaktiviert	Deaktiviert
MinOccurs=0	Deaktiviert	Aktiviert
Nillable	Aktiviert	Deaktiviert

Um ein spezifisches, optionales Argument im Dienstaufruf einzuschließen, klicken Sie auf den Knoten und wählen **Argument im Aufruf berücksichtigen** aus. Die Knoten für die eingeschlossenen Argumente erscheinen in Blau. Nicht eingeschlossene Argumente erscheinen in Grau.

Wenn Sie ein Element auf einer übergeordneten Ebene einschließen, werden automatisch alle dieser Ebene untergeordneten erforderlichen Nillable-Elemente mit eingeschlossen. Bei einem untergeordneten Element werden automatisch das übergeordnete Element sowie alle erforderlichen Elemente bzw. Nillable-Elemente auf dieser Ebene eingeschlossen. Wenn Sie in einem übergeordneten Element **Automatische Werte generieren** definieren, stellt VuGen für diese untergeordneten Elemente Werte zur Verfügung, die im übergeordneten Element eingeschlossen sind.

Hinweis: VuGen stellt über die Toolkit-Implementierung fest, ob die Elemente erforderlich oder optional sind. Dies ist u. U. nicht immer mit den Attributen des Elements in der WSDL-Datei konsistent.

Optionale Choice-Elemente

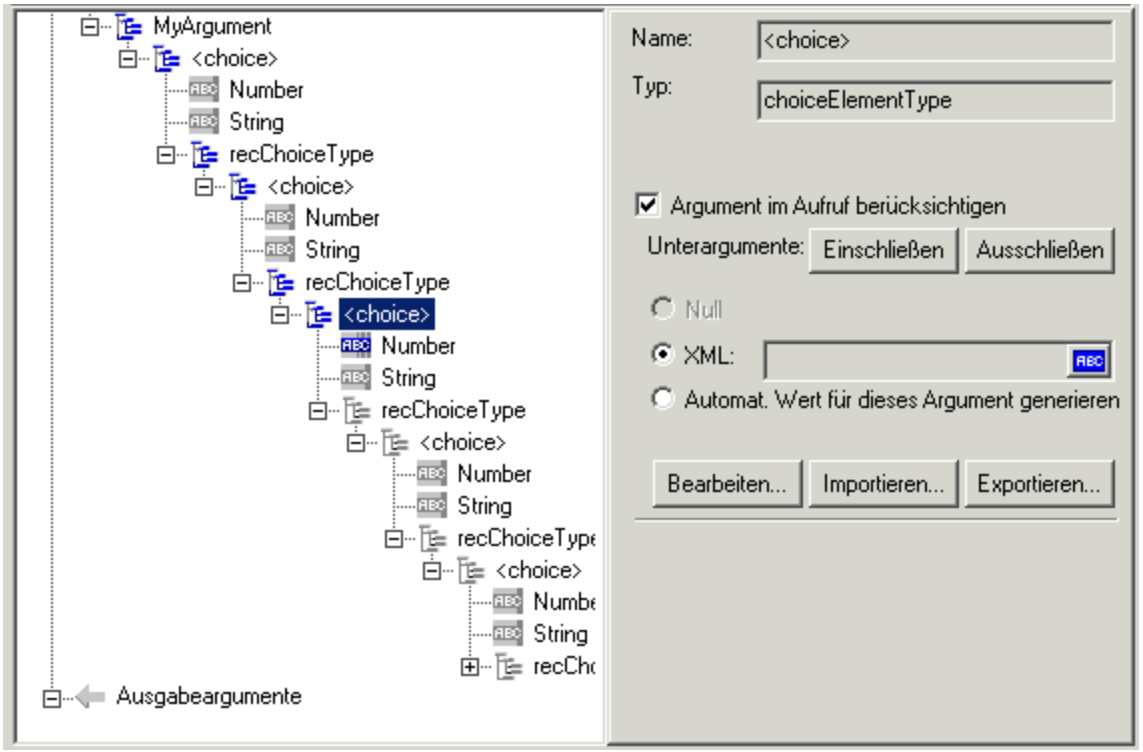
Ein Choice-Element in einer WSDL definiert einen Satz mit Elementen, von denen nur eines in der SOAP-Meldung erscheint. In einigen Fällen ist eines der Choice-Elemente optional, während die anderen es nicht sind. Sie können Sie ein Choice-Element auswählen und verhindern, dass das entsprechende optionale Element im SOAP-Umschlag erscheint. Wählen Sie im **Schrittenavigator** ein Choice-Element und heben Sie die Auswahl der Option **Argument im Aufruf berücksichtigen** auf. Löschen Sie in der Skriptansicht die Zeile, die das Choice-Argument definiert.

Rekursive Elemente

Mit dem Dialogfeld **Eigenschaften** können Sie die Ebene der rekursiven Elemente steuern, die im Webdienstaufruf eingeschlossen werden sollen.

Um eine bestimmte Ebene sowie die darunter liegenden Ebenen auszuschließen, wählen Sie den untersten übergeordneten Knoten, den Sie einschließen möchten, und wählen **Argument im Aufruf berücksichtigen** aus. VuGen schließt die ausgewählten Knoten sowie die erforderlichen untergeordneten und alle übergeordneten Knoten ein.

Im folgenden Beispiel sind drei Ebenen mit Choice-Argumenten eingeschlossen, während die anderen Knoten unberücksichtigt bleiben. Ausgeschlossene Knoten werden grau unterlegt angezeigt.



Base64-Argumente

Bei der Base 64-Codierung handelt es sich um eine Codierungsmethode, die zum Darstellen binärer Daten als ASCII-Text verwendet wird. Da es sich bei SOAP-Umschlägen um Volltext handelt, können Sie diese Codierung zum Darstellen von binären Daten als Text innerhalb der SOAP-Umschläge verwenden.

Wenn VuGen ein WSDL-Element vom Typ **base64Binary** erkennt, können Sie einen codierten Wert eingeben. Beim Angeben eines Werts haben Sie die Wahl zwischen zwei Möglichkeiten:

- **Aus Datei beziehen.** Referenzieren Sie einen Dateinamen.
- **Verschlüsselten Text einbetten.** Geben Sie den zu codierenden Text an.

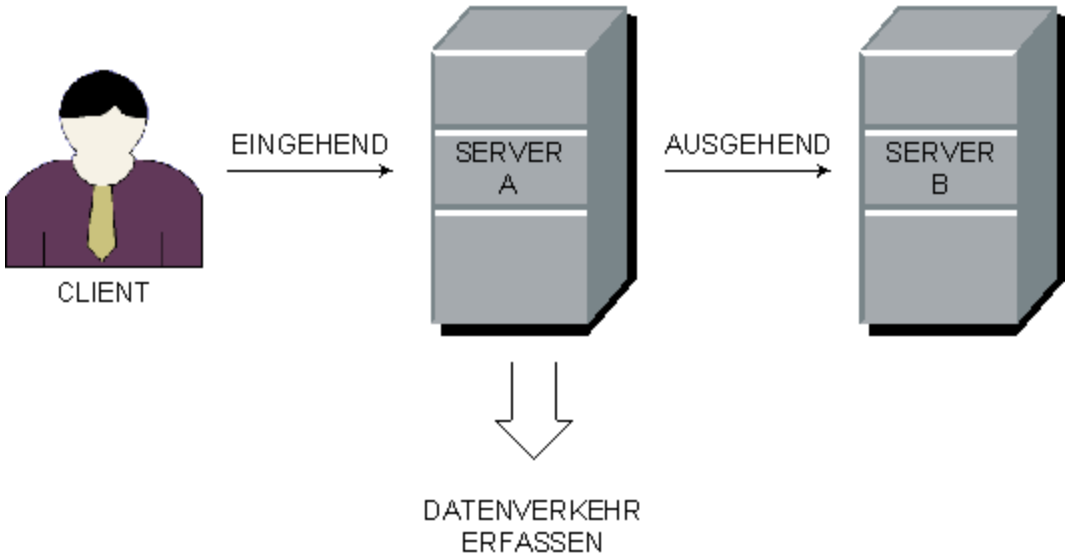
Weitere Informationen finden Sie unter "Dialogfeld "Base64-Daten verarbeiten - Einfache Daten"" auf Seite 960.

Übersicht über Serververkehrskripts

Das Hauptaugenmerk liegt beim Testen von Unternehmensumgebungen und anderen komplexen Systemen auf der Messung des Leistungsverhaltens auf der Clientseite. Normalerweise erfasst VuGen die Aktionen, die in Anwendungen oder Browsern ausgeführt werden, und generiert daraus ein Skript, das die Clientaktionen und Serveranfragen emuliert.

In bestimmten Testumgebungen ist es nicht möglich, die Aktionen der Clientanwendung aufzuzeichnen und daraus die Serveranfragen abzurufen. Dies kann daran liegen, dass der Server als Client fungiert, oder weil Sie keinen Zugriff auf die Clientanwendung haben. In diesen Fällen können Sie ein Skript mithilfe der VuGen-Funktion **Datenverkehr analysieren** erstellen.

Die Funktion **Datenverkehr analysieren** analysiert eine Sammlungsdatei für den Server-Netzwerkverkehr und erstellt ein Skript zur Emulation der Anfragen, die an bzw. von dem Server gesendet wurden. Die Schritte zum Erstellen eines Skripts durch Analyse des Serververkehrs werden unter "Erstellen eines Skripts durch Analysieren des Datenverkehrs" auf Seite 945 beschrieben.



Es gibt zwei Typen von Emulationen: **Eingehender Datenverkehr** und **Ausgehender Datenverkehr**.

Skripts für **eingehenden Verkehr** emulieren Situationen, in denen Anfragen an den Server gesendet werden, aber kein Zugriff auf die Clientanwendung besteht, beispielsweise aufgrund von

Sicherheitseinschränkungen. Die exakteste Lösung in einem solchen Fall besteht darin, ein Skript auf Grundlage des Verkehrs zu generieren, der von der Clientseite **an** den Server übermittelt wird.

Bei eingehendem Servernetzwerkverkehr geben Sie die IP-Adresse des Servers und die Portnummer der Anwendung an. VuGen analysiert den gesamten an den Server gerichteten Datenverkehr, extrahiert die relevanten Meldungen und generiert ein Skript. Im Schaubild oben können Sie, falls der Client nicht verfügbar ist, ein Skript für eingehenden Verkehr erstellen, um die Anfragen zu emulieren, die bei **Server A** eingehen.

Skripts für **ausgehenden Verkehr** emulieren Server, die als Client für einen anderen Server fungieren. Bei einem Anwendungsserver mit einer Reihe von internen Servern empfiehlt es sich, die Kommunikation zwischen den Servercomputern zu emulieren, z. B. zwischen **Server A** und **Server B** im Schaubild oben. Die Lösung besteht in diesem Fall darin, ein Skript auf Grundlage des Verkehrs zu generieren, der als Ausgabe **von** einem bestimmten Server gesendet wird.

Wenn Sie ein Skript für ausgehenden Verkehr erstellen, geben Sie die IP-Adresse des Servers an, dessen ausgehender Verkehr emuliert werden soll. VuGen extrahiert dann den Datenverkehr, der als Ausgabe von diesem Server gesendet wird. Im Schaubild oben könnte ein Skript für ausgehenden Datenverkehr beispielsweise die Anfragen emulieren, die **Server A** an **Server B** übermittelt.

- "Aufzeichnungsdateien" unten
- "Filtern von Datenverkehr" auf der nächsten Seite
- "Daten auf sicheren Servern" auf Seite 940

Aufzeichnungsdateien

Eine Aufzeichnungsdatei ist eine Verfolgungsdatei, in der der gesamte TCP-Verkehr im Netzwerk protokolliert wird. Mithilfe einer Sniffer-Anwendung erhalten Sie ein Abbild des gesamten Netzwerkverkehrs. Der Sniffer zeichnet alle Ereignisse innerhalb des Netzwerks auf und speichert diese in einer Aufzeichnungsdatei.

Um ein kleineres, besser handhabbares Skript zu generieren, versuchen Sie, die Aufzeichnung des Netzwerkverkehrs auf den Zeitraum zu beschränken, in dem Sie Aktionen innerhalb der betreffenden Anwendung ausführen.

Hinweis: Aufzeichnungsdateien enthalten keinen Prüfschleifen-Netzwerkverkehr (Loopback).

Sie können Aufzeichnungsdateien mithilfe des Befehlszeilenprogramms, aber auch mit beliebigen anderen Aufzeichnungstools generieren.

Das VuGen-Befehlszeilenprogramm **lrtcpdump** befindet sich im Ordner **bin** des Produkts. Für jede Betriebssystemplattform ist ein eigenes Befehlszeilenprogramm vorhanden: **lrtcpdump.exe** (Windows), **lrtcpdump.hp9**, **lrtcpdump.ibm**, **lrtcpdump.linux** oder **lrtcpdump.solv4**.

Externe Aufzeichnungstools

Die meisten UNIX-Betriebssysteme verfügen über ein integriertes Aufzeichnungstool. Darüber hinaus gibt es viele Aufzeichnungstools wie **Ethereal/tcpdump**, die zum Download zur Verfügung

stehen.

Achten Sie bei der Verwendung externer Tools darauf, dass sämtliche Paketdaten vollständig aufgezeichnet werden.

Bei einigen Dienstprogrammen müssen zusätzliche Argumente angegeben werden. Bei **tcpdump** muss beispielsweise das Argument **-s 0** angegeben werden, damit die Datenpakete vollständig ohne Trunkierungen aufgezeichnet werden.

Fehlerbehebung für fehlende Pakete

Problem: In Ihrem Skript fehlen Schritte, die Sie in einer Aufzeichnungsdatei aufgezeichnet haben.

Unter **Ausgabebereich > Codegenerierung** wird folgende Warnung angezeigt:

```
Warning: One or more responses are missing or have missing packets. Therefore, a
step may appear to be missing in the script.
This issue can be caused if the recording was stopped before all the responses w
ere received.
If the script is generated from a .pcap file, check if the file has missing pack
ets.
```

Dieser Fehler kann durch unnötige Netzwerkaktivität auf dem aufgezeichneten Computer verursacht werden, was dazu führen kann, dass die aufzeichnende Anwendung Pakete löscht.

Schritte für die Behebung: Stellen Sie sicher, dass auf dem aufzeichnenden Computer im Hintergrund kein unnötiger Netzwerkverkehr stattfindet.

Hinweis: Wenn der Server Last aufweist, die verhindert, dass zeitnah Antworten empfangen werden, können Sie ein Skript erstellen, das auf Anforderungen basiert, indem Sie einfach die Registrierung bearbeiten.

Rufen Sie **regedit** auf, und fügen Sie an folgender Stelle einen DWORD-Schlüssel hinzu:

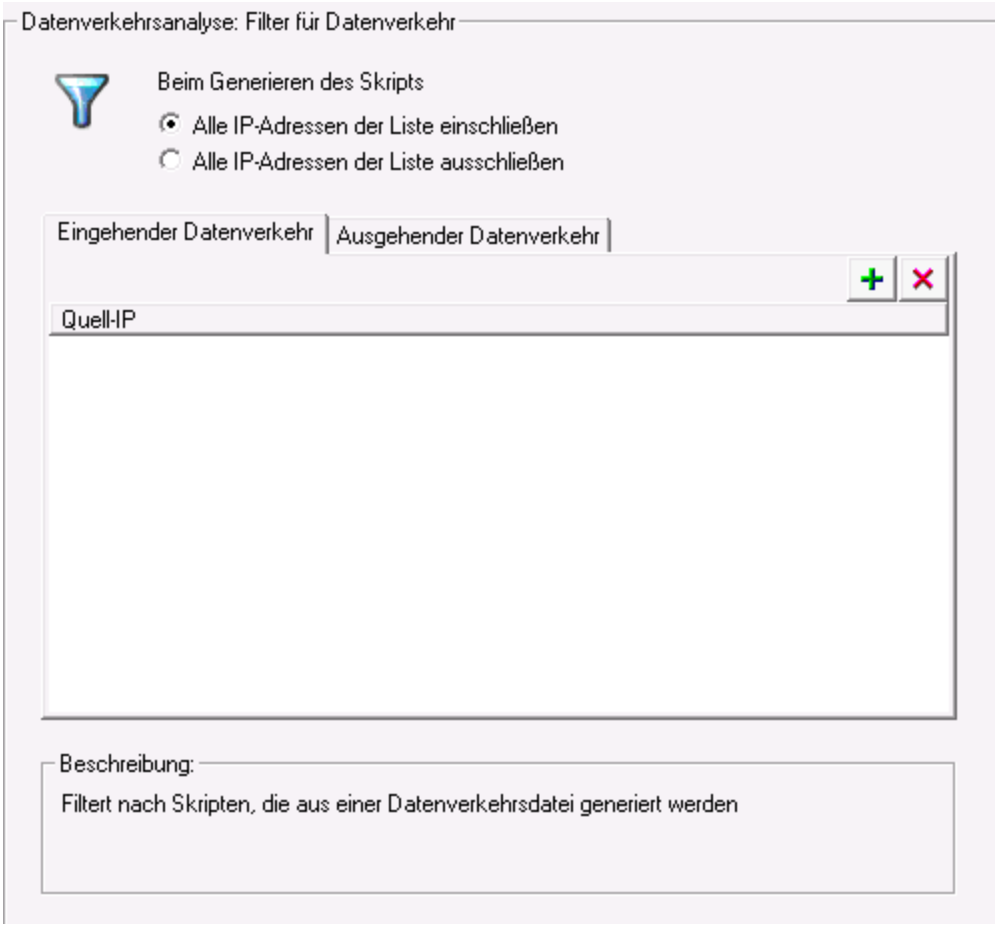
```
Pfad:
Software\\Mercury Interactive\\LoadRunner\\Protocols\\HTTP\\Analyzer\
Schlüssel:
AllowAutomaticOutApiEvents = 1
```

Filtern von Datenverkehr

Sie können einen Filter festlegen, um nur bestimmte Anforderungen im ein- und ausgehenden Datenverkehr eines Servers zu analysieren, indem Sie eine bestimmte IP-Adresse und einen Port angeben.

Tipp: Verschiedene externe Aufzeichnungstools ermöglichen das Filtern der IP-Adressen während der Aufzeichnung des Datenverkehrs.

Sie filtern Anfragen, indem Sie die relevanten Host-IP-Adressen auswählen. Der Filter arbeitet einschließend oder ausschließend, d. h. Sie schließen entweder nur die IP-Adresse in der Liste ein oder Sie schließen die IP-Adressen in der Liste aus (und damit alle anderen ein).



Weitere Informationen finden Sie unter "Erstellen eines Skripts durch Analysieren des Datenverkehrs" auf Seite 945.

Daten auf sicheren Servern

Zur Analyse des Datenverkehrs eines sicheren Servers müssen Sie ein Zertifikat angeben, das den privaten Schlüssel des Servers enthält.

Wenn der Datenverkehr per SSL verschlüsselt wird, müssen Sie für die Entschlüsselung eine Zertifikatsdatei und ein Kennwort bereitstellen. Wenn Sie Verkehr von mehreren Servern in das Skript aufnehmen möchten, müssen Sie für jede IP-Adresse, die per SSL verschlüsselt wird, jeweils ein Zertifikat und Kennwort angeben.

Weitere Informationen finden Sie unter "Erstellen eines Skripts durch Analysieren des Datenverkehrs" auf Seite 945.

Hinzufügen von Inhalt

In dieser Aufgabe wird beschrieben, wie Inhalt, beispielsweise Webdienstaufrufe, zu einem Webdienste-Vuser-Skript hinzugefügt wird.

Voraussetzungen

Erstellen Sie ein leeres Webdienst-Vuser-Skript. Klicken Sie auf **Datei > Neues Skript und neue Lösung** und wählen Sie das Protokoll **Webdienste** aus. Sie können ein Vuser-Skript mit einem einzelnen Protokoll oder ein Skript mit mehreren Protokollen erstellen.

Aufzeichnen einer Sitzung (optional)

1. Klicken Sie auf die Schaltfläche **Aufzeichnen** in der VuGen-Symbolleiste oder drücken Sie Strg+R, um den Aufzeichnungs-Assistenten mit der Seite **Dienste angeben** zu öffnen.

   Informationen zur Benutzeroberfläche finden Sie unter "Bildschirm "Dienste angeben"" auf Seite 948.

2. Fügen Sie Diensten zur Liste hinzu. Klicken Sie auf **Importieren**, um eine WSDL für den Test zu laden. Geben Sie den Speicherort der WSDL-Datei an.

   Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Dienst importieren"" auf Seite 1010.

3. Klicken Sie auf die Schaltfläche **Weiter**. Geben Sie den Speicherort der Anwendung und allen weiteren relevanten Argumenten an. Informationen hierzu finden Sie unter "Dialogfeld "Aufzuzeichnende Anwendung angeben"" auf Seite 948.

Hinzufügen eines neuen Dienstaufrufs (optional)

1. Importieren Sie einen Dienst. Klicken Sie auf **Dienste verwalten**, um auf das Dialogfeld **Importieren** zuzugreifen.

   Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Dienst importieren"" auf Seite 1010.

2. Platzieren Sie den Cursor an der gewünschten Stelle in Ihrem Skript (**Editor**) oder in den Testschritten (**Schrittenavigator**).

3. Klicken Sie auf die Schaltfläche **Dienstaufruf hinzufügen**. Das Dialogfeld **Neuer Webdienstaufruf** wird geöffnet.

4. Wählen Sie im Abschnitt **Webdienstaufruf auswählen** einen Dienst, einen Portnamen und einen Vorgang aus.

5. Um anzugeben, dass der Endpunkt von der standardmäßigen Zieladresse abweicht, wählen Sie **Adresse überschreiben** aus und fügen den neuen Endpunkt ein, an den die Anforderungen übermittelt werden sollen.

6. Erweitern Sie die Knoten und geben Sie die Argumentwerte an. Um Beispielwerte für alle Eingabeargumente zu generieren, wählen Sie den Knoten **Eingabeargumente** aus und klicken Sie auf **Erzeugen**. Informationen zum Bearbeiten, Importieren oder Exportieren der XML-Struktur des Elements finden Sie unter "Zuweisen von Werten zu XML-Elementen" auf der nächsten Seite.

7. Klicken Sie zum Parametrieren eines Eingabeargumentes auf den Knoten und wählen Sie die Option **Wert** aus. Klicken Sie auf das ABC-Symbol und fahren Sie mit der Parametrierung fort. Weitere Informationen finden Sie unter "Parameter" auf Seite 436.

8. Wählen Sie den Knoten **Konfiguration der Transportschicht** aus, um erweiterte Optionen anzugeben, z. B. JMS-Transport für SOAP-Meldungen (nur für das Axis-Toolkit), asynchrones Messaging oder WS-Adressierung. Weitere Informationen finden Sie unter "Senden von Nachrichten über JMS" auf Seite 985.

Hinzufügen von Anhängen (optional)

1. Um einen Anhang zu einem Eingabeargument hinzuzufügen, wählen Sie im linken Bereich einen Vorgang aus. Wählen Sie **Zu Anforderung hinzufügen (Eingabe)** aus. VuGen fordert Sie zur Eingabe von Informationen über den Anhang auf und fügt den Anhang in der Strukturansicht der Methode hinzu. Details finden Sie im "Dialogfeld "Eingabeanhang hinzufügen"" auf Seite 959.

2. Um einen Ausgabeanhang zum Speichern der Ausgabeargumente anzugeben, wählen Sie im linken Bereich einen Vorgang aus. Wählen Sie **Empfangene speichern (Ausgabe)** aus. Wählen Sie die gewünschte Option aus: **Alle Anhänge speichern** oder **Anhänge nach Index speichern** - basierend auf der Indexnummer, beginnend mit 1. Weitere Informationen finden Sie unter "Webdienstaufruf - Anhänge" auf Seite 933.

3. Um die Eigenschaften eines Anhangs mit Eingabe- oder Ausgabeargumenten zu bearbeiten, klicken Sie auf den Anhang im linken Fensterbereich und geben im rechten Fensterbereich die erforderlichen Informationen ein.

Angeben von SOAP-Kopfzeilen (optional)

Wählen Sie den Knoten **Benutzerdefinierte SOAP-Kopfzeile** im linken Bereich aus und aktivieren Sie die Option **SOAP-Kopfzeile verwenden**. Sie müssen für jedes einzelne Element eine SOAP-Kopfzeile angeben. Klicken Sie zum Erstellen einer eigenen Kopfzeile auf **Bearbeiten** und bearbeiten Sie den XML-Code. Klicken Sie zum Importieren einer XML-Datei für die SOAP-Kopfzeile auf **Importieren**.

Importieren von SOAP (optional)

1. Importieren Sie einen Dienst falls vorhanden. Klicken Sie auf **Dienste verwalten**, um auf das Dialogfeld **Importieren** zuzugreifen. Weitere Informationen finden Sie im "Dialogfeld "Dienst importieren"" auf Seite 1010.

2. Klicken Sie auf die Schaltfläche **SOAP importieren**, um das Dialogfeld **SOAP importieren** zu öffnen.

3. Suchen Sie nach der XML-Datei, die die gewünschte SOAP-Anfrage darstellt.

4. Wählen Sie die Art des Schritts aus, der erstellt werden soll: Erstellen eines Webdienstaufrufs oder Erstellen einer SOAP-Anforderung. Zum Erstellen eines Webdienstaufrufs müssen Sie zuerst mindestens eine WSDL-Datei importieren, die die Operation in der SOAP-Anfragedatei beschreibt. Klicken Sie auf **SOAP anzeigen**, wenn Sie die SOAP-Anfrage vor dem Laden anzeigen möchten.

5. Klicken Sie auf **Laden**, um die XML-Elementwerte zu importieren.

   Legen Sie bei einem Webdienstaufruf die Eigenschaften für den Dienstaufruf wie unter "New Web Service Call Dialog Box" beschrieben fest.

   Geben Sie bei einer SOAP-Anforderung den URL sowie andere relevante Parameter an.

6. Wenn bei einem Webdienstaufruf mehrere Dienste mit identischen Vorgangsnamen (Methoden) vorliegen, wählen Sie den Dienst aus, dessen SOAP-Datenverkehr importiert werden soll.

7. Klicken Sie auf **OK**, um den neuen Schritt im Skript zu erstellen.

Analysieren von Serververkehr (optional)

Klicken Sie zum Erstellen eines Skripts durch Analyse einer Datei mit einem Abbild des Serververkehrs auf **Datenverkehr analysieren**.

Weitere Informationen finden Sie unter "Übersicht über Serververkehrskripts" auf Seite 937.

Zuweisen von Werten zu XML-Elementen

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie XML-Elemente verwenden, indem Sie Code manuell bearbeiten, eine externe Datei importieren und den XML-Code zur späteren Verwendung exportieren.

1. ### Voraussetzungen

   Importieren Sie einen Dienst und erstellen Sie einen neuen Webdienstaufruf. Klicken Sie im Schrittenavigator mit der rechten Maustaste auf einen Schritt und wählen Sie **Argumente anzeigen** aus.

2. ### Auswählen des Elements

   Wählen Sie im linken Bereich einen komplexen Typ oder ein Arrayargument aus. Klicken Sie im rechten Bereich auf **XML**. Im Feld **XML** wird der XML-Code als einzelne Zeichenfolge angezeigt.

3. ### Importieren einer Datei - optional

   Um eine zuvor gespeicherte XML-Datei zu importieren, klicken Sie auf **Importieren** und geben

den Dateipfad an.

4. **Bearbeiten der XML-Elemente - optional**

   Klicken Sie zum Bearbeiten der XML-Struktur und -Elementwerte auf **Bearbeiten**. Der XML-Editor wird geöffnet. Klicken Sie zum Speichern einer zuvor gespeicherten XML-Datei auf **Datei importieren**.

   - Um den Code manuell zu bearbeiten, wechseln Sie zur Registerkarte **Textansicht**.
   - Um den XML-Code über eine grafische Benutzeroberfläche zu bearbeiten, klicken Sie auf **Schrittenavigator**. Fügen Sie über das Kontextmenü untergeordnete Elemente und gleichgeordnete Elemente hinzu und benennen Sie den Knoten um. Klicken Sie im Kontextmenü auf **Einfügen**, um ein neues Element vor und nach dem ausgewählten Element hinzuzufügen.

5. **Exportieren einer Datei - optional**

   Um die XML-Daten in einer Datei zu speichern, so dass Sie für weitere Tests verwendet werden können, klicken Sie auf **Exportieren** und geben den Dateipfad an.

Automatisches Erstellen eines Tests

Im Rahmen dieser Aufgabe wird das Erstellen von Anforderungen oder Tests zum Überprüfen Ihres Diensts beschrieben.

1. **Öffnen des Assistenten**

   Wählen Sie **Datei > Neu** aus, um das Dialogfeld **Neuer virtueller Benutzer** zu öffnen. Wählen Sie im linken Bereich **SOA-Testgenerator** aus und klicken Sie auf **Erstellen**.

2. **Hinzufügen eines Diensts**

   Fahren Sie mit dem nächsten Bildschirm fort und klicken Sie auf **Hinzufügen**, um mindestens einen Dienst hinzuzufügen. Wenn Ihr Dienst noch nicht bereit ist, können Sie einen emulierten Dienst verwenden. Weitere Informationen finden Sie unter "Hinzufügen und Verwalten von Diensten" auf Seite 1004. Klicken Sie auf die Schaltfläche **Weiter**.

3. **Auswählen von Testaspekten**

   Erweitern Sie die Knoten und wählen Sie die gewünschten Testaspekte aus. Klicken Sie auf **Weiter**.

4. **Angeben eines Speicherorts**

   Geben Sie einen Testnamen und einen Speicherort für die Testskripts an: **HP ALM** oder ein **lokales Dateisystem**. Wenn Sie ALM angegeben haben, klicken Sie auf **Verbinden**, um sich beim Server anzumelden und auf **Durchsuchen**, um nach dem Testknoten zu suchen.

5. **Abschließen der Testgenerierung**

   Überprüfen Sie die Zusammenfassung und schließen Sie Skripts für die Erstellung ein oder aus. Klicken Sie auf **Generieren**.

6. **Öffnen der Skripts**

   Überprüfen Sie im letzten Bildschirm die Liste der erstellten Skripts und geben Sie an, welche Skripts geöffnet werden sollen. Klicken Sie auf **Fertig stellen**.

Erstellen eines Skripts durch Analysieren des Datenverkehrs

Im Rahmen dieser Aufgabe wird die Erstellung eines Skripts mithilfe einer Datei für den Netzwerkverkehr beschrieben.

1. **Erstellen einer Aufzeichnungsdatei auf einer Windows-Plattform - optional**

   Erstellen Sie eine Aufzeichnungsdatei, in der der gesamte TCP-Verkehr im Netzwerk auf einer Windows-Plattform protokolliert wird. Verwenden Sie ein Aufzeichnungstool (als Download verfügbar) oder das im **bin**-Ordner des Produkts bereitgestellte Tool **lrtcpdump**.<*Plattform*>.**Hinweis:** Sie müssen WinPcap vor der Verwendung dieses Befehls installieren. Weitere Informationen finden Sie unter "Weitere Komponenten" auf Seite 1108.

   a. Führen Sie das Dienstprogramm in einem Befehlsfenster **lrtcpdump -f<***Dateiname***>.cap** aus. **lrtcpdump** fordert Sie auf, eine Netzwerkkarte auszuwählen. Verwendung unter Windows: **lrtcpdump.exe -i interface -f filename**. Sie müssen einen Dateinamen angeben. Wenn Sie keine Schnittstelle angeben, werden Sie aufgefordert eine auszuwählen.

   b. Geben Sie die Nummer der gewünschten Schnittstellenkarte ein (falls mehrere vorhanden) und drücken Sie die EINGABETASTE.

   c. Führen Sie typische Aktionen innerhalb Ihrer Anwendung aus.

   d. Wechseln Sie wieder zur Eingabeaufforderung zurück und drücken Sie erneut die EINGABETASTE, um die Aufzeichnung zu beenden.

   e. Legen Sie die Aufzeichnungsdatei im Netzwerk an einem Speicherort ab, auf den von VuGen aus zugegriffen werden kann.

2. **Erstellen einer Aufzeichnungsdatei auf einer Linux-Plattform (optional)**

   Erstellen Sie eine Aufzeichnungsdatei, in der der gesamte TCP-Verkehr im Netzwerk auf einer Linux-Plattform protokolliert wird.

a. Suchen Sie im Produktverzeichnis **bin** nach dem Dienstprogramm **lrtcpdump** für Ihre Plattform. Kopieren Sie es in ein Verzeichnis, auf das Sie von Ihrem Linux-Computer aus zugreifen können. Kopieren Sie z. B. bei einer HP-Plattform die Datei **lrtcpdump.hp9**. Stellen Sie bei Verwendung von FTP sicher, dass der binäre Übertragungsmodus aktiviert ist.

b. Wechseln Sie zum Erteilen der Ausführungsberechtigungen zum Root-Benutzer: **chmod 755 lrtcpdump.<***Plattform***>**

c. **lrtcpdump** verwendet bei Vorhandensein mehrerer Schnittstellenkarten die erste in alphabetischer Reihenfolge. Verwenden Sie den Befehl **ifconfig**, um alle vorhandenen Schnittstellenkarten aufzulisten.

d. Führen Sie das Dienstprogramm mit der kompletten Syntax unter Angabe von Schnittstelle und Dateiname aus. Beispielsweise **lrtcpdump.hp9 -ietho -f<***Dateiname***>.cap**. Die Aufzeichnung des Netzwerkverkehrs wird gestartet.

e. Führen Sie typische Aktionen innerhalb Ihrer Anwendung aus.

f. Wechseln Sie zum Befehlszeilenfenster zurück, in dem **lrtcpdump** ausgeführt wird, und folgen Sie den Anweisungen zum Beenden der Aufzeichnung.

g. Legen Sie die Aufzeichnungsdatei im Netzwerk an einem Speicherort ab, auf den von VuGen aus zugegriffen werden kann.

3. ## Öffnen des Assistenten für die Datenverkehrsanalyse

   Klicken Sie auf die Schaltfläche **Datenverkehr analysieren**, oder wählen Sie **Vuser > Datenverkehr analysieren**. Weitere Informationen finden Sie unter "Bildschirm "Dienste angeben"" auf Seite 948.

4. ## Importieren eines Diensts (optional)

   Fügen Sie einen oder mehrere Dienste zur Liste hinzu (optional). Klicken Sie auf **Importieren**, um eine WSDL-Datei zu laden. Details finden Sie im "Dialogfeld "Dienst importieren"" auf Seite 1010.

   Klicken Sie auf die Schaltfläche **Weiter**.

5. ## Angeben von Informationen zum Datenverkehr

   Geben Sie eine Aufzeichnungsdatei und den Abschnitt des Skripts an, in den Sie den Datenverkehr laden möchten: **vuser_init**, **Action** oder **vuser_end**.

   Geben Sie an, ob Sie eingehenden oder ausgehenden Datenverkehr analysieren möchten. Geben Sie den Server an, dessen Datenverkehr Sie analysieren möchten.

   Weitere Informationen finden Sie unter "Dialogfeld "Aufzuzeichnende Anwendung angeben"" auf Seite 948.

6. **Filtern der IP-Adressen (optional)**

   Klicken Sie auf die Schaltfläche **Filteroptionen**, um die Aufzeichnungsoptionen zu öffnen und anzugeben, welche IP-Adressen ignoriert oder eingeschlossen werden sollen.

   Weitere Informationen finden Sie unter "Aufzeichnungsoptionen" auf Seite 164.

7. **Konfigurieren von SSL (optional)**

   Klicken Sie auf die Schaltfläche **SSL-Konfiguration**, um SSL-Zertifikate hinzuzufügen. Dies ist erforderlich, um Datenverkehr von einem sicheren Server zu analysieren.

   Details finden Sie im "Dialogfeld "SSL-Konfiguration"" auf Seite 966.

Erstellen von Aufzeichnungsdateien mit "lrtcpdump"

In dieser Aufgabe wird beschrieben, wie Sie mit dem Dienstprogramm "lrtcpdump" eine Aufzeichnungsdatei erstellen können.

1. **Installieren von WinPcap**

   WinPcap ermöglicht die Erfassung von Netzwerkverkehr in einer Datei. WinPcap wird für die Server Side Recording-Funktion im Web Services-Protokoll von VuGen verwendet. Weitere Informationen finden Sie unter http://www.winpcap.org.

   WinPCap kann aus **LoadRunnerInstallationDVD\Additional Components\WinPcap** installiert werden.

2. **Starten von "lrtcpdump.exe"**

   a. Öffnen Sie eine Befehlszeile. Ziehen Sie das Dienstprogramm "lrtcpdump" in die Befehlszeile.

      Das VuGen-Befehlszeilendienstprogramm **lrtcpdump** befindet sich im Ordner **LoadRunnerInstallationDVD\Additional Components\lrtcpdump**. Für jede Betriebssystemplattform ist ein eigenes Befehlszeilenprogramm vorhanden: **lrtcpdump.exe** (Windows), **lrtcpdump.hp9**, **lrtcpdump.ibm**, **lrtcpdump.linux** oder **lrtcpdump.solv4**.

   b. Geben Sie die Eigenschaften für Schnittstelle (-i) und Dateiname (-f) in der Befehlszeile an.

      Mit der Schnittstelleneigenschaft (-i) wird die Schnittstelle angegeben, die Sie zum Aufzeichnen des Datenverkehrs verwenden. Diese Eigenschaft ist optional.

      Hinweis: Wenn Sie die Schnittstelleneigenschaft nicht in der Befehlszeile definieren, werden Sie nach dem Drücken der Eingabetaste aufgefordert, eine auszuwählen.

      Mit der Dateinameneigenschaft (-f) wird angegeben, wo (Dateiname und Pfad) die Sicherung des Datenverkehrs gespeichert werden soll. Diese Eigenschaft ist erforderlich.

c. Drücken Sie die Eingabetaste, um die Aufzeichnung des Datenverkehrs zu starten. Nachdem Sie die Ausführung Ihres Geschäftsprozesses abgeschlossen haben, kehren Sie zur Befehlszeile zurück und drücken die Eingabetaste, um die Aufzeichnung zu beenden.

Bildschirm "Dienste angeben"

In diesem Dialogfeld können Sie die grundlegenden Informationen eingeben, die für das Starten einer Skriptaufzeichnung erforderlich sind.

Zugriff	Schaltfläche **VuGen > Aufzeichnung starten**
Relevante Aufgaben	"Hinzufügen von Inhalt" auf Seite 941

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
Details...	Öffnet das Dialogfeld **Dienste verwalten** für die Bereitstellung weiterer Informationen zum Dienst. Weitere Informationen finden Sie unter "Dialogfeld "Dienste verwalten"" auf Seite 1006.
Importieren	Öffnet das Dialogfeld **Dienst importieren**. Weitere Informationen finden Sie im "Dialogfeld "Dienst importieren"" auf Seite 1010. Informationen zur Benutzeroberfläche erhalten Sie unter "Dialogfeld "Dienst importieren"" auf Seite 1010.
Löschen	Entfernt den ausgewählten Dienst aus der Liste.
<Diensteliste>	Eine Liste der verfügbaren Dienste: • **Dienstname.** Der systemeigene Name des Diensts. • **WSDL-Speicherort.** Die Quelle der WSDL-Datei.

Dialogfeld "Aufzuzeichnende Anwendung angeben"

Über dieses Dialogfeld können Sie die grundlegenden Informationen angeben, die für das Starten einer Webdienstskript-Aufzeichnung erforderlich sind.

Zugriff	**VuGen > Aufzeichnung starten**, danach **Weiter**
Relevante Aufgaben	"Hinzufügen von Inhalt" auf Seite 941

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Erweiterte Optionen...	Öffnet das Dialogfeld **Aufzeichnungsoptionen**. Informationen zur Benutzeroberfläche finden Sie unter "Aufzeichnungsoptionen" auf Seite 164.
Standardwebbrowser aufzeichnen	Zeichnet die Standardbrowseraktionen auf. Geben Sie die Anfangs-URL-Adresse ein oder klicken Sie auf die Schaltfläche **Durchsuchen**, um zu einem Speicherort zu navigieren. Das Webdiensteprotokkoll unterstützt nur IE als Standardbrowser.
Beliebige Anwendung aufzeichnen	Zeichnet eine beliebige Win32-Anwendung auf. darüber hinaus können Sie Folgendes angeben: • **Aufzuzeichnendes Programm.** Wählen Sie den Browser, die Internet-Anwendung oder die Win32-Anwendung aus, die Sie aufzeichnen möchten. • **Programmargumente** (nur Win32-Anwendungen). Befehlszeilenargumente für die Anwendung. Wenn Sie beispielsweise **plus32.exe** mit den Befehlszeilenoptionen **peter@neptune** angeben, wird der Benutzer **Peter** beim Starten von **plus32.exe** mit dem Server **Neptune** verbunden. • **Arbeitsverzeichnis.** Ein Arbeitsordner für die Anwendung (nur wenn seitens der Anwendung erforderlich).
In Aktion aufzeichnen	Der Abschnitt für die Aufzeichnung: **vuser_init**, **Action** oder **vuser_end**. Verwenden Sie für zu wiederholende Aktionen den Abschnitt **Action** aus. Verwenden Sie für Initialisierungsschritte **vuser_init**.
Anwendungsstart aufzeichnen	In den folgenden Fällen ist es möglicherweise ratsam, den Start nicht aufzeichnen: • Wenn Sie mehrere Aktionen aufzeichnen; denn in diesem Fall muss der Start in nur einer Aktion ausgeführt werden. • In Fällen, in denen Sie zu einem spezifischen Punkt in der Anwendung navigieren möchten, bevor das Aufzeichnen gestartet wird. • Wenn Sie in ein vorhandenes Skript aufzeichnen.

Dialogfeld "SOAP importieren"

Über dieses Dialogfeld können Sie basierend auf einer SOAP-Datei einen Testschritt erstellen.

Zugriff	Führen Sie einen der folgenden Schritte durch: • Klicken Sie auf SOAP importieren • **SOA-Tools > SOAP importieren**
Relevante Aufgaben	"Hinzufügen von Inhalt" auf Seite 941 "Hinzufügen von Webdienst-Skriptinhalt - Übersicht" auf Seite 931

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
	Durchsuchen. Suchen Sie die XML-Datei mit dem SOAP-Datenverkehr.
	Lädt die Elementwerte aus der SOAP-Datei.
	Öffnet das Dialogfeld **Dienste verwalten** zum Importieren und Konfigurieren von Diensten.
	Öffnet die SOAP-Datei zur Ansicht in einem Browser.
<Aufruftyp>	Der Aufruftyp, der im Skript/Test erstellt werden soll: • **Webdienstaufruf**. Erfordert das Importieren eines Diensts. • **SOAP-Anforderung.** Erstellt im Skript einen Schritt vom Typ **soap_request**.
Eigenschaften von SOAP-Anforderungen	Die Eigenschaften der SOAP-Anforderung (wird nur für Aufrufe des Typs **SOAP-Anforderung** angezeigt). Geben Sie Folgendes an: • **URL.** Der URL oder die IP-Adresse des Servers, an den die Anforderung übermittelt werden soll. • **SOAP-Aktion.** Die in die Anforderung aufzunehmende SOAP-Aktion (bei mehreren Aktionen). • **Antwortparameter.** Ein Parametername für die Speicherung der Antwort der SOAP- oder Webdienstaufrufanforderung.

Dialogfeld "Neuer Webdienstaufruf"

In diesem Dialogfeld können Sie einen neuen Webdienstaufruf erstellen und konfigurieren.

Zugriff	Öffnen Sie ein Webdienst-Vuser-Skript und klicken Sie anschließend auf **SOA-Tools > Dienstaufruf hinzufügen** oder klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Dienstaufruf hinzufügen** .
Wichtige Informationen	Um über die Eigenschaften des Webdienstaufrufs auf vorhandene Webdienstaufrufe zuzugreifen, wählen Sie einen Schritt im **Schrittenavigator** und anschließend im Kontextmenü **Eigenschaften** aus.
Relevante Aufgaben	"Hinzufügen von Inhalt" auf Seite 941

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<Dienstargumentstruktur> (linker Bereich)	Eine erweiterbare Strukturhierarchie des Diensts, die die folgenden Knoten enthält: • <Vorgangsname> • Konfiguration der Transportschicht • Benutzerdefinierte SOAP-Kopfzeile • Eingabeargumente • Ausgabeargumente
<Parameterwerte> (rechter Bereich)	Ermöglicht es Ihnen, Werte für alle Knoten im linken Bereich festzulegen und auszuwählen.
Webdienstaufruf auswählen	Ermöglicht das Festlegen der folgenden Elemente: • **Dienst.** Eine Dropdownliste aller importierten Dienste, deren Name von der WSDL abgeleitet ist. • **Portname.** Eine Dropdownliste der verfügbaren Ports, über die die Anforderung gesendet werden soll. • **Vorgang.** Eine Dropdownliste der Vorgänge des Diensts. • **Zieladresse**. Der standardmäßige Endpunkt des Diensts. • **Adresse überschreiben.** Ermöglicht Ihnen die Eingabe einer alternativen Endpunktadresse im Feld **Zieladresse**.

Knoten <Vorgangsname>

Ermöglicht Ihnen die Erstellung von Beispielwerten für die Eingabeargumente des Vorgangs. Darüber hinaus können Sie Anhänge hinzufügen.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Methode	Der Name des ausgewählten Vorgangs (schreibgeschützt).
Automatische Werte generieren	Erstellt automatisch Beispielwerte für alle Eingabeargumente, basierend auf ihrem Datentyp.
Anhänge	Verarbeitet Eingabe- und Ausgabeanhänge: • **Zu Anforderung hinzufügen (Eingabe).** Fügt der Anforderung eine Datei oder einen Parameterwert hinzu. • **Empfangene speichern (Ausgabe).** Speichert die Antwort in einem Parameter.
Schritteigenschaften.	Führt die folgenden Dienstaufrufeigenschaften und ihre Werte auf: • WSDL-Speicherort • Dienstname • Portname • Zieladresse • SOAP-Aktion • SOAP-Namespace

Knoten "Konfiguration der Transportschicht"

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
HTTP/S-Transport	Legt die Transportmethode auf **HTTP** oder **HTTPS-Transport** fest.
Async.-Unterstützung	Kennzeichnet den Webdienstaufruf als asynchrone Nachricht, die durch ein Ereignis aktiviert wurde: **Async.-Ereignis** Ein beliebiger Name für ein Ereignis. **Hinweis:** Fügen Sie dem Skript einen Schritt **Webdienst - Auf Ereignis warten** hinzu, um das Wiedergabemodul anzuweisen, auf das Ereignis zu warten.

Forts.

WSA-Unterstützung	Aktiviert die WS-Adressierung. Verwenden Sie für eine Antwort eine der folgenden Optionen: • **WS-A-Antwort an.** Eine IP-Adresse des Servers, der Antworten soll, wenn das Ereignis eintritt. • **Auto-Erkennung.** Antwort an den aktuellen Host, wenn das Ereignis eintritt. Diese Option empfiehlt sich, wenn dasselbe Skript auf unterschiedlichen Computern ausgeführt werden soll. **Tipp:** Lassen Sie das Feld **Async.-Ereignis** leer, wenn Sie die WS-Adressierung im synchronen Modus verwenden möchten. Entfernen Sie in der Skriptansicht das Argument **AsyncEvent**. Hierdurch wird das Wiedergabemodul angewiesen, die Skriptausführung zu blockieren, bis die vollständige Antwort vom Server empfangen wurde.
JMS-Transport	Legt die Transportmethode für *synchrone* Nachrichten auf **JMS** fest. Weitere Informationen finden Sie unter "Testen von Transportschichten für Webdienste – Übersicht" auf Seite 967. **Hinweis:** Für *asynchrone* JMS-Nachrichten fügen Sie dem Skript manuell einen Schritt **JMS - Nachricht senden - Warteschlange** oder **JMS - Nachricht empfangen - Warteschlange** hinzu, um die Nachrichtenwarteschlangeninformation einzurichten.
JMS-Warteschlangen überschreiben	Ermöglicht es Ihnen, die Anforderungs- und Antwortwarteschlangen bereitzustellen.
Anforderungswarteschlange	Der Warteschlangenname für die Anforderungsnachricht.
Antwortwarteschlange	Der Warteschlangenname für die Antwortnachricht.

Knoten "Benutzerdefinierte SOAP-Kopfzeile"

Ermöglicht es Ihnen, zusätzliche, von der Anwendung generierte Elemente anzugeben, die in den SOAP-Umschlag einer HTTP-Nachricht aufgenommen werden sollen. Weitere Informationen über die Aufgabe finden Sie unter "Hinzufügen von Inhalt" auf Seite 941.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Öffnet einen XML-Editor, mit dem Sie den XML-Code der SOAP-Kopfzeile anzeigen und bearbeiten können.

Forts.

Importieren...	Öffnet das Dialogfeld **XML-Datei zum Import auswählen**.
Exportieren...	Öffnet das Dialogfeld **Exportieren von SOAP-Kopfzeile in Datei**.
	Öffnet das Dialogfeld **Parameter auswählen oder erstellen**.
SOAP-Kopfzeile verwenden	Schließt eine SOAP-Kopfzeile in die HTTP-Anforderung ein.
Kopfzeile	Die Kopfzeilenquelle: • Für eine importierte Datei: Kopfzeilenelement, wie es in der importierten Datei erscheint. • Für einen Parameter: Der Parametername (in geschweiften Klammern)

Knoten "Eingabeargumente"

Ermöglicht es Ihnen, die Eigenschaften festzulegen und Werte für alle Eingabeargumente zu erstellen.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Alle einschließen	Umfasst alle Argumente der Methode im Webdienstaufruf.
Zurücksetzen	Setzt die Argumente auf ihren ursprünglichen Zustand zurück. Sie werden nicht in den Aufruf einbezogen und sie werden auf die Werte in der WSDL festgelegt.
Generieren	Erstellt Beispieldaten für alle Eingabeargumente.
Argument bearbeiten	Öffnet den Bereich für die Bearbeitung des Werts des ausgewählten Arguments.
Name	Der Name des Vorgangs (schreibgeschützt)
Argumentliste	Eine Liste der Eingabeargumente. • Einfache Parameter • Arrays (zeigt nur die oberste Ebene an).

Knoten <Eingabeargumentname>

Wenn Sie ein Eingabeargument auswählen, können Sie im rechten Ausschnitt Argumentwerte angeben.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Hinzufügen...	Öffnet das "Dialogfeld "Arrayelemente hinzufügen"" auf Seite 960 zum Hinzufügen eines neuen Arrayelements zum Eingabeargument (wird nur angezeigt, wenn ein übergeordneter Knoten im Eingabearray ausgewählt wurde).
Löschen...	Entfernt die ausgewählten Arrayelemente im Eingabeargument (wird nur angezeigt, wenn ein übergeordneter Knoten im Eingabearray ausgewählt wurde).
Einschließen	Schließt die untergeordneten Argumente des ausgewählten Arguments in den Webdienstaufruf ein. Diese Option ist nur für ein Argument mit untergeordneten Argumenten aktiviert, für das die Option **Argument im Aufruf berücksichtigen** aktiviert ist.
Ausschließen...	Schließt die untergeordneten Argumente des Arguments aus dem Webdienstaufruf aus.
Bearbeiten...	Öffnet einen XML-Editor zur Bearbeitung des XML-Codes, der die Argumentwerte enthält. Es werden nur die Änderungen an den Elementwerten und an der Anzahl der Arrayelemente gespeichert.
Importieren...	Öffnet das Dialogfeld **XML-Datei zum Import auswählen**.
Exportieren...	Öffnet das Dialogfeld **Exportieren von Argument-XML in Datei**.
	Öffnet das Dialogfeld **Parameter auswählen oder erstellen**.
Name	Der Name des Arguments oder Arrays.
Argument im Aufruf berücksichtigen	Schließt das Argument in den Aufruf ein. Klicken Sie für Arrays auf **Einschließen**, um dem Aufruf die untergeordneten Argumente hinzuzufügen. Klicken Sie zum Ausschließen aller auslassbaren Argumente auf **Ausschließen**.
Typ	Der in der WSDL festgelegte Typ des Arguments. Wenn die WSDL abgeleitete Typen enthält, erscheint dieses Feld als Dropdownliste. Weitere Informationen finden Sie unter "Spezielle Argumenttypen" auf Seite 934.
Null	Setzt das Nillable-Attribut auf **true**.

Forts.

XML (nur für Arrays)	• **XML.** Aktiviert die Schaltflächen **Bearbeiten**, **Importieren** und **Exportieren**. Durch das Bearbeiten von XML können Sie die Werte eines Arguments manuell einfügen. Klicken Sie auf das Symbol **ABC**, um die gesamte XML-Struktur durch einen einzelnen Parameter vom Typ XML zu ersetzen. **Hinweis**: Bei diesem Importvorgang werden zuvor exportierte XML-Dateien, und keine SOAP-Standarddateien verarbeitet. • **Automatisch Wert für dieses Argument generieren.** Fügt automatisch Werte für alle untergeordneten Elemente ein. • **Hinzufügen**/**Löschen**. Fügt Elemente im Array hinzu oder entfernt sie.
Wert (für Nicht-Arrayelemente)	Der Argumentwert. Klicken Sie zum Parametrieren dieses Werts auf das Symbol **ABC** (nur für Nicht-Arrays verfügbar).
Automatisch Wert für dieses Argument generieren	Erstellt einen Beispielwert für das ausgewählte Argument.

Knoten "Ausgabeargumente"

Ermöglicht es Ihnen, die Eigenschaften aller Ausgabeargumente anzuzeigen.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Öffnet den Bereich für die Bearbeitung des Werts des ausgewählten Arguments.
Name	Der Name des Vorgangs (schreibgeschützt)

Forts.

Negativtest	**Ermöglicht Negativtests.** Bestätigt, dass die Anwendung eine Aufgabe nicht ausführt, für deren Ausführung sie nicht entwickelt wurde. In diesen Fällen müssen Sie sicherstellen, dass die Anwendung einen SOAP-Fehler und kein SOAP-Ergebnis ausgegeben hat. Wählen Sie eine **Erwartete Antwort** aus. • **SOAP-Ergebnis.** Eine SOAP-Antwort auf die Anfrage. • **SOAP-Fehler.** Eine Rückmeldung, die darauf hinweist, dass die SOAP-Anfrage ungültig war. Negativtests beziehen sich ausschließlich auf SOAP-Fehler. • **HTTP-Fehler.** Ein HTTP-Fehler, z. B. "Seite nicht gefunden", der in keinem Zusammenhang mit dem Webdienst steht. Weitere Informationen finden Sie unter "Negative Testing Overview".
Argumentliste	Eine Liste der Ausgabeargumente und der entsprechenden Parameter, in denen die Werte gespeichert werden.

Knoten <Ausgabeargumentname>

Ermöglicht es Ihnen, einen Parameter für die Speicherung des Werts des Ausgabeargumente anzugeben.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Hinzufügen...	Öffnet das "Dialogfeld "Arrayelemente hinzufügen"" auf Seite 960 zum Hinzufügen eines neuen Arrayelements zum Ausgabeargument (wird nur angezeigt, wenn ein übergeordneter Knoten im Ausgabearray ausgewählt wurde). Weitere Informationen finden Sie unter "Dialogfeld "Arrayelemente hinzufügen"" auf Seite 960.
Löschen	Entfernt die ausgewählten Arrayelemente im Ausgabeargument (wird nur angezeigt, wenn ein übergeordneter Knoten im Ausgabearray ausgewählt wurde).
Name	Der Name des Arguments oder Arrays.
Rückgegebenen Wert in Parameter speichern	Speichert den Wert des ausgewählten Arguments in einem Parameter. Um einen benutzerdefinierten Parameternamen anzugeben, ändern Sie die Standardangabe für Param_<*arg_name*> im Feld **Parameter**.
Null	Setzt den Wert des aktuellen Arguments auf **nil=true**.
XML (nur für Arrays)	XML-Code, der die Argumentwerte enthält. Klicken Sie zum Parametrieren dieses Werts auf das Symbol **ABC** (nur für Arrays verfügbar).

Knoten "Ausgabeanhänge"

Ermöglicht es Ihnen, die Eigenschaften für Ausgabeanhangparameter festzulegen. Diese Option wird nur angezeigt, wenn Sie Ausgabeanhänge im "<Operation Name> Node" aktiviert haben.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Fügt ein neues indexbasiertes Ausgabeargument hinzu. Diese Option ist nur verfügbar, wenn Sie **Anhänge nach Index speichern** ausgewählt haben.
Alle Anhänge speichern	Speichert alle Ausgabeanhänge mit den folgenden Eigenschaften in einem Parameter: • **Inhalt.** Ein bearbeitbarer Name für den Parameter, in dem der Anhang gespeichert wird. • **Inhaltstyp.** Der Parametertyp (schreibgeschützt). • **Inhalts-ID.** Eine eindeutige ID für den Parameter (schreibgeschützt).
Anhänge nach Index speichern	Speichert die Ausgabeanhänge in indexbasierten Parametern. Wählen Sie zum Festlegen des Index einen der Parameter aus und ändern Sie die Indexnummer im rechten Ausschnitt.

Knoten <Ausgabeargument>

Ermöglicht es Ihnen, die Eigenschaften für Ausgabeanhangparameter festzulegen. Diese Option wird nur angezeigt, wenn Sie Ausgabeanhänge im "<Operation Name> Node" aktiviert haben.

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Löscht den ausgewählten Anhangparameter. Wenn Sie die Anhänge nach Index gespeichert haben, wird nur das ausgewählte Element entfernt.
Index	Eine Indexnummer für den Parameter. Dieses Feld ist nur aktiviert, wenn Sie **Anhänge nach Index speichern** im "Output Attachments Node" ausgewählt haben.
Inhalt	Ein bearbeitbarer Name für den Parameter, in dem der Anhang gespeichert wird.
Inhaltstyp	Der Inhaltstyp des Parameters (schreibgeschützt).
Inhalts-ID	Eine eindeutige Inhalts-ID für den Parameter (schreibgeschützt).

Dialogfeld "Eingabeanhang hinzufügen"

Über diese Seite können Sie Ihren Webdienstanforderungen Eingabeanhänge hinzufügen.

Zugriff	Klicken Sie auf Dienstaufruf hinzufügen und wählen Sie den obersten Knoten aus - den Vorgangsnamen. Wählen Sie **Zu Anforderung hinzufügen (Eingabe)** im Abschnitt **Anhänge** aus.
Wichtige Informationen	Sie müssen einen Dienst importieren, bevor Sie einem Dienstaufruf einen Anhang hinzufügen. Hintergrundinformationen finden Sie unter "Webdienstaufruf - Anhänge" auf Seite 933.
Relevante Aufgaben	"Hinzufügen von Inhalt" auf Seite 941

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Datenquelle	Der Speicherort der Daten. • **Datei.** Der Speicherort der Datei: ▪ Absoluter Pfad: Der vollständige Pfad der Datei. Beachten Sie, dass diese Datei für alle Rechner, auf denen das Skript ausgeführt wird, zugänglich sein muss. ▪ Relativer Pfad: Ein Dateiname (empfohlen). Wird diese Methode während der Wiedergabe gewählt, sucht VuGen im Ordner des Skripts nach dem Anhang. Um den Anhang im Ordner des Skripts hinzuzufügen, wählen Sie **Datei > Dateien zum Skript hinzufügen** und geben den Dateinamen an. • **Parameter.** Der Name eines Parameters, in dem die Daten enthalten sind.
Inhaltstyp	Der Inhaltstyp der Datei, in der die Daten enthalten sind. Die Option **Automatisch ermitteln** weist VuGen an, automatisch den Inhaltstyp zu ermitteln. Im Feld **Wert** sind manuelle Eingaben zulässig und es wird eine Dropdownliste mit allgemeinen Inhaltstypen bereitgestellt.
Inhalts-ID	Ein eindeutige ID für den Anhang. Standardmäßig erstellt VuGen diese ID automatisch. Optional können Sie eine andere ID im Feld **Wert** angeben.

Dialogfeld "Arrayelemente hinzufügen"

Über diese Seite können Sie Elemente zu einem Argumentarray hinzufügen, das dieselbe Struktur aufweist wie das vorhandene Array. Diese Option ist für Eingabe- und für Ausgabeargumente verfügbar.

Bei Eingabeelementen können Sie den Werten des neuen Arrays ein vorhandenes Element zugrunde legen.

Zugriff	Klicken Sie auf Dienstaufruf hinzufügen. Wählen Sie einen Argumentknoten aus, bei dem es sich um ein Array handelt.
Wichtige Informationen	Ihre Argumentstruktur muss ein Array aufweisen, damit dieses Dialogfeld angezeigt wird.
Relevante Aufgaben	"Hinzufügen von Inhalt" auf Seite 941

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Name	Der Name und der Index des übergeordneten Knotens des Arrays.
Startindex	Der Index, dem neue Arrayelemente hinzugefügt werden sollen.
Elemente	Die Anzahl der identischen Arrayelemente, die der Argumentstruktur hinzugefügt werden sollen.
Werte aus Index kopieren	Erstellt die neuen Arrayelemente mit den Werten eines spezifischen Arrayelements (nur für Eingabeargumente verfügbar).

Dialogfeld "Base64-Daten verarbeiten - Einfache Daten"

Über dieses Dialogfeld können Sie die Codierungsoptionen für Ihre einfachen Base64-Daten festlegen.

Zugriff	Für einfache, nicht komplexe Base64-Werte: • Wählen Sie in den Eigenschaften des Webdienstaufrufs ein Eingabeargument des Typs **Base64** aus. • Wählen Sie die Option **Wert** aus. • Wählen Sie **Verschlüsselten Text einbetten** aus. • Klicken Sie auf die Schaltfläche **Durchsuchen**.
Wichtige Informationen	Verwenden Sie für ein komplexes Array das "Dialogfeld "Base64-Daten verarbeiten - Komplexe Daten"" unten.
Relevante Aufgaben	"Hinzufügen von Inhalt" auf Seite 941

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Decode to File ...	Ermöglicht es Ihnen, den decodierten Text in einer Datei zu speichern.
Text to encode	Verwenden Sie für komplexe Daten das "Dialogfeld "Base64-Daten verarbeiten - Komplexe Daten"" unten.
Encoding Options	Eine Liste der Codierungsmethoden. **Standard:** Unicode (UTF-8)
Encoded data	Die codierte Version der Daten aus dem Bereich mit dem zu codierenden Text.

Dialogfeld "Base64-Daten verarbeiten - Komplexe Daten"

In diesem Dialogfeld können Sie die Codierungsoptionen für Ihre komplexen Base64-Daten festlegen.

Zugriff	Für komplexe Base64-Werte: • Wählen Sie in den Eigenschaften des Webdienstaufrufs ein komplexes Eingabeargument des Typs **Base64** aus. • Wählen Sie die Option **Wert** aus und klicken Sie auf das Parametersymbol. • Ersetzen Sie den Wert durch einen Parameter • Klicken Sie mit der rechten Maustaste auf das Parametersymbol im Feld **Wert** und wählen Sie **Parametereigenschaften** aus. • Klicken Sie auf die Schaltfläche **Daten bearbeiten**. • Klicken Sie in der Spalte mit den festgelegten Werten auf die Schaltfläche **B64**.
Wichtige Informationen	Verwenden Sie für einfache, nicht komplexe Daten das "Dialogfeld "Base64-Daten verarbeiten - Einfache Daten"" auf Seite 960.
Relevante Aufgaben	"Hinzufügen von Inhalt" auf Seite 941

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Codieren	Codiert die angegebene Datei.
Decode to File ...	Ermöglicht es Ihnen, decodierte Daten in einer Datei zu speichern. Dabei handelt es sich normalerweise um Daten, die bei der Wiedergabe abgerufen wurden.
File	Codieren Sie die Datei nach Referenz oder nach Inhalt. • **File path.** Die zu codierende Datei. • **Link to file.** Referenziert die Datei mit den Werten. Wenn keine Angabe vorhanden ist, wird der Inhalt der angegebenen Datei verwendet. VuGen kopiert den Inhalt in den Skriptordner. **Tipp:** Aktivieren Sie für Text mit mehr als 10 KB **Link to file**.
Text	Codiert die angegebene Textzeichenfolge. • **Text to encode.** Der zu codierende Base64-Text. Während Sie den Text eingeben, codiert VuGen ihn im Ausschnitt **Encoded data**. • **Encoding Options.** Eine Liste der Codierungsmethoden. Standard: Unicode (UTF-8).

Aspektliste

In der folgenden Tabelle sind die verfügbaren Testaspekte aufgeführt:

Aspektname	Beschreibung
Positive Testing	Generiert einen umfassenden Positivtest, mit dem alle Vorgänge des Diensts überprüft werden.
Standard Compliance	Überprüft den Dienst hinsichtlich der Einhaltung von Branchenstandards wie WS-I und SOAP.
Service Interoperability	Testet die Interoperabilität der Vorgänge des Diensts mit allen unterstützten Webdienstetoolkits. Umfasst die folgenden untergeordneten Aspekte: • **.NET Framework.** Testet, ob die Dienste vollständige Interoperabilität mit dem .NET Framework WSE 2-Toolkit aufweisen, indem alle Vorgänge mit Standard-/erwarteten Werten aufgerufen werden. • **Axis/Java Based Web Services.** Testet, ob die Dienste vollständige Interoperabilität mit Axis 1.3 Web Services Framework aufweisen, indem alle Vorgänge mit Standard-/erwarteten Werten aufgerufen werden.
Security Testing	Testet die Dienstsicherheit. Umfasst die folgenden untergeordneten Aspekte: • **SQL Injection Vulnerability.** Überprüft, ob der Dienst Anfälligkeiten für SQL-Injections aufweist, indem SQL-Anweisungen und Fehler in relevante Parameter eingeschleust werden. • **Cross-site Scripting (XSS).** Versucht, den Dienst hacken, indem Code in eine Website eingeschleust wird, der die Funktionalität beeinträchtigt.
Boundary Testing	Erstellt Tests mithilfe der Negativtesttechnik, um Daten, Typen, Parameter und die eigentliche SOAP-Nachricht zu ändern und so die Grenzen des Dienst zu erfassen. Umfasst die folgenden untergeordneten Aspekte: • **Extreme Values.** Stellt Datentypen bereit, die für die Dienste nicht zulässig sind, und stellt sicher, dass sie nicht akzeptiert werden. • **Null Values.** Stellt NULL-Parameter für den Dienst bereit, um sicherzustellen, dass sie nicht akzeptiert werden.

Forts.

Performance Testing	Umfasst die folgenden untergeordneten Aspekte: • **Stress Testing.** Test zum Ermitteln der möglichen Spitzenlasten der Anwendung. • **Overload Sustainability Testing.** Testet, wie gut die für die Anwendung zugeordnete Hardware die Anzahl der Benutzer unterstützt, die voraussichtlich darauf zugreifen werden. • **Volume Testing.** Testet, ob das System eine große Dateneingabe verarbeiten kann. • **Longevity Test.** Testet, ob das System eine konsistente Anzahl von gleichzeitig ausgeführten Vusern bewältigen kann, die für einen Zeitraum von mindestens 24 Stunden Transaktionen durchführen und die Kapazitäten nahezu vollständig ausnutzen. • **Scalability Testing.** Wiederholte Belastungs-, Überlastungs-, Volumen- und Lebensdauertests mit unterschiedlichen Server- oder Netzwerkhardwarekonfigurationen.

Bildschirm "Dienste angeben"

Mithilfe dieses Assistenten können Sie Webdienste auswählen, die Ihrem datenverkehrbasierten Skript zugeordnet werden sollen.

Zugriff	Schaltfläche **Datenverkehr analysieren**
Relevante Aufgaben	"Hinzufügen von Inhalt" auf Seite 941

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
Details...	Öffnet das Dialogfeld **Dienste verwalten** für die Bereitstellung weitere Informationen zu dem Dienst. Weitere Informationen finden Sie unter "Dialogfeld "Dienste verwalten"" auf Seite 1006.
Importieren	Öffnet das Dialogfeld **Dienst importieren**. Weitere Informationen finden Sie unter "Dialogfeld "Dienst importieren"" auf Seite 1010.
Löschen	Entfernt den ausgewählten Dienst aus der Liste.

<Diensteliste>	Eine Liste der verfügbaren Dienste: • **Dienstname.** Der systemeigene Name des Diensts. • **WSDL-Speicherort.** Die Quelle der WSDL-Datei.

Bildschirm "Informationen zum Datenverkehr angeben"

Über diesen Assistentenbildschirm können Sie die Aufzeichnungsdatei für den eingehenden und ausgehenden Datenverkehr angeben.

Zugriff	Schaltfläche **Datenverkehr analysieren**, **Weiter**
Relevante Aufgaben	"Hinzufügen von Inhalt" auf Seite 941

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Durchsuchen. Ermöglicht es Ihnen, eine Aufzeichnungsdatei auszuwählen, die importiert werden soll.
	Öffnet den Knoten **Filter für Datenverkehr** im Dialogfeld **Aufzeichnungsoptionen**. Auf diese Weise können Sie angeben, welche IP-Adressen für das Skript berücksichtigt werden sollen und welche nicht. Weitere Informationen finden Sie unter "Aufzeichnungsoptionen" auf Seite 164.
	Öffnet das "Dialogfeld "SSL-Konfiguration"" auf der nächsten Seite, über das Sie SSL-Zertifikate zum Analysieren von Datenverkehr von einem sicheren Server hinzufügen können.
Aufzeichnungsdatei	Der Name einer Aufzeichnungsdatei, die den Serververkehr enthält und normalerweise mit der Erweiterung **.cap** endet.
Eingehender Datenverkehr	IP-Adresse und Port des Servers, dessen eingehender Verkehr analysiert werden soll.
Ausgehender Datenverkehr	IP-Adresse und Port des Servers, dessen ausgehender Verkehr analysiert werden soll.
In Aktion aufzeichnen	Der Abschnitt, für den das Skript erstellt werden soll: **vuser_init**, **Action** oder **vuser_end**. Verwenden Sie für zu wiederholende Aktionen den Abschnitt **Action** aus. Verwenden Sie für Initialisierungsschritte **vuser_init**.

Dialogfeld "SSL-Konfiguration"

In diesem Dialogfeld können Sie die SSL-Zertifikate für Ihre Datenverkehrsdatei konfigurieren.

Zugriff	Schaltfläche **Datenverkehr analysieren**, **Weiter**
Relevante Aufgaben	"Hinzufügen von Inhalt" auf Seite 941

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
	Hinzufügen eines Zertifikats. Fügt eine neue Zeile zur Zertifikatsliste hinzu.
	Löschen. Entfernt das ausgewählte Zertifikat.
<Zertifikatsliste>	Die Eigenschaften des Zertifikats. Geben Sie Folgendes an: • **IP.** IP-Adresse des Servers, der analysiert wird. • **Port.** Port des Servers, der analysiert wird. • **Datei.** Der Pfad der Zertifikatsdatei (mit der Erweiterung **.pem**), die den privaten Schlüssel enthält. Verwenden Sie die Schaltfläche **Durchsuchen**, um nach der Datei zu suchen. • **Kennwort.** Ein Kennwort zum Entschlüsseln der Zertifikatsdatei. Informationen zum Konvertieren von Zertifikaten finden Sie nachfolgend unter **Zertifikat-Dienstprogramm**.
	Öffnet die Registerkarte **Convert Certificate** (Zertifikat konvertieren) des SSL-Dienstprogramms, mit dem Sie Zertifikate aus den Formaten PKCS #12 und X.509 in das PEM-Format konvertieren können. Weitere Informationen zum Konvertieren des Zertifikats in das PEM-Format finden Sie unter Converting Certificates oder unter **Hilfe > LoadRunner-Funktionsreferenz > Web-Vuser-Funktionen (WEB) > Web-Vuser-Funktionen > web_set_certificate_ex**.

Webdienste - Vorbereiten von Skripts für die Wiedergabe

Übersicht über die Vorbereitung für die Wiedergabe

Nachdem Sie ein Skript mit Webdienstaufrufen erstellt haben, können Sie dieses für die Wiedergabe vorbereiten.

Dabei besteht die Möglichkeit, das Skript mit angepassten Fehler- und Protokollmeldungen oder mit Transaktionen zu erweitern. Ferner können Sie Ihr Skript mit JMS-Funktionen, **jms_<Suffix>** oder XML-Funktionen, **lr_xml_<Suffix>** ergänzen. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Mithilfe der Laufzeiteinstellungen können Sie das Verhalten echter Benutzer noch genauer emulieren. Sie enthalten allgemeine Einstellungen, aber auch Einstellungen, die nur für das Web Service-Protokoll gelten. Weitere Informationen finden Sie unter "Laufzeiteinstellungen" auf Seite 330.

In einigen Fällen müssen Sie u. U. das Ergebnis eines Webdienstaufrufs als Eingabe für einen anderen Aufruf verwenden. Hierzu speichern Sie das Ergebnis in einem Parameter, den Sie anschließend referenzieren. Weitere Informationen finden Sie unter "Vorbereiten von Skripts für die Wiedergabe" auf Seite 984.

Testen von Transportschichten für Webdienste – Übersicht

Webdienste bedienen sich einer Reihe unterschiedlicher Transportschichten. Die Transportschicht ist das Protokoll, das für den Austausch von Nachrichten mit dem Server genutzt wird.

In VuGen ist es möglich, die Transportschicht für Dienste zu konfigurieren. Die Transportschichten HTTP/HTTPS und JMS (Java Message Service) werden vollständig unterstützt.

Benutzerhandler ermöglichen es, SOAP-Anfragen und -Antworten zu verarbeiten und ihnen ein benutzerdefiniertes Verhalten zuzuweisen. Weitere Informationen finden Sie unter "Benutzerhandler" auf Seite 979.

- "Senden von Nachrichten über HTTP/HTTPS" unten

Senden von Nachrichten über HTTP/HTTPS

HTTP wird verwendet, um Anforderungen von einem Webclient, in der Regel einem Browser, an einen Webserver zu übermitteln. Der angeforderte Webinhalt wird ebenfalls per HTTP vom Server an den Client zurückgegeben.

HTTPS dient der sicheren Kommunikation zwischen Client und Server. Normalerweise werden Kreditkartentransaktionen und andere Vorgänge, bei denen es zum Austausch sensibler Daten kommt, über dieses Protokoll abgewickelt.

In der Regel arbeitet der Anfrage/Antwort-Mechanismus synchron. Beim synchronen Messaging blockiert das Wiedergabemodul die Skriptausführung, bis der Server die Antwort sendet. Im asynchronen Modus führt das Wiedergabemodul das Skript aus, ohne auf die Antwort des Servers auf vorangegangene Nachrichten zu warten.

Für die HTTP/HTTPS-Transportmethode kann bei asynchronen Aufrufen die WS-Adressierung verwendet werden. Weitere Informationen finden Sie unter "WS-Adressierung" auf Seite 971.

JMS-Transport – Übersicht

JMS ist ein J2EE-Standard für den Austausch von Nachrichten (entweder Text oder Java-Objekte) zwischen Java-Clients.

Es gibt zwei unterschiedliche Kommunikationsszenarios:

Peer-to-Peer. Wird auch als **Point-to-Point** bezeichnet. In JMS wird das Point-to-Point-Messaging implementiert, indem eine Nachrichtenwarteschlange als Ziel einer Nachricht definiert wird. Nachrichten werden an die Nachrichtenwarteschlange übermittelt, und der Empfänger erhält die Nachricht dann aus der Warteschlange.

Publish-Subscribe. Jede Meldung wird von einem Veröffentlicher unter einem ausgewählten Thema an eine Vielzahl von Abonnenten gesendet. Die Abonnenten erhalten nur die Nachrichten, die gesendet wurden, nachdem das Abonnement abgeschlossen wurde.

VuGen unterstützt Point-to-Point-Verbindungen durch den Austausch von JMS-Nachrichten mit einer Warteschlange.

Bevor Meldungen über JMS ausgetauscht werden können, müssen eine Reihe von Elementen zur Beschreibung der Transportmethode konfiguriert werden:

- **JNDI initial context factory.** Der Klassenname der Fabrikklasse, die einen anfänglichen Kontext zum Auffinden der JMS-Ressourcen, z. B. JMS-Verbindungsfabrik oder JMS-Warteschlange, herstellt.
- **JNDI provider.** Der URL des Dienstanbieters, der zum Auffinden der JMS-Ressourcen, z. B. JMS-Verbindungsfabrik oder JMS-Warteschlange, verwendet wird.
- **JMS connection factory.** Der JNDI-Name der JMS-Verbindungsfabrik.

Darüber hinaus können Sie einen Zeitüberschreitungswert für empfangene Nachrichten und die Anzahl der JMS-Verbindungen pro Prozess festlegen.

Diese Optionen können in den JMS-Laufzeiteinstellungen konfiguriert werden. Weitere Informationen finden Sie unter "JMS > Erweitert (Knoten)" auf Seite 388.

In diesem Abschnitt werden außerdem folgende Themen behandelt:

- "JMS-Skriptfunktionen" unten
- "JMS-Nachrichtenstruktur" auf der nächsten Seite

JMS-Skriptfunktionen

VuGen verwendet zur Implementierung der JMS-Transportmethode die vorhandenen API-Funktionen. Alle Funktionen beginnen mit dem Präfix **jms**:

Funktionsname	Beschreibung
jms_publish_ message_topic	Veröffentlicht Nachrichten zu einem bestimmten Thema.
jms_receive_ message_ queue	Empfängt eine Nachricht aus einer Warteschlange.
jms_receive_ message_topic	Empfängt die zu einem bestimmten Thema veröffentlichten Nachrichten als Abonnement.
jms_send_ message_ queue	Sendet eine Nachricht an eine Warteschlange.
jms_send_ receive_ message_ queue	Sendet eine Nachricht an eine bestimmte Warteschlange und empfängt eine Nachricht aus einer bestimmten Warteschlange.
jms_ subscribe_ topic	Erstellt ein Abonnement für ein Thema.
jms_set_ general_ property	Legt eine allgemeine Eigenschaft im Benutzerkontext fest.
jms_set_ message_ property	Legt eine JMS-Kopfzeile oder -Eigenschaft für die nächste zu sendende Nachricht fest bzw. verwendet eine JMS-Kopfzeile oder -Eigenschaft zum Filtern von empfangenen Nachrichten.

Die JMS-Schritte bzw. -Funktionen stehen nur beim manuellen Erstellen von Skripts zur Verfügung – JMS-Nachrichten, die zwischen Client und Server ausgetauscht werden, können nicht aufgezeichnet werden.

Im Gegensatz zu Peer-to-Peer-Verbindungen, bei denen Nachrichtenwarteschlangen verwendet werden, werden die publish-subscribe-Funktionen **jms_publish_message_topic**, **jms_ subscribe_topic** und **jms_receive_message_topic** nicht für Webdienstaufrufe unterstützt. Um diese Funktionen zusammen mit Webdienstaufrufen einzusetzen, müssen Sie manuell Benutzer-

Handler einrichten, um die Nutzdaten in den JMS-Nachrichten zu generieren. Weitere Informationen finden Sie unter "Erstellen eines Benutzerhandlers" auf Seite 991.

Weitere Informationen zur JMS-Funktion finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**; Sie können auch eine Funktion auswählen und dann **F1** drücken).

JMS-Nachrichtenstruktur

Jede JMS-Nachricht besteht aus den folgenden Elementen:

- Die **Kopfzeile** enthält Standardattribute (Correlation ID, Priority, Expiration date).
- **Eigenschaften**. Benutzerdefinierte Attribute.
- **Textkörper**. Text- oder Binärdaten.

Für den Textkörper von JMS-Nachrichten stehen mehrere Formate zur Verfügung. Zwei gebräuchliche Formate sind **TextMessage** und **BytesMessage**.

Wenn Sie das Standardverhalten ändern möchten, überschreiben Sie es mithilfe der Funktion **jms_set_general_property**, bevor Sie die Nachricht senden. Sie können für die Eigenschaft JMS_MESSAGE_TYPE die Werte **TextMessage**, **BytesMessage** oder **Default** festlegen. Beispiel:

jms_set_general_property("step1","**JMS_MESSAGE_TYPE**","BytesMessage");

Weitere Informationen finden Sie in der Function Reference **(Hilfe > Funktionsreferenz)**.

Asynchrone Nachrichten – Übersicht

Mit VuGen kann sowohl synchrones als auch asynchrones Messaging emuliert werden.

Beim synchronen Messaging blockiert das Wiedergabemodul die Skriptausführung, bis der Server die Antwort sendet. Im asynchronen Modus führt das Wiedergabemodul das Skript aus, ohne auf die Antwort des Servers auf vorangegangene Nachrichten zu warten.

In diesem Abschnitt werden außerdem folgende Themen behandelt:

- "Senden von asynchronen Aufrufen über HTTP/HTTPS" unten
- "WS-Adressierung" auf der nächsten Seite

Senden von asynchronen Aufrufen über HTTP/HTTPS

Im folgenden Abschnitt wird beschrieben, wie asynchrone Aufrufe über HTTP/HTTPS übermittelt werden. Sie verwenden den Schritt **Wait for Event**, um Vuser zu zwingen, vor dem Fortfahren die Antwort auf vorherige asynchrone Anfragen abzuwarten. Der Listener blockiert die Ausführung des Dienstes, bis der Server antwortet.

Folgende Optionen können beim Hinzufügen von "Webdienst - Auf Ereignis warten"-Schritten konfiguriert werden:

- **Quantifier.** Der Quantifizierer gibt an, ob der Vuser abwarten muss, bis auf alle Ereignisse (**ALL**), oder nur bis auf das erste (**ANY**) Ereignis eine Antwort erfolgt. **ANY** gibt den Namen des ersten Ereignis zurück, das eine Antwort empfangen hat. **ALL** gibt einen der Ereignisnamen zurück.

- **Timeout**. Der Zeitüberschreitungswert in Millisekunden. Wenn innerhalb des angegebenen Zeitraums auf keines der Ereignisse Antworten erfolgen, dann gibt der Schritt **web_service_wait_for_event** den Wert NULL zurück.

- **Events**. Eine Liste aller asynchronen Ereignisse, auf die gewartet werden soll.

Wenn ein Skript mit asynchronem Messaging ausgeführt wird, enthält das Wiedergabeprotokoll Informationen über die Ereignisse und die Ein- und Ausgabeargumente.

Weitere Informationen über die Aufgabe finden Sie unter "Senden von Nachrichten über HTTP/S" auf Seite 987.

Beim Einrichten einer asynchronen Nachricht können Sie mithilfe der WS-Adressierung die Adresse angeben, an die der Dienst beim Auftreten eines Ereignisses antwortet. Weitere Informationen finden Sie unter "WS-Adressierung" unten.

WS-Adressierung

Die WS-Adressierung ist eine Spezifikation, die es Webdiensten ermöglicht, Adressinformationen auszutauschen. Hierzu werden Webdienst-Endpunkte angegeben, um eine durchgängige Endpunktidentifizierung für Nachrichten zu garantieren. Auf diese Weise können Nachrichten in Netzwerken übermittelt werden, die über zusätzliche Verarbeitungsknoten, z. B. Endpunkt-Manager, Firewalls und Gateways, verfügen. Die WS-Adressierung unterstützt für Webdienstnachrichten die synchrone und asynchrone Transportmethode.

Für die WS-Adressierung muss die Adresse **WSAReplyTo** angegeben werden. Dies ist die Adresse, an die der Dienst antworten soll.

Ein optionales Argument vom Typ **WSAAction** ermöglicht Ihnen, eine SOAP-Aktion für den Fall festzulegen, dass über die Transportschicht keine Nachrichten übertragen werden können.

Das folgende Beispiel zeigt eine typische SOAP-Nachricht mit WS-Adressierung, die von VuGen im Hintergrund implementiert wird.

```
<S:Envelope xmlns:S="http://www.w3.org/2003/05/soap-envelope"
                xmlns:wsa="http://www.w3.org/2004/12/addressing">
    <S:Header>
        <wsa:MessageID>
            http://example.com/SomeUniqueMessageIdString
        <wsa:MessageID>
        <wsa:ReplyTo>
            <wsa:Address>http://myClient.example/someClientUser</wsa:Address>
        </wsa:ReplyTo>
```

```
<wsa:Address>http://myserver.example/DemoErrorHandler</wsa:Address>
        </wsa:FaultTo>
        <wsa:To>http://myserver.example/DemoServiceURI</wsa:To>
        <wsa:Action>http://myserver.example/DoAction</wsa:Action>
    </S:Header>
    <S:Body>
         <!-- Body of SOAP request message -->
    </S:Body>
</S:Envelope>
```

Im folgenden Beispiel antwortet der Server bei Auftreten von **Event_1** an die Schnittstelle **212.199.95.138**.

```
web_service_call("StepName=Add_101",
        "SOAPMethod=Calc.CalcSoap.Add",
        "ResponseParam=response",
        "AsyncEvent=Event_1",                "WSAReplyTo=212.199.95.138",
        "WSDL=http://lab1/WebServices/CalcWS/Calc.asmx?wsdl",
        "UseWSDLCopy=1",
        "Snapshot=t1153825715.inf",
        BEGIN_ARGUMENTS,
            "first=1",
            "second=2",
        END_ARGUMENTS,
        BEGIN_RESULT,
            "AddResult=Param_AddResult1",
        END_RESULT,
        LAST);
```

WS-Adressierungsaufrufe können im asynchronen oder asynchronen Modus ausgegeben werden. Lassen Sie unter den Optionen für die Transportschicht das Feld **Async Event** leer, wenn Sie die WS-Adressierung im synchronen Modus verwenden möchten. Entfernen Sie in der Skriptansicht das Argument **AsyncEvent**. Hierdurch wird das Wiedergabemodul angewiesen, die Skriptausführung zu blockieren, bis die vollständige Antwort vom Server empfangen wurde.

Weitere Informationen über die Aufgabe finden Sie unter "Senden von Nachrichten über HTTP/S" auf Seite 987.

Datenbankintegration – Übersicht

Beim Testen von Webdiensten ist die Verwendung exakter und aktueller Daten von großer Bedeutung. Ein veralteter Daten-Snapshot kann bereits ungültig oder nicht mehr maßgeblich sein.

Mithilfe der Datenbankintegration greifen Sie während des Tests auf Werte in einer Datenbank zu und gewährleisten damit, dass die Daten wirklich aktuell sind.

Die Funktionen für die Datenbankintegration empfehlen sich in folgenden Situationen:

- "Herstellen einer Verbindung zu einer Datenbank" unten
- "Verwenden von Daten, die aus SQL-Abfragen abgerufen wurden" unten
- "Validieren von Datenbankwerten nach einem Webdienstaufruf" auf Seite 975
- "Überprüfen von Rückgabewerten anhand einer Datenbank" auf Seite 977
- "Durchführen von Aktionen in Datasets" auf Seite 978

Herstellen einer Verbindung zu einer Datenbank

Fügen Sie zum Herstellen einer Verbindung zu einer Datenbank einen Verbindungsschritt in Ihr Skript ein. Ein integrierter Connection String Generator führt Sie durch den Erstellungsprozess für eine Zeichenfolge zur Herstellung einer Datenbankverbindung; diese Zeichenfolge ist speziell auf Ihre Datenbank und die zugehörigen Anmeldeinformationen abgestimmt. Vor dem Einfügen des Schrittes können Sie Ihre Verbindung zunächst testen.

Wenn Sie Ihr Skript mit Iterationen ausführen, wiederholen virtuelle Benutzer lediglich den Abschnitt **Action** des Skripts. Wenn Sie den Schritt zum Herstellen der Datenbankverbindung in den Abschnitt **Action** einfügen, wird er während des Tests für jede Iteration wiederholt. Virtuelle Benutzer wiederholen lediglich den Abschnitt **Action** des Skripts, nicht aber die Abschnitte **vuser_init** und **vuser_end**. Es empfiehlt sich daher, den Schritt zum Herstellen der Datenbankverbindung in den Abschnitt **vuser_init** einzufügen und den Schritt zum Trennen der Verbindung, **lr_db_disconnect**, in den Abschnitt **vuser_end**.

Wenn Sie nur eine einzige Abfrage ausführen müssen und einen Bildlauf durch die Daten durchführen, können Sie auch den Schritt **Database: ExecuteSQL Query** in den Abschnitt **vuser_init** einfügen.

Weitere Informationen über die Aufgabe finden Sie unter "Senden von Nachrichten über JMS" auf Seite 985.

Verwenden von Daten, die aus SQL-Abfragen abgerufen wurden

In diesem Szenario ruft der Test Daten aus der Datenbank ab und verwendet sie an einer späteren Stelle im Skript, z. B. für Webdienstaufrufe. Da das Skript die Daten in jedem Testlauf abruft, sind sie immer aktuell und maßgeblich.

In der folgenden Tabelle wird ein typischer Skriptablauf dargestellt:

Schritt	API-Funktion
Datenbankverbindung herstellen	lr_db_connect
SQL-Abfrage ausführen	lr_db_executeSQLStatement

Daten abrufen und speichern	lr_db_getvalue nach <param_name>
Webdienstaufruf	web_service_call mit {<param_name>}
Datenbankverbindung trennen	lr_db_disconnect

Es gibt zwei Möglichkeiten für die Durchführung einer Iteration in den Ergebnissen:

- Speichern der Ergebnisse bei jeder Iteration als einfachen Parameter
- Verwenden von integrierten VuGen-Iterationen zum Durchführen eines Bildlaufs in den Daten

Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Im folgenden Beispiel wird eine Verbindung zwischen dem Abschnitt **vuser_init** und der Datenbank hergestellt und eine Datenbankabfrage durchgeführt.

```
vuser_init()
{
lr_db_connect("StepName=myStep",
    "ConnectionString=Initial Catalog=MyDB;Data Source=mylab.net;user id =sa ;pa
ssword = 12345;" ,
    "ConnectionName=MyConnection",
    "ConnectionType=SQL",
    LAST);
lr_db_executeSQLStatement("StepName=MyStep",
        "ConnectionName=MyConnection",
        "SQLQuery=SELECT * FROM Addresses",
        "DatasetName=ds1",
        LAST);
    return 0;
}
```

Trennen Sie am Ende des Tests die Verbindung zur Datenbank im Abschnitt **vuser_end**.

```
vuser_end()
{
    lr_db_connect("StepName=myStep",
        "ConnectionString=Initial Catalog=MyDB;Data Source=LAB1.devlab.net;user
id =sa ;password = soarnd1314;" ,
        "ConnectionName=MyConnection",
        "ConnectionType=SQL",
        LAST);
    return 0;
}
```

Die wiederholt auszuführenden Schritte werden in den Abschnitt **Action** eingefügt. Beachten Sie die Verwendung des Arguments **Row**. Im ersten Aufruf der Datenbank geben Sie die erste Zeile mit dem Argument **Row=next** an. Verwenden Sie zum Abrufen eines weiteren Werts in derselben Zeile das Argument **current**.

```
Action()
{
```

```
    lr_db_getvalue("StepName=MyStep",
        "DatasetName=ds1",
        "Column=Name",
        "Row=next",
        "OutParam=nameParam",
        LAST);
    lr_db_getvalue("StepName=MyStep",
        "DatasetName=ds1",
        "Column=city",
        "Row=current",
        "OutParam=cityParam",
        LAST);
/* Use the values that you retrieved from the database in your Web Service call
*/
    web_service_call( "StepName=EchoAddr_101",
        "SOAPMethod=SanityService|SanityServiceSoap|EchoAddr",
        "ResponseParam=response",
        "Service=SanityService",
        "ExpectedResponse=SoapResult",
        "Snapshot=t1227168459.inf",
        BEGIN_ARGUMENTS,
        "xml:addr="
            "<addr>"
                "<name>{nameParam}</name>"
                "<street></street>"
                "<city>{cityParam}</city>"
                "<state></state>"
                "<zip></zip>"
            "</addr>",
        END_ARGUMENTS,
        BEGIN_RESULT,
        END_RESULT,
        LAST);
return 0;
}
```

Validieren von Datenbankwerten nach einem Webdienstaufruf

In diesem Szenario führt der Test einen Webdienstaufruf zum Ändern einer Datenbank im Back-End aus. Ziel dieses Szenarios ist die Überprüfung der resultierenden Werte in der Datenbank auf Korrektheit.

In der folgenden Tabelle wird ein typischer Skriptablauf dargestellt:

Schritt	**API-Funktion**
Datenbankverbindung herstellen	lr_db_connect (im Abschnitt **vuser_init**)

Webdienstaufruf	web_service_call
SQL-Abfrage ausführen	lr_db_executeSQLStatement
Daten abrufen und speichern	lr_db_getvalue nach <param_name>
Daten überprüfen	lr_checkpoint
Datenbankverbindung trennen	lr_db_disconnect (im Abschnitt **vuser_end**)

Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Das folgende Beispiel veranschaulicht den Prozess der Datenüberprüfung:

```
Action()
{
/* A Web Service call that modifies a database on the back end. */
    web_service_call( "StepName=addAddr_102",
        "SOAPMethod=Axis2AddrBookService|Axis2AddrBookPort|addAddr",
        "ResponseParam=response",
        "Service=Axis2AddrBookService",
        "ExpectedResponse=SoapResult",
        "Snapshot=t1227169681.inf",
        BEGIN_ARGUMENTS,
        "xml:arg0="
            "<arg0>"
                "<name>{Customers}</name>"
                "<city>{City}</city>"
            "</arg0>",
        END_ARGUMENTS,
        LAST);
/* Query the database by the cusotmer name that was modified by the Web Servic
e*/
    lr_db_executeSQLStatement("StepName=MyStep",
        "ConnectionName=MyConnection",
        "SQLQuery=SELECT * FROM Addresses WHERE name = '{Customers}' ",
        "DatasetName=ds1",
        LAST);
/* Get the values retrieved by the database query. */
    lr_db_getvalue("StepName=MyStep",
        "DatasetName=ds1",
        "Column=Name",
        "Row=current",
        "OutParam=CustomerName",
        LAST);
/* Compare the actual value with the expected value stored in the database. */
    lr_checkpoint("StepName=validateCustomer",
        "ActualValue={Customers}",
        "ExpectedValue={CustomerName}",
        "Compare=Equals",
        "StopOnValidationError=false",
```

```
        LAST);
return 0;
}
```

Überprüfen von Rückgabewerten anhand einer Datenbank

In diesem Szenario führt der Benutzer einen Webdienstaufruf aus, der eine XML-Antwort zurückgibt. Ziel dieses Szenarios ist die Validierung der Antwort anhand von erwarteten Werten. Die erwarteten Werte sind in einer Datenbank gespeichert. Das Skript ruft die erwarteten Ergebnisse aus der Datenbank ab und vergleicht sie anschließend mit der tatsächlich ausgegebenen Antwort.

In der folgenden Tabelle wird ein typischer Skriptablauf dargestellt:

Schritt	**API-Funktion**
Datenbankverbindung herstellen	lr_db_connect (im Abschnitt **vuser_init**)
Webdienstaufruf	web_service_call mit **Result=<result_param>**
SQL-Abfrage ausführen	lr_db_executeSQLStatement
Erwartete Daten abrufen	lr_db_getvalue nach <param_name>
Daten validieren	soa_xml_validate mit XPATH-Prüfpunkten.
Datenbankverbindung trennen	lr_db_disconnect (im Abschnitt **vuser_end**)

Mithilfe des XML-Validierungstools können Sie einen Prüfpunkt für die Antwortdaten erstellen. Verwenden Sie für die Erstellung des Validierungsschritts den Datenbankparameter, den Sie mit der Funktion **lr_db_getvalue** abgerufen haben.

Das folgende Beispiel veranschaulicht eine typische Validierung von Daten, die von einem Webdienstaufruf zurückgegeben wurden. Der Schritt **Validation** dient dem Vergleich der tatsächlich erwarteten Ergebnisse:

```
Action()
{
    web_service_call( "StepName=GetAddr_102",
        "SOAPMethod=AddrBook|AddrBookSoapPort|GetAddr",
        "ResponseParam=response",
        "Service=AddrBook",
        "ExpectedResponse=SoapResult",
        "Snapshot=t1227172583.inf",
        BEGIN_ARGUMENTS,
        "Name=abcde",
        END_ARGUMENTS,
        BEGIN_RESULT,
        END_RESULT,
        LAST);
```

```
    lr_db_executeSQLStatement("StepName=MyStep",
        "ConnectionName=MyConnection",
        "SQLQuery=SELECT * FROM Addresses WHERE name = 'abcde' ",
        "DatasetName=ds1",
        LAST);
    lr_db_getvalue("StepName=MyStep",
        "DatasetName=ds1",
        "Column=Name",
        "Row=current",
        "OutParam=CustomerName",
        LAST);
    soa_xml_validate ("StepName=XmlValidation_1146894916",
        "Snapshot=t623713af7a594db2b5fef43da68ad59d.inf",
        "XML={GetAddrAllArgsParam}",
        "StopOnValidationError=0",
        BEGIN_CHECKPOINTS,
            CHECKPOINT,"XPATH=/*[local-name(.)='GetAddr'][1]/*[local-name(.)='Re
sult'][1]/*[local-name(.)='name'][1]","Value_Equals={CustomerName}",
        END_CHECKPOINTS,
        LAST);
    return 0;
}
```

Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Durchführen von Aktionen in Datasets

Mit VuGen können Sie Aktionen in Datasets durchführen, die von SQL-Abfragen zurückgegeben werden.

Die Funktion **lr_db_dataset_action** führt folgende Aktionen aus:

- **Zurücksetzen.** Positioniert den Cursor im ersten Datensatz des Datasets.
- **Entfernen.** Gibt für das Dataset reservierten Speicherplatz frei.
- **Print.** Gibt den Inhalt des gesamten Datasets im Wiedergabeprotokoll und sonstigen Testberichtübersichten aus.

Hinweis: Wenn Sie mithilfe der Funktion **lr_db_getvalue** Binärdaten abrufen, kann deren Inhalt nicht mit der Aktion **Print** ausgegeben werden.

Weitere Informationen zur Syntax sowie zur Verwendung der Funktion finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Anpassung – Übersicht

VuGen stellt eine Reihe erweiterter Funktionen zur Anpassung des Skriptverhaltens zur Verfügung. Diese Funktionen lassen sich aufteilen in Benutzerhandler und Konfigurationsdateien.

Benutzerhandler ermöglichen es, SOAP-Anfragen und -Antworten zu verarbeiten und ihnen ein benutzerdefiniertes Verhalten zuzuweisen. Weitere Informationen finden Sie unter weiter unten.

Mithilfe von Konfigurationsdateien können Sie erweiterte Einstellungen anpassen, z. B. Sicherheitsinformationen und die WSE-Konfiguration.

- "Benutzerhandler" unten
- "Benutzerdefinierte Konfigurationsdateien" auf Seite 981

Benutzerhandler

Benutzerhandler sind offene APIs, mit deren Hilfe die folgenden Operationen ausgeführt werden können:

- Abrufen und Festlegen der SOAP-Umschläge für Anfrage/Antwort
- Überschreiben der Transportschicht
- Abrufen und Festlegen des Inhaltstyps für Anfrage/Antwort
- Abrufen und Festlegen von Werten für LoadRunner-Parameter
- Abrufen von Konfigurationsargumenten aus einem Skript
- Ausgeben von Meldungen an das Ausführungsprotokoll
- Abbrechen einer Ausführung mit dem Status "Fehlgeschlagen"

Sie können einen Benutzerhandler direkt im Skript einrichten oder über eine DLL implementieren. Ein Handler kann lokal oder global angewendet werden.

Weitere Informationen über die Aufgabe finden Sie unter "Erstellen eines Benutzerhandlers" auf Seite 991.

Beispiele für Benutzerhandler finden Sie unter "Beispiele für Benutzerhandler" auf Seite 981.

Definition von Handlerfunktionen

Bei der Standardimplementierung eines Benutzerhandlers definieren Sie eine Benutzerhandlerfunktion innerhalb Ihres Vuser-Skripts mit folgender Syntax:

```
int MyScriptFunction(const char* pArgs, int isRequest)
```

Das Argument **pArgs** enthält die Zeichenfolge, die im Argument **UserHandlerArgs** der Funktion **web_service_call** angegeben ist.

Im Argument **isRequest** wird angegeben, ob die Funktion während der Verarbeitung einer Anfrage (1) oder einer Antwort (0) für einen SOAP-Umschlag aufgerufen wird.

Der Inhalt des SOAP-Umschlags wird sowohl für Anfragen als auch Antworten an den Parameter **SoapEnvelopeParam** übergeben. Nachdem der SOAP-Umschlag von der Funktion verarbeitet wurde, speichern Sie den Inhalt in demselben Parameter.

Verwenden Sie zum Aufrufen der Handlerfunktion den Funktionsnamen als Wert für das Argument **UserHandlerFunction** im entsprechenden Schritt für den Webdienstaufruf. Weitere Informationen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Rückgabecodes von Ereignishandlern

VuGen erkennt die folgenden Rückgabecodes für die Handlerfunktion.

Rückgabecode		Beschreibung
LR_HANDLER_ SUCCEEDED	0	Der Handler wurde erfolgreich ausgeführt, jedoch ohne Änderungen am SOAP-Umschlag.
LR_HANDLER_FAILED	1	Der Handler ist fehlgeschlagen, die Verarbeitung sollte abgebrochen werden.
LR_HANDLER_ SUCCEEDED_AND_ MODIFIED	2	Der Handler wurde erfolgreich ausgeführt; der aktualisierte SOAP-Umschlag wurde in **SoapEnvelopeParam** gespeichert.

Im folgenden Beispiel nimmt ein Skripthandler Änderungen am Umschlag der ausgehenden SOAP-Anfrage vor:

```
//This function processes the SOAP envelope before sending it to the server.
int MyScriptFunction(const char* pArgs, int isRequest)
{
    if (isRequest == 1) {
        //Get the request that is going to be sent
        char* str = lr_eval_string("{SoapEnvelopeParam}");
        //Manipulate the string...
        //Assign the new request content
        lr_save_string(str, "SoapEnvelopeParam");
        return LR_HANDLER_SUCCEEDED_AND_MODIFIED;
    }
    return LR_HANDLER_SUCCEEDED;
}
Action()
{
    //Instruct the web_service_call to use the handler
    web_service_call( "StepName=EchoAddr_102",
        "SOAPMethod=SpecialCases.SpecialCasesSoap.EchoAddr",
        "ResponseParam=response",
        "userHandlerFunction=MyScriptFunction",
        "Service=SpecialCases",
        "Snapshot=t1174304648.inf",
```

```
        BEGIN_ARGUMENTS,
        "xml:addr="
            "<addr>"
                "<name>abcde</name>"
                "<street>abcde</street>"
                "<city>abcde</city>"
                "<state>abcde</state>"
                "<zip>abcde</zip>"
            "</addr>",
        END_ARGUMENTS,
        BEGIN_RESULT,
        END_RESULT,
        LAST);
    return 0;
```

Benutzerdefinierte Konfigurationsdateien

Mithilfe von Konfigurationsdateien können Sie erweiterte Einstellungen anpassen, z. B. Sicherheitsinformationen und die WSE-Konfiguration. Sie ermöglichen Ihnen, das Testverhalten zur Laufzeit zu steuern.

Die .NET-Standardkonfigurationsdatei, **mmdrv.exe.config**, befindet sich im VuGen-Installationsverzeichnis. Einige Anwendungen besitzen ihre eigene Konfigurationsdatei, **app.config**.

Sie können den Testlauf noch weiter anpassen, indem Sie Ein- und Ausgabe filtern. Darüber hinaus können Sie Sicherheitsinformationen konfigurieren, z. B. Token-Informationen, und ob unsignierte Testzertifikate zulässig sind oder nicht.

Weitere Informationen über die Aufgabe finden Sie unter "Anpassen von Konfigurationsdateien" auf Seite 994.

Beispiele für Benutzerhandler

In diesem Abschnitt werden einige gängige Anwendungsfälle für Benutzerhandler vorgestellt.

.NET-Filter

Sie können .NET-Filter mithilfe des Benutzerhandlermechanismus auf Meldungen anwenden.

Wenn Sie mit Microsofts **Web Service Enhancements** (WSE) 2.0 vertraut sind, können Sie einen .NET-Filter erstellen und diesen für ein- und ausgehende SOAP-Nachrichten registrieren. Ein .NET-Filter ist eine Klasse, die sich aus **Microsoft.Web.Services2.SoapInputFilter** oder **Microsoft.Web.Services2.SoapOutputFilter** ableitet. Durch Überschreiben der Funktion **ProcessMessage** für diese Klasse können Sie Textkörper und Kopfzeile des Umschlags prüfen und bearbeiten.

Erweitern Sie die unten dargestellte Datei **default.cfg** um die folgenden Zeilen, um den Filter global für das gesamte Skript zu definieren.

```
[UserHandler]
Function=LrWsSoapFilterLoader
Args=<Filters InputFilterClass="class name" InputFilterLib="lib name" OutputFilt
erClass="class name" OutputFilterLib="lib name" />
Order=BeforeSecurity/AfterSecurity/AfterAttachments
```

Der Parameter **InputFilterClass** gibt den Namen der Klasse an. **InputFilterLib** gibt den Namen der Assembly an, zu der die Klasse gehört. Beispiel:

```
    web_service_call(
        ...
        "UserHandlerName=LrWsSoapFilterLoader",
        "UserHandlerArgs=<Filters InputFilterClass=\"MyFilterNamespace.MyFilterC
lassName\" InputFilterLib=\"MyAssemblyName\" />",
        BEGIN_ARGUMENTS,
        ...
        END_ARGUMENTS,
        ...
        );
```

Verwenden Sie **SoapOutputFilter**, um ausgehende Anfragen vom Typ **web_service_call** zu untersuchen, und **SoapInputFilter**, um die Antwort des Servers zu untersuchen. Verwenden Sie **InputFilterClass** und **InputFilterLib**, wenn sich Ihr Filter von **SoapInputFilter** ableitet, bzw. **OutputFilterClass** und **OutputFilterLib**, wenn sich der Filter von **SoapOutputFilter** ableitet.

Fügen Sie die folgenden Argumente zur Funktion **web_service_call** hinzu, um den Filter für einen bestimmten Schritt zu definieren.

```
UserHandlerName= LrWsSoapFilterLoader
UserHandlerArgs=<Filters InputFilterClass=\"class name\" InputFilterLib=\"lib na
me\" OutputFilterClass=\"class name\" OutputFilterLib=\"lib name\" />
UserHandlerOrder=BeforeSecurity/AfterSecurity/AfterAttachments
```

Überschreiben der Transportschicht

Das folgende Beispiel zeigt, wie eine Benutzerhandlerfunktion die Transportschicht überschreibt. In diesem Fall sendet VuGen die SOAP-Anfrage nicht automatisch per HTTP-Transport, sondern übernimmt die im benutzerdefinierten Handler angegebene Transportmethode.

Nachdem Sie eine Antwort erhalten haben, legen Sie den Antwortumschlag mit folgendem Befehl fest:

```
lr_save_string(someResponseEnvelopeStr, "SoapEnvelopeParam");
```

Wenn Sie eine alternative Transportschicht anwenden möchten, geben Sie **ReplaceTransport** als Wert für das Argument **UserHandlerOrder** an. Definieren Sie die Transportschicht im Handler.

```
web_service_call(

...

"UserHandlerFunction=<Transport HandlerFunction>",

"UserHandlerArgs=<handler arguments>",
```

```
"UserHandlerOrder=ReplaceTransport"

...

LAST);
```

Hinzufügen von MIME-Anhängen

Für Webdienstskripts, die auf dem .NET-Toolkit basieren, werden MIME-Anhänge von der Infrastruktur nicht unterstützt. Mithilfe des Handlermechanismus können Sie Funktionen für MIME-Anhänge zu .NET-Skripts hinzufügen.

In den folgenden Abschnitten wird beschrieben, wie Sie das Senden und Empfangen von MIME-Anhängen für das .NET-Toolkit gestalten können. MIME-Anhänge können in einer einzigen Operation empfangen und gesendet werden.

Senden von MIME-Anhängen

Fügen Sie den fett formatierten Code zur Funktion **web_service_call** hinzu, um einen MIME-Anhang zu übermitteln:

```
web_service_call( "StepName=EchoComplex_101",
  "SOAPMethod=SimpleService|SimpleServiceSoap|EchoComplex",
  "ResponseParam=response",
  "Service=SimpleService",
  "UserHandlerName=LrWsAttachmentsHandler",  "UserHandlerArgs=ATTACHMENT_ADD; AT
TACHMENTS_FORMAT_MIME;   ContentType=text/plain; FileName=C:\\temp\\results.disc
omap",
  "ExpectedResponse=SoapResult",
  "Snapshot=t1208947811.inf",
  BEGIN_ARGUMENTS,
  "xml:cls="
   "<cls>"
    "<i>123456789</i>"
    "<s>abcde</s>"
   "</cls>",
  END_ARGUMENTS,
  BEGIN_RESULT,
  END_RESULT,
   LAST);
```

Bearbeiten Sie die Parameter **FileName** und **ContentType** so, dass sie den tatsächlichen Dateipfad und Inhaltstypen angeben.

Empfangen von MIME-Anhängen

Fügen Sie folgenden Code zur Funktion **web_service_call** hinzu, um einen MIME-Anhang zu empfangen:

```
"UserHandlerName=LrWsAttachmentsHandler",
 "UserHandlerArgs=ATTACHMENT_SAVE_ALL;ParamNamePrefix=attach;"
```

Senden und Empfangen von MIME-Anhängen

Modifizieren Sie den Webdienstaufruf wie im Folgenden gezeigt, um einen MIME-Anhang in

derselben **web_service_call**-Funktion zu senden und zu empfangen:

```
"UserHandlerName=LrWsAttachmentsHandler",
 "UserHandlerArgs=ATTACHMENT_SAVE_ALL;ParamNamePrefix=attach; ATTACHMENT_ADD; AT
TACHMENTS_FORMAT_MIME; ContentType=text/plain; FileName=C:\\temp\\results.discom
ap",
```

Vorbereiten von Skripts für die Wiedergabe

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie das Skript für die Wiedergabe vorbereiten und es dann ausführen.

Zuordnen von Werten für Eingabeparameter

Speichern Sie das Ausgabeergebnis zunächst in einem Parameter, den Sie dann später in einem Webdienstaufruf referenzieren.

1. **Speichern des Ausgabeparameters**

   a. Doppelklicken Sie im **Schrittenavigator** auf den Webdienstaufruf, dessen Ausgabeergebnis Sie verwenden möchten, um seine Eigenschaften anzuzeigen.

   b. Wählen Sie im linken Fensterbereich das Ausgabeargument aus, dessen Wert in einem Parameter gespeichert werden soll.

   c. Wählen Sie im rechten Fensterbereich **Save returned value in parameter** aus. Geben Sie im Feld **Parameter** einen Namen ein.

2. **Verwenden des gespeicherten Parameters als Eingabe**

   a. Doppelklicken Sie im **Schrittenavigator** auf den Webdienstaufruf, dessen Eingabeparameter Sie festlegen möchten.

   b. Wählen Sie im linken Fensterausschnitt das Eingabeargument aus, für das der gespeicherte Parameter verwendet werden soll.

   c. Wählen Sie im rechten Fensterbereich **Wert** aus und klicken Sie auf das Symbol **abc**. Das Dialogfeld **Select or Create Parameter** wird geöffnet.

   d. Wählen Sie den gespeicherten Ausgabeparameter aus der Liste **Parametername** aus.

   e. Um einen Eingabeparameter in der Skriptansicht anzugeben, wählen Sie den zu ersetzenden Wert aus und wählen im Kontextmenü den Befehl **Vorhandenen Parameter verwenden** aus. Wählen Sie einen der verfügbaren Parameter.

Festlegen der Laufzeiteinstellungen (optional)

Öffnen Sie die Laufzeiteinstellungen (F4), um die JMS- und VM-Einstellungen zu konfigurieren. Klicken Sie auf den Knoten **JMS > Erweitert**. Informationen zur Benutzeroberfläche erhalten Sie unter "JMS > Erweitert (Knoten)" auf Seite 388.

Konfigurieren von XSDs mit Any-Elementen (optional)

Bei Webdiensten, die über ein XSD-Schema mit einem **Any**-Element vom Typ `<xsd:element name="<Any_element>" type="xsd:anyType" />` verfügen, müssen Sie sich vergewissern, dass Ihr Skript mit dem nachstehenden Modell übereinstimmt:

```
BEGIN_ARGUMENTS,
                        "xml:Any_element="
                             "<Any_element>"
                             "<string>the string to send</string>"
                             "</Any_element>",
END_ARGUMENTS,
```

Das eigentliche SOA-Protokoll kann sich von dem obigen Modell leicht unterscheiden, wird jedoch fehlerfrei ausgeführt, solange Ihr Skript mit diesem Modell übereinstimmt.

Ferner besteht die Möglichkeit, komplexe Elementtypen für den Typ **<any>** zu senden. Beispiel:

```
"xml:Any_element="
                                    "<Any_element>"
                                    "<myComplexTypeName>"
                                    "<property1>123</property1>"
                                    "<property2>456</property2>"
                                    "</myComplexTypeName>"
                                    "</Any_element>",
```

Ausführen des Skripts

Klicken Sie auf **Wiedergeben > Ausführen**. Beobachten Sie das Ausgabeprotokoll, ob dort wichtige Meldungen angezeigt werden.

Überprüfen der Testergebnisse

Nach Abschluss des Testlaufs werden die Testergebnisse automatisch in einem Fenster angezeigt. Ein X kennzeichnet einen fehlgeschlagenen Schritt.

Wenn Sie einen Knoten erweitern, erhalten Sie detaillierte Informationen über die SOAP-Antwort und die Prüfpunkte. Weitere Informationen finden Sie unter "Anzeigen von Wiedergabeergebnissen" auf Seite 513.

Senden von Nachrichten über JMS

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie Nachrichten mithilfe der JMS-Transportmethode übermitteln.

1. ### Öffnen der Schritteigenschaften

   Wählen Sie im **Schrittenavigator** den Schritt aus, für den Sie den Transport festlegen möchten. Wählen Sie anschließend im Kontextmenü **Argumente anzeigen** aus.

2. ### Auswählen der JMS-Transportmethode

Wählen Sie zunächst den Knoten **Konfiguration der Transportschicht** und dann **JMS-Transport** aus.

Weitere Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "Neuer Webdienstaufruf"" auf Seite 950.

3. **Festlegen der Laufzeiteinstellungen (optional)**

Konfigurieren Sie die Laufzeiteinstellung für das Skript. Informationen hierzu finden Sie unter "JMS > Erweitert (Knoten)" auf Seite 388.

4. **Senden von synchronen JMS-Nachrichten (optional)**

Nachdem Sie einen Webdienstaufruf erstellt und die JMS als Transportmethode festgelegt haben, sendet VuGen die JMS-Nachrichten im synchronen Modus. Machen Sie ggf. Angaben zur Warteschlange.

5. **Senden von asynchronen JMS-Nachrichten (optional)**

Für die Implementierung von asynchronen Nachrichten über JMS werden Anfragen und Antworten mithilfe von JMS-Schritten gesendet bzw. abgerufen, und nicht über Webdienstaufrufe.

a. Setzen Sie den Cursor im Skript an die gewünschte Stelle. Wählen Sie **Insert > New Step** aus und erweitern Sie den Knoten **JMS Functions**.

b. Wählen Sie eine JMS-Funktion aus: **JMS Send Message Queue** sendet eine Nachricht an eine Warteschlange. **JMS Receive Message Queue** empfängt eine Nachricht aus einer Warteschlange.

c. Klicken Sie auf **OK**, um die Eigenschaften für die JMS-Funktionen zu konfigurieren.

d. Geben Sie einen Namen für die Warteschlange an. Klicken Sie auf **OK**, um die JMS-Funktion zu erzeugen.

Weitere Informationen zu diesen Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**; Sie können auch eine Funktion auswählen und dann **F1** drücken).

6. **Senden von Nachrichten über JMS mithilfe von SOAP-Nachrichten (optional)**

So senden Sie Nachrichten über JMS mithilfe einer SOAP-Nachricht und ohne Webdienstaufruf:

a. Zeichnen Sie SOAP-Nachrichten mithilfe eines Standardwebprotokolls auf.

b. Setzen Sie den Cursor im Skript an die gewünschte Stelle. Wählen Sie **Insert > New Step** aus und erweitern Sie den Knoten **JMS Functions**.

c. Wählen Sie eine JMS-Funktion aus: **Send Message Queue** oder **JMS Receive Message Queue**.

d. Klicken Sie auf **OK**, um die Eigenschaften für die JMS-Funktionen zu konfigurieren.

e. Geben Sie einen Namen für die Warteschlange an. Klicken Sie auf **OK**, um die JMS-Funktion zu erzeugen.

Weitere Informationen hierzu finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**; Sie können auch eine Funktion auswählen und dann **F1** drücken).

Senden von Nachrichten über HTTP/S

In dieser Aufgabe wird beschrieben, wie Nachrichten mithilfe der HTTP-Transportmethode übermittelt werden.

1. **Öffnen der Schritteigenschaften**

   Wählen Sie im **Schrittenavigator** den Schritt aus, für den Sie den Transport festlegen möchten. Wählen Sie anschließend im Kontextmenü **Argumente anzeigen** aus.

2. **Auswählen der HTTP/S-Transportmethode**

   Wählen Sie zunächst den Knoten **Konfiguration der Transportschicht** und dann **HTTP/S-Transport** aus.

3. **Senden einer synchronen HTTP-Meldung (optional)**

   Erstellen Sie für HTTP-Nachrichten im synchronen Modus einen standardmäßigen Webdienstaufruf und aktivieren Sie die Option **Async.-Unterstützung** nicht.

4. **Senden von asynchronen HTTP-Nachrichten (optional)**

   a. Wählen Sie **HTTP/S Transport** aus und aktivieren Sie die Option **Async Support**.

   b. Geben Sie im Feld **Async Event** einen Ereignisnamen ein.

   c. Klicken Sie auf **OK**, um den Webdienstaufruf zu erzeugen.

   d. Fügen Sie einen **Wait for Event**-Schritt hinzu. Wählen Sie **Insert > New Step** und dann **Web Service Wait for Event** aus.

   e. Geben Sie einen Schrittnamen, einen Quantifizierer und einen Zeitüberschreitungswert an. Klicken Sie auf **Add** und fügen Sie dann den Namen des in vorherigen Schritt erstellten Ereignisses ein.

   In der Skriptansicht gibt VuGen asynchrones Messaging durch Hinzufügen des Parameters **AsyncEvent** an.

```
web_service_call("StepName=EchoString_101",
        "SOAPMethod=EchoRpcEncoded.EchoSoap.EchoString",
        "ResponseParam=response1",
```

```
"Service=ExtendedECHO_rpc_encoded",
"AsyncEvent=Event_1",
"Snapshot=t1157371707.inf",
BEGIN_ARGUMENTS,
"sec=7",
"strString=mytext",
END_ARGUMENTS,
BEGIN_RESULT,
"EchoStringResult=first_call",
END_RESULT,
LAST);
```

Das Kennzeichen **AsyncEvent** zwingt den Vuser, die Antwort auf vorherige asynchrone Dienstanfragen abzuwarten.

5. **Senden einer asynchronen Meldung mithilfe der WS-Adressierung (optional)**

   a. Wählen Sie die **Async.-Unterstützung** aus und geben Sie im Feld **Async.-Ereignis** einen Namen für das Ereignis an. Hier können Sie einen beliebigen Namen eingeben.

   b. Aktivieren Sie die Option **WSA Support**. Geben Sie im Feld **WS-A-Antwort an** eine IP-Adresse ein. Soll der aktuelle Host verwendet werden, geben Sie **autodetect** ein. **Autodetect** empfiehlt sich, wenn dasselbe Skript auf unterschiedlichen Computern ausgeführt werden soll. Der Server antwortet an die angegebene Adresse, wenn das Ereignis auftritt.

   c. Klicken Sie auf **OK**, um die Einstellungen zu speichern.

   d. Weisen Sie den Vuser an, auf ein Ereignis zu warten. Wählen Sie **Insert > New Step** aus und fügen Sie nach dem Schritt für den Webdienstaufruf einen **Web Service Wait For Event**-Schritt hinzu.

   e. Geben Sie einen Schrittnamen, einen Quantifizierer und einen Zeitüberschreitungswert an. Klicken Sie zum Hinzufügen eines Ereignisnamens auf **Add**. Der Webdienst wartet das angegebene Ereignis ab, bevor er antwortet.

   f. Mithilfe der Schaltflächen **Edit**, **Move Up** und **Move Down** können Sie Änderungen an den Ereignissen vornehmen.

Definieren einer Testmethode

In dieser Aufgabe wird beschrieben, wie Sie eine Testmethode auswählen.

1. **Öffnen der Schritteigenschaften**

   Klicken Sie im Schrittenavigator mit der rechten Maustaste auf die zu testende Antwort und

wählen Sie **Argumente anzeigen** aus.

2. **Auswählen eines Arguments**

   Wählen Sie den Knoten **Ausgabeargument** aus. Weitere Informationen finden Sie unter "Dialogfeld "Neuer Webdienstaufruf"" auf Seite 950.

3. **Festlegen einer Testmethode und Auswählen einer erwarteten Antwort**

   - Wenn nur Negativtests ausgeführt werden sollen, aktivieren Sie das Kontrollkästchen **Negative Testing** und wählen Sie **SOAP Fault** unter **Expected Response** aus.
   - Wenn beliebige SOAP-Antworten zugelassen werden sollen, aktivieren Sie das Kontrollkästchen **Negative Testing** und wählen Sie **Any SOAP** unter **Expected Response** aus.
   - Wenn nur Positivtests ausgeführt werden sollen, deaktivieren Sie das Kontrollkästchen **Negative Testing**.

4. **Überprüfen der Funktion im Skript**

   In der Skriptansicht wird die Testmethode im Argument **ExpectedResponse** angegeben. Im Beispiel unten führt das Skript einen Negativtest durch, angegeben durch den Wert **SoapFault**:

```
web_service_call("StepName=AddAddr_101",
        "SOAPMethod=AddrBook|AddrBookSoapPort|AddAddr",
        "ResponseParam=response",
        "Service=AddrBook",
        "ExpectedResponse=SoapFault",
        "Snapshot=t1189409011.inf",
        BEGIN_ARGUMENTS,
        END_ARGUMENTS,
        BEGIN_RESULT,
        END_RESULT,
        LAST);
```

5. **Auswerten des SOAP-Fehlerwerts**

   Bei Auftreten eines SOAP-Fehlers während der Skriptwiedergabe speichert VuGen den Fehler in dem Parameter **response**. Sie können diesen Wert mithilfe der Funktion **lr_xml_find** auswerten.

   Im folgenden Beispiel überprüft die Funktion **lr_xml_find** auf das Vorhandensein eines SOAP-Fehlers vom Typ **VersionMismatch** und gibt eine Meldung zurück.

```
lr_xml_find("XML={response}",
                "FastQuery=/Envelope/Body/Fault/faultString ",
                "Value=VersionMismatch",
    LAST);
```

```
if (soap_fault_cnt >0)
     lr_output_message{"A Version Mismatch SOAP Fault occurred")
```

Weitere Informationen zur Funktion **lr_xml_find** finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Hinzufügen einer Datenbankverbindung

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie in der Strukturansicht eine Datenbankverbindung hinzufügen.

1. **Öffnen des Lösungs-Explorers**

   Wählen Sie **Ansicht > Lösungs-Explorer** aus.

2. **Auswählen eines Abschnitts**

   Wählen Sie den gewünschten Abschnitt aus: **vuser_init** oder **Action**. Um zu verhindern, dass der Verbindungsaufbau mit jeder Iteration wiederholt wird, platzieren Sie den Schritt im Abschnitt **vuser_init**.

3. **Einfügen eines Datenbankverbindungsschritts**

   Wählen Sie **Entwurf > In Skript einfügen > Neuer Schritt** aus. Wählen Sie den Schritt **lr_db_connect** aus. Das Dialogfeld **Database Connection** wird geöffnet. Machen Sie folgende Angaben: **Step Name**, **Connection Name** und **Data Provider** (OLEDB oder SQL).

4. **Erstellen einer Datebankverbindungszeichenfolge**

   a. Klicken Sie auf **Connection String Generator**, um eine umgebungsspezifische Zeichenfolge zur Herstellung einer Datenbankverbindung zu erstellen.

   b. Geben Sie die Verbindungseigenschaften an:

      - **Server Name**
      - **Database Name**
      - **Authentication** method: **Windows Authentication** oder **User/password**.
      - **Username** und **Password**

   c. Klicken Sie auf **Test Connection**, um die angegebenen Informationen auf Korrektheit zu überprüfen.

   d. Wählen Sie einen **SQL Provider** (OLEDB oder SQL) aus und klicken Sie auf **Generate**.

5. **Überprüfen der Funktion im Skript**

   Überprüfen Sie, ob die Funktion **lr_db_connect** in das Skript eingefügt wurde.

Erstellen eines Benutzerhandlers

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie einen Benutzerhandler für Ihr Skript erstellen.

1. **Voraussetzung – Erstellen eines Webdienstaufrufs**

   Importieren Sie eine WSDL-Datei und erstellen Sie einen Standard-Webdienstaufruf. Weitere Informationen finden Sie unter "Hinzufügen von Webdienst-Skriptinhalt - Übersicht" auf Seite 931.

2. **Definieren einer Benutzerhandlerfunktion**

   Definieren Sie einen Benutzerhandler vor dem Webdienstaufruf:

   ```
   int MyScriptFunction(const char* pArgs, int isRequest)
   {
   ...
   }
   ```

3. **Aufrufen der Benutzerhandlerfunktion**

   Geben Sie zum Aufrufen der Handlerfunktion den Funktionsnamen als Wert für das Argument **UserHandlerFunction** im entsprechenden Schritt für den Webdienstaufruf an.

   ```
   web_service_call(

   ...

   "UserHandlerFunction=MyScriptFunction",

   "UserHandlerArgs=<handler arguments>",



   LAST);
   ```

4. **Auswerten der Handlerfunktion**

   Werten Sie den Handler-Rückgabecode aus, um zu ermitteln, ob der Handler erfolgreich war. Verwenden Sie die Rückgabecodes so wie unter "Benutzerhandler" auf Seite 979 beschrieben.

   ```
   //This function processes the SOAP envelope before sending it to the server.
   int MyScriptFunction(const char* pArgs, int isRequest)
   {
       if (isRequest == 1) {
           //Get the request that is going to be sent
           char* str = lr_eval_string("{SoapEnvelopeParam}");
           //Manipulate the string...
           //Assign the new request content
           lr_save_string(str, "SoapEnvelopeParam");
   ```

```
        return LR_HANDLER_SUCCEEDED_AND_MODIFIED;
    }
    return LR_HANDLER_SUCCEEDED;
}
```

5. ## Erstellen einer DLL-Datei (optional)

   Um einen Benutzerhandler über eine DLL zu definieren, suchen Sie im Ordner **include** des Produkts nach der API-Kopfzeilendatei **LrWsHandlerAPI.h**.

   Sie können ein Visual Studio-Beispielprojekt aus dem Ordner **samples/WebServices/SampleWsHandler** als Vorlage zum Erstellen eines Handlers nutzen. In diesem Beispiel wird der Anfrage- und Antwortumschlag abgerufen und in einem Parameter gespeichert. Öffnen Sie es in Visual Studio und passen Sie es nach Bedarf an. Wenn es nicht erforderlich ist, die Anfrage bzw. Antwort in einem Parameter zu speichern, können Sie den entsprechenden Abschnitt aus dem Beispiel entfernen.

   Speichern Sie das Beispiel nach der Bearbeitung und kompilieren Sie die DLL. Beim Kompilieren des Projekts legt Visual Studio die Datei **<name_des_benutzerhandlers>.DLL** im Verzeichnis **bin** ab. Wenn Sie das Projekt an einem anderen Speicherort kompilieren oder vorhaben, die DLL von einem Computer auf einen anderen zu kopieren, sollten Sie sie ebenfalls im Ordner **bin** ablegen.

6. ## Konfigurieren des Benutzerhandlers (optional)

   Sie können den DLL-Benutzerhandler entweder global oder lokal deklarieren.

   Wenn der Handler global für alle Anfragen im Skript gelten soll, fügen Sie den folgenden Abschnitt zur Datei **default.cfg** (diese befindet sich in demselben Ordner wie das Skript) hinzu.

   ```
   [UserHandler]
   Function=<name>
   Args=<arguments>
   Order=<BeforeSecurity/AfterSecurity/AfterAttachments>
   ```

   - **Name.** Der Name der DLL.
   - **Args.** Eine Liste der Konfigurationsargumente für den Handler. Verwenden Sie zum Abrufen der Argumente des Handlers die Methode **GetArguments**.
   - **Order.** Die Reihenfolge, in welcher der Benutzerhandler in Anfragen von Vusern verarbeitet wird: **Before Security**, **After Security** oder **After Attachments**. Sie können dieses Argument auch dazu verwenden, die Transportschicht zu überschreiben, indem Sie den Wert **Replace Transport** eingeben.

     Hinweis: Durch Festlegen der Eigenschaft **UserHandlerFunction** in einer Funktion vom Typ **web_service_call** werden die Definitionen in der **.cfg**-Datei überschrieben.

Standardmäßig werden Benutzerhandler vor den Sicherheitsfunktionen verarbeitet. Bei Anfragenachrichten verarbeiten Vuser Anhangshandler nach dem Sicherheitshandler. Bei Antworten verarbeiten Vuser die Handler in umgekehrter Reihenfolge. Normalerweise spielt die Reihenfolge keine Rolle, jeder Wert ist also zulässig.

Geben Sie zum Überschreiben der Transportschicht **Order=Replace Transport** und dann den neuen Transporthandler an. Wenn Sie den Transporthandler als getrennte DLL implementieren, wird die Funktion **HandleRequest** aufgerufen. Die Funktion **HandleResponse** hingegen wird ignoriert.

Fügen Sie die folgenden Argumente zur Funktion **web_service_call** hinzu, wenn Sie den Handler lokal für eine bestimmte Anfrage verwenden möchten:

```
UserHandlerName=<name1>
UserHandlerArgs=<args1>
UserHandlerOrder=<BeforeSecurity/AfterSecurity/AfterAttachments/Replace
                                  Transport>
```

7. ## Kopieren des Benutzerhandlers auf alle erforderlichen Computer

Vergewissern Sie sich, dass von allen Load Generator-Systemen aus, auf denen Skripts mit entsprechenden Aufrufen ausgeführt werden, auf die Benutzerhandler-DLL zugegriffen werden kann. Sie können sie beispielsweise in den Ordner **/bin** des Produkts kopieren.

Wenn Sie das Skript auf einen anderen Computer kopieren, werden die Handlereinstellungen beibehalten, da diese im Skriptordner festgelegt wurden.

8. ## Implementieren des Benutzerhandlers (optional)

Zur Implementierung eines Benutzerhandlers müssen die Funktionen **HandleRequest** oder **HandleResponse** verwendet werden. Beide Funktionen kennen nur einen Parameter, **context**, dessen Eigenschaften Sie im Handler festlegen können. Verwenden Sie die Get-Funktionen zum Abrufen von Eigenschaften und die Set-Funktionen, um Daten aus dem Wiedergabeframework an die Handler oder zwischen den Handlern zu übergeben.

- **GetEnvelope.** Ruft den Inhalt des Umschlags ab. Beispiel:
  ```
  const char * pEnvelope = context->GetEnvelope();
  ```
- **GetEnvelopeLength.** Ruft die Länge des Umschlags ab.
- **SetEnvelope.** Legt den Inhalt und die Länge des Umschlags fest. Beispiel:
  ```
  string str("MySoapEnvelope...");
  context->SetEnvelope(str.c_str(), str.length());
  ```
- **SetContentType.** Legt einen neuen Wert für den Inhalt der HTTP-Kopfzeile fest.
- **LogMessage.** Gibt eine Meldung an das Wiedergabeprotokoll aus.
- **GetArguments.** Ruft die für den aktuellen Handler definierten Konfigurationsargumente ab, um sie an die DLL weiterzugeben.

- **GetProperty.** Ruft einen benutzerdefinierten Eigenschaftenwert ab.
- **SetProperty.** Legt einen benutzerdefinierten Eigenschaftenwert fest.

Weitere Informationen finden Sie in den Kommentaren in der Datei **LrWsHandlerAPI.h** im Ordner **include** des Produkts.

Anpassen von Konfigurationsdateien

In den folgenden Schritten wird beschrieben, wie Sie Konfigurationsdateien anpassen. Weitere Informationen finden Sie unter "Benutzerdefinierte Konfigurationsdateien" auf Seite 981.

Suchen nach der Konfigurationsdatei

Ermitteln Sie den Speicherort der Konfigurationsdatei. Die .NET-Standardkonfigurationsdatei, **mmdrv.exe.config**, befindet sich im **bin**-Ordner des Produkts. Einige Anwendungen besitzen ihre eigene Konfigurationsdatei, **app.config**.

Speichern der Konfigurationsdatei der Anwendung

Wenn die Anwendung über eine eigene Konfigurationsdatei verfügt:

- Um die Konfigurationseinstellungen global auf alle Skripts anzuwenden, speichern Sie die Datei **app.config** unter dem Namen **mmdrv.exe.config** im Verzeichnis **bin**, indem Sie die vorhandene Datei überschreiben.
- Wenn Sie die Konfigurationseinstellungen lokal nur auf das vorliegende Skript anwenden möchten, kopieren Sie die Datei **app.config** in den Skriptordner. Hierdurch wird die Datei **mmdrv.exe.config** überschrieben. Sie ist dann immer mit diesem Skript verknüpft, selbst wenn Sie sie auf andere Computer kopieren.

Einstellen der Sicherheit (optional)

Standardmäßig sind in VuGen unsignierte Zertifikate zur Vereinfachung von Tests zulässig. Setzen Sie das Kennzeichen **allowTestRoot** im Abschnitt <security> auf **false**, um unsignierte Zertifikate auszuschließen.

```
<security>
   <x509 storeLocation="currentuser" alllowTestRoot="false"
```

Snapshots für Webdienste - Übersicht

Auf dem Protokoll für Webdienste basierende Vuser-Skripts verwenden den Snapshot-Bereich von VuGen.

- Eine Einführung zum Snapshot-Bereich finden Sie unter "Snapshot-Bereich - Übersicht" auf Seite 62.

- Details zum Arbeiten mit dem Snapshot-Bereich finden Sie unter "Arbeiten mit Snapshots" auf Seite 65.
- Details zur Standardbenutzeroberfläche des Snapshot-Bereichs finden Sie unter "Snapshot-Bereich" auf Seite 109.

Im Snapshot-Bereich können Sie Snapshots von Webdienstaufrufen anzeigen. Wenn Sie den Snapshot-Bereich für ein Webdienstskript aufrufen, werden die Snapshot-Daten auf der linken Seite des Snapshot-Bereichs in einer Strukturansicht und auf der rechten Seite in einer Textansicht dargestellt.

Die Strukturansicht auf der linken Seite des Snapshot-Bereichs setzt sich aus mehreren Knoten zusammen. Ein Symbol links von den einzelnen Knoten gibt Aufschluss über den Knotentyp:

- **Element:** Der Knoten stellt ein Element in der XML-Datei dar.
- **Attribut:** Der Knoten stellt ein Attribut in der XML-Datei dar.
- **Wert:** Der Knoten stellt einen Wert in der XML-Datei dar.

Neben den grundlegenden Funktionen des Snapshot-Bereichs bietet der Snapshot-Bereich für Webdienstskripts weitere Funktionen. Die Benutzeroberfläche hierfür wird nachfolgend beschrieben.

Zugriff	Wählen Sie **Ansicht > Snapshot** aus oder klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Snapshot-Bereich anzeigen** .
Relevante Aufgaben	"Vorbereiten von Skripts für die Wiedergabe" auf Seite 984

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
	Zeigt einen Snapshot der vom Server zurückgegebenen SOAP-Antwort an.
	Zeigt einen Snapshot der SOAP-Anforderung an, die vom Webdienstaufruf an den Server gesendet wurde.
	Öffnet das Dialogfeld **XPath-Suche** an, in dem Sie eine XPath-Suche nach dem Snapshot ausführen können.
	Zeigt die vorherigen oder nächsten Ergebnisse der XPath-Suche an.
	Zeigt die XML-Knoten vom Typ **Attribut** in der Strukturansicht des Snapshots an oder blendet sie aus.

	Zeigt die XML-Knoten vom Typ **Wert** in der Strukturansicht des Snapshots an oder blendet sie aus.

<Kontextmenü>	• **Auswahl kopieren.** Kopiert den in der Textansicht markierten Text in die Zwischenablage. • **In Community suchen.** Führt mit dem in der Textansicht als Suchzeichenfolge markierten Text eine Community-Suche aus. Weitere Informationen zum Ausführen einer Community-Suche finden Sie unter "Editor - Übersicht" auf Seite 59. • **XPath kopieren.** Kopiert in der Strukturansicht den XPath des ausgewählten Knotens in die Zwischenablage. Kopiert in der Textansicht den XPath des XML-Elements, in dem sich der Cursor befindet, in die Zwischenablage. • **Vollständigen Wert kopieren.** Kopiert in der Strukturansicht den vollständigen XML-Code des ausgewählten Knotens in die Zwischenablage. Kopiert in der Textansicht den vollständigen XML-Code des XML-Elements, in dem sich der Cursor befindet. • **XML-Prüfung einfügen.** Öffnet das Dialogfeld **XML-Prüfung einfügen**, in dem Sie einen **XML suchen**-Schritt in das Vuser-Skript einfügen können. Diese Option ist nur in der Antwortansicht verfügbar. Diese Option ist nur für Knoten vom Typ **Attribut** und **Wert** verfügbar. • **Wert in Parameter speichern.** Öffnet das Dialogfeld **Wert in Parameter speichern**, in dem Sie den ausgewählten Wert in einem einfachen Parameter speichern können. Diese Option ist nur in der Antwortansicht verfügbar. Diese Option ist nur für Knoten vom Typ **Attribut** und **Wert** verfügbar. • **XML in Parameter speichern.** Öffnet das Dialogfeld **Wert in Parameter speichern**, in dem Sie den ausgewählten Wert in einem XML-Parameter speichern können. Diese Option ist nur in der Antwortansicht verfügbar. • **Korrelation erstellen.** Öffnet die Registerkarte **Korrelation** in Design Studio. Der im Snapshot-Bereich markierte Text wird als manueller Korrelationseintrag in Design Studio angezeigt. Weitere Informationen finden Sie unter "Manuelles Korrelieren von Skripts" auf Seite 290. Diese Option ist nur in der Antwortansicht verfügbar.

	Diese Option ist in der Strukturansicht für Knoten vom Typ **Attribut** und **Wert** verfügbar und in der Textansicht, wenn der Text markiert ist. • **Korrelationsregel erstellen.** Öffnet das Dialogfeld **Als Regel hinzufügen**, in dem Sie den markierten Text als Teil einer Korrelationsregel hinzufügen können. Weitere Informationen finden Sie unter "Registerkarte "Korrelation" [Design Studio] - Übersicht" auf Seite 282. Diese Option ist nur in der Antwortansicht verfügbar. Diese Option ist in der Strukturansicht für Knoten vom Typ **Attribut** und **Wert** verfügbar und in der Textansicht, wenn der Text markiert ist.

Dialogfeld "Datenbankverbindung"

In diesem Dialogfeld können Sie eine Zeichenfolge erstellen, über die Sie eine Verbindung mit Ihrer Datenbank herstellen können.

Zugriff	Klicken Sie im Dialogfeld **Datenbankverbindung** auf den Generator für die Verbindungszeichenfolge.
Relevante Aufgaben	"Senden von Nachrichten über JMS" auf Seite 985
Siehe auch	"Dialogfeld "Generator für Verbindungszeichenfolge"" auf der nächsten Seite

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Datenbankverbindung	Öffnet den Generator für die Verbindungszeichenfolge. Weitere Informationen finden Sie unter "Dialogfeld "Generator für Verbindungszeichenfolge"" auf der nächsten Seite.
Schrittname.	Name oder IP-Adresse des Datenbankservers.
Verbindungszeichenfolge	Die zum Herstellen der Verbindung zur Datenbank verwendete Zeichenfolge. Verwenden Sie den Generator für die Verbindungszeichenfolge.
Datenanbieter	Der SQL-Anbieter: **OLEDB** oder **SQL**.

Dialogfeld "Generator für Verbindungszeichenfolge"

In diesem Dialogfeld können Sie eine Zeichenfolge erstellen, über die Sie eine Verbindung mit Ihrer Datenbank herstellen können.

Zugriff	Klicken Sie im Dialogfeld **Datenbankverbindung** auf den Generator für die Verbindungszeichenfolge.
Relevante Aufgaben	"Senden von Nachrichten über JMS" auf Seite 985
Siehe auch	"Dialogfeld "Datenbankverbindung"" auf der vorherigen Seite

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Verbindung testen...	Testet die Datenbankverbindung.
Generieren	Generiert die Zeichenfolge für die Datenbankverbindung und fügt sie im Dialogfeld **Datenbankverbindung** in das Feld **Verbindungszeichenfolge** ein.
Servername	Name oder IP-Adresse des Datenbankservers.
DB-Name	Der Name der Datenbank.
Authentifizierung	Die Authentifizierungsmethode für die Datenbank: **Windows-Authentifizierung** oder **Benutzer/Kennwort**. • **Benutzername und Kennwort**. Die Anmeldeinformationen für die Datenbank.
SQL-Anbieter	Der SQL-Anbieter: **OLEDB** oder **SQL**.

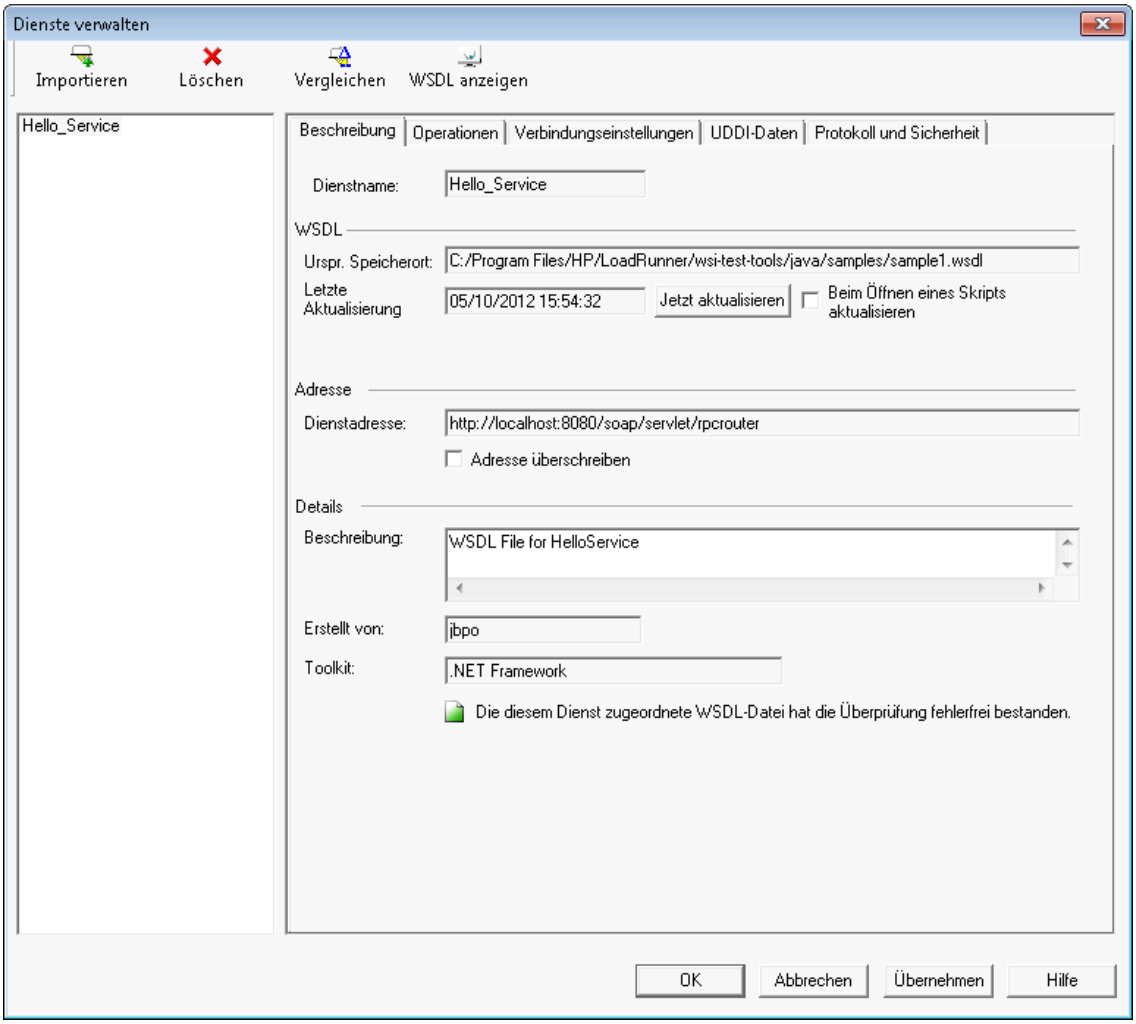
Webdienste - Verwalten von Diensten

Verwalten von Diensten – Übersicht

Im Fenster **Dienste verwalten** können Sie eine Liste von Diensteinträgen für das aktuelle Skript verwalten. Sie können die Diensteinträge anzeigen und ihre Eigenschaften festlegen.

Sie können der Liste durch Import von WSDL-Dateien Diensteinträge hinzufügen. Wenn Sie eine WSDL-Datei zur Liste hinzufügen, erstellt VuGen eine Arbeitskopie, die zusammen mit dem Skript gespeichert wird. Diese gilt nicht global. Aus diesem Grund müssen Sie für jedes Skript, das Sie erstellen, die gewünschten WSDL-Dateien importieren.

Das Fenster **Service Management** enthält Zugriffsschaltflächen, über die sich Dienste umgehend importieren, löschen und vergleichen lassen.



Über die Registerkarten können Sie die WSDL-Eigenschaften festlegen.

Beschreibung

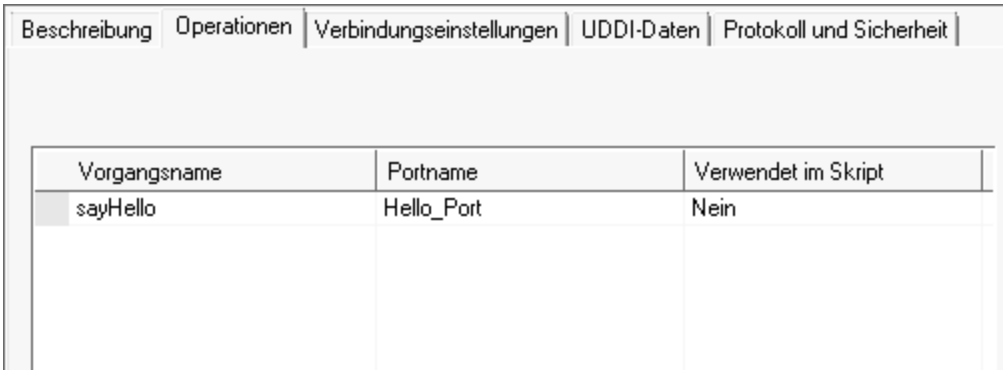
Auf der Registerkarte **Beschreibung** werden Informationen über den Dienst angezeigt:

- **Urspr. Speicherort.** Die ursprüngliche Quelle der WSDL-Datei (schreibgeschützt).
- **Dienstname.** Der Name des Webdienstes (schreibgeschützt).
- **Letzte Aktual. vom Original.** Das Datum, an dem die lokale Kopie zuletzt aus der ursprünglichen Quelle aktualisiert wurde (schreibgeschützt). Sie können die Version entweder manuell aktualisieren oder sie automatisch abrufen lassen, wenn Sie den Test wieder öffnen.
- **Dienstadresse.** Eine Endpunktadresse, an die die Anfrage gesendet wird. Falls erforderlich, können Sie auch den in der WSDL-Datei angegebenen Endpunkt überschreiben.

- **Erstellt von**. Der Name des Benutzers, der den Dienst ursprünglich importiert hat (schreibgeschützt).
- **Toolkit.** Das dem Skript zugewiesene Toolkit. Sie legen dies vor dem Import der ersten WSDL-Datei (schreibgeschützt) fest.

Vorgänge

Jeder der importierten Dienste kann mehrere Operationen definieren. Auf der Registerkarte **Operationen** wird angegeben, welche Operationen für den aus dem Bereich auf der linken Seite ausgewählten Dienst verwendet werden.



Verbindungseinstellungen

In einigen Fällen werden die WSDL-Dateien auf sicheren Sites gespeichert, für die eine Authentifizierung erforderlich ist. Manchmal wird auf WSDL-Dateien auch über einen Proxyserver zugegriffen.

VuGen unterstützt den sicheren Import von WSDL-Dateien und den Zugriff auf WSDL-Dateien über Proxyserver. Die folgenden Sicherheits- und Authentifizierungsmethoden werden unterstützt:

- SSL
- Standard- und NTLM-Authentifizierung
- Kerberos für das .NET-Toolkit

Weitere Informationen über das Konfigurieren von Verbindungsinformationen während des WSDL-Imports finden Sie unter "Hinzufügen und Verwalten von Diensten" auf Seite 1004.

UDDI-Daten

Sie können die Details zum UDDI-Server für jeden Dienst anzeigen, den Sie aus der UDDI-Registry importiert haben.

Die schreibgeschützten Informationen geben den URL des UDDI-Servers, die UDDI-Version und den Dienstschlüssel an.

Protokoll- und Sicherheitseinstellungen

Die Registerkarte **Protokoll und Sicherheit** enthält die Details des Sicherheitsszenarios, das auf das Skript angewendet wird. Wenn Sie kein Szenario ausgewählt haben, wird standardmäßig kein

Szenario angewendet (**<Kein Szenario>**). Weitere Informationen finden Sie unter "Hinzufügen und Verwalten von Diensten" auf Seite 1004.

In diesem Abschnitt werden außerdem folgende Themen behandelt:

- "Importieren von Diensten" unten
- "Vergleichsberichte" auf der nächsten Seite

Importieren von Diensten

VuGen ermöglicht Ihnen das Importieren von Diensten, um Tests auf hoher Ebene mit Webdienstaufrufschritten zu erstellen. Normalerweise besteht der erste Schritt beim Erstellen eines Skripts im Import einer WSDL-Datei.

Für den Importvorgang müssen die folgenden Informationen angegeben werden:

- **Quelle.** Die Quelle der WSDL-Datei: URL, Datei, UDDI oder Application Lifecycle Management. UDDI (Universal Description, Discovery, and Integration) ist ein universelles Repository für Dienste. Service Broker registrieren und kategorisieren veröffentlichte Webdienste und stellen Suchfunktionen zur Verfügung. Die UDDI-Business Registry ist ein Beispiel eines Service Brokers für Webdienste, die durch WSDL definiert werden.
- **Speicherort**. Der Pfad bzw. URL der WSDL-Datei, entweder manuell eingegeben oder aus der Verzeichnisstruktur ausgewählt.
- **Toolkit.** Das Toolkit, das permanent mit allen Diensten im Skript für alle folgenden Import- und Wiedergabevorgänge verknüpft wird (steht nur beim ersten Dienst zur Verfügung, der zum Skript hinzugefügt wird). Durch Auswahl eines Toolkits wird VuGen angewiesen, "echten" Clientverkehr mithilfe eines "echten" Toolkits – keiner Emulation – zu übermitteln.

  VuGen unterstützt Toolkits für .NET-Framework mit WSE 2, Version SP3, und Axis/Java-basierte Web Services Framework-Toolkits. Das Skript wird von VuGen mithilfe des tatsächlichen .NET- oder Axis-Toolkits importiert, aufgezeichnet und abgespielt. Standardmäßig verwendet VuGen eine automatische Erkennungsfunktion, um das am besten geeignete Toolkit zu ermitteln.
- **Verbindungseinstellungen.** Angaben zu Authentifizierung oder Proxyserver. Diese Einstellung ist hilfreich bei WSDLs, die auf abgesicherten Servern liegen oder auf die über einen Proxyserver zugegriffen werden muss.

Falls VuGen während des Imports ein Problem mit der WSDL-Datei feststellt, erhalten Sie eine Warnmeldung und werden aufgefordert, den Bericht zu öffnen. Der Bericht enthält die aufgetretenen Fehler mitsamt Detailinformationen.

Weitere Informationen über die Aufgabe finden Sie unter "Hinzufügen und Verwalten von Diensten" auf Seite 1004.

Vergleichsberichte

VuGen listet die Unterschiede zwischen den Dateien in einem Vergleichsbericht auf.

Sie können in den Vergleichseinstellungen festlegen, welche Elemente für den Vergleich nicht berücksichtigt werden sollen. Weitere Informationen finden Sie im "Dialogfeld "Vergleich von XML/WSDL"" auf Seite 1012.

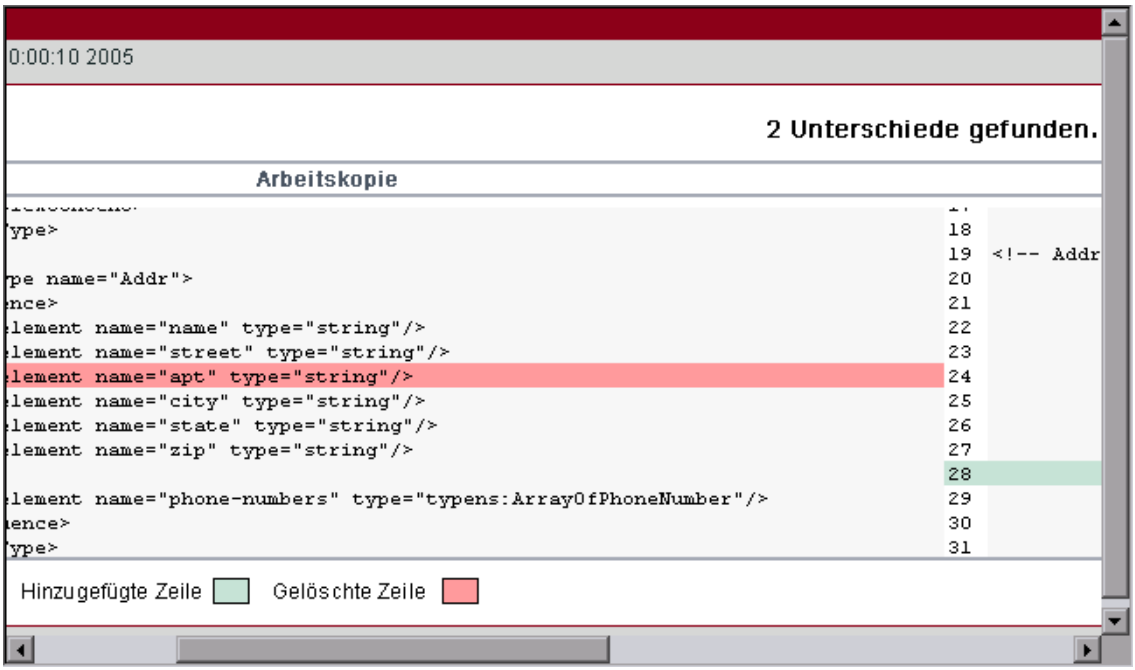
Vergleichsberichte für WSDL-Dateien bestehen aus zwei Spalten – **Arbeitskopie** und **Originaldatei**. **Arbeitskopie** bezieht sich auf die mit dem Skript gespeicherte WSDL-Arbeitskopie, während **Originaldatei** die WSDL am Originalspeicherort (Netzwerkdateipfad oder URL) bezeichnet.

In XML-Vergleichsberichten enthalten die Spalten jeweils den Pfad einer XML-Datei.

In Vergleichsberichten wird zur Markierung der Unterschiede folgende Legende verwendet:

- **Gelb.** Änderungen an einem vorhandenen Element (wird in beiden Versionen angezeigt).
- **Grün.** Ein neu hinzugefügtes Element (wird in der Originaldatei angezeigt).
- **Rosa.** Ein gelöschtes Element (wird in der Arbeitskopie angezeigt).

Im folgenden Beispiel wurde Zeile 24 aus der Originaldatei gelöscht und Zeile 28 wurde hinzugefügt.



Webreferenzanalyse

Viele WSDL-Dateien enthalten Verweise auf andere Dateien, z. B. XSD- und andere XML-Dateien. Vor dem Ausführen des Skripts empfiehlt es sich, die Verfügbarkeit dieser Dateien zu überprüfen.

Die WSDL-Referenzanalyse von VuGen prüft die WSDL auf Abhängigkeiten und zeigt diese im Hauptfenster und in einer eigenen Protokolldatei an.

Der Analyzer verpackt die WSDL-Datei und die referenzierten Dateien zusammen in ein Zip-Archiv. Die bestehenden Abhängigkeiten werden in einer Protokolldatei gespeichert, deren Dateipfad im Analyzer-Fenster angezeigt wird.

Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "WSDL-Referenzanalyse"" auf Seite 1012.

Weitere Informationen über die Aufgabe finden Sie unter "Analysieren von Abhängigkeiten in der WSDL-Datei" auf der nächsten Seite.

Hinzufügen und Verwalten von Diensten

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie eine Liste von Diensten erstellen, die Sie im Verlauf Ihres Tests aufrufen. Im Fenster **Dienste verwalten** können Sie Dienste importieren und die zugehörigen Einstellungen konfigurieren.

Öffnen des Dialogfelds "Dienste verwalten"

Wählen Sie **SOA-Tools > Dienste verwalten** oder klicken Sie auf die entsprechende Schaltfläche in der Symbolleiste, um das Dialogfeld **Dienste verwalten** zu öffnen.

Importieren eines Dienstes

Klicken Sie auf **Import**. Wählen Sie im Dialogfeld **Dienst importieren** die WSDL-Quelle aus und öffnen Sie den zugehörigen Speicherort.

Bei Importen vom Typ **URL** wird nach dem Klicken auf die Schaltfläche **Durchsuchen** ein neues Browserfenster geöffnet. Navigieren Sie zu der WSDL-Datei und schließen Sie den Browser. Der URL wird nun im Feld für den Speicherort übernommen. Details finden Sie im "Dialogfeld "Dienst importieren"" auf Seite 1010.

Wenn zur Nutzung des ausgewählten Dienstes eine Authentifizierung oder ein Proxy erforderlich ist, nehmen Sie die entsprechenden Einstellungen vor, bevor Sie die WSDL importieren. Erweitern Sie das Dialogfeld **Dienst importieren** und klicken Sie auf **Konfigurieren**. Details finden Sie im "Dialogfeld "Verbindungseinstellungen"" auf Seite 1009.

Wiederholen Sie diesen Schritt für alle Dienste, die während des Tests aufgerufen werden sollen.

Vertraut machen mit der WSDL

Machen Sie sich mit der WSDL vertraut. Schauen Sie sich die enthaltenen Details wie unter "Dialogfeld "Dienste verwalten"" auf Seite 1006 angegeben an.

Klicken Sie auf **WSDL anzeigen**, um die lokal gespeicherte WSDL-Datei im Internet Explorer zu öffnen und zu studieren.

Überprüfen auf Änderungen in der WSDL (optional)

Überprüfen Sie mit dem Vergleichstool, ob sich die WSDL seit dem letzten Import oder der letzten Aktualisierung geändert hat.

Legen Sie zunächst die Optionen für den Vergleich fest. Wählen Sie **SOA-Tools > SOA-Einstellungen > Vergleich von XML/WSDL** aus. Geben Sie an, welche Unterschiede ignoriert werden sollen. Weitere Informationen finden Sie unter "Dialogfeld "Vergleich von XML/WSDL"" auf Seite 1012.

Klicken Sie im Dialogfeld **Dienste verwalten** auf **Vergleichen**, um einen Bericht zu öffnen, in dem die Arbeitskopie der WSDL mit der Originaldatei verglichen wird.

Wenn Sie im Vergleichsbericht Abweichungen feststellen, klicken Sie auf **Jetzt aktualisieren**, um die aktuelle Version der WSDL vom Originalspeicherort abzurufen.

Überschreiben der Dienstadresse (optional)

Schauen Sie sich die Adresse im Feld **Dienstadresse** an. Das ist die standardmäßige Endpunktadresse, so wie sie aus der WSDL abgerufen wird. Wenn Sie sie überschreiben möchten, klicken Sie auf **Adresse überschreiben** und geben Sie eine andere Endpunktadresse für die Dienstanfrage ein.

Um die Standardadresse wiederherzustellen, deaktivieren Sie die Option **Adresse überschreiben**. Details finden Sie im "Dialogfeld "Dienste verwalten"" auf der nächsten Seite.

Festlegen eines Sicherheitsszenarios (optional)

Klicken Sie auf die Registerkarte **Protokoll und Sicherheit**, um WS-Security oder ein anderes Sicherheitsszenario zu verwenden.

Weitere Informationen finden Sie unter "Webdienste - Sicherheit" auf Seite 1013.

Analysieren von Abhängigkeiten in der WSDL-Datei

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie mithilfe der Referenzanalyse Abhängigkeiten in der WSDL-Datei ausfindig machen können. Informationen zur Benutzeroberfläche finden Sie unter "Dialogfeld "WSDL-Referenzanalyse"" auf Seite 1012.

1. **Öffnen der Referenzanalyse**

   Wählen Sie **SOA-Tools > WSDL-Referenzanalyse** aus.

2. **Auswählen von Quelle und Ziel**

   Geben Sie im Feld **Auswählen der WSDL-Datei** den Speicherort der WSDL an, die analysiert werden soll.

   Geben Sie im Feld **Ausgabedateipfad** den Speicherort für die Zip-Datei an.

3. **Starten der Analyse**

Klicken Sie auf **Analyse starten**. Die Analysefunktion listet alle Abhängigkeiten zusammen mit den zugehörigen Pfaden im Ausgabefenster auf.

4. **Anzeigen des Protokolls**

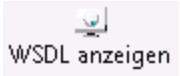
   Zeigen Sie die Ergebnisse im Protokollfenster an. Um die Ergebnisse zu löschen und eine weitere Analyse auszuführen, klicken Sie auf **Protokoll löschen**.

Dialogfeld "Dienste verwalten"

In diesem Dialogfeld können Sie Ihre WSDLs verwalten, Authentifizierungsinformationen angeben und ein Sicherheitsszenario festlegen.

Zugriff	Führen Sie einen der folgenden Schritte durch: • Klicken Sie in der VuGen-Symbolleiste auf die Schaltfläche **Dienste verwalten** . • **SOA-Tools > Dienste verwalten**
Relevante Aufgaben	"Hinzufügen und Verwalten von Diensten" auf Seite 1004

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
	Öffnet das Dialogfeld **Dienst importieren**.
	Entfernt den ausgewählten Dienst aus der Liste.
	Öffnet den WSDL-Vergleichsbericht, in dem die Arbeitskopie und die Originalkopie der WSDL nebeneinander angezeigt werden. Informationen zum Konfigurieren der Vergleichseinstellungen finden Sie unter "Dialogfeld "Vergleich von XML/WSDL"" auf Seite 1012.
	Zeigt die WSDL in einem Browser an.
<WSDL-Liste>	Eine Liste der importierten WSDLs.

Registerkarte "Beschreibung"	Zeigt Informationen zur WSDL an: Endpunktadresse, Toolkit und Aktualisierungsinformationen.
Registerkarte "Operationen"	Listet die Operationen des Dienstes auf: **Vorgangsname**, **Portname** und **Verwendet im Skript** (**Ja** oder **Nein**). Klicken Sie auf einen der Spaltenköpfe, um die Vorgänge anhand der Daten in dieser Spalte zu sortieren. Wenn Sie noch einmal darauf klicken, wird die Sortierreihenfolge umgekehrt.
Registerkarte "Verbindungseinstellungen"	Hier können Sie Authentifizierungseinstellungen für den Computer festlegen, von dem Sie einen Dienst importieren. Weitere Informationen finden Sie im "Dialogfeld "Dienste verwalten"" auf der vorherigen Seite. **Hinweis:** Dies gilt nur für URL- und UDDI-Importe.
Registerkarte "UDDI-Daten"	UDDI-Server, UDDI-Version und Dienstschlüssel. Weitere Informationen finden Sie im "Dialogfeld "Dienste verwalten"" auf der vorherigen Seite.
Registerkarte "Protokoll- und Sicherheitseinstellungen"	Hier können Sie Sicherheitsszenarios für Ihre Webdienstaufrufe anzeigen und festlegen. Weitere Informationen finden Sie weiter unten.

Registerkarte "Beschreibung"

Die folgenden Elemente werden auf der Registerkarte **Beschreibung** angezeigt:

Element der Oberfläche	Beschreibung
Jetzt aktualisieren	Ruft die aktuelle Version der WSDL von ihrem Originalspeicherort ab.
Erstellt von	Der Name, mit dem Sie sich angemeldet haben. Sie können dieses Feld bearbeiten und einen anderen Namen angeben. Dies kann bei der Sortierung der Dienste in Berichten hilfreich sein (schreibgeschützt).
Beschreibung	Ein bearbeitbares Feld, in das Sie Informationen zum Dienst eingeben können.
Letzte Aktualisierung vom Original	Zeitpunkt der letzten WSDL-Aktualisierung (schreibgeschützt).
Urspr. Speicherort	Der ursprüngliche Speicherort, von dem die WSDL importiert wurde (schreibgeschützt).
Adresse überschreiben	Hier können Sie einen anderen Endpunkt für den Dienst im Feld **Dienstadresse** eingeben.

Dienstadresse	Der aus der WSDL-Datei geladene Endpunkt des Dienstes, an den die Anforderungen gerichtet werden (schreibgeschützt). Wählen Sie **Adresse überschreiben** aus, um die Standardadresse zu überschreiben.
Dienstname	Der systemeigene Dienstname in der WSDL-Datei, der standardmäßig beim Dienstimport angezeigt wird (schreibgeschützt).
Toolkit	Das dem Skript zugewiesene Toolkit. Sie legen das Toolkit beim Import des Dienstes fest (schreibgeschützt).
Beim Öffnen eines Skripts aktualisieren	Aktualisiert die WSDL anhand der Originalversion bei jedem Öffnen des Skripts.

Registerkarte "Verbindungseinstellungen"

Die folgenden Elemente sind enthalten:

Element der Oberfläche	Beschreibung
Authentifizierung	**Authentifizierungseinstellungen verwenden:** Hier können Sie die Anmeldeinformationen für die Authentifizierung eingeben. • **Benutzername, Kennwort.** Der Benutzername und das Kennwort, die zum Abrufen der WSDL verwendet werden. **Tipp:** Geben Sie für Benutzer außerhalb der Standarddomäne den Domänennamen vor dem Benutzernamen an. Beispiel: `domäne1/alex_qc`.
Proxy	**Proxyeinstellungen verwenden**. Hier können Sie Details und Anmeldeinformationen für den Proxy eingeben. • **Server.** Name oder IP-Adresse des Proxyservers. • **Port.** Port für den Zugriff auf die WSDL. • **Benutzername, Kennwort.** Benutzername und Kennwort, die zur Authentifizierung verwendet werden. Geben Sie für Benutzer außerhalb der Standarddomäne den Domänennamen vor dem Benutzernamen an. Beispiel: `domäne1/alex_qc`.

Registerkarte "UDDI-Daten"

Die folgenden Elemente sind enthalten:

Element der Oberfläche	Beschreibung

Dienstschlüssel	Eine eindeutige Kennzeichnung für den Dienst auf dem UDDI-Server, anhand derer die Dienstdefinition bei der Aktualisierung des Dienstes ausfindig gemacht werden kann.
UDDI-Server	Die URL-Adresse und Version des UDDI-Servers, von dem die Dienstdefinition importiert wird.
UDDI-Version	Die Version der UDDI-Registrierung: 2 oder 3.

Dialogfeld "Verbindungseinstellungen"

Hier können Sie Anmeldeinformationen für die Authentifizierung sowie Informationen zum Proxyserver für den Computer angeben, auf dem die WSDL gespeichert ist.

Zugriff	• Bei einem neuen Dienst: Wählen Sie **SOA-Tools > Dienste verwalten** aus. Klicken Sie auf die Schaltfläche **Importieren**. Klicken Sie im Dialogfeld **Dienst importieren** auf **Verbindungseinstellungen**. • Bei vorhandenen Diensten: Wählen Sie im Dialogfeld **Dienste verwalten** einen Dienst aus, und klicken Sie dann auf die Registerkarte **Verbindungseinstellungen**.
Wichtige Informationen	Steht nur für Dienste zur Verfügung, die über einen URL oder UDDI importiert wurden.
Relevante Aufgaben	"Hinzufügen und Verwalten von Diensten" auf Seite 1004

Die folgenden Elemente sind enthalten:

Element der Oberfläche	Beschreibung
Authentifizierung	**Authentifizierungseinstellungen verwenden:** Hier können Sie die Anmeldeinformationen für die Authentifizierung eingeben. • **Benutzername, Kennwort.** Benutzername und Kennwort, die zum Abrufen der WSDL verwendet werden. **Tipp:** Geben Sie für Benutzer außerhalb der Standarddomäne den Domänennamen vor dem Benutzernamen an. Beispiel: `domäne1/alex_qc`.

Proxy	**Proxyeinstellungen verwenden**. Hier können Sie Details und Anmeldeinformationen für den Proxy eingeben. • **Server.** Name oder IP-Adresse des Proxyservers. • **Port.** Port für den Zugriff auf die WSDL. • **Benutzername, Kennwort.** Benutzername und Kennwort, die zur Authentifizierung verwendet werden. Geben Sie für Benutzer außerhalb der Standarddomäne den Domänennamen vor dem Benutzernamen an. Beispiel: `domäne1/alex_qc`.

Dialogfeld "Dienst importieren"

Hier können Sie WSDLs aus einem Dateisystem, über einen URL, Application Lifecycle Management, UDDI oder über Systinet importieren.

Zugriff	Führen Sie einen der folgenden Schritte durch: • Wählen Sie **Dienste > Neu > Dienst importieren** aus. • Wählen Sie im Kontextmenü **Neu > Dienst importieren** aus.
Relevante Aufgaben	"Hinzufügen und Verwalten von Diensten" auf Seite 1004

Die folgenden Elemente sind enthalten:

Element der Oberfläche	Beschreibung
	Durchsuchen. Je nachdem, was Sie als Importquelle ausgewählt haben, können Sie hier im Dateisystem, über einen Browser, in der UDDI-Registrierung oder dem Application Lifecycle Management-Repository nach einem Dienst suchen.
	Öffnet das Dialogfeld **Verbindungseinstellungen**, wo Sie die Einstellungen für Authentifizierung und Proxy für den Server konfigurieren können, auf dem die WSDL gespeichert ist. Weitere Informationen finden Sie unter "Dialogfeld "Verbindungseinstellungen"" auf der vorherigen Seite.
	Hier können Sie ein Toolkit für den Test auswählen. Optionen: **Automatisch**, **.NET** oder **Axis**. Bei **Automatisch** wird anhand eines Algorithmus das am besten geeignete Toolkit ermittelt.
	Startet den Importvorgang.

Forts.

WSDL auswählen von	Speicherort der WSDL. Suchen Sie nach den entsprechenden Angaben oder geben Sie sie von Hand ein: • **URL:** Vollständiger URL. Geben Sie hier einen vollständigen URL ein, keine verkürzte Version. • **Datei:** Vollständiger Pfad und Dateiname. • **UDDI:** ID in der UDDI-Registrierung. Die Durchsuchen-Schaltfläche öffnet das "Dialogfeld "Nach einem Dienst in UDDI suchen"" unten.

Dialogfeld "Nach einem Dienst in UDDI suchen"

In diesem Dialogfeld können Sie gezielt nach einem bestimmten Dienst innerhalb der UDDI-Registrierung suchen.

Zugriff	• Klicken Sie im Dialogfeld **Dienste verwalten** auf **Importieren**. • Wählen Sie im Dialogfeld **Importieren** im Abschnitt **WSDL auswählen von** die Option **UDDI** aus. • Klicken Sie auf .
Relevante Aufgaben	"Hinzufügen und Verwalten von Diensten" auf Seite 1004

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
	Startet die Suche nach einem Dienst anhand des Textes, der im Feld **Der vollst. Dienstname oder ein Teil davon** eingegeben wurde.
<Dienstliste>	Eine alphabetisch geordnete Liste aller Liste, die mit der Suchzeichenfolge übereinstimmen. Die Tabelle hat die folgenden Spalten: **Dienstname**, **Dienstschlüssel**, **Dienstbeschreibung**, **Dienst-WSDL**.

Forts.

Der vollst. Dienstname oder ein Teil davon	Zeichenfolge mit dem Namen des gesuchten Dienstes oder einem Teil davon. Es ist hier nicht erforderlich, Platzhalterausdrücke anzugeben. Die Suche kann durch die folgenden Optionen eingegrenzt werden: • **Genaue Übereinstimmung.** Der Dienstname muss genau mit der eingegebenen Zeichenfolge übereinstimmen. • **Groß-/Kleinschreibung beachten.** Bei der Suche wird nach Groß- oder Kleinschreibung unterschieden.
UDDI-Serverabfrageadresse	Der vollständige Pfad für die Abfrage auf dem UDDI-Server.
UDDI Version 2/3	Die UDDI-Version des Dienstes, der in der Liste angezeigt werden soll.

Dialogfeld "Vergleich von XML/WSDL"

In diesem Dialogfeld können Sie die Einstellungen für den Vergleich zwischen mehreren Versionen einer WSDL-Datei konfigurieren. So können Sie das Vergleichstool beispielsweise anweisen, bestimmte Unterschiede, wie etwa Groß-/Kleinschreibung, Kommentare etc., zu ignorieren.

Zugriff	**SOA-Tools > SOA-Einstellungen > Vergleich von XML/WSDL**.
Relevante Aufgaben	"Hinzufügen und Verwalten von Diensten" auf Seite 1004

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Nur Unterschiede anzeigen	Im Bericht werden nur die Unterschiede angezeigt, keine Übereinstimmungen.
Groß-/Kleinschreibung ignorieren	Unterschiede bei der Groß-/Kleinschreibung werden nicht als Unterschiede angezeigt.
Kommentare ignorieren	Unterschiede bei den Kommentaren werden ignoriert.
Verarbeitungsanweis. ignorieren	Unterschiede bei den Verarbeitungsanweisungen werden ignoriert.
Namespaces ignorieren	Unterschiede bei den Namespaces werden ignoriert.

Dialogfeld "WSDL-Referenzanalyse"

In diesem Dialogfeld können Sie die Abhängigkeiten ermitteln, die in einer WSDL-Datei bestehen.

Zugriff	**SOA-Tools > WSDL-Referenzanalyse**
Relevante Aufgaben	"Analysieren von Abhängigkeiten in der WSDL-Datei" auf Seite 1005

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
	Startet die Analyse und gibt die Ergebnisse im Protokollfenster aus.
	Löscht den Inhalt des Protokollfensters und der Protokolldatei.
	Öffnet den Ordner mit der Ausgabedatei.
<Protokollfenster>	Ein laufendes Protokoll der aktuell ausgeführten Referenzanalyse.
WSDL-Datei auswählen	Der lokale Pfad oder URL der zu analysierenden WSDL-Datei.
Ausgabedateipfad	Der Speicherort für die auszugebende Zip-Datei.

Webdienste - Sicherheit

Konfigurieren der Sicherheitseinstellungen – Übersicht

Die große Herausforderung liegt in der Erstellung von skalierbaren und zugleich sicheren Webdienstanwendungen. Einen sicheren Webdienst erhalten Sie, wenn Sie die Nachricht über ein sicheres Übertragungsprotokoll wie Secure Sockets Layer (SSL) versenden, was allerdings nur bei Punkt-zu-Punkt-Verbindungen möglich ist.

Für den sicheren Versand Ihrer Nachrichten unterstützt VuGen mehrere Sicherheitsmechanismen – Sicherheitstoken (WS-Security) und SAML.

Weitere Informationen zu Token finden Sie weiter unten unter . Weitere Informationen zu SAML finden Sie unter "SAML-Sicherheitsoptionen" auf Seite 1017.

In der Tabelle unten finden Sie Hinweise zur Verwendung der beiden Modelle.

Legacy-Modell	Szenariobasiertes Modell
Sie arbeiten mit einem Skript, das auf dem Legacy-Modell basiert.	Sie testen einen WCF-Dienst.

Sie testen einen in Framework-Anwendungen wie .NET 2.0, Axis oder sonstigen Toolkits erstellten Dienst.	Sie testen einen Dienst, der in einem aktuellen Framework, z. B. Axis2 oder Metro (WSIT), erstellt wurde.
Sie benötigen für WS-Security-Token lediglich Steuerungsmöglichkeiten der unteren Ebene.	Ihr Dienst verwendet erweiterte Spezifikationen wie WS-SecureConversation oder WS-Trust.
Sie haben Probleme mit der Verwendung des neuen Modells oder sind der Meinung, dass die Funktionen des Legacy-Modells Ihrem Bedarf besser entsprechen.	Sie haben Probleme mit der Verwendung des Legacy-Modells oder sind der Meinung, dass die Funktionen des neuen Modells besser geeignet sind.

Hinweis: Wenn sich Ihre WSDL an einem gesicherten Speicherort befindet, müssen Sie die Sicherheitsinformationen im Dialogfeld **Dienste verwalten** bereitstellen. Weitere Informationen finden Sie im "Dialogfeld "Verbindungseinstellungen"" auf Seite 1009.

Sicherheitstoken und Verschlüsselung

Mithilfe der WS-Security-Spezifikationen können Sie in die eigentliche SOAP-Nachricht Sicherheitsanmeldeinformationen einfügen. Sie erreichen dies, indem Sie einen Client anweisen, Sicherheitsanmeldeinformationen von einer für den Absender und den Empfänger vertrauenswürdigen Quelle abzurufen. Wenn der Absender einer SOAP-Nachricht eine Anfrage sendet, werden diese Sicherheitsanmeldeinformationen, auch **Sicherheitstoken** genannt, in die SOAP-Nachricht eingefügt. Sobald die SOAP-Anfrage beim Webserver eingeht, muss dieser keine weiteren Anfragen zur Überprüfung der Integrität des Absenders mehr aussenden. Der Server überprüft die Authentizität der Anmeldeinformationen, bevor der Webdienst die Anwendung ausführen darf. Die Tatsache, dass die Quelle der Anmeldeinformationen nicht erneut überprüft werden muss, erhöht die Skalierbarkeit der Anwendung beträchtlich.

Zur weiteren Erhöhung der Sicherheit von Webdiensten werden für SOAP-Nachrichten häufig digitale Signaturen oder Verschlüsselungsmethoden eingesetzt. Eine digitale Signatur gewährleistet einem Empfänger, dass die SOAP-Nachricht während der Übermittlung nicht geändert wurde. Eine Verschlüsselung der SOAP-Nachricht erhöht die Sicherheit eines Webdiensts insofern, als eine andere Person als der beabsichtigte Empfänger Schwierigkeiten beim Lesen des Nachrichteninhalts haben wird.

Der Sicherheitsmechanismus für Webdienste weist den Nachrichten Sicherheitstoken zu. Dieser Mechanismus unterstützt verschiedene Sicherheitstokenformate und kann somit zahlreiche Authentifizierungsanforderungen erfüllen. Es kann beispielsweise vorkommen, dass ein Client einen Identitätsnachweis oder ein Sicherheitszertifikat bereitstellen muss.

Zur Unterstützung von WS-Security ermöglicht VuGen die Erstellung von Sicherheitstoken für Ihr Skript. Sie können mehrere Token erstellen und Eigenschaften für sie festlegen. Nach der Erstellung können Sie den Token zum Signieren oder Verschlüsseln einer SOAP-Nachricht verwenden.

Manchmal müssen Sie den Token nicht explizit versenden; Sie verwenden ihn dann lediglich zum Signieren oder Verschlüsseln, ohne ihn in tatsächlich in die Kopfzeile des SOAP-Umschlags einzubinden. Mit der Option **Add** können Sie angeben, ob der Token tatsächlich versendet werden soll.

Die verfügbaren Token sind **Username and Password**, **X.509 Certificate**, **Kerberos Ticket**, **Kerberos2 Ticket**, **Security Context Token** und **Derived Token**. Die von Ihnen anzugebenden Informationen variieren je nach Token.

- **User Name and Password.** Der Token **User Name and Password** enthält Informationen zur Identifizierung von Benutzern zum Zwecke der Authentifizierung: **Benutzername** und **Kennwort**.

Sie können zudem anhand der Kennwortoptionen angeben, wie das Kennwort zur Authentifizierung an den Server übermittelt werden soll: **SendPlainText**, **SendNone** oder **SendHashed**.

- **X.509 Certificate.** Dieser Sicherheitstoken basiert auf einem X.509-Zertifikat. Zertifikate können Sie entweder über eine Zertifizierungsstelle wie VeriSign Inc. beziehen oder Sie richten einen eigenen Zertifizierungsdienst ein. Die meisten Windows-Server unterstützen die Public-Key-Infrastruktur (PKI) für die Erstellung von Zertifikaten. Diese können Sie dann entweder durch eine Zertifizierungsstelle signieren lassen oder Sie können unsignierte Zertifikate verwenden.

Wenn Sie einen X.509-Token zu einem Vuser-Skript hinzufügen, geben Sie folgende Argumente an: **LogicalName**, **Store Name**, **Key identifier type**, **Key identifier value** und **Store Location**.

- **Kerberos Ticket/Kerberos2 Ticket.** (Ab Windows 2003 bzw. XP SP1.) Das Kerberos-Protokoll wird für die gegenseitige Authentifizierung von Benutzern und Diensten in einem offenen, unsicheren Netzwerk verwendet. Es verschlüsselt und signiert die Anmeldedaten von Benutzern über gemeinsame geheime Schlüssel. Die Anmeldeinformationen werden von einer dritten Partei, einem so genannten KDC (Kerberos Key Distribution Center) authentifiziert. Nach der Authentifizierung kann der Benutzer ein Ticket für den Zugriff auf einen oder mehrere Dienste im Netzwerk anfordern. Das Ticket enthält die verschlüsselte, authentifizierte Identität des Benutzers. Die Tickets werden über die Anmeldeinformationen des aktuellen Benutzers angefordert.

VuGen unterstützt Token, die auf Kerberos- und Kerberos2-Sicherheitstoken basieren. Kerberos- und Kerberos2-Token unterscheiden sich in erster Linie dadurch, dass Kerberos2 die Security Support Provider-Schnittstelle (SSPI) verwendet und für den Wechsel der Clientidentität keine erhöhten Berechtigungen benötigt. Darüber hinaus lassen sich Kerberos2-Sicherheitstoken zum Sichern von SOAP-Nachrichten verwenden, die an einen in einer Webfarm ausgeführten Webdienst gesendet werden.

Wenn Sie einen Kerberos-Token an das Vuser-Skript senden, geben Sie einen **logischen Namen** für den Token sowie den **Hostnamen** und den **Domänennamen** des Webdienstcomputers an.

- **Security Context Token.** Hierbei handelt es sich um Sicherheitstoken, die bis zu ihrem Ablauf beliebig oft eingesetzt werden können. Absender von SOAP-Nachrichten können

Sicherheitskontexttoken zum Signieren und/oder Verschlüsseln einer Reihe von SOAP-Nachrichten verwenden, die zwischen einem Absender und einem empfangenden Webdienst übertragen werden (Unterhaltung). Dieser Tokentyp hat folgende Vorteile:

- Solange der Sicherheitskontexttoken gültig ist, kann ihn der Absender der SOAP-Nachricht zum Signieren und/oder Verschlüsseln von SOAP-Nachrichten einsetzen, die er an den Ziel-Webdienst sendet.
- Sicherheitskontexttoken basieren auf einem symmetrischen Schlüssel, wodurch sie für die digitale Signierung bzw. Verschlüsselung von SOAP-Nachrichten effizienter eingesetzt werden können als asymmetrische Schlüssel.
- Sicherheitskontexttoken können bei einem Sicherheitstokendienst durch Senden einer SOAP-Nachricht an einen anderen Sicherheitstokendienst angefordert werden.

Wenn Sie einen **Sicherheitskontext** zum Vuser-Skript hinzufügen, geben Sie Werte für die folgenden Argumente an: **LogicalName**, **Base Token**, **Issuer Token**, **End Point URI** und **Add applies to**.

- **Derived Token.** Bei einem abgeleiteten Token handelt es sich um einen Token, der auf einem bereits vorhandenen Token basiert, mit Ausnahme von X.509-Token, die keine Ableitung unterstützen. Machen Sie folgende Angaben: **LogicalName** und **Derived From**-Token. Wenn Sie den Ursprungstoken entfernen, ist auch der abgeleitete Token nicht mehr verfügbar. Hinweis: Abgeleitete Token lassen sich nicht rekursiv verwenden.

Weitere Informationen zum Tokenattribut finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Hinzufügen der Sicherheitsrichtlinie

Wenn Sie einem Abschnitt Ihres Skripts eine Sicherheitsrichtlinie hinzufügen möchten, schließen Sie die entsprechenden Schritte in die Schritte **Webdienst - Sicherheitseinstellungen festlegen** und **Webdienst - Auswahl der Sicherheitseinstellungen aufheben** ein.

Wenn Sie einen Schritt vom Typ **Webdienst - Sicherheitseinstellungen festlegen** zu einem Schritt Ihres Skripts hinzufügen, fügt VuGen eine **web_service_set_security**-Funktion ein, die Argumente mit den Token, Nachrichtensignaturen und der Verschlüsselung enthält, die Sie definiert haben.

```
web_service_set_security(
   SECURITY_TOKEN, "Type=USERNAME", "TokenName=mytoekn1",    "UserName=bob", "Pa
ssword=123", "PasswordOptions=SendNone",    "Add=True", LAST);
```

Folgende Argumente unterstützen keine Parametrierung: **Token Type, Logical Name, Base Token, Issuer Token** oder **Derive From**.

Arbeiten mit Nachrichtensignaturen und verschlüsselten Daten

Wenn Sie einen Sicherheitstoken zu einer SOAP-Nachricht hinzufügen, wird dieser in Form eines XML-Elements zur WS-Security-SOAP-Kopfzeile der Nachricht hinzugefügt.

Die Nachricht wird jedoch unverschlüsselt dargestellt und erfordert daher zusätzliche Sicherheit. Dies gilt insbesondere bei unverschlüsseltem Versand der Anmeldeinformationen, einschließlich des Kennworts, wie es bei rollenbasierter Sicherheit der Fall ist.

Es gibt zwei Methoden des Schutzes von Daten, digitale Signaturen und Verschlüsselung.

- **Digitale Signaturen.** Anhand von digitalen Signaturen können Nachrichtenempfänger verifizieren, dass Nachrichten seit ihrer Signierung nicht geändert wurden. Die digitale Signatur innerhalb der SOAP-Nachricht weist in der Regel das XML-Format auf. Der Empfänger überprüft die Signatur und vergewissert sich, dass sie gültig ist. In gewissen Umgebungen, wie z. B. unter WSE, wird die Signatur auf dem Empfängercomputer automatisch überprüft.

- **Verschlüsselung.** Anhand von digitalen XML-Signaturen kann zwar sichergestellt werden, dass eine SOAP-Nachricht nach dem Signieren nicht verändert wurde, eine Signatur bietet jedoch keine Verschlüsselungsmechanismus. Die Nachricht befindet sind weiterhin im unverschlüsselten XML-Format. Um die Nachricht so zu schützen, dass sie nicht dargestellt wird, sollten Sie sie verschlüsseln. Dadurch wird verhindert, dass unbefugte Personen das Kennwort des Benutzers sehen und in dessen Besitz gelangen können.

Parametrierung wird von Nachrichtensignaturen und Verschlüsselungsargumenten nicht unterstützt. Weitere Informationen zum Hinzufügen von Nachrichtensignaturen und Verschlüsselungstechniken zu Ihrem Skript finden Sie unter "Hinzufügen von Sicherheitsfunktionen zu einem Webdienstskript" auf Seite 1035.

SAML-Sicherheitsoptionen

VuGen unterstützt SAML (Security Assertion Markup Language) für Webdienste. SAML ist ein XML-Standard für den Austausch von sicherheitsbezogenen Informationen, so genannter **Assertionen**, zwischen Geschäftspartnern über das Internet. Die Assertionen können Attributanweisungen, Authentifizierung, bedingte Anweisungen und bedingte Autorisierungsanweisungen enthalten.

SAML verwendet eine vermittelte Authentifizierung mit einem vom STS (Security Token Service) ausgegebenen Sicherheitstoken. Der STS ist ein für den Client und den Webdienst vertrauenswürdiger Dienst zur Bereitstellung gemeinsam verwendbarer Sicherheitstoken. SAML-Token sind die für Webdienstsicherheit von Bedeutung, da sie eine plattformübergreifende Interoperabilität sowie die Möglichkeit des Informationsaustauschs zwischen Clients und Diensten bereitstellen, die sich in unterschiedlichen Sicherheitsdomänen befinden.

Sie können die SAML-Einstellungen für ein gesamtes Skript oder Teile davon festlegen. Weitere Informationen finden Sie unter "Hinzufügen von SAML-Sicherheit" auf Seite 1036.

Hinweis: SAML-Sicherheit und die Standardsicherheit für Webdienste (d. h. ein **Webdienst - Sicherheitseinstellungen festlegen**-Schritt) können nicht demselben Schritt zugewiesen werden. Fügen Sie zum Entfernen der Webdienstsicherheit einen **Webdienst - Auswahl der Sicherheitseinstellungen aufheben**-Schritt ein.

Signieren von SAML-Assertionen

VuGen stellt eine Methode zum Signieren von SAML-Assertionen bereit. Als Eingabe sind die unsignierte Assertion, eine Zertifikatsdatei sowie das optionale Kennwort erforderlich. Als Ergebnis gibt VuGen anschließend eine signierte SAML-Assertion aus. Weitere Informationen über die Aufgabe finden Sie unter "Hinzufügen von SAML-Sicherheit" auf Seite 1036.

Richtliniendateien

SAML-Richtliniendateien entsprechen dem WSE 3.0-Standard und definieren die Attributwerte für die SAML-Sicherheit. Standardmäßig verwendet VuGen die im Ordner **dat** des Installationsverzeichnisses befindliche Datei **samlPolicy.config**.

Sie können SAML-Sicherheitsinformationen manuell in den Eigenschaftendialog eingeben oder einen Verweis auf eine Richtliniendatei erstellen, die sämtliche Sicherheitsinformationen enthält. Sie können auf der Grundlage der Datei **samlPolicy.config** Ihre eigene Richtliniendatei erstellen.

Sie können die Richtliniendatei dahingehend ändern, dass sie Werte für die Sicherheitsparameter, wie z. B. den Benutzernamen und die Zertifikatsinformationen, enthält. Wenn Sie in Ihr Skript einen SAML-Sicherheitsschritt einfügen und dabei explizite Werte für die Sicherheitsargumente angeben, werden die Werte in der Richtliniendatei überschrieben.

Bei Änderungen an der Standardrichtliniendatei empfiehlt es sich, die neue Richtliniendatei in den Ordner des Skripts zu kopieren. Speichern Sie die benutzerdefinierten Richtliniendateien unter der Erweiterung **.config**, damit sie auch dann mit dem Skript verbunden bleiben, wenn sie auf anderen Computern ausgeführt oder über LoadRunner Controller aufgerufen werden.

Im SAML STS-Beispiel auf der MSDN-Website erfahren Sie mehr über SAML-Richtliniendateien. Wenn Sie das Verhalten einer SAML-Föderation emulieren möchten, kopieren Sie die Datei **samlFederationPolicy.config** aus dem Datenordner in den Ordner Ihres Skripts und geben Sie sie als Richtliniendatei an.

Sicherheitsszenarios – Übersicht

VuGen ermöglicht Ihnen den Test von Webdiensten, die erweiterte Sicherheit und WS-Spezifikationen verwenden. Diese Art von Diensten können auf unterschiedlichen Plattformen geschrieben werden, z. B. WCF (Windows Communication Foundation), Metro (WSIT) und Axis2. Für WCF-Dienste unterstützt VuGen darüber hinaus proprietäre Standards und Transportmethoden.

Sie können die Unterstützung durch Einrichten eines Sicherheitsszenarios aktivieren. Jedes Szenario stellt eine typische Umgebung dar, in der Webdienstaufrufe verwendet werden. VuGen stellt eine Reihe integrierter, allgemeingültiger Sicherheitsszenarios zur Verfügung. Die Szenarioeinstellungen werden einzeln auf jeden der Dienste angewendet.

Bei den integrierten Szenarios können Sie über die Benutzeroberfläche je nach Bedarf Sicherheitsinformationen eingeben. Darüber hinaus können Sie erweiterte Einstellungen, z. B. für Sicherheit, Transportmethode und Proxyserver, konfigurieren.

Wenn keines der Szenarios zu Ihrer Umgebung passt, können Sie das allgemeine benutzerdefinierte Szenario verwenden.

Eine praktische Anleitung zum Auswählen eines Szenarios finden Sie unter Tips and Guidelines.

Auswählen eines Sicherheitsmodells

VuGen unterstützt zwei Modelle für die Konfiguration der Sicherheitsfunktionen für Ihre Webdienstaufrufe: *Legacy* und *Scenario*. In diesemKapitel wird das Szenario-Sicherheitsmodell vorgestellt. Das Legacy-Modell sieht das manuelle Hinzufügen von Schritten vom Typ **Webdienst - Sicherheitseinstellungen festlegen** bzw. der Funktion **web_service_set_security** vor.

In der Tabelle unten finden Sie Hinweise zur Verwendung der beiden Modelle.

Legacy-Modell	Szenariobasiertes Modell
Sie arbeiten mit einem Skript, das auf dem Legacy-Modell basiert.	Sie testen einen WCF-Dienst.
Sie testen einen in Framework-Anwendungen wie .NET 2.0, Axis oder sonstigen Toolkits erstellten Dienst.	Sie testen einen Dienst, der in einem aktuellen Framework, z. B. Axis2 oder Metro (WSIT), erstellt wurde.
Sie benötigen für WS-Security-Token lediglich Steuerungsmöglichkeiten der unteren Ebene.	Ihr Dienst verwendet erweiterte Spezifikationen wie WS-SecureConversation oder WS-Trust.
Sie haben Probleme mit der Verwendung des neuen Modells oder sind der Meinung, dass die Legacy-Funktionen ausreichen.	Sie haben Probleme mit der Verwendung des neuen Modells oder sind der Meinung, dass die Funktionen des Legacy-Modells besser geeignet sind.

Private, importierte und freigegebene Szenarios

Wenn Sie einem Dienst ein Sicherheitsszenario zuweisen möchten, verwenden Sie das Fenster **Dienste verwalten**. Auf der Registerkarte **Protokoll und Sicherheit** können Sie Sicherheitsszenarios für einzelne Dienste erstellen und bearbeiten.

Sie haben drei Möglichkeiten zur Auswahl eines Szenarios:

- **Privates Szenario.** Erstellen Sie ein neues Szenario, indem Sie eines der systemeigenen Szenarios auswählen und dann für den gewünschten Webdienst anpassen.
- **Importiertes Szenario.** Verwenden Sie ein Szenario, das zu einem früheren Zeitpunkt erstellt wurde. Dieses Szenario kann bearbeitet werden. Änderungen an dem ursprünglichen Szenario haben keine Auswirkungen auf die von Ihnen bearbeitete Version.
- **Freigegebenes Szenario.** Laden Sie ein bereits von einem anderen Benutzer konfiguriertes Sicherheitsszenario von einem Remotespeicherort oder aus dem Dateisystem. Freigegebene Szenarios können im Fenster **Dienste verwalten** nicht bearbeitet werden. Änderungen durch einen anderen Benutzer wirken sich auf Ihre Umgebung aus. Normalerweise empfiehlt sich diese Option, wenn Sie bereits einige Zeit mit dem Produkt arbeiten und gespeicherte

Szenariodateien vorhanden sind.

Szenariokategorien

Ein Szenario beschreibt die Konfiguration eines Webdienstes. Hierzu gehören Informationen zur Sicherheit, Codierung, Proxyservern etc. VuGen enthält einen Sicherheitsszenario-Editor, mit dem Sie die einzelnen Szenarioeinstellungen konfigurieren können.

Anhand der folgenden Abschnitte können Sie ermitteln, welche Szenarios am besten für welche Dienste geeignet sind. Wenn Sie sich nicht sicher sind, welches Szenario Sie auswählen sollen, empfehlen wir die Verwendung des Szenarios **Benutzerdefinierte Bindung**. Weitere Informationen finden Sie unter "Die Szenarios "Benutzerdefinierte Bindung"" auf Seite 1025.

Verwenden Sie die Standardeinstellung **<Kein Szenario>** für:

- einfache Webdienste, für die keine erweiterten Standards erforderlich sind.
- Skripts, die auf dem Legacy-Sicherheitsmodell basieren.
- Webdienste, für die spezifische Sicherheitseinstellungen erforderlich sind, die in keinem der vorhandenen Szenarios enthalten sind.

Wenn Sie ein systemeigenes Szenario auswählen und bei der Skriptwiedergabe Fehler auftreten, kann es sein, dass kein Szenario erforderlich war und das Problem eine andere Ursache hat. Setzen Sie die Einstellung auf **<Kein Szenario>** zurück.

Die integrierten Sicherheitsszenarios werden in folgende Kategorien eingeteilt:

Hauptszenarios

Die folgende Tabelle enthält Informationen zum integrierten Hauptszenario.

Szenarioname	Anwendung
Einfaches SOAP	• Webdienste, für die keine erweiterten Standards erforderlich sind. • Webdienste, für die die Version der WS-Adressierung angegeben werden muss.

Geben Sie bei diesem Szenario die Version an, falls der Dienst die WS-Adressierung verwendet.

Sicherheitsszenarios

Die folgende Tabelle enthält Informationen zum integrierten Sicherheitsszenario.

Szenarioname	Anwendung
Authentifizierung durch Benutzername	• Client wird durch Benutzername und Kennwort auf Nachrichtenebene authentifiziert

Geben Sie bei diesem Szenario Benutzername und Kennwort an und, falls der Dienst WS-Adressierung verwendet, auch die Version.

WCF-Szenarios

In der Tabelle unten finden Sie die Szenarios für Webdienste, die WCF verwenden. Szenarios, die auf WSHttpBinding basieren, werden anhand der Clientauthentifizierungsmethode unterteilt. Wenn der Client beispielsweise einen Benutzernamen und ein Kennwort an den Server übermittelt, wählen Sie das Szenario **Username (Nachrichtenschutz)** aus. In der Benutzeroberfläche können Sie die Identitätsdaten in Form eines Benutzernamens oder ggf. durch ein Zertifikat angeben.

WCF-Szenarioname	Anwendung
WSHttpBinding - Keine Authentifizierung	• Der Client verwendet für die Verschlüsselung das X.509-Zertifikat des Servers. • Der Client wird nicht authentifiziert. • Für Verbindungen werden möglicherweise erweiterte Standards, wie z. B. sichere Konversation und MTOM, verwendet.
WSHttpBinding - Windows-Authentifizierung	• Client und Server verwenden die Windows-Authentifizierung. • Die Sicherheit basiert auf Aushandlungen durch Kerberos oder SPNEGO. • Für Verbindungen werden möglicherweise erweiterte Standards, wie z. B. sichere Konversation und MTOM, verwendet.
wsHttpBInding - Authentifizierung durch Zertifikat	• Der Client verwendet für die Verschlüsselung das X.509-Zertifikat des Servers. • Der Client verwendet für Signaturen sein eigenes X.509-Zertifikat. • Für Verbindungen werden möglicherweise erweiterte Standards, wie z. B. sichere Konversation und MTOM, verwendet.

WSHttpBinding - Authentifizierung durch Benutzername (Nachrichtenschutz)	• Der Client verwendet für die Verschlüsselung das X.509-Zertifikat des Servers. • Der Client wird durch einen Benutzernamen und ein Kennwort authentifiziert. • Für Verbindungen werden möglicherweise erweiterte Standards, wie z. B. sichere Konversation und MTOM, verwendet.
WSHttpBinding - Authentifizierung durch Benutzername (Transportschutz)	• SSL ist aktiviert. • Der Client wird durch einen Benutzernamen und ein Kennwort authentifiziert. • Für Verbindungen werden möglicherweise erweiterte Standards, wie z. B. sichere Konversation und MTOM, verwendet.
WSFederationHttpBinding	• Der Client wird durch den STS unter Verwendung eines vordefinierten Szenarios authentifiziert. • Der Client verwendet das vom STS übermittelte Token für die Authentifizierung beim Server.
Custom Binding	• Webdienst, der WS-*-Standards verwendet. • WCF-Dienste mit beliebiger Konfiguration.

Optimierungsszenarios

Die folgende Tabelle enthält Informationen zum integrierten Optimierungsszenario.

Szenarioname	Anwendung
MTOM	• MTOM-aktivierte Webdienste. • Webdienste, für die die Version der WS-Adressierung angegeben werden muss.

Geben Sie bei MTOM-Szenarios die Version an, falls der Dienst die WS-Adressierung verwendet.

WCF-Szenarioeinstellungen

In diesem Abschnitt werden die für die WCF-Sicherheitsszenarios erforderlichen Werte erörtert:

Das WsHttpBinding-Szenario

Keine Authentifizierung (anonym)

In diesem Szenario verwendet der Client zur Verschlüsselung von Nachrichten das Zertifikat des Servers; der Client wird nicht authentifiziert.

Wählen Sie nur eine der folgenden Einstellungen aus:

- **Dienstanmeldeinformationen aushandeln.** Handelt das Zertifikat des Webdienstes mit dem Server aus.
- **Dienstzertifikat angeben.** Das Dienstzertifikat kann ausgewählt werden. Weitere Informationen finden Sie unter "Dialogfeld "Zertifikat auswählen"" auf Seite 1046. Wenn Sie diese Option auswählen, wird die Option **Dienstanmeldeinformationen aushandeln** deaktiviert.

Machen Sie Angaben zum DNS.

- **Erwarteter Server-DNS.** Die erwartete Identität des Servers auf Grundlage seines DNS. Diese kann **localhost** lauten oder eine IP-Adresse bzw. ein Servername sein. Es kann sich auch um den allgemeinen Namen handeln, auf den das Zertifikat ausgestellt wurde.

Windows-Authentifizierung

Dieses WCF-Szenario nutzt die Windows-Authentifizierung.

Sie deklarieren die erwartete Identität des Servers anhand seiner **SPN**- oder **UPN**-Identitäten. Nutzen Sie dieses Szenario für den Test von WCF-Diensten, bei denen die Standardkonfiguration ohne benutzerdefinierte Anpassung verwendet wird.

Authentifizierung durch Zertifikat

Im WCF-Szenario WSHttpBinding verwendet der Client zur Verschlüsselung von Nachrichten das X.509-Zertifikat des Servers und sein eigenes Zertifikat als Signatur.

Wählen Sie nur eine der folgenden Einstellungen aus:

- **Dienstanmeldeinformationen aushandeln.** Handelt das Zertifikat des Webdienstes mit dem Server aus.
- **Dienstzertifikat angeben.** Das Dienstzertifikat kann ausgewählt werden. Weitere Informationen finden Sie unter "Dialogfeld "Zertifikat auswählen"" auf Seite 1046. Wenn Sie diese Option auswählen, wird die Option **Dienstanmeldeinformationen aushandeln** deaktiviert.

Machen Sie Angaben zum DNS:

- **Erwarteter Server-DNS.** Die erwartete Identität des Servers auf Grundlage seines DNS. Diese kann **localhost** lauten oder eine IP-Adresse bzw. ein Servername sein. Es kann sich auch um den allgemeinen Namen handeln, auf den das Zertifikat ausgestellt wurde.

Authentifizierung durch Benutzername (Nachrichtenschutz)

Im WCF-Szenario WSHttpBinding verwendet der Client zur Verschlüsselung von Nachrichten das X.509-Zertifikat des Servers und übermittelt zu seiner eigenen Authentifizierung einen Benutzernamen und ein Kennwort.

Machen Sie folgende Angaben:

- **Benutzername. Kennwort.** Benutzername und Kennwort des Clients.

Wählen Sie nur eine der folgenden Einstellungen aus:

- **Dienstanmeldeinformationen aushandeln.** Handelt das Zertifikat des Webdienstes mit dem Server aus.
- **Dienstzertifikat angeben.** Das Dienstzertifikat kann ausgewählt werden. Weitere Informationen finden Sie unter "Dialogfeld "Zertifikat auswählen"" auf Seite 1046. Wenn Sie diese Option auswählen, wird die Option **Dienstanmeldeinformationen aushandeln** deaktiviert.

Machen Sie Angaben zum DNS:

- **Erwarteter Server-DNS.** Die erwartete Identität des Servers auf Grundlage seines DNS. Diese kann **localhost** lauten oder eine IP-Adresse bzw. ein Servername sein. Es kann sich auch um den allgemeinen Namen handeln, auf den das Zertifikat ausgestellt wurde.

Authentifizierung durch Benutzername (Transportschutz)

Im WCF-Szenario **WSHttpBinding** ist SSL aktiviert. Der Client wird auf Nachrichtenebene durch Benutzername und Kennwort authentifiziert.

Machen Sie folgende Angaben:

- **Benutzername. Kennwort.** Benutzername und Kennwort des Clients.

Das Verbundszenario

Im Szenario **WSFederationHttpBinding** wird der Client mithilfe von STS (Security Token Service) authentifiziert und erhält ein Sicherheitstoken. Mithilfe dieses Token authentifiziert der Client sich dann bei dem Anwendungsserver.

Aus diesem Grund sind zwei Bindungen erforderlich, eine mit STS und eine mit dem Anwendungsserver.

Verwenden Sie zunächst den Sicherheitsszenario-Editor, um eine STS-Bindung zu definieren. Weitere Informationen finden Sie unter "Erstellen und Verwalten von Sicherheitsszenarios" auf Seite 1037. Geben Sie für die Bindung mit dem Anwendungsserver die gewünschte Datei im Feld **Referenzierte Datei** an.

Geben Sie für das Verbundszenario die folgenden Serverinformationen an:

- **Transport.** HTTP oder HTTPS
- **Codierung.** Text oder MTOM

Geben Sie für das Verbundszenario die folgenden Sicherheitsinformationen an:

- **Authentifizierungsmodus.** IssuedToken, IssuedTokenForCertificate, IssuedTokenForSslNegotiated, IssuedTokenOverTransport oder SecureConversation
- **Bootstraprichtl.** IssuedToken, IssuedTokenForCertificate, IssuedTokenForSslNegotiated oder IssuedTokenOverTransport

Geben Sie für das Verbundszenario die folgenden Identitätsinformationen an:

- **Serverzertifikat.** Das Serverzertifikat kann ausgewählt werden. Weitere Informationen finden Sie im "Dialogfeld "Zertifikat auswählen"" auf Seite 1046.
- **Erwarteter Server-DNS.** Die erwartete Identität des Servers auf Grundlage seines DNS. Diese kann **localhost** lauten oder eine IP-Adresse bzw. ein Servername sein.

Geben Sie für das Verbundszenario die folgenden Informationen zum STS (Security Token Service) an:

- **Ausstelleradresse.** Die Adresse des STS-Ausstellers. Diese kann **localhost** lauten oder eine IP-Adresse bzw. ein Servername sein.
- **Referenzierte Bindung.** Die Datei, in der die Bindung für den Kontakt zum STS (Security Token Service) referenziert wird.

Die Szenarios "Benutzerdefinierte Bindung"

Das Szenario **Benutzerdefinierte Bindung** ermöglicht den höchsten Grad der benutzerdefinierten Anpassung. Da es auf dem WCF-Szenario **customBinding** basiert, können Sie mit ihm fast alle WCF-Dienste testen, aber auch Dienste auf anderen Plattformen (z. B. Java), die Spezifikationen vom Typ **WS - <*spec_name*>** verwenden.

Verwenden Sie das Szenario **Benutzerdefinierte Bindung** zur Konfiguration eines benutzerdefinierten Szenarios, das mit keinem der vordefinierten Sicherheitsszenarios übereinstimmt.

Geben Sie für das Szenario **Benutzerdefinierte Bindung** die folgenden Serverinformationen an:

- **Transport.** HTTP, HTTPS, TCP oder NamedPipe
- **Codierung.** Text, MTOM oder WCF Binary

Machen Sie folgende Angaben zur Sicherheit:

- **Authentifizierungsmodus.** None, AnonymousForCertificate, AnonymousForSslNegotiated, CertificateOverTransport, Kerberos, KerberosOverTransport, MutualCertificate, MutualSslNegotiated, SecureConversation, SspiNegotiated, UserNameForCertificate, UserNameForSslNegotiated, UserNameOverTransport oder SspiNegotiatedOverTransport
- **Bootstraprichtl.** Geben Sie für den Authentifizierungstyp **SecureConversation** eine Bootstraprichtlinie an: AnonymousForCertificate, AnonymousForSslNegotiated, CertificateOverTransport, Kerberos, KerberosOverTransport, MutualCertificate, MutualSslNegotiated, SspiNegotiated, UserNameForCertificate, UserNameForSslNegotiated, UserNameOverTransport oder SspiNegotiatedOverTransport
- **Netzwerksicherheit.** Die Netzwerksicherheitsmethode. Wählen Sie **Keine**, **Windows-Datenstr.Sicherheit** oder **SSL-Datenstr.Sicherheit** aus. Belassen Sie bei Diensten mit HTTP-Transport die Standardeinstellung **Keine**. Wählen Sie zur Aktivierung von SSL für HTTP den HTTPS-Transport aus.

Wenn Ihr Webdienst **Reliable Messaging** verwendet, aktivieren Sie diese Option und wählen Sie **Sortiert** oder **Nicht sortiert** aus.

Identitäten

Für die Sicherheitseinstellungen müssen Sie möglicherweise Identitätsinformationen entweder für den Client oder den Server bzw. für beide angeben.

Identitätsinformationen für den Client sind beispielsweise ein Benutzername und ein Kennwort oder ein **X.509**-Zertifikat.

Geben Sie je nach Dienst die erforderlichen Authentifizierungsdetails an:

Benutzername, **Kennwort**, **Serverzertifikat**, **Clientzertifikat** oder eine benutzerdefinierte Windows-Identität. Informationen zum Auswählen von Zertifikaten finden Sie unter "Dialogfeld "Zertifikat auswählen"" auf Seite 1046.

Für einige Szenarios ist es erforderlich, die erwartete Identität des Servers anhand seiner DNS-, SPN- oder UPN-Identität zu deklarieren.

- **DNS.** Geben Sie den Namen eines Servers oder **localhost** an.
- **SPN.** Geben Sie die SPN-Identität im Format `Domäne\Computer` an.
- **UPN.** Geben Sie die UPN-Identität im Format `benutzer@domäne` an.

Nachdem Sie die grundlegenden Werte eingestellt haben, können Sie erweiterte Attribute wie unter "Erweiterte Szenarioeinstellungen" unten beschrieben festlegen.

Erweiterte Szenarioeinstellungen

In diesem Abschnitt werden die erweiterten Szenarioeinstellungen erläutert. Mit ihnen können Sie ein Sicherheitsszenario in den Bereichen Codierung, erweiterte Standards, Sicherheit oder HTTP und Proxy anpassen.

Nicht alle Einstellungen sind für alle Szenarios relevant. Einige können je nach Szenario deaktiviert sein oder nicht angezeigt werden.

Codierung

Auf der Registerkarte **Codierung** können Sie den Kodierungstyp für die Nachrichten festlegen: **Text**, **MTOM** oder **Binary**. Die Standardeinstellung ist **Text**.

Für jede dieser Codierungsmethoden können Sie eine Version der WS-Adressierung auswählen:

- Keine
- WSA 1.0
- WSA 04/08

Erweiterte Standards

Auf dieser Registerkarte können Sie die erweiterten WS-Standards festlegen, z. B. Reliable Messaging und die Option **Über Adresse**.

Wenn der Dienst die Spezifikation **WS-ReliableMessaging** implementiert, aktivieren Sie die Option **Reliable Messaging** und legen Sie die folgenden Optionen fest:

- **Geordnetes Reliable Messaging**. Gibt an, ob die Reliable Messaging-Sitzung geordnet werden soll.
- **Reliable Messaging-Version. WSReliableMessagingFebruary2005** oder **WSReliableMessaging11**.

Über Adresse

Manchmal ist es erforderlich, eine Nachricht an einen Zwischendienst zu senden, der diese an den tatsächlichen Server übergibt. Dieser Mechanismus kann auch zum Tragen kommen, wenn Sie eine Nachricht an einen Debugproxy senden. Dies entspricht dem Verhalten von WCF **clientVia**.

Hier kann es sinnvoll sein, die physikalische Adresse, an die die Nachricht tatsächlich übermittelt wird, von der logischen Adresse, an die die Nachricht gerichtet ist, zu trennen. Die logische Adresse kann die physikalische Adresse des letzten Servers oder ein beliebiger Name sein. Sie wird in der SOAP-Meldung folgendermaßen dargestellt:

<wsa:Action>http://myLogicalAddress<wsa:Action>

Die logische Adresse kann über die Benutzeroberfläche abgerufen werden. Standardmäßig ist es die in der WSDL-Datei angegebene Adresse. Sie können diese Adresse im Dialogfeld **Dienste verwalten** überschreiben.

Sicherheit

Die erweiterten Sicherheitseinstellungen entsprechen den **WS-Security**-Spezifikationen.

Für Sicherheitsszenarios, die auf dem WCF-Szenario **WSHttpBinding** basieren, können Sie die folgenden Optionen konfigurieren:

- **Sichere Sitzung aktivieren.** Erstellen Sie einen Sicherheitskontext unter Verwendung des

WS-SecureConversation-Standards.

- **Dienstanmeldeinformationen aushandeln.** Proprietäre WCF-Aushandlungen werden zugelassen, um die Sicherheitsdetails des Dienstes auszuhandeln.

Für die WCF-Szenarios **WSHttpBinding, Benutzerdefinierte Bindung** und **WSFederationHttpBinding** können Sie das Standardalgorithmuspaket und die Schutzebene festlegen:

Attribut	Bedeutung	Mögliche Werte
Standard-Algorithmuspaket	Der Algorithmus, der für die symmetrische/ asymmetrische Verschlüsselung verwendet wird. Links finden Sie die Werte aus der SecurityAlgorithmSuite-Konfiguration in WCF:	• Basic128 • Basic128Rsa15 • Basic128Sha256 • Basic128Sha256Rsa15 • Basic192 • Basic192Rsa15 • Basic192Sha256 • Basic192Sha256Rsa15 • Basic256 • Basic256Rsa15 • Basic256Sha256 • Basic256Sha256Rsa15 • TripleDes • TripleDesRsa15 • TripleDesSha256 • TripleDesSha256Rsa15
Schutzebene	Soll der SOAP-Textkörper verschlüsselt/signiert werden?	None, Sign und EncryptAndSign (Standard)

Für die WCF-Szenarios **Benutzerdefinierte Bindung** oder **WSFederationHttpBinding** können Sie die Sicherheitseinstellungen detaillierter konfigurieren. Die Tabelle unten beschreibt die Optionen und die zugehörigen Werte:

Attribut	Bedeutung	Mögliche Werte
Message Protection Order	Die Reihenfolge von Signierung und Verschlüsselung.	• SignBeforeEncrypt • SignBeforeEncrypt-AndEncryptSignature • EncryptBeforeSign
Version für Meldungssicherheit	Die WS-Security-Version.	Eine Liste der aktuellen Versionen.
Sicherheitskopfzeilenlayout	Das Layout der Nachrichtenkopfzeile.	• Strict • Lax • LaxTimeStampFirst • LaxTimeStampLast
Schlüsselentropiemodus	Der Entropiemodus für den Sicherheitsschlüssel.	• Client Entropy • Security Entropy • Combined Entropy

Folgende Optionen stehen zur Verfügung:

- **Abgeleitete Schlüssel anfordern.** Gibt an, ob abgeleitete Schlüssel angefordert werden oder nicht.
- **Aufhebung des Sicherheitskontexts anfordern.** Das Deaktivieren dieser Option setzt voraus, dass zustandsorientierte Sicherheitstoken in der **WS-SecureConversation**-Sitzung (falls aktiviert) verwendet werden.
- **Zeitstempel einschließen.** Fügt der Kopfzeile einen Zeitstempel hinzu.
- **Serialisiertes Token bei Antwort zulassen.** Mit der Antwort wird ein serialisiertes Token gesendet.
- **Signaturbestätigung anfordern.** Weist den Server an, in der Antwort eine Signaturbestätigung zu übermitteln.

Bei X.509-Zertifikaten können Sie Werte für die folgenden Elemente angeben:

Attribut	Bedeutung	Mögliche Werte

Einschlussmodus für X509	Gibt an, wann das X509-Zertifikat hinzuzufügen ist.	• Always to Recipient • Nie • Once • AlwaysToInitiator
Verweisart für X509	Die Art, wie das Zertifikat referenziert werden soll.	• Internal • External
Abgeleitete Schlüssel für X509 erforderlich	Sollen in X509-Zertifikaten abgeleitete Schlüssel angefordert werden?	• Aktiviert - Ja • Deaktiviert - Nein
Typ der Klausel für X509-Schlüsselidentifikation	Die Art der Klausel, mit deren Hilfe der X509-Schlüssel identifiziert wird.	• Any • Thumbprint • IssuerSerial • SubjectKeyIdentifier • RawDataKeyIdentifier

HTTP und Proxy

Auf dieser Registerkarte können Sie die HTTP- und Proxyinformationen für Ihren Test festlegen.

HTTP (S) Transport

Die Tabelle unten beschreibt die unter **HTTP(S) Transport** verfügbaren Optionen:

Option	Bedeutung	Mögliche Werte
Übertragungsmodus	Die Transfermethode für Anforderungen/Antworten.	Buffered, Streamed, StreamedRequest, StreamedResponse
Max. Antwortgröße (KB)	Die maximale Größe der Antwort, bevor diese verkettet wird	Standardeinstellung: 65 KB
Cookies zulassen	Cookies werden aktiviert.	Aktiviert/Deaktiviert
Keep-Alive aktiviert	Keep-alive-Verbindungen werden aktiviert.	Aktiviert/Deaktiviert

Authentifizierungsschema	HTTP-Authentifizierungsmethode	None, Digest, Negotiate, NTLM, IntegratedWindows Authentication, Basic, Anonymous
Bereich	Der Bereich des Authentifizierungsschemas.	Beliebiger URL
Clientzertifikat anfordern	Bei SSL-Verschlüsselung wird ein Zertifikat angefordert.	Aktiviert/Deaktiviert

Proxyinformationen

Wenn für die Transportmethode des Webdienstes ein Proxyserver vorgesehen ist, können Sie die zugehörigen Details auf dieser Registerkarte festlegen. In der Tabelle unten werden die Proxyoptionen erläutert:

Option	Bedeutung	Mögliche Werte
Standard-Webproxy verwenden	Es werden die Standardeinstellungen für Proxyserver auf dem Computer verwendet.	Aktiviert/Deaktiviert
Proxy auf lokalem Computer umgehen	Der Proxyserver wird ignoriert, wenn der Dienst auf dem lokalen Computer ausgeführt wird.	Aktiviert/Deaktiviert
Proxyadresse	Der Proxyserver.	Beliebiger URL
Proxy-Authentifizierungsschema	HTTP-Authentifizierungsmethode auf dem Proxyserver.	None, Digest, Negotiate, NTLM, IntegratedWindows Authentication, Basic, Anonymous

WCF-Erweiterungsmöglichkeiten

Sie können bei Verwendung von **customBinding** Ihre eigene Auswahl für Bindung, Verhalten oder Kanal verwenden, indem Sie den `assemblyPath` und `typeName` in der Konfigurationsdatei `<Skriptverzeichnis>/WSDL/@config/[IhreKonfig].stss` definieren.

Das Attribut `assemblyPath` sollte entweder den vollständigen Pfad der dll oder den relativen Pfad zum Skriptverzeichnis enthalten.

Das Attribut `typeName` sollte den vollständigen Typnamen enthalten. `ns.typeName`.

Bindung

Benennen Sie das Szenarioattribut im Protokollelement und geben Sie die Attribute `assemblyPath` und `typeName` an.

Die zum Binden verwendete Klasse wird von `System.ServiceModel.Channels.Binding` vererbt.

Kanal

Fügen Sie unter dem Anpassungsknoten ein neues Element hinzu. Sie können dem Element einen beliebigen Namen geben, es muss aber die beiden Attribute `assemblyPath` und`typeName` enthalten.

Die zum Binden verwendete Klasse wird von `System.ServiceModel.Channels.BindingElement` vererbt.

Hinweis: Diese Methode funktioniert nur mit customBinding-Szenarios.

Verhalten

Fügen Sie unter dem Verhaltenselement (das sich unter `endpointBehavior` befindet) ein neues Element ein und fügen Sie die beiden Attribute `assemblyPath` und `typeName` hinzu.

Um das neue Element zu binden, implementieren Sie die Klasse `System.ServiceModel.Description.IEndpointBehavior`.

Hinweis: Wird die Klasse von `System.ServiceModel.Description.ClientCredentials` vererbt, werden auch die Clientanmeldedaten dieser Klasse verwendet.

Beispiele für Kanal und Verhalten

```
<protocols scenario="customBinding" uiType="customBinding" xmlns="http://hp/ServiceTest/config">
<mode>Private</mode>
<customization>
<textMessageEncoding />
<preferlrhttpTransport />
<myChannel assemblyPath="CustomChannel.dll" typeName="CustomChannel.WCFChannel" />
</customization>
<behaviors>
<endpointBehaviors>
<behavior>
<clientVia viaUri="qwqwq" />
<myBehavior assemblyPath="CustomBehavior.dll" typeName="CustomBehavior.WCFbehavior" />
</behavior>
</endpointBehaviors>
```

```
</behaviors>
</protocols>
```

Ein Beispiel für das Überschreiben der gesamten Bindung (die Konfiguration enthält möglicherweise nur eine Zeile):

```
<protocols scenario="userBinding" assemblyPath="WCFBinding.dll" typeName=" W
CFBinding.Binding"/>
```

Vorbereiten von Sicherheitsszenarios für die Ausführung

Parametrieren von Sicherheitselementen

Sie können die Sicherheitselemente innerhalb eines Skripts unabhängig voneinander parametrieren. Bei einem Sicherheitsszenario, das auf Benutzernamen basiert, kann es beispielsweise sinnvoll sein, für jeden Vuser oder jede Iteration einen anderen Benutzernamen zu verwenden.

Schützen von benutzerdefinierten Kopfzeilen

Wenn für einen Vorgang SOAP-Kopfzeilen verwendet werden, signiert oder verschlüsselt VuGen diese nicht automatisch. Um ein Schutzschema, z. B. eine Signatur oder Verschlüsselung, einzubeziehen, müssen Sie folgende Informationen zum **behavior**-Element in der Konfigurationsdatei des Szenarios (.stss) manuell hinzufügen:

- Die "oapAction" für den entsprechenden Vorgang
- Name und Namespace der XML-Kopfzeile
- Die Schutzebene

Das folgende Beispiel zeigt eine ausgehende Nachricht mit der soapAction **http://mySoapAction**. Das XML-Element **header1** aus dem Namespace **http://myServiceNamespace** ist verschlüsselt und signiert. Das **header2**-Element aus demselben Namespace ist nur signiert.

```
<protocols ...>
              ...
              <behaviors>
                    <contractBehaviors>
                        <behavior>
                            <channelProtectionBehavior>
                                <protectionRequirements action="http://mySoapAct
ion">
                                    <incomingEncryptionParts>
```

```
                                    <header localName="header1" namespace="h
ttp://myServiceNamespace" />
                                </incomingEncryptionParts>

                                <incomingSignatureParts>
                                    <header localName="header1" namespace="
http://myServiceNamespace " />
                                    <header localName="header2" namespace="
http://myServiceNamespace " />
                                </incomingSignatureParts>

                            </protectionRequirements>

                        </channelProtectionBehavior>
                    </behavior>
                </contractBehaviors>
            </behaviors>
        </protocols>
```

Emulieren von Benutzern mit Iterationen

Bei vielen der Sicherheitsszenarios wird eine Sitzung mit einem Server hergestellt. Jedes Szenario, das beispielsweise **WS-SecureConversation** verwendet, stellt eine Serversitzung her. Diese Sitzung wird hergestellt, wenn die erste Operation ausgeführt wird, und geschlossen, wenn das Skript beendet wird. Standardmäßig schließt VuGen alle Sitzungen nach jeder Iteration und öffnet diese wieder, wenn die nächste Iteration beginnt. Dies setzt voraus, dass bei jeder Iteration eine neue Sitzung und ein neuer Vuser simuliert wird.

Wenn Sie mit mehreren Iterationen arbeiten, ist dies jedoch nicht unbedingt wünschenswert – es ist unter Umständen sinnvoller, die ursprüngliche Sitzung beizubehalten und nicht für jede Iteration eine neue Sitzung zu öffnen. Dies gilt für das Ausführen von Lasttests mithilfe von LoadRunner Controller oder wenn Sie mehrere Iterationen in den Laufzeiteinstellungen festlegen.

Sie können dieses Verhalten überschreiben, so dass nur bei der ersten Iteration eine neue Sitzung erstellt wird. Alle weiteren Iterationen nutzen dann die vorhandene Sitzung. Hierdurch wird ein Benutzer emuliert, der in einer Sitzung mehrfach hintereinander dieselbe Aktion ausführt.

Wählen Sie den Simulationsmodus aus, der am besten geeignet ist für das, was Sie simulieren möchten. Wenn Sie beispielsweise einen Lasttest simulieren, bei dem ein Großteil der Aktionen von einem Benutzer mehrfach hintereinander ausgeführt wird, verwenden Sie die oben beschriebene Konfiguration. Wenn Sie sich nicht sicher sind, belassen Sie die Standardeinstellungen.

Hinzufügen von Sicherheitsfunktionen zu einem Webdienstskript

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie die Sicherheitseinstellungen für Webdienstaufrufe konfigurieren. Weitere Informationen zur Webdienstsicherheit finden Sie unter "Konfigurieren der Sicherheitseinstellungen – Übersicht" auf Seite 1013.

Einfügen eines neuen Webdienstsicherheitsschritts

1. Platzieren Sie den Cursor an der Stelle, der Sie die Sicherheitseinstellungen hinzufügen möchten. In der Regel empfiehlt es sich, den Cursor im Abschnitt **vuser_init** zu platzieren, damit die Sicherheit für das gesamte Skript übernommen wird. Wenn Sie nur bestimmten Aufrufen Sicherheitseinstellungen hinzufügen möchten, platzieren Sie den Cursor an der gewünschten Stelle.

2. Wählen Sie die Option **Einfügen> Neuer Schritt** aus, um das Dialogfeld **Schritt hinzufügen** zu öffnen.

3. Wählen Sie **Webdienst-Sicherheitseinstellungen festlegen** aus und klicken Sie auf **OK**. Das Dialogfeld **Sicherheitseigenschaften festlegen** wird geöffnet.

Hinzufügen eines Token (optional)

1. Klicken Sie auf **Sicherheitstoken hinzufügen**, um einen neuen Token hinzuzufügen. Das Dialogfeld **Token hinzufügen** wird geöffnet.

2. Wählen Sie einen Tokentyp aus. Weitere Informationen finden Sie unter "Sicherheitstoken und Verschlüsselung" auf Seite 1014.

   Weisen Sie dem Token im Feld **Logischer Name** einen beliebigen Namen zu, der in VuGen zur Identifizierung verwendet wird.

   Fügen Sie alle relevanten Informationen hinzu, z. B. **Benutzername** und **Kennwort** für Token vom Typ **Benutzername und Kennwort**.

   Wenn der Token explizit an die Kopfzeile des SOAP-Umschlags gesendet werden soll, wählen Sie **True** aus. Wenn der Token aus der Kopfzeile des SOAP-Umschlags entfernt werden soll, wählen Sie **False** aus.

Hinzufügen einer Signatur oder Verschlüsselung (optional)

1. Klicken Sie auf **Hinzufügen > Nachrichtensignatur** oder **Hinzufügen > Verschlüsselte Daten**.

2. Wählen Sie einen Token aus, der mit der Meldungssignatur oder Verschlüsselung verwendet werden soll. Signaturen und Verschlüsselungen setzen die Angabe eines Tokens voraus, der zuvor als verschlüsselnder/signierender Token definiert wurde.

3. Geben Sie einen Zieltoken an, oder lassen Sie das betreffende Feld leer, um die Signatur bzw.

Verschlüsselung auf den gesamten Textkörper anzuwenden. Weitere Informationen finden Sie unter "Sicherheitstoken und Verschlüsselung" auf Seite 1014.

Festlegen einer Nachrichtenzeitüberschreitung (optional)

Wählen Sie zur Angabe einer Gültigkeit für das Nachrichtenpaket die Option **Gültigkeitsdauer** aus und geben Sie die Zeit in Sekunden an.

Aufheben der Sicherheitseinstellungen (optional)

Wenn Sie die Sicherheitseinstellungen an einer bestimmten Stelle im Skript aufheben möchten, fügen Sie an der Stelle einen **Webdienst - Auswahl der Sicherheitseinstellungen aufheben**-Schritt hinzu.

Hinzufügen von SAML-Sicherheit

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie die SAML-Sicherheit für Ihre Webdienstaufrufe hinzufügen. Weitere Informationen zu SAML finden Sie unter "SAML-Sicherheitsoptionen" auf Seite 1017.

Informationen zur Syntax finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

1. **Einfügen eines neuen Webdienstsicherheitsschritts**

   a. Platzieren Sie den Cursor an der Stelle, der Sie die Sicherheitseinstellungen hinzufügen möchten.

   b. Wählen Sie die Option **Einfügen > Neuer Schritt** aus, um das Dialogfeld **Schritt hinzufügen** zu öffnen.

   c. Wählen Sie **Webdienst - SAML-Einstellungen festlegen** aus und klicken Sie auf **OK**. Ein Eigenschaften-Dialogfeld wird geöffnet.

2. **Einfügen einer SAML-Assertion**

   Um eine SAML-Assertionsmethode hinzuzufügen, fügen Sie einen Schritt vom Typ **Webdienst - SAML bestätigen** im Dialogfeld **Schritt hinzufügen** ein (**Einfügen > Neuer Schritt**). Geben Sie die unsignierte Assertion, eine Zertifikatsdatei und ein Kennwort (optional) an.

3. **Festlegen der Sicherheitsrichtlinien (optional)**

   Geben Sie entweder eine Richtliniendatei an oder lassen Sie das Feld leer, um die Standardeinstellung zu verwenden. Wenn Sie Werte von Hand eingeben, überschreiben diese alle Einstellungen aus der Richtliniendatei. Geben Sie die URL-Adresse des Ausstellers an, auch als **STS-URL** bekannt.

4. **Aufheben der SAML-Sicherheitseinstellungen (optional)**

   Um die Einstellungen an einem bestimmten Punkt im Skript zu entfernen, fügen Sie einen Schritt vom Typ **Webdienst - Auswahl von SAML-Einstellungen aufheben** ein.

Erstellen und Verwalten von Sicherheitsszenarios

In den folgenden Schritten wird beschrieben, wie Sie ein Sicherheitsszenario für einen bestimmten Dienst erstellen und anpassen.

1. **Öffnen des Dialogfelds für Sicherheitsszenariodaten**

   a. Klicken Sie auf **Manage Services**. Wählen Sie im Bereich auf der linken Seite den Dienst aus, für den Sie ein Sicherheitsszenario festlegen möchten. Importieren Sie ggf. wie unter "Dialogfeld "Dienst importieren"" auf Seite 1010 beschrieben einen Dienst.

   b. Aktivieren Sie die Registerkarte **Protocol and Security** und klicken Sie auf die Schaltfläche **Edit Data**. Das Dialogfeld **Security Scenario Data** wird geöffnet.

2. **Erstellen eines Szenarios (falls noch nicht vorhanden)**

   a. Aktivieren Sie die Option **Privates Szenario** und wählen Sie dann eines der vorhandenen Sicherheitsszenarios für den aktuellen Dienst aus.

   b. Wählen Sie im Feld **Szenariotyp** ein Szenario aus. Weitere Informationen finden Sie unter "Auswählen eines Sicherheitsmodells" auf Seite 1019.

   c. Geben Sie die erforderlichen Werte für Ihr Szenario ein. Weitere Informationen finden Sie unter "WCF-Szenarioeinstellungen" auf Seite 1022.

   d. Klicken Sie zum Angeben eines Zertifikats (dies gilt nur für einige der Szenarios) auf die Durchsuchen-Schaltfläche neben dem Feld **Clientzertifikat** oder **Dienstzertifikat angeben**, um das Dialogfeld **Zertifikat auswählen** zu öffnen. Details finden Sie im "Dialogfeld "Zertifikat auswählen"" auf Seite 1046.

      - Um ein Zertifikat aus einer Datei abzurufen, wählen Sie **Datei** und wählen Sie dann den gewünschten Pfad aus.

      - Um ein Zertifikat aus einem Windows-Speicher abzurufen, wählen Sie **Windows-Speicher** aus. Wählen Sie einen Speicherort und einen Namen für den Speicher aus. Geben Sie eine Suchzeichenfolge ein. Wenn Sie nach allen vorhandenen Zertifikaten suchen möchten, lassen Sie das Feld **Suchtext** leer. Wenn Sie nach einem bestimmten Zertifikat suchen, geben Sie einen Teil des Zertifikatsnamen ein. Geben Sie nach Bedarf ein Kennwort für den privaten Schlüssel an. Klicken Sie auf **Find**, um eine Liste der im Speicher gefundenen Zertifikate zu erzeugen.

3. **Laden eines Sicherheitsszenarios (wenn bereits Szenarios vorhanden sind)**

   a. Wenn Sie ein vorhandenes Szenario durch Bearbeiten anpassen möchten, wählen Sie **Privates Szenario.** Klicken Sie auf **Importieren**. Wählen Sie im Dialogfeld **Shared Scenario** eines der vorhandenen Szenarios aus. Bearbeiten Sie die Einstellungen ggf. wie unter "WCF-Szenarioeinstellungen" auf Seite 1022 beschrieben.

b. Wenn Sie ein vorhandenes Szenario unverändert übernehmen möchten, wählen Sie **Shared scenario** aus. Klicken Sie auf die Schaltfläche **Browse**, um im Dialogfeld **Shared Scenario** ein gespeichertes Szenario auszuwählen.

Hinweis: Wenn jemand das Original einer freigegebenen Szenariodatei bearbeitet, wirken sich die Änderungen auf Ihr Skript aus.

4. Konfigurieren von erweiterten Einstellungen (optional)

Klicken Sie auf **Erweitert**, um die erweiterten Einstellungen für Proxy, Codierung etc. zu konfigurieren. Für die meisten Szenarios eignen sich die Standardeinstellungen am besten. Weitere Informationen finden Sie unter "Erweiterte Szenarioeinstellungen" auf Seite 1026. Klicken Sie auf **OK**, um das Sicherheitsszenario zu speichern.

5. Bearbeiten eines bestehenden Sicherheitsszenarios (optional)

Verwenden Sie den Sicherheitsszenario-Editor, wenn Sie Sicherheitsszenarios erstellen oder anpassen möchten, die global für alle Skripts und nicht nur einen bestimmten Dienst verfügbar sein sollen. Mit dem Editor können Sie das Szenario außerdem speichern, sodass andere Benutzer darauf zugreifen können.

a. Wählen Sie **SOA-Tools** > **Sicherheitsszenario-Editor** aus.

b. Klicken Sie auf die Schaltfläche **Laden**, und wählen Sie eine vorhandene Szenariodatei (**.stss**) aus.

c. Bearbeiten Sie die Szenarioeinstellungen nach Bedarf.

d. Klicken Sie auf die Schaltfläche **Speichern** bzw. **Speichern unter**.

6. Schützen von SOAP-Kopfzeilen (optional)

Bearbeiten Sie das **behavior**-Element in der Szenariokonfigurationsdatei manuell.

a. Öffnen Sie in VuGen die Skriptansicht. Wählen Sie **Ansicht > Skriptansicht** aus.

b. Klicken Sie mit der rechten Maustaste auf eine Stelle im Skript und wählen Sie dann **Skriptverzeichnis öffnen** aus dem Kontextmenü aus.

c. Suchen Sie im Ordner **WSDL/@config** nach der Konfigurationsdatei des Szenarios (<*dienstname*>**.stss**).

d. Bearbeiten Sie den behavior-Abschnitt in der Datei. Weitere Informationen finden Sie unter "Schützen von benutzerdefinierten Kopfzeilen" auf Seite 1033.

7. Festlegen des Iterationsmodus (optional)

So konfigurieren Sie die Umgebung so, dass dieselbe Sitzung für alle Iterationen verwendet wird:

a. Öffnen Sie den Stammordner des Skripts: Klicken Sie in der Skriptansicht mit der rechten Maustaste in das Skript und wählen Sie **Skriptverzeichnis öffnen** aus dem Kontextmenü aus.

b. Öffnen Sie die Datei **default.cfg** in einem Text-Editor.

c. Fügen Sie im Abschnitt **[WebServices]** eine Zeile unterhalb von **Toolkit** ein. Wenn Sie das Axis-Toolkit verwenden oder andere Einstellungen konfiguriert haben, kann der Inhalt der Datei von dem oben gezeigten abweichen.

```
[WebServices]
Toolkit=.NET
SimulateNewUserInNewIteration=0
```

d. Speichern und schließen Sie die Datei.

Weitere Informationen finden Sie unter "Emulieren von Benutzern mit Iterationen" auf Seite 1034.

Parametrieren von Sicherheitselementen

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie die Sicherheitselemente in einem Skript unabhängig voneinander parametrieren.

1. ## Öffnen des Sicherheitsszenario-Editors

   Wählen Sie **SOA Tools > Sicherheitsszenario-Editor** aus.

2. ## Einrichten eines Sicherheitsszenarios für jeden Vuser

   Richten Sie ein Sicherheitsszenario für jeden Vuser ein. Eine Beschreibung finden Sie unter "Erstellen und Verwalten von Sicherheitsszenarios" auf Seite 1037. Es wird empfohlen, die Szenarios nach dem Muster **Benutzer1**, **Benutzer2** usw. zu benennen und sie in einem neuen Ordner unter **%script root%/WSDL/referencedConfig** zu speichern.

3. ## Öffnen der Parameterliste und Erstellen eines Parameters

   Wählen Sie **Vuser > Parameterliste** aus. Erstellen Sie einen neuen Parameter im Format **<Dienstname>_shared_config**. Ersetzen Sie <Dienstname> durch den Namen des Dienstes, den Sie testen. Achten Sie dabei auf die Groß-/Kleinschreibung. Um den genauen Namen des Dienstes zu ermitteln, klicken Sie auf **Dienste verwalten**, um eine Liste der Dienste anzuzeigen.

4. ## Hinzufügen von Parameterwerten

   Fügen Sie in der Wertetabelle in jeder der Zeilen die Dateinamen der Sicherheitsszenarios mitsamt der .stss-Erweiterung als Parameterwert hinzu. Sie können einen Pfad angeben, der relativ zum Skriptordner ist. Klicken Sie auf **Zeile hinzu**, wenn Sie mehrere Werte hinzufügen möchten. Schließen Sie das Dialogfeld **Parameterliste**.

5. **Aufrufen des Parameters**

   a. Klicken Sie auf **Dienste verwalten** und aktivieren Sie die Registerkarte **Protokoll und Sicherheit**. Klicken Sie auf **Daten bearbeiten**.

   b. Wählen Sie **Freigegebenes Szenario** aus. Klicken Sie auf die Schaltfläche **Durchsuchen** und geben Sie den Parameternamen, **<Dienstname>_shared_config**, im Testfeld ein.

Dialogfeld "Sicherheitseigenschaften festlegen"

In diesem Dialogfeld können Sie die Sicherheitseinstellungen für Webdienstaufrufe konfigurieren.

Zugriff	**VuGen > Einfügen > Neuer Schritt > Webdienst-Sicherheitseinstellungen festlegen.** Klicken Sie auf **OK**.
Relevante Aufgaben	"Hinzufügen von Sicherheitsfunktionen zu einem Webdienstskript" auf Seite 1035
Siehe auch	"Hinzufügen von SAML-Sicherheit" auf Seite 1036
Wichtige Informationen	Wenn Sie in der Datei **mmdrv.config** Werte für Schlüsselalgorithmen oder Sitzungsalgorithmen für ein vorhandenes Skript bearbeitet haben, werden diese Werte durch die Standardwerte des Systems ersetzt.

Registerkarte WS-Sicherheit

Nachfolgend werden die Elemente der Oberfläche beschrieben (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
<**Tokentabelle**>	Zeigt eine eindeutige Nummer, den Typ und den Namen aller hinzugefügten Token an.

Sicherheits-token hinzufügen

Ermöglicht das Auswählen eines Tokentyps:

Benutzername und Kennwort

Element der Oberfläche	Beschreibung
Name des Token	Ein aussagekräftiger Name für das Token.
Mit Zufallszahl	Wird diese Option gewählt, wird zum Signieren der Kommunikation eine beliebige Zahl verwendet.
Benutzername	Geben Sie den Benutzernamen an.
Kennwort	Geben Sie das Kennwort an.
Kennworttyp	Geben Sie einen der folgenden Kennworttypen an: • Text • Hash • Keine
Zeitstempelformat	Geben Sie das Zeitstempelformat an: • Vollständig • Erstellt • Keine

X.509-Zertifikatstoken

Element der Oberfläche	Beschreibung
Name des Token	Ein aussagekräftiger Name für das Token.
Zertifikat	Wird diese Option gewählt, wird zum Signieren der Kommunikation eine beliebige Zahl verwendet.
Referenztyp	Geben Sie den Benutzernamen an.

Kerberos-Token

Element der Oberfläche	Beschreibung
Name des Token	Ein aussagekräftiger Name für das Token.
Host	Der Hostname des Servers, gegen den Sie die Authentifizierung durchführen möchten. In den meisten Fällen ist dies der Hostabschnitt des Dienst-URL.
Domäne	Die Windows-Domäne des Servers, gegen den Sie die Authentifizierung durchführen möchten.

Kerberos2-Token

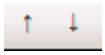
Element der Oberfläche	Beschreibung
Name des Token	Geben Sie den Namen des Token an.
Host	Der Hostname des Servers, gegen den Sie die Authentifizierung durchführen möchten. In den meisten Fällen ist dies der Hostabschnitt des Dienst-URL.
Domäne	Die Windows-Domäne des Servers, gegen den Sie die Authentifizierung durchführen möchten.

Nachrichten-signatur hinzufügen

Element der Oberfläche	Beschreibung
Signaturtoken	Das für die Signierung verwendete Token, normalerweise ein Token des Typs X.509. Treffen Sie in der Liste aller hinzugefügten Token eine Auswahl.
Kanonisierungs-algorithmus	Ein URL für den Algorithmus, der für die Kanonisierung verwendet wird. In einer Dropdownliste stehen die allgemeinen Algorithmen zur Verfügung. Wenn Sie nicht sicher sind, welchen Wert Sie verwenden sollen, behalten Sie die Standardeinstellung bei.
Transformations-algorithmus	Ein URL für den Transformationsalgorithmus, der auf die Nachrichtensignatur angewendet wird. In einer Dropdownliste stehen die allgemeinen Algorithmen zur Verfügung. Wenn Sie nicht sicher sind, welchen Wert Sie verwenden sollen, behalten Sie die Standardeinstellung bei.
Liste eingeschl. Namespaces	Eine Liste kommagetrennter Präfixe, die als inklusiv behandelt werden (optional).
Zu signierende Elemente Die zu signierenden SOAP-Elemente: SOAP-Text, Zeitstempel und WS-Adressierung.	
XPath (optional)	Ein XPath, um die zu signierenden Teile der Nachricht anzugeben. Werden keine Informationen angegeben, werden die im Feld für die Signaturoptionen ausgewählten Elemente signiert. Zum Beispiel `//*[local-name(.)='Body']`.
Token (optional)	Das zu signierende Zieltoken. Treffen Sie in der Dropdownliste aller hinzugefügten Token eine Auswahl. Für die meisten Dienste muss dieses Feld leer bleiben.

Nachrichten-verschlüsselung hinzufügen

Element der Oberfläche	Beschreibung
Ver-schlüsselungs-token	Das für die Verschlüsselung verwendete Token, normalerweise ein Token des Typs X.509. Sie können aus einer Liste aller zuvor erstellten Token eine Auswahl treffen.
Ver-schlüsselungs-typ	Gibt an, ob das gesamte Zielelement oder nur sein Inhalt verschlüsselt wird.
Schlüssel-algorithmus	Der für die Verschlüsselung des Sitzungsschlüssels verwendete Algorithmus: RSA15 oder RSAOAEP. **Hinweis:** Wenn Sie die Datei **mmdrv.config** mit einem benutzerdefinierten Schlüsselalgorithmuswert für ein vorhandenes Skript bearbeitet haben, wird dieser Wert durch den Systemstandardwert RSA15 ersetzt.
Sitzungs-algorithmus	Der für die Verschlüsselung der SOAP-Nachricht verwendete Algorithmus. Sie können aus einer Liste allgemeiner Werte eine Auswahl treffen. **Hinweis:** Wenn Sie die Datei **mmdrv.config** mit einem benutzerdefinierten Schlüsselalgorithmuswert für ein vorhandenes Skript bearbeitet haben, wird dieser Wert durch den Systemstandardwert AES128 ersetzt.
Zu verschlüsselnde Elemente	
XPath (optional)	Ein XPath, um die zu verschlüsselnden Teile der Nachricht anzuzeigen. Werden keine Informationen eingegeben, wird nur der SOAP-Text verschlüsselt.
Token (optional)	Der Name des verschlüsselten Tokens. In einem Dropdownfeld steht eine Liste aller hinzugefügten Token zur Verfügung. Für die meisten Dienste muss dieses Feld leer bleiben.

	Löschen einer Tokendefinition aus der Tabelle.
	**Nach oben/unten**. Positionierungswerkzeuge, mit denen Sie die Priorität der Sicherheitselemente festlegen können. **Hinweis:** Vergewissern Sie sich, dass die Sicherheitselemente in der Reihenfolge ihrer Priorität sortiert sind.
Zeitstempel ausschließen	Entfernt den Zeitstempel aus der SOAP-Kopfzeile, bevor das Sicherheitselement an den Server gesendet wird.

WS-Adressierung

Die Registerkarte **WS-Adressierung** gibt an, ob der Dienst die WS-Adressierung verwendet. Wenn ja, wird die Versionsnummer angegeben. Sie können auch die IP-Adresse des Servers angeben, an den die Antwort gesendet werden soll.

Dialogfeld "Sicherheitsszenario-Editor"

Mithilfe dieses Dialogfelds können Sie die Sicherheitsszenarios für Ihr Skript festlegen.

Zugriff	**SOA-Tools > Sicherheitsszenario-Editor**
Wichtige Informationen	Darüber hinaus können Sie auch Szenarios für einen bestimmten Dienst festlegen. Weitere Informationen finden Sie unter "Erstellen und Verwalten von Sicherheitsszenarios" auf Seite 1037.
Relevante Aufgaben	"Erstellen und Verwalten von Sicherheitsszenarios" auf Seite 1037

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
	Neu. Setzt den Editor zurück, damit Sie ein neues Sicherheitsszenario festlegen können. Wenn Sie Änderungen am aktuellen Szenario vorgenommen haben, werden Sie aufgefordert, diese zu speichern.
	Laden. Öffnet ein freigegebenes Szenario, das unter einem URL oder in einer Datei gespeichert ist.
	Speichern. Speichert die Szenariodatei. Wenn Sie die Datei nicht bereits gespeichert haben, werden Sie aufgefordert, einen Namen anzugeben.

	Speichern unter. Speichert die Szenariodatei an einem anderen Speicherort.
	Hilfe. Öffnet die Online-Hilfe zum Thema "Sicherheitsszenarios".
	Schließen. Schließt das Dialogfeld.
	Öffnet das Dialogfeld **Erweiterte Einstellungen**, wo Sie die Codierung, Reliable Messaging, sichere Sitzungsinformationen und die Proxykonfiguration festlegen können. Weitere Informationen finden Sie unter "Erweiterte Szenarioeinstellungen" auf Seite 1026.
Szenariotyp	Der Typ des Sicherheitsszenarios: Kein Szenario oder eine Unterkategorie von Haupt-, Sicherheits-, WCF- oder Optimierungsszenario.

Dialogfeld "Zertifikat auswählen"

In diesem Dialogfeld können Sie nach Zertifikaten suchen, die als Datei oder in einem Windows-Speicher vorliegen.

Zugriff	Sie können ein Szenario, für das ein Zertifikat erforderlich ist, auf eine der folgenden Arten auswählen: • Öffnen Sie den Sicherheitsszenario-Editor: Wählen Sie **SOA-Tools > Sicherheitsszenario-Editor** aus. • Aktivieren Sie im Dialogfeld **Dienste verwalten** die Registerkarte **Protokoll und Sicherheit** und klicken Sie auf die Schaltfläche **Daten bearbeiten**. Wählen Sie ein WCF-Szenario mit einem Client- oder Dienstzertifikat aus, z. B. **WsHttpBinding** oder **Federation**. Klicken Sie neben dem Feld **Zertifikat** auf die Durchsuchen-Schaltfläche.
Wichtige Informationen	Gilt nur für Sicherheitsszenarios, bei denen Sie Client-, Server- oder Dienstzertifikate angeben können.
Relevante Aufgaben	• "Erstellen und Verwalten von Sicherheitsszenarios" auf Seite 1037. • "Erstellen und Verwalten von Sicherheitsszenarios" auf Seite 1037.

Auswählen des Zertifikats aus einer Datei

Wenn Sie die Option **Datei** auswählen, weist die Benutzeroberfläche folgende Elemente auf:

Element der Oberfläche	Beschreibung

...	**Durchsuchen.** Ermöglicht Ihnen das Suchen nach einer Zertifikatsdatei mit den Erweiterungen **.pem**, **.arm**, **.der** oder **.pfx**.
Datei	Der vollständige Pfad der Zertifikatsdatei.
Kennwort (optional)	Das für den Zugriff auf das Zertifikat erforderliche Kennwort.

Auswählen eines Zertifikats aus dem Windows-Speicher

Wenn Sie die Option **Windows-Speicher** auswählen, weist die Benutzeroberfläche folgende Elemente auf (Elemente ohne Beschriftung werden in spitzen Klammern dargestellt):

Element der Oberfläche	Beschreibung
Suchen	Startet die Suche nach dem Zertifikat.
Importieren aus	Der Speicherort des Zertifikats. • **Windows-Speicher** • **File**
Speicherort	Der Speicherort, beispielsweise **Aktueller Benutzer**.
Speichername	Der Speichername, beispielsweise **AuthRoot**.
Suchtext	Der Suchtext, nach dem im Zertifikatsnamen gesucht wird.
Kennwort (optional)	Das für den Zugriff auf das Zertifikat erforderliche Kennwort.
<Zertifikatsliste>	Eine Liste der Zertifikate im Windows-Speicher, sortiert nach Thema, Aussteller, Privat, Speicherort und Speichername.

Sicherheitsszenarios für Webdienste

In diesem Abschnitt werden gängige Sicherheitsszenarios beschrieben.

Authentifizieren mit einem Benutzernamentoken

Im folgenden Beispiel wird der Versand eines Benutzernamen/Kennwort-Tokens auf Nachrichtenebene (Benutzernamentoken) veranschaulicht. Der Benutzername lautet "John", das Kennwort "1234".

```
    web_service_set_security(
        SECURITY_TOKEN, "Type=USERNAME","LogicalName=myToken", "UserName=John",
"Password=1234", "PasswordOptions=SendPlainText", "Add=True",
        LAST);
```

Signieren eines bestimmten Elements mit einem X.509-Zertifikat

Es ist möglich, nur ein bestimmtes Element einer Nachricht zu signieren. Im folgenden Beispiel geschieht dies mithilfe eines XPATH-Ausdrucks:

```
    web_service_set_security(
        SECURITY_TOKEN, "Type=X509","LogicalName=myCert", "StoreName=My", "IDTyp
e=SubjectName", "IDValue=CN=myCert", "StoreLocation=CurrentUser", "Add=True",
        MESSAGE_SIGNATURE, "UseToken=myCert", "TargetPath=//*[local-name(.)='som
eElement' and namespace-uri(.)='http://myNamespace']",
        LAST);
```

Signieren mit einem X.509-Zertifikat

Das folgende Beispiel zeigt ein Skript, das eine digitale Signatur mithilfe eines X.509-Zertifikats erstellt.

```
    web_service_set_security(
        SECURITY_TOKEN, "Type=X509","LogicalName=myCert", "StoreName=My", "IDTyp
e=SubjectName", "IDValue=CN=myCert", "StoreLocation=CurrentUser", "Add=True",
        MESSAGE_SIGNATURE, "UseToken=myCert",
        LAST);
```

Hinweis: Ihr Zertifikat muss im Zertifikatspeicher von Windows installiert sein. Im obigen Beispiel müssen Sie den Namen des tatsächlichen Speichers, den Speicherort sowie den Namen des Antragstellers Ihres Zertifikats angeben.

Verschlüsseln mit einem Zertifikat

Im folgenden Beispiel wird eine Nachricht mit dem X.509-Zertifikat des Dienstes verschlüsselt.

```
    web_service_set_security(
        SECURITY_TOKEN, "Type=X509","LogicalName=serviceCert", "StoreName=My", "
IDType=SubjectName", "IDValue=CN=serviceCert", "StoreLocation=CurrentUser", "Add
=False",
        ENCRYPTED_DATA, "UseToken=serviceCert",
        LAST);
```

Nach der Angabe der Details Ihres X.509-Zertifikats können Sie einen bestimmten XPATH in der Nachricht verschlüsseln.

Da die Schlüsselkennung des Antragstellers generiert werden soll, wird Wert für die Option **Add** auf **False** eingestellt.

Authentifizieren mit einem Benutzernamentoken und Verschlüsselung mit einem X.509-Zertifikat

Im folgenden Beispiel wird ein Benutzernamentoken an den Dienst gesendet und mit dem X.509-Zertifikat des Dienstes verschlüsselt:

```
web_service_set_security(
        SECURITY_TOKEN, "Type=X509","LogicalName=serviceCert", "StoreName=My", "
IDType=SubjectName", "IDValue=CN=serviceCert", "StoreLocation=CurrentUser", "Add
=True",
```

```
        SECURITY_TOKEN, "Type=USERNAME","LogicalName=myUser", "UserName=John", "
Password=1234", "PasswordOptions=SendPlainText", "Add=True",
        ENCRYPTED_DATA, "UseToken=serviceCert", "TargetToken=myUser",
        LAST);
```

Durch die Eigenschaften **UseToken** und **TargetToken** wird angegeben, welcher Token verwendet und welcher verschlüsselt werden soll. Ihre jeweiligen Werte verweisen auf die Eigenschaft **LogicalName** der Token.

Verschlüsseln und Signieren einer Nachricht

In diesem Beispiel wird eine Nachricht mit einem privaten Schlüssel signiert und anschließend mit dem öffentlichen Schlüssel des Dienstes verschlüsselt.

```
web_service_set_security(
        SECURITY_TOKEN, "Type=X509","LogicalName=myCert", "StoreName=My", "IDTyp
e=SubjectName", "IDValue=CN=myCert", "StoreLocation=CurrentUser", "Add=True",
        SECURITY_TOKEN, "Type=X509","LogicalName=serverToken", "StoreName=My", "
IDType=SubjectName", "IDValue=CN=serverCert", "StoreLocation=CurrentUser", "Add=
False",
        MESSAGE_SIGNATURE, "UseToken=myCert",
        ENCRYPTED_DATA, "UseToken=serverCert",
    LAST);
```

Referenzieren eines X.509-Zertifikats mit einem Hash-Wert

Manchmal kann nicht mit dem Namen des Antragstellers auf ein Zertifikat verwiesen werden. In diesem Beispiel wird das Zertifikat mithilfe von dessen eindeutigem Hash-Wert referenziert.

```
web_service_set_security(
   SECURITY_TOKEN, "Type=X509","LogicalName=serviceCert", "StoreName=My",    "ID
Type=Base64KeyID", "IDValue=pOl0+1iuotKLlO91nhjDg5reEw0=",    "StoreLocation=Cur
rentUser", "Add=False",
   ENCRYPTED_DATA, "UseToken=serviceCert",
 LAST);
```

Fehlerbehebung und Einschränkungen – Webdienste

In diesem Abschnitt werden die Fehlerbehebung und die für das Webdienstprotokoll geltenden Einschränkungen beschrieben.

- Die Option **Standardwebbrowser aufzeichnen** im Aufzeichnungs-Assistenten wird nur für Internet Explorer unterstützt.
- Für große SOAP-Umschläge sind Aufzeichnungs- und Wiedergabe-Snapshots deaktiviert.
- Die Funktion **SOAP importieren** wird nicht für Umschläge unterstützt, die einzelne Elemente größer als 500 KB enthalten.
- Aufzeichnungsanforderungen mit Anhängen oder Sicherheit werden nicht unterstützt.

- Beim Axis-Toolkit werden Webdienstaufrufe, die sowohl Anhänge als auch Sicherheit enthalten, nicht unterstützt.
- Beim .NET-Toolkit wird bei asynchronen Aufrufen SOAP Version 1.2 nicht unterstützt.
- Sie können Zeichenfolgen mit bis zu 10 KB für die Codierung in base64 eingeben. Wenn Ihre Zeichenfolge länger ist, verwenden Sie die Option **Aus Datei beziehen**.
- VuGen unterstützt Webdienstmeldungen über die JMS-Meldungswarteschlange; JMS-Themen jedoch nicht.
- JMS Bindings Extensions werden nicht unterstützt.
- Alle Dienste in Ihrem Skript müssen dasselbe Sicherheitsszenario verwenden. Dies kann über die Registerkarte **Protokoll und Sicherheit** konfiguriert werden.
- Asynchrone Webdienstaufrufe und benutzerdefinierte Benutzer-Handler werden für WCF nicht unterstützt.
- LoadRunner kann keine Skripts wiedergeben, welche die Funktion **soa_xml_validate** enthalten.
- Wenn Sie den Aktualisierungsdienst verwenden, werden in den Schritten, die sich bereits im Skript befinden, die aktualisierten Eigenschaften erst angezeigt (in der Schrittargument-Ansicht), wenn Sie die Anwendung schließen und erneut öffnen. Wenn Sie die Anwendung erneut geöffnet haben, sind die Schrittargumente aktualisiert. Wenn das Skript beim Ausführen des Aktualisierungsdiensts geöffnet ist, löst die Anwendung in der Schrittargument-Ansicht eine Ausnahme aus.

  Problemumgehung: Schließen Sie die Skriptdatei beim Ausführen des Aktualisierungsdiensts, oder öffnen Sie den Test erneut nach dem Ausführen des Aktualisierungsdiensts.
- Ein Webdienstskript wird möglicherweise nicht geöffnet, wenn Sie die WDSL mit dem Axis-Toolkit importieren.

  Problemumgehung: Importieren Sie die WDSL mit dem .NET-Toolkit.

  Gehen Sie wie folgt vor, wenn die Skripts nicht neu erstellt werden können:

  a. Erstellen Sie einen neuen Test.

  b. Importieren Sie die WSDL mit dem .NET-Toolkit.

  c. Wechseln Sie zu dem Verzeichnis des neuen Skripts.

  d. Kopieren Sie den Ordner "WSDL", und fügen Sie ihn in das Verzeichnis des alten Skripts ein.

  e. Öffnen Sie im Verzeichnis des alten Skripts die Datei 'default.cfg'.

f. Ersetzen Sie unter der Überschrift [WebServices] den Text "Toolkit=Axis" durch "Toolkit=.Net".

Fehlerbehebung und Einschränkungen – Webdienste

In diesem Abschnitt werden die Fehlerbehebung und die für das Webdienstprotokoll geltenden Einschränkungen beschrieben.

- Die Option **Standardwebbrowser aufzeichnen** im Aufzeichnungs-Assistenten wird nur für Internet Explorer unterstützt.
- Für große SOAP-Umschläge sind Aufzeichnungs- und Wiedergabe-Snapshots deaktiviert.
- Die Funktion **SOAP importieren** wird nicht für Umschläge unterstützt, die einzelne Elemente größer als 500 KB enthalten.
- Aufzeichnungsanforderungen mit Anhängen oder Sicherheit werden nicht unterstützt.
- Beim Axis-Toolkit werden Webdienstaufrufe, die sowohl Anhänge als auch Sicherheit enthalten, nicht unterstützt.
- Beim .NET-Toolkit wird bei asynchronen Aufrufen SOAP Version 1.2 nicht unterstützt.
- Sie können Zeichenfolgen mit bis zu 10 KB für die Codierung in base64 eingeben. Wenn Ihre Zeichenfolge länger ist, verwenden Sie die Option **Aus Datei beziehen**.
- VuGen unterstützt Webdienstmeldungen über die JMS-Meldungswarteschlange; JMS-Themen jedoch nicht.
- JMS Bindings Extensions werden nicht unterstützt.
- Alle Dienste in Ihrem Skript müssen dasselbe Sicherheitsszenario verwenden. Dies kann über die Registerkarte **Protokoll und Sicherheit** konfiguriert werden.
- Asynchrone Webdienstaufrufe und benutzerdefinierte Benutzer-Handler werden für WCF nicht unterstützt.
- LoadRunner kann keine Skripts wiedergeben, welche die Funktion **soa_xml_validate** enthalten.
- Wenn Sie den Aktualisierungsdienst verwenden, werden in den Schritten, die sich bereits im Skript befinden, die aktualisierten Eigenschaften erst angezeigt (in der Schrittargument-Ansicht), wenn Sie die Anwendung schließen und erneut öffnen. Wenn Sie die Anwendung erneut geöffnet haben, sind die Schrittargumente aktualisiert. Wenn das Skript beim Ausführen des Aktualisierungsdiensts geöffnet ist, löst die Anwendung in der Schrittargument-Ansicht eine Ausnahme aus.

  Problemumgehung: Schließen Sie die Skriptdatei beim Ausführen des Aktualisierungsdiensts, oder öffnen Sie den Test erneut nach dem Ausführen des Aktualisierungsdiensts.
- Ein Webdienstskript wird möglicherweise nicht geöffnet, wenn Sie die WDSL mit dem Axis-

Toolkit importieren.

Problemumgehung: Importieren Sie die WDSL mit dem .NET-Toolkit.

Gehen Sie wie folgt vor, wenn die Skripts nicht neu erstellt werden können:

a. Erstellen Sie einen neuen Test.

b. Importieren Sie die WSDL mit dem .NET-Toolkit.

c. Wechseln Sie zu dem Verzeichnis des neuen Skripts.

d. Kopieren Sie den Ordner "WSDL", und fügen Sie ihn in das Verzeichnis des alten Skripts ein.

e. Öffnen Sie im Verzeichnis des alten Skripts die Datei 'default.cfg'.

f. Ersetzen Sie unter der Überschrift [WebServices] den Text "Toolkit=Axis" durch "Toolkit=.Net".

Windows-Sockets-Protokoll

Übersicht über das Aufzeichnen von Windows-Sockets

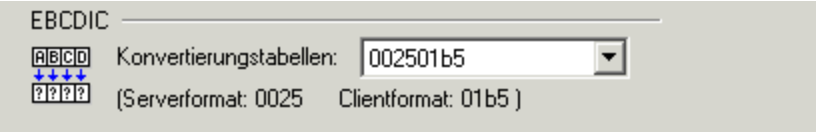
Das Windows-Sockets-Protokoll unterstützt Anwendungen, die über das TCP/IP-Protokoll mithilfe einer WinSock-DLL von Microsoft kommunizieren. Das WinSock-Protokoll ermöglicht es Ihnen, die eigentlichen Daten anzuzeigen, die von den Puffern gesendet und empfangen wurden.

Das WinSock-Protokoll zeichnet Funktionen auf, die sich auf die Sockets, Datenpuffer und die Windows-Sockets-Umgebung beziehen. Mit VuGen zeichnen Sie die API-Aufrufe Ihrer Anwendung in der Datei **Winsock.dll** oder **Wsock32.dll** auf. Beispielsweise können Sie ein Skript erstellen, indem Sie die Aktionen einer telnet-Anwendung aufzeichnen. Nach der Erstellung eines Winsock-Vuser-Skripts können Sie die aufgezeichneten Puffer als Rohdatendatei oder als Snapshot anzeigen. Weitere Informationen finden Sie unter "Windows-Sockets-Daten" auf der nächsten Seite oder "Windows-Sockets-Snapshots - Übersicht" auf Seite 1054.

Konvertierungstabellen

Sie können Windows-Socket-Daten im EBCDIC-Format über eine Konvertierungstabelle anzeigen.

Eine Konvertierungstabelle ermöglicht es Ihnen, das Format für die Aufzeichnung anzugeben, wenn Sie WinSock als Einzelprotokoll verwenden, und Sie können sie für die Codegenerierung heranziehen, wenn Sie ein WinSock-Mehrfachprotokoll verwenden. Dieses gilt für Benutzer, die auf Mainframecomputern oder AS/400-Servern ausgeführt werden. Sowohl Server als auch Client beziehen das Datenformat aus den auf Ihrem System installierten Übersetzungstabellen. Daten im ASCII-Format erfordern keine Übersetzung.



Die ersten vier Zeichen des Listenfeldelements stehen für das Serverformat. Die letzten vier Zeichen stehen für das Clientformat. Im Beispiel oben wurde als Übersetzungstabelle `002501b5` ausgewählt. Das Serverformat `0025` und das Clientformat `01b5` bezeichnet eine Datenübertragung vom Server zum Client. Bei einer Datenübertragung vom Client zum Server muss das Element mit dem umgekehrten Format `01b50025` ausgewählt werden. Das bedeutet, dass das Clientformat `01b5` in das Serverformat `0025` zu übersetzen ist.

Die Übersetzungstabellen befinden sich im Ordner **ebcdic** des VuGen-Installationsordners. Wenn Ihr System andere Übersetzungstabellen verwendet, kopieren Sie sie in den Ordner **ebcdic**.

Weitere Informationen zum Auswählen einer Konvertierungstabelle mithilfe der Aufzeichnungsoptionen finden Sie unter "Knoten "WinSock"" auf Seite 248.

Windows-Sockets-Daten

Wenn Sie mit VuGen ein Windows-Sockets-Vuser-Skript erstellen, werden Ihre Aktionen in drei Skriptabschnitten aufgezeichnet: **vuser_init**, **Action** und **vuser_end**. Zusätzlich zum Vuser-Skript erstellt VuGen auch die folgenden Datendateien:

- **snapshotdata.ws** enthält die Daten, die während der Aufzeichnungssitzung übertragen oder empfangen wurden. Im Snapshot-Bereich von VuGen wird der Inhalt der Datendatei angezeigt. Der Inhalt der Datei **snapshotdata.ws** darf nicht geändert werden.

- **data.ws** enthält die während der Wiedergabesitzungen übertragenen Daten und erwartungsgemäß empfangenen Daten. Sie können im Editor mit der rechten Maustaste auf einen beliebigen Schritt klicken und anschließend **Argumente anzeigen** auswählen, um den Pufferinhalt anzuzeigen, der für den ausgewählten Schritt in der Datei **data.ws** gespeichert ist. In der Registerkarte **Textansicht** des geöffneten Dialogfelds können Sie die für jeden Datenpuffer gespeicherten Daten bearbeiten.

Einige LRS-Funktionen, z. B. **lrs_receive** und **lrs_send**, verarbeiten die Daten, die tatsächlich zwischen Servern und Clients übertragen werden. Die empfangenen oder übertragenen Daten werden in Datenpuffern gespeichert, die mitunter sehr umfangreich sind. Um die Darstellung des Vuser-Skripts möglichst einfach zu halten, werden die eigentlichen Daten in externen Dateien und nicht in der C-Datei gespeichert. Bei einer Datenübertragung werden die Daten aus der externen Datei in einen temporären Puffer kopiert.

Die externe Datei **data.ws** enthält den Inhalt aller temporären Puffer. Der Inhalt der Puffer wird in Form von fortlaufenden Einträgen gespeichert. Diese werden mit einer Kennung als gesendet oder empfangen und mit dem Pufferdeskriptor versehen. Die LRS-Funktionen verwenden die Pufferdeskriptoren für den Zugriff auf die Daten.

Die Deskriptoren verfügen über eines der folgenden Formate:

```
recv buf index   number of bytes received
send buf index   number of bytes sent
```

Der Pufferindex beginnt bei 0 (null). Alle folgenden Puffer werden fortlaufend durchnummeriert (1,2,3…) - unabhängig davon, ob es sich um Sende- oder Empfangspuffer handelt.

Das folgende Beispiel zeigt eine **lrs_receive**-Funktion, die in einer Vuser-Sitzung aufgezeichnet wurde:

```
lrs_receive("socket1", "buf4", LrsLastArg)
```

In diesem Beispiel verarbeitet die Funktion **lrs_receive** Daten, die von **socket1** empfangen wurden. Die Daten wurde im fünften Empfangseintrag (buf4) gespeichert. Beachten Sie, dass die Indexnummer bei null beginnt. Im entsprechenden Bereich der Datei **data.ws** wird der Puffer mit seinem Inhalt angezeigt.

```
recv buf4 39
    "\xff\xfb\x01\xff\xfb\x03\xff\xfd\x01"
    "\r\n"
    "\r\n"
    "SunOS UNIX (sunny)\r\n"
    "\r"
    "\x0"
    "\r\n"
    "\r"
    "\x0"
```

Weitere Informationen über die Aufgabe finden Sie unter "Anzeigen und Ändern von Windows-Socket-Puffern" auf Seite 1058.

Windows-Sockets-Snapshots - Übersicht

Auf dem Windows-Sockets-Vuser-Protokoll basierende Vuser-Skripts verwenden den Snapshot-Bereich von VuGen.

- Eine Einführung zum Snapshot-Bereich finden Sie unter "Snapshot-Bereich - Übersicht" auf Seite 62.
- Details zum Arbeiten mit dem Snapshot-Bereich finden Sie unter "Arbeiten mit Snapshots" auf Seite 65.
- Details zur Benutzeroberfläche des Snapshot-Bereichs finden Sie unter "Snapshot-Bereich" auf Seite 109.

Wenn Sie ein Windows-Sockets-Vuser-Skript erstellen, stehen die Standardfunktionen des Snapshot-Bereichs von VuGen zur Verfügung. Für Windows-Sockets-Vuser-Skripts zeigt der Snapshot-Bereich Snapshots der aufgezeichneten Datenpuffer an.

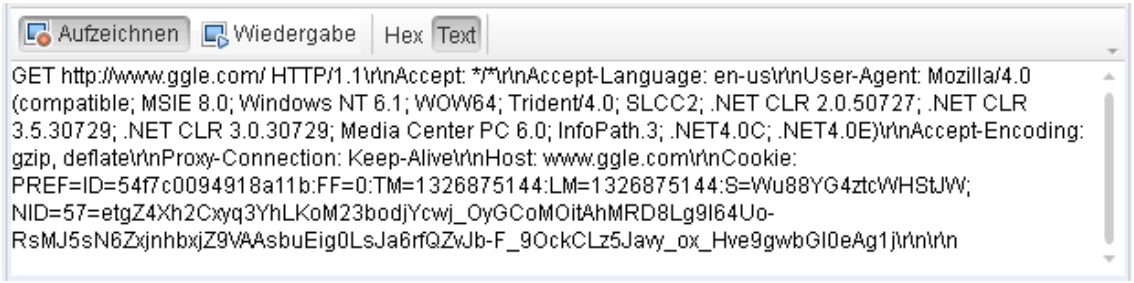
So zeigen Sie einen bestimmten Puffer im Snapshot-Bereich an:

- Wählen Sie im Editor den Schritt aus, der eine Referenz zu dem erforderlichen Puffer enthält.
- Doppelklicken Sie im Schrittenavigator auf den Schritt, der eine Referenz zu dem erforderlichen Puffer enthält.

Sie können die Puffersnapshots in der **Textansicht** oder in der **Hexadezimalansicht** anzeigen.

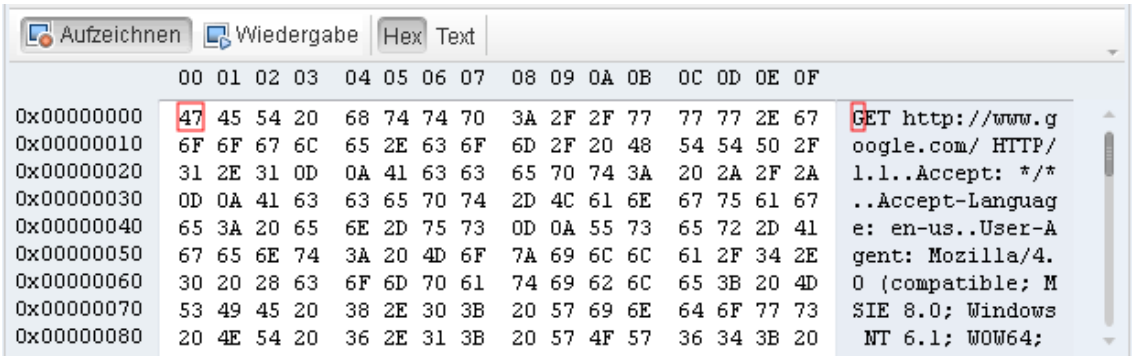
Hinweis: VuGen speichert die Snapshot-Daten schreibgeschützt. Sie können den Inhalt der Snapshots nicht ändern. Sie können jedoch die Pufferdaten ändern, die mit beliebigen Schritten in einem Vuser-Skript verbunden sind. Um die Pufferdaten zu ändern, klicken Sie im Editor mit der rechten Maustaste auf den erforderlichen Schritt und wählen Sie **Argumente anzeigen** aus. In der Registerkarte **Textansicht** des geöffneten Dialogfelds können Sie die Pufferdaten ändern. Weitere Informationen finden Sie unter "Anzeigen und Ändern von Windows-Socket-Puffern" auf Seite 1058.

In der **Textansicht** werden die Pufferdaten als Text angezeigt.



In der **Hexadezimalansicht** werden die Pufferdaten in Hexadezimaldarstellung angezeigt. Die Daten werden in drei Spalten dargestellt:

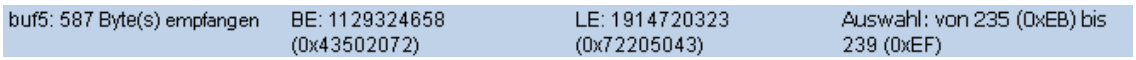
- Die linke Spalte enthält den Offset des ersten Zeichens in jeder Zeile.
- Die mittlere Spalte enthält den Hexadezimalwert der Daten.
- Die rechte Spalte stellt die Daten im ASCII-Format dar.



Die Statusleiste unter dem Puffersnapshot stellt Informationen zu dem Puffer und den im Puffer ausgewählten Daten bereit:

- **Puffernummer.** Die Nummer des angezeigten Puffers.
- **Puffergröße.** Die Gesamtanzahl der Byte im Puffer.
- **Puffertyp.** Der Typ des Puffers: **empfangen** oder **gesendet**.

- **Daten.** Der Wert der im Puffer ausgewählten Daten im Dezimal- und Hexadezimalformat. Es werden sowohl Big Endian- als auch Little Endian-Sequenzen angezeigt.
- **Offset.** Der Offset der ausgewählten Daten vom Anfang des Puffers. Wenn Sie mehrere Byte auswählen, zeigt der Offset den Auswahlbereich an.



Navigieren in den Pufferdaten

- Um einen bestimmten Offset im Puffer aufzurufen (absolut), klicken Sie auf **Gehe zu Offset**. Geben Sie im Dialogfeld **Gehe zu Offset** einen Offsetwert ein und klicken Sie auf **Übernehmen**.
- Um zu einer Position relativ zu dem ausgewählten Eintrag zu springen, klicken Sie auf **Gehe zu**. Klicken Sie im Dialogfeld **Gehe zu Offset** auf **Bewegen um**, legen Sie die Anzahl der Byte für die Bewegung fest und klicken Sie anschließend auf **Übernehmen**.

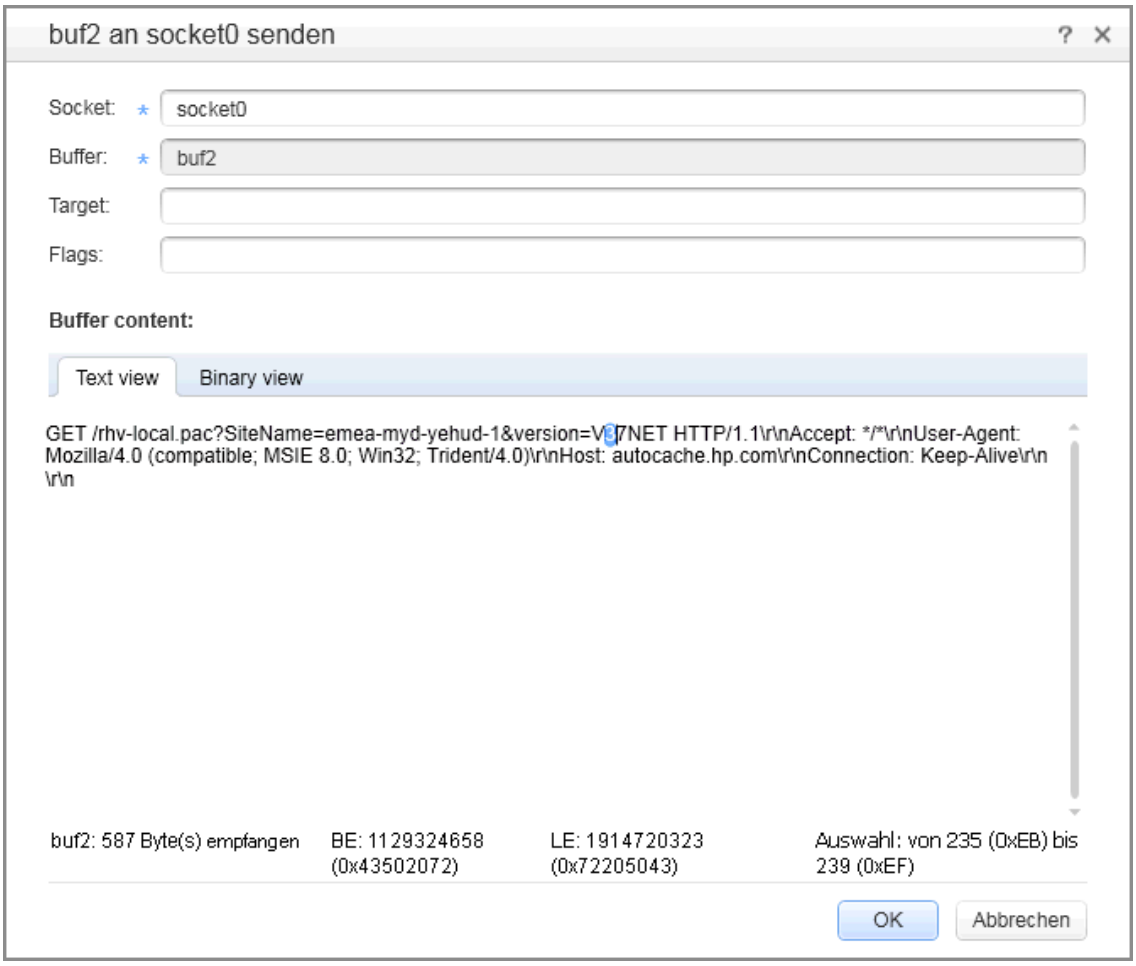
Datennavigationstools

VuGen bietet Ihnen die Funktionalität **Gehe zu** als Unterstützung der Navigation durch die Daten im Snapshot-Bereich. Damit können Sie leichter bestimmte Werte im Snapshot erkennen und analysieren. Sie können sich durch Angabe eines Offsets durch den Datenpuffer bewegen. Dabei können Sie die Datenposition als absoluten Wert angeben oder relativ zur aktuellen Cursorposition im Puffer. Sie können außerdem einen Datenbereich auswählen, indem Sie den Start- und den Endoffset angeben. Weitere Informationen zu den Optionen des Dialogfelds finden Sie unter "Dialogfeld "Gehe zu Offset"" auf Seite 1063.

Bearbeiten von Pufferdaten

Sie können die Pufferdaten auf alle gängigen Arten bearbeiten: durch Kopieren, Einfügen und Löschen. Um die Pufferdaten zu bearbeiten, klicken Sie im Editor mit der rechten Maustaste auf den erforderlichen Schritt und wählen Sie **Argumente anzeigen** aus. Anschließend können Sie den Bearbeitungsvorgang auf der Registerkarte **Textansicht** des jetzt geöffneten Dialogfelds durchführen. Sie können keine Bearbeitungen in der Registerkarte **Binäre Ansicht** vornehmen.

Hinweis: Sie dürfen nur die Pufferdaten bearbeiten, nicht die schreibgeschützten Snapshotdaten.



Wenn Sie Daten aus einem Puffer kopieren, ermöglicht VuGen den Kopiervorgang entweder im hexadezimalen oder im dezimalen Format. Beim Einfügen von Daten in einen Puffer können Sie in VuGen das Format der Daten angeben: Single-Byte, 2-Byte oder 4-Byte.

Aufzeichnen eines Windows-Sockets-Skripts

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie eine Windows-Sockets-Aufzeichnung einrichten und die Sitzung aufzeichnen.

1. **Öffnen der Aufzeichnungsoptionen - optional**

   Wählen Sie nach dem Erstellen eines WinSock-Skripts **Aufzeichnen > Aufzeichnungsoptionen** aus und klicken Sie auf den Knoten **WinSock**.

2. **Auswählen einer Konvertierungstabelle - optional**

   Wählen Sie im Abschnitt **EBCDIC** eine Konvertierungstabelle aus. Wenn Ihre Daten im ASCII-Format vorliegen, wählen Sie die Option **Keine** aus, andernfalls konvertiert VuGen die ASCII-Daten. Weitere Informationen finden Sie unter "Konvertierungstabellen" auf Seite 1052.

3. **Ausschließen aller nicht relevanten Sockets - optional**

   Fügen Sie der Liste im Abschnitt **Ausschlusseinstellungen** nicht relevante Sockets hinzu. Es wird empfohlen, nur solche Hosts und Ports auszuschließen, die im Test keine Auswirkungen auf die Serverlast haben, ähnlich wie der lokale Host und der DNS-Port 53, die standardmäßig ausgeschlossen sind.

   Um die Einträge von der Aufzeichnung auszuschließen, aber in das Protokoll aufzunehmen, deaktivieren Sie die Option **Ausgeschlossenes Socket nicht in Protokoll aufnehmen**.

   Informationen zur Benutzeroberfläche erhalten Sie unter "Knoten "WinSock"" auf Seite 248.

4. **Festlegen eines Latenzzeit-Schwellenwerts - optional**

   Geben Sie einen Latenzzeit-Schwellenwert an. Wenn VuGen eine Aktionspause erkennt, die die Schwellenwertzeit unterschreitet, wird keine **Latenzzeit**-Schritt/**lr_think_time**-Funktion generiert. Details finden Sie im "Knoten "WinSock"" auf Seite 248.

5. **Aufzeichnen der Sitzung**

   Zeichnen Sie die Sitzung auf und speichern Sie das Skript.

6. **Parametrieren des Skripts - optional**

   Ersetzen Sie über das Kontextmenü die aufgezeichneten Werte durch Parameter. Weitere Informationen finden Sie in "Parameter" auf Seite 436.

7. **Neugenerieren des Skripts - optional**

   Wenn Sie das Skript neu generieren müssen, beispielsweise um einen ausgeschlossenen **host:port** aufzunehmen oder wenn die Übersetzung nicht korrekt ist:

   - Wählen Sie **Aufzeichnen > Skript neu generieren** aus.
   - Klicken Sie auf den Link **Optionen**.
   - Wählen Sie unter **Allgemein** den Eintrag **Protokolle** aus und stellen Sie dann unter **Aktive Protokolle** sicher, dass das Kontrollkästchen **Windows Sockets** aktiviert ist.
   - Wählen Sie unter **Sockets** den Eintrag **Winsock** aus und ändern Sie die Einstellungen nach Bedarf.

   Hinweis: Die Optionen für das Neugenerieren von Skripts steht nur für Mehrfachprotokollskripts zur Verfügung.

Anzeigen und Ändern von Windows-Socket-Puffern

Mithilfe der folgenden Schritte können Sie WinSock-Pufferdaten anzeigen und ändern sowie durch die Daten navigieren.

Ändern von Pufferdaten

Sie können im Dialogfeld **Argumente anzeigen** die Pufferdaten für bestimmte **lrs**-Schritte in einem Vuser-Skript ändern. Sie können das Dialogfeld **Argumente anzeigen** verwenden, um Pufferdaten für die folgenden Schritte zu ändern:

- lrs_length_receive
- lrs_length_send
- lrs_receive
- lrs_receive_ex
- lrs_send

Weitere Informationen zu diesen Schritten finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Um das Dialogfeld **Argumente anzeigen** für beliebige der zuvor genannten Schritte anzuzeigen, klicken Sie mit der rechten Maustaste im Editor auf einen Schritt und wählen **Argumente anzeigen** aus. Es wird ein Dialogfeld mit den Pufferdaten im Abschnitt **Pufferinhalt** des Dialogfelds angezeigt.

Weitere Informationen zum Bearbeiten von Pufferdaten in Windows-Sockets-Schritten finden Sie unter "Bearbeiten von Pufferdaten" auf Seite 1056.

> Beachten Sie, dass Sie im Bereich **Snapshot** keine Daten ändern können.

Anzeigen und Ändern der Daten in der Datei data.ws

Doppelklicken Sie im Lösungs-Explorer auf die Datei **data.ws**. Der Inhalt der Datei **data.ws** wird im VuGen-Editor angezeigt. Ändern Sie die Daten direkt im Editor. Weitere Informationen finden Sie unter "Windows-Sockets-Daten" auf Seite 1053.

> Beachten Sie, dass Sie die Dateien **data.ws** zwar ändern können, dies jedoch nicht empfohlen wird.

Anzeigen der Daten im Snapshot-Bereich

1. Stellen Sie sicher, dass der Snapshot-Bereich angezeigt wird.
2. Öffnen Sie das Vuser-Skript im Editor und wählen Sie einen Schritt aus oder doppelklicken Sie im Schrittenavigator auf einen Eintrag. Der entsprechende Snapshot wird im Snapshot-Bereich angezeigt. Sie können die Snapshot-Daten nicht bearbeiten.

Navigieren in den Snapshot-Daten

Um in den Pufferdaten zu navigieren, zeigen Sie den Snapshot-Bereich an und klicken auf **Gehe zu**. Das Dialogfeld **Gehe zu Offset** wird geöffnet.

- Um einen bestimmten Offset im Puffer aufzurufen (absolut), wählen Sie **Gehe zu Offset** aus und geben Sie einen Offsetwert an. Klicken Sie auf **Übernehmen**.
- Um zu einer Stelle relativ zur Cursorposition zu springen, klicken Sie auf **Bewegen um** und geben Sie Anzahl der Bytes an, um die Sie den Cursor bewegen möchten. Geben Sie einen positiven Wert ein, um sich vorwärts im Puffer zu bewegen. Geben Sie einen negativen Wert ein, um sich rückwärts im Puffer zu bewegen. Klicken Sie auf **Übernehmen**.
- Wenn Sie einen Datenbereich im Puffer auswählen möchten, klicken Sie auf **Bereich auswählen von** und geben Sie den Start- und den Endoffset ein. Klicken Sie auf **Übernehmen**.

Einfügen von Daten in einen Puffer

Sie können einen numerischen Wert in einen Datenpuffer eingeben. Die Dateneingabe ist als Single-Byte-, Double-Byte- oder 4-Byte-Wert möglich. In den folgenden Schritten wird das Einfügen eines Werts in einen Datenpuffer beschrieben.

1. Kopieren Sie die einzufügenden numerischen Daten in die Zwischenablage.
2. Klicken Sie im Editor mit der rechten Maustaste auf einen Schritt und wählen Sie **Argumente anzeigen** aus.
3. Vergewissern Sie sich im angezeigten Dialogfeld, dass die Registerkarte **Textansicht** angezeigt wird.
4. Klicken Sie unter **Pufferinhalt** mit der rechten Maustaste auf die Position im Puffer, an der Sie die Daten einfügen möchten, und wählen Sie anschließend **Erweitert > Byte einfügen** oder **Erweitert > Kurz einfügen (2 Bytes)** oder **Erweitert > Lang einfügen (4 Bytes)** aus.
5. Klicken Sie auf **OK**. VuGen fügt die Daten in den Puffer ein.

Kopieren und Einfügen von Datenblöcken

Sie können die Pufferdaten als Zeichen, Dezimalzahlen oder Hexadezimalzahlen ändern. Weitere Informationen finden Sie unter "Bearbeiten von Pufferdaten" auf Seite 1056.

Beachten Sie, dass Sie Pufferdaten nur bearbeiten können, wenn Sie die Schrittargumente anzeigen. Im Snapshot-Bereich können Sie Pufferdaten nicht bearbeiten.

1. Klicken Sie im Editor mit der rechten Maustaste auf einen Schritt und wählen Sie **Argumente anzeigen** aus. Es wird ein Dialogfeld mit den Pufferdaten im Abschnitt **Pufferinhalt** des Dialogfelds angezeigt.
2. So kopieren Sie Pufferdaten:
   - Wählen Sie zum Kopieren von Pufferdaten als Zeichen ein oder mehrere Byte aus und drücken Sie Strg+C.
   - Wählen Sie zum Kopieren als Dezimalzahl ein oder mehrere Byte aus, klicken Sie mit der rechten Maustaste in die Auswahl und wählen Sie **Erweitert > Als Dezimalzahl kopieren**.
   - Wählen Sie zum Kopieren als Hexadezimalzahl ein oder mehrere Byte aus, klicken Sie mit der rechten Maustaste in die Auswahl und wählen Sie **Erweitert > Als Hexadezimalzahl kopieren** aus.
3. So fügen Sie Daten ein:
   - Zum Einfügen als Zeichen drücken Sie Strg+V.
   - Klicken Sie zum Einfügen von Daten als Single-Byte-Wert (angenommene Datengröße im Pufferspeicher) mit der rechten Maustaste an die gewünschte Stelle im Puffer und wählen Sie **Erweitert > Byte einfügen**.
   - Zum Einfügen im Kurzformat (2 Byte) klicken Sie im Puffer mit der rechten Maustaste an die gewünschte Stelle und wählen **Erweitert > Kurz einfügen (2 Bytes)** aus.
   - Zum Einfügen im Langformat (4 Byte) klicken Sie im Puffer mit der rechten Maustaste an die gewünschte Stelle und wählen **Erweitert > Lang einfügen (4 Bytes)** aus.
4. Um Daten zu löschen, markieren Sie die Daten in der Textansicht, klicken mit der rechten Maustaste in die Auswahl und wählen **Löschen** aus.

Datenpuffer

Wenn Sie VuGen zum Erstellen eines Windows-Sockets-Vuser-Skripts verwenden, erstellt VuGen die Datendatei **data.ws**. Diese Datei enthält die übertragenen Daten. Es wird erwartet, dass diese Datei während der Wiedergabesitzungen empfangen wird. Sie können im Editor mit der rechten Maustaste auf einen beliebigen Schritt klicken und anschließend **Argumente anzeigen** auswählen, um den Pufferinhalt anzuzeigen, der für den ausgewählten Schritt in der Datei **data.ws** gespeichert ist. In der Registerkarte **Textansicht** des geöffneten Dialogfelds können Sie die für jeden Datenpuffer gespeicherten Daten bearbeiten.

Die Datendatei **data.ws** weist das folgende Format auf:

- Dateikopfzeile
- Liste der Puffer und ihrer Inhalte

Die Dateikopfzeile enthält eine interne Versionsnummer für das Format der Datendatei. Die aktuelle Version ist 2. Wenn Sie die Daten mit einer Datendatei der Version 1 öffnen möchten, gibt VuGen einen Fehler aus.

```
;WSRData 2 1
```

Jedem Eintrag wird eine Kennung vorangestellt, die angibt, ob die Daten empfangen oder gesendet wurden. Darauf folgen der Pufferdeskriptor und die Anzahl der empfangenen Bytes (nur für **lrs_ receive**-Funktionen). Der Pufferdeskriptor enthält eine Nummer zur Kennzeichnung des Puffers.

Beispiel:

```
recv buf5   25
```

Gibt an, dass der Puffer empfangene Daten enthält. Die Nummer des Eintrags lautet 5. Damit wird angegeben, dass es sich bei diesem Empfangsvorgang um die sechste Datenübertagung handelt (da der Index bei null beginnt). Die empfangene Datenmenge beträgt 25 Byte.

Haben Ihre Daten das ASCII-Format, werden dem Deskriptor die eigentlichen ASCII-Daten nachgestellt, die durch die Sockets übertragen wurden.

Wenn Ihre Daten das EBCDIC-Format haben, müssen Sie von einer Suchtabelle übersetzt werden. Weitere Informationen zum Festlegen der Übersetzungstabelle finden Sie unter "Knoten "WinSock"" auf Seite 248. EBCDIC-Werte, deren ASCII-Werte nach der Übersetzung druckbar sind, werden als ASCII-Zeichen dargestellt. Wenn allerdings der ASCII-Wert einem nicht druckbaren Zeichen entspricht, zeigt VuGen den ursprünglichen EBCDIC-Wert an.

```
recv buf6 39
    "\xff\xfb\x01\xff\xfb\x03\xff\xfd\x01"
    "\r\n"
    "SunOS UNIX (sunny)\r\n"
```

Im Folgenden werden die Kopfzeile, die Deskriptoren und die Daten einer typischen Datendatei dargestellt:

```
;WSRData 2 1
send buf0 48
    "\xff\xfd\x01\xff\xfd\x03\xff\xfb\x03\xff\xfb\x18"
recv buf1 15
    "\xff\xfd\x18\xff\xfd\x1f\xff\xfd"
    "#"
    "\xff\xfd"
    "'"
    "\xff\xfd"
    "$"
send buf2 24
    "\xff\xfb\x18"
```

Dialogfeld "Gehe zu Offset"

Über dieses Dialogfeld können Sie zu einer bestimmten Position in den aufgezeichneten Daten wechseln.

Zugriff	Klicken Sie in der Symbolleiste des Snapshot-Bereichs auf **Gehe zu**.
Relevante Aufgaben	"Anzeigen und Ändern von Windows-Socket-Puffern" auf Seite 1058

Nachfolgend werden die Elemente der Oberfläche beschrieben:

Element der Oberfläche	Beschreibung
Aktueller Offset	Der aktuelle Offset des Cursors (schreibgeschützt).
Gehe zu Offset	Wechselt zu einem bestimmten, absoluten Offset in den Daten.
Bewegen um...Byte	Springt um die angegebene Anzahl an Byte zu einer Position relativ zur Cursorposition. Positive Werte geben eine Vorwärtsbewegung an. Negative Werte geben eine Rückwärtsbewegung an.
Bereich auswählen von...bis...	Wählt einen Datenbereich im Puffer aus.
Übernehmen	Verschiebt den Cursor zum angegebenen Offset.

MMS-Vuser-Skripts (Multimedia Messaging Service) – Übersicht

Der MMS-Protokoll ist nützlich zum Austausch von MMS-Nachrichten zwischen mobilen Geräten.

MMS (Multimedia Messaging Service) ist eine Erweiterung des SMS-Protokolls. Während SMS-Meldungen nur Text enthalten können, ermöglicht MMS Ihnen, Meldungen mit ganz unterschiedlichem Inhalt an MMS-fähige Handgeräte zu senden und auf diesen zu empfangen. Bei diesem Inhalt kann es sich um Text, Klänge, E-Mails, Bilder, Videoclips und sogar Streamingdaten handeln. Es ist ebenfalls möglich, Multimedianachrichten von einem Mobiltelefon an eine E-Mail-Adresse zu senden.

Eine MMS-Meldung enthält typischerweise eine Sammlung von Anhängen. Während SMS-Meldungen auf 160 Byte begrenzt sind, kann eine MMS-Meldung mehrere MB groß sein. MMS erfordert in der Regel ein 3G-Netzwerk, damit große Meldungen zugestellt werden können.

Um eine MMS-Meldung zu empfangen, erhält ein Mobiltelefon eine MMS-Benachrichtigung über SMS. Die SMS-Meldung kann über verschiedene SMS-Protokolle wie z. B. SMPP, UCP und CIMD2 empfangen werden. Die SMS-Meldung enthält einen eindeutigen Pfad zu der MMS-Meldung, die in der Datenbank des MMSC-Servers gespeichert wird, und das Mobiltelefon verwendet diesen Pfad, um die Meldung vom SMSC herunterzuladen. Die aktuelle Version von VuGen unterstützt den Empfang von MMS-Benachrichtigungen über die SMPP-Schnittstelle.

MMS-Vuser-Skripts unterstützen die Versionen 1.0 und 1.1 des MMS-Protokolls gemäß Definition der OMA (Open Mobile Alliance). Unter Verwendung von MMS-Vusern können Sie MMS-Meldungen an den MMSC-Server direkt über das HTTP-Protokoll oder über das WAP-Protokoll durch ein WAP-Gateway senden.

MMS-Funktionen (Multimedia Messaging Service) emulieren das Senden und Empfangen von MMS-Meldungen. Alle Funktionen beginnen mit dem Präfix **mm**. Ausführliche Informationen zur Syntax dieser Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Ausführen eines MMS-Szenarios im Controller

Ein MMS-Szenario (Multimedia Messaging Service) erfordert eine Befehlszeileneinstellung.

So legen Sie die MMS-Befehlszeileneinstellung fest:

1. Klicken Sie im Bildschirm **Szenariozeitplan** auf **Details**. Das Dialogfeld **Gruppeninformationen** wird angezeigt.
2. Wenn das Feld **Befehlszeile** nicht angezeigt wird, klicken Sie auf die Schaltfläche **Mehr**.
3. Fügen Sie am Ende des Befehlszeilentexts Folgendes hinzu: `-usingwininet yes`
4. Klicken Sie auf **OK**, um die Befehlszeilenoption zu übernehmen.

Weiterführende Themen

Manuelles Programmieren eines Skripts mithilfe des VuGen-Editors

Übersicht über das manuelle Programmieren von Skripts

VuGen ermöglicht Ihnen neben dem Aufzeichnen einer Sitzung auch das Programmieren eigener Funktionen im Skript. Sie können hierzu die Vuser-API oder die Standardfunktionen für die Programmierung verwenden. Mit den Vuser-API-Funktionen können Sie Informationen über Vuser erfassen. Beispielsweise können Sie Vuser-Funktionen verwenden, um die Serverleistung zu messen, die Serverlast zu kontrollieren, einen Debugging-Code hinzuzufügen oder Laufzeitinformationen über die Vuser, die am Test oder an der Überwachung teilnehmen, abzurufen.

In diesem Kapitel wird die Programmierung eines Vuser-Skripts mit dem VuGen-Editor unter Einbeziehung der Bibliotheken oder Klassen Ihrer Anwendung beschrieben.

Sie können ein Vuser-Skript auch in Visual C und Visual Basic programmieren. In diesen Umgebungen entwickeln Sie Ihr Vuser-Skript innerhalb Ihrer Entwicklungsanwendung und importieren die Funktionsbibliotheken der Vuser-API. Weitere Informationen finden Sie unter "Erstellen von Skripts in externen IDEs" auf Seite 1068.

Erstellen Sie für die Programmierung eines benutzerdefinierten Skripts zunächst ein Skeleton-Skript. Dieses setzt sich aus den drei Hauptabschnitten eines Skripts zusammen: **init**, **actions** und **end**. Diese Abschnitte sind leer und werden manuell mit Funktionen gefüllt.

Sie können leere Skripts für die Programmiersprachen C und Java erstellen.

C-Vuser-Skripts

In C-Vuser-Skripts können Sie beliebigen C-Code eingeben, der den ANSI-Standardkonventionen entspricht. Wählen Sie zum Erstellen eines leeren C-Vuser-Skripts im Dialogfeld "Neues Skript erstellen" den Eintrag **C Vuser** aus. VuGen erstellt ein leeres C-Vuser-Skript:

```
Action1()
{
    return 0;
}
```

Sie können C-Vuser-Funktionen in allen Vuser-Skripttypen einsetzen, die C-Funktionen verwenden.

Eine C-Referenz mit Informationen zur Syntax und Beispielen von häufig verwendeten C-Funktionen finden Sie in der Function Reference (**Hilfe > Funktionsreferenz**).

Richtlinien für die Verwendung von C-Funktionen

Für C-Vuser-Skripts gelten alle ANSI-C-Standardkonventionen, einschließlich Kontrollstrukturen und Syntax. Sie können dem Skript wie in anderen C-Programmen Kommentare und bedingte Anweisungen hinzufügen. Mithilfe von ANSI-C-Konventionen deklarieren und definieren Sie Variablen.

Der C-Interpreter für die Ausführung von Vuser-Skripts erkennt die ANSI-C-Standardsprache. Er unterstützt jedoch nicht die Microsoft-Erweiterungen zu ANSI C.

Beachten Sie folgende Einschränkungen, bevor Sie C-Funktionen zu einem Vuser-Skript hinzufügen:

- Ein Vuser-Skript kann die Adresse einer seiner Funktionen nicht als Callback an eine Bibliotheksfunktion übergeben.
- Die Funktionen **stdargs**, **longjmp** und **alloca** werden in Vuser-Skripts nicht unterstützt.
- Vuser-Skripts unterstützen keine Strukturargumente und Rückgabetypen. Zeiger auf Strukturen sind hingegen zulässig.
- Literale Zeichenfolgen in Vuser-Skripts sind schreibgeschützt. Bei dem Versuch, in eine literale Zeichenfolge zu schreiben, wird eine Zugriffsverletzung generiert.
- C-Funktionen, die keine Ganzzahlen zurückgeben, müssen in CAST-Funktionen umgewandelt werden. Beispiel:
  ```
  extern char * strtok();
  ```

Aufrufen von libc-Funktionen

Sie können in einem Vuser-Skript **libc**-Funktionen aufrufen. Wenn der für die Ausführung von Vuser-Skripts verwendete Interpreter allerdings keine Microsoft-Erweiterungen zu ANSI C unterstützt, können Sie die Include-Dateien von Microsoft nicht verwenden. Entweder Sie schreiben eigene Prototypen oder Sie wenden sich an den HP Customer Support und fordern ANSI-kompatible Include-Dateien an, die Prototypen für **libc**-Funktionen enthalten.

Verknüpfungsmodus

Der für die Ausführung von Vuser-Skripts eingesetzte C-Interpreter arbeitet mit einem "verzögerten" Verknüpfungsmodus, d. h. dass eine Funktion noch nicht beim Start der Skriptausführung, wohl aber kurz vor ihrer Verwendung definiert werden muss. Beispiel:

```
lr_load_dll("mydll.dll");
    myfun();  /* defined in mydll.dll -- can be called directly,
                 immediately after myfun.dll is loaded. */
```

Java-Vuser

In Java-Vuser-Skripts können Sie beliebigen Java-Standardcode eingeben. Wählen Sie zum Erstellen eines leeren Java-Vuser-Skripts im Dialogfeld "Neuer virtueller Benutzer" den Eintrag **Java-Vuser** aus. VuGen erstellt ein leeres Java-Skript:

```
import lrapi.lr;
public class Actions
{
    public int init() {
        return 0;
    }
    public int action() {
        return 0;
    }
    public int end() {
        return 0;
    }
}
```

Hinweis: Bei Java-Vusern kann lediglich die **Actions**-Klasse bearbeitet werden. Die Actions-Klasse verfügt über drei Methoden: **init**, **action** und **end**. Geben Sie Initialisierungscode in der **init**-Methode, Geschäftsprozesse in der **action**-Methode und Bereinigungscode in der **end**-Methode ein.

.NET-Vuser

Sie können ein leeres .NET-Vuser-Skript für den .NET-Code anlegen. Mit diesem Skripttyp können Sie Ihre vorhandene .NET-Anwendung in VuGen integrieren. Wählen Sie zum Erstellen eines leeren .NET-Vuser-Skripts im Dialogfeld "Neues Skript erstellen" den Eintrag **.NET** aus.

In einem .NET-Vuser-Skript ist die Standardsprache C#. Wurde das Skript aus einer aufgezeichneten Sitzung generiert, können Sie die Skriptsprache C# in VB.NET ändern, indem Sie **Sprache Visual Basic .NET** unter **Aufzeichnen > Aufzeichnungsoptionen > Allgemein > Skript** auswählen und das Skript neu generieren.

Tipp: Klicken Sie auf die Schaltfläche , um das Skript in Visual Studio zu bearbeiten.

Das folgende Beispiel zeigt den Abschnitt "Action" eines leeren .NET-Skripts:

```
namespace Script
 {
  public partial class VuserClass
  {
    public int Action()
  {
  // Add your code here
```

```
    return 0;
   }
  }
 }
```

Hinweis: Geben Sie den Geschäftsprozesscode in die Methode **Action** ein. Fügen Sie Initialisierungscode zur Methode **vuser_init** hinzu und bereinigen Sie den Code für die Methode **vuser_end**.

Erstellen von Skripts in externen IDEs

Erstellen von Vuser-Skripts oder LoadRunner-Tests in Visual Studio oder Eclipse

In diesem Kapitel wird beschrieben, wie ein Vuser-Skript oder ein LoadRunner-Test durch Programmierung in den Umgebungen Visual Studio oder Eclipse entwickelt wird.

LoadRunner enthält Add-Ins, über die Sie mit Visual Studio oder Eclipse arbeiten können.

Es gibt zwei Typen von Add-Ins:

- Visual Studio-IDE-Add-In
- Visual Studio-/Eclipse-IDE-Add-In für Entwickler

Mit dem einfachen **Visual Studio-IDE-Add-In** können Sie ein Vuser-Skript in der Visual Studio-Umgebung erstellen. Sie programmieren das Skript mithilfe von standardmäßigen LoadRunner- oder C#-Funktionen in Ihrer systemeigenen Umgebung. Das Skript können Sie dann einem LoadRunner-Szenario hinzufügen.

Mit dem **Visual Studio-IDE-Add-In für Entwickler** können Sie einen LoadRunner-Test direkt in Visual Studio oder Eclipse erstellen. Sie programmieren das Skript mithilfe von standardmäßigen LoadRunner-, C#- oder Java-Funktionen in Ihrer systemeigenen Umgebung. Wenn Sie den Test abgeschlossen haben, können Sie Laufzeiteinstellungen hinzufügen und den Test wiedergeben. Der Test wird in einem NUnit (Visual Studio)- oder JUnit (Eclipse)-Test gespeichert. Diese Tests können Sie direkt vom LoadRunner-Controller aus als System- oder Einheitentests einem Szenario hinzufügen.

Diese Add-Ins sind im DVD-Hauptordner des LoadRunner-Produkts unter dem Ordner **Additional Components** verfügbar. Stellen Sie sicher, dass Sie das richtige Visual Studio-Add-In für Ihre Version von Visual Studio auswählen.

Sobald Sie das einfache **Visual Studio-IDE-Add-In** installiert haben, können Sie in Visual Studio ein neues LoadRunner-Skript erstellen.

Alternativ können Sie das Skript von VuGen aus an Visual Studio oder Eclipse senden. Wenn Sie beispielsweise ein Skript in VuGen erstellt haben und feststellen, dass Sie die Funktionen Ihrer systemeigenen Umgebung benötigen, können Sie das Skript über die Schaltflächen **In Visual Studio öffnen** oder **In Eclipse öffnen** in der entsprechenden Anwendung öffnen. Damit dies möglich ist, müssen Sie zunächst das einfache **Visual Studio-IDE-Add-In** installieren. (Für Eclipse installiert LoadRunner das Add-In intern beim ersten Auswählen von **In Eclipse öffnen**). Weitere Informationen finden Sie unter "Debuggen von .NET-Vuser-Skripts" auf Seite 795 oder "Öffnen von Java-Vuser-Skripts in Eclipse" auf Seite 733.

Beim Arbeiten in Visual Studio oder Eclipse ist die gesamte LoadRunner-API über den Objektbrowser verfügbar. Informationen zu allen LoadRunner-API-Funktionen zum Programmieren von Skripts finden Sie unter Function Reference (**Hilfe > Funktionsreferenz**).

Erstellen eines Vuser-Skripts in Visual Studio

Mithilfe der Add-Ins von LoadRunner können Sie ein Vuser-Skript in Visual Studio in VB, C++ oder C# erstellen.

So erstellen Sie ein Vuser-Skript in Visual Studio:

1. Installieren Sie das IDE-Add-In für Ihre Version von Microsoft Visual Studio vom Ordner Download-/DVD-Ordner **Additional Components** aus. Beispiel: Additional Components\IDE Add-Ins\MS Visual Studio .NET\LRVS2010IDEAddInSetup.exe.

2. Wählen Sie in Visual Studio die entsprechende Vorlage aus den installierten Vorlagen (**LoadRunner VB|C++|C# .NET Vuser**) aus. Visual Studio erstellt ein neues Projekt mit einer Klasse und einer Vorlage für einen Vuser sowie die Skriptdatei (**<name>.usr**). Die Vorlage enthält drei Abschnitte: **Initialisieren**, **Aktionen** und **Beenden**.

   Das folgende Beispiel zeigt eine Visual C#-Vorlage:

```
public int Initialize()
      {
          // TO DO: Add virtual user's initialization routines
            return lr.PASS;
      }
      public int Actions()
      {
          // TO DO: Add virtual user's business process actions
          return lr.PASS;
      }
      public int Terminate()
      {
          // TO DO: Add virtual user's termination routines
          return lr.PASS;
      }
```

3. Fügen Sie der Vorlage in den TODO-Abschnitten Code hinzu.

4. Öffnen Sie den Objektbrowser (Menü **View**). Erweitern Sie den LoadRunner-Knoten (z. B. **Interop.LoadRunner**), um die LoadRunner-Elemente anzuzeigen. Fügen Sie dem Skript die gewünschten Elemente hinzu, z. B. Transaktionen, Rendezvous-Punkte und Meldungen.

5. Erweitern Sie das Symbolleistenmenü (**Vuser**), und fügen Sie dem Skript Laufzeiteinstellungen und Parameter hinzu. Weitere Informationen finden Sie unter "Allgemein > Ausführungslogik (Knoten)" auf Seite 368 oder "Dialogfeld "Parameterliste"" auf Seite 470.

6. Verwenden Sie das Menü "Vuser", um das Skript wiederzugeben und seine Funktionalität zu testen.

7. Wählen Sie **Vuser > Lastszenario erstellen** aus, um ein LoadRunner-Szenario mit dieser ".usr"-Datei zu verwenden.

8. Sie können das LoadRunner-Projekt auch als DLL-Datei erstellen, die im selben Ordner wie das Projekt gespeichert wird. Auf diese DLL können Sie direkt aus einem LoadRunner-Szenario verweisen.

Erstellen eines Vuser-Skripts in Eclipse

Mithilfe der Add-Ins von LoadRunner können Sie Java-Vuser-Skript in Eclipse erstellen. Sie können in VuGen beginnen und dann Eclipse öffnen oder Sie arbeiten von vorneherein in Eclipse.

Voraussetzung

1. Stellen Sie sicher, dass auf Ihrem Computer JDK 1.7 (JRE 7) installiert ist. Besuchen Sie **java.com**, um Ihre Version zu überprüfen oder die erforderliche Version herunterzuladen. Öffnen Sie Eclipse nach der Installation, und wählen Sie **Window > Preferences** aus. Navigieren Sie zum Knoten **Java > Installed JREs**. Wenn **jre7** in der Liste **Installed JREs** nicht aufgeführt ist, klicken Sie auf **Add**, und verwenden Sie den Assistenten, um den Ordner hinzuzufügen (z. B. "c:\Programme\Java\jre7"). Aktivieren Sie in der Liste "Installed JREs" auf das Kontrollkästchen für **jre7**, um Eclipse zur Verwendung dieser Version anzuweisen. Schließen Sie Eclipse.

Möglichkeit 1: Erstellen eines Skripts in VuGen und Weiterentwicklung in Eclipse

1. Öffnen Sie VuGen und erstellen Sie ein neues Java-Sprachskript.

2. Zeichnen Sie Schritte auf oder fügen Sie sie hinzufügen wie gewohnt hinzu.

3. Klicken Sie auf der Symbolleiste auf die Schaltfläche "In Eclipse öffnen" . Wenn Sie Eclipse zum ersten Mal aus VuGen heraus öffnen, werden sie in einer Meldung dazu aufgefordert, den Speicherort von Eclipse Standort anzugeben. Eclipse wird mit dem aktuellen Skript im Bereich **Package Explorer** geöffnet.

Möglichkeit 2: Erstellen des Skripts in Eclipse

1. Kopieren Sie die Datei **hp.lr.vugeneclipse42addin.jar** von der DVD aus dem Verzeichnis **Additional Components\IDE Add-Ins \EclipseAddin** in den Eclipse-Ordner **dropins**. Extrahieren Sie die Dateien aus der jar-Datei. (Alternativ können Sie Eclipse mindestens einmal innerhalb von VuGen öffnen. Anschließend können Sie weitere Java-Skripts direkt in Eclipse erstellen, ohne das Add-In manuell hineinkopieren zu müssen. Dazu folgen Sie den Anweisungen in Option 1).

2. Öffnen Sie Eclipse. Wählen Sie **Datei > Neu > Projekt** aus, und erweitern Sie den Knoten **LoadRunner-Skripts**. Sie können beliebige Java-Protokollskripts erstellen: Java-Vuser, Java Record Replay oder Java over HTTP.

Entwicklung des Skripts in Eclipse

1. Erweitern Sie den Knoten des Skripts, und wählen Sie den Knoten **default package > Action.java** aus. Codieren Sie das Skript wie aus dem Eclipse-Editor gewohnt in den entsprechenden Abschnitten.

2. Erweitern Sie im Skriptknoten im Package Explorer **Referenced Libraries > classes > lrapi >**, um auf die gewünschten LoadRunner-Elemente zuzugreifen, z. B. Transaktionen, Rendezvous-Punkte und Meldungen.

3. Erweitern Sie das Menü **Vuser** (dazu müssen Sie ggf. den Namen des übergeordneten Skripts im Package Explorer auswählen), und erweitern Sie das Skript mit Laufzeiteinstellungen und Parametern. Weitere Informationen finden Sie unter "Allgemein > Ausführungslogik (Knoten)" auf Seite 368 oder "Dialogfeld "Parameterliste"" auf Seite 470.

4. Speichern Sie das Szenario und führe Sie es aus. Wählen Sie **Vuser > Vuser ausführen** aus, um das Skript zu testen. Wählen Sie dann **Vuser > Lastszenario erstellen** aus, um es vom Controller aus auszuführen. Beachten Sie, dass Sie das Skript nicht mehr in VuGen öffnen können, wenn Sie es in Eclipse bearbeiten.

Entwickeln eines Einheitentests mit Visual Studio (NUnit-Test)

Mithilfe der Entwicklungs-Add-Ins von LoadRunner können Sie einen NUnit-Test in Visual Studio in VB, C++ oder C# erstellen.

So erstellen Sie ein NUnit-Test in Visual Studio:

1. Installieren Sie das IDE-Add-In für Microsoft Visual Studio vom Ordner Download-/DVD-Ordner **Additional Components** aus. Beispiel: DVD\Additional Components\IDE Add-Ins Dev\MS Visual Studio .NET\LRVS2010IDEAddInDevSetup.exe.

2. Wählen Sie in Visual Studio die entsprechende Vorlage aus den installierten Vorlagen

(**LoadRunner VB|C++|C# .NET Vuser**) aus. Visual Studio erstellt ein neues Projekt mit einer Klasse und einer Vorlage für einen Vuser sowie die Skriptdatei (**<name>.usr**). Die Vorlage enthält drei Abschnitte: **Initialisieren**, **Aktionen** und **Beenden**.

Das folgende Beispiel zeigt eine Visual C#-Vorlage:

```
public int Initialize()
      {
          // TO DO: Add virtual user's initialization routines
            return lr.PASS;
      }
      public int Actions()
      {
          // TO DO: Add virtual user's business process actions
          return lr.PASS;
      }
      public int Terminate()
      {
          // TO DO: Add virtual user's termination routines
          return lr.PASS;
      }
```

3. Fügen Sie der Vorlage in den TODO-Abschnitten Code hinzu.
4. Öffnen Sie den Objektbrowser (Menü **View**). Erweitern Sie den LoadRunner-Knoten (z. B. **Interop.LoadRunner**), um die LoadRunner-Elemente anzuzeigen. Fügen Sie dem Skript die gewünschten Elemente hinzu, z. B. Transaktionen, Rendezvous-Punkte und Meldungen.
5. Erweitern Sie das Symbolleistenmenü, Vuser, und fügen Sie dem Skript Laufzeiteinstellungen und Parameter hinzu. Weitere Informationen finden Sie unter "Allgemein > Ausführungslogik (Knoten)" auf Seite 368 oder "Dialogfeld "Parameterliste"" auf Seite 470.
6. Führen Sie den Test aus, um seine Funktion zu prüfen.
7. Erstellen Sie das LoadRunner-Projekt auch als DLL-Datei, die im selben Ordner wie das Projekt gespeichert wird.
8. Fügen Sie die DLL-Datei als Komponententest in ein LoadRunner-Szenario ein.

Entwickeln eines Einheitentests mit Eclipse (JUnit- oder Selenium-Test)

Mithilfe der Entwicklungs-Add-Ins von LoadRunner können Sie einen JUnit-Test in Eclipse erstellen.

So erstellen Sie einen Einheitentest in Eclipse:

1. Stellen Sie sicher, dass auf Ihrem Computer JDK 1.7 (JRE 7) installiert ist. Besuchen Sie **oracle.com**, um Ihre Version zu überprüfen oder die erforderliche Version herunterzuladen. Öffnen Sie Eclipse nach der Installation, und wählen Sie **Window > Preferences** aus. Navigieren Sie zum Knoten **Java > Installed JREs**. Wenn **jre7** in der Liste **Installed JREs** nicht aufgeführt ist, klicken Sie auf **Add**, und verwenden Sie den Assistenten, um den Ordner hinzuzufügen (z. B. "c:\Programme\Java\jre7"). Aktivieren Sie in der Liste "Installed JREs" auf das Kontrollkästchen für **jre7**, um Eclipse zur Verwendung dieser Version anzuweisen.

2. Führen Sie das Entwicklungs-Add-in für Eclipse, LREclipseIDEAddInDevSetup.exe, aus dem Eclipse-Ordner "download/DVD" heraus aus: **Additional Components\IDE Add-Ins Dev**. Führen Sie nach der Installation des Eclipse-Add-ins den folgenden Befehl über die Befehlszeile aus, um den Plugin-Cache neu aufzubauen: **Eclipse.exe -clean**.

3. Öffnen Sie in Eclipse den Selenium- oder JUnit-Test.

4. Codieren Sie den Test, wie Sie es normalerweise in Eclipse tun würden.

5. Erstellen Sie Ihre Java-Klassen.

6. Wählen Sie **DevOps-Vuser > LoadRunner-API-Referenz hinzufügen**, um die gewünschten LoadRunner-Funktionen neben Transaktionen, Rendezvous-Punkten und Nachrichten zum Skript hinzuzufügen.

7. Erweitern Sie das Menü **DevOps-Vuser** und fügen Sie dem Skript Laufzeiteinstellungen und Parameter hinzu. Weitere Informationen finden Sie unter "Allgemein > Ausführungslogik (Knoten)" auf Seite 368 oder "Dialogfeld "Parameterliste"" auf Seite 470.

8. Wählen Sie **DevOps-Vuser > Vuser ausführen**, um den Test in Eclipse auszuführen und seine Funktion zu überprüfen.

9. Starten Sie LoadRunner Controller über das Menü **DevOps-Vuser** oder fügen Sie den Test zu einem bereits geöffneten Controller-Szenario hinzu.

10. Fügen Sie die Klassendatei zu einem beliebigen Zeitpunkt als Komponententest in ein LoadRunner-Szenario ein.

Verwenden von DLLs und Anpassen von VuGen

Aufrufen von Funktionen in externen DLLs

Sie können Funktionen aufrufen, die in externen DLLs definiert sind. Durch Aufrufen von externen Funktionen über Ihr Skript können Sie die Speichernutzung Ihres Skripts und die gesamte Laufzeit reduzieren.

Zum Aufrufen der externen Funktion laden Sie die DLL, in der die Funktion definiert ist.

Sie können eine DLL mithilfe einer der folgenden Methoden laden:

- Lokal - für ein Skript mit der Funktion **lr_load_dll**. Weitere Informationen über die Aufgabe finden Sie unter "Lokales Laden einer DLL" unten.
- Global - für alle Skripts, durch Hinzufügen von Anweisungen zur Datei **vugen.dat**. Weitere Informationen über die Aufgabe finden Sie unter "Globales Laden einer DLL" auf der nächsten Seite.

Lokales Laden einer DLL

Im Rahmen dieser Aufgabe wird beschrieben, wie die Funktion **lr_load_dll** zum Laden einer DLL in ein Vuser-Skript verwendet wird. Nachdem die DLL geladen ist, können Sie jede in der DLL definierte Funktion aufrufen, ohne sie in Ihrem Skript deklarieren zu müssen.

Aufrufen einer in einer DLL definierten Funktion

1. Fügen Sie in einem C-Vuser-Skript die Funktion **lr_load_dll** hinzu, um die DLL am Anfang Ihres Skripts zu laden. Fügen Sie die Anweisung am Anfang des Abschnitts *vuser_init* ein. **lr_load_dll** ersetzt die Funktion **ci_load_dll**.

   Verwenden Sie die folgende Syntax:

   ```
   lr_load_dll(Bibliotheksname);
   ```

   Beachten Sie, dass DLLs für Linux-Plattformen als Shared Libraries (gemeinsame Bibliotheken) bekannt sind. Die Erweiterung der Bibliotheken hängt von der Plattform ab.

2. Rufen Sie die in der DLL definierte Funktion an der entsprechenden Stelle in Ihrem Skript auf.

   Im folgenden Beispiel wird die in **orac1.dll** definierte Funktion **insert_vals** nach Erstellen der Tabelle **Test_1** aufgerufen.

   ```
   int LR_FUNC Actions(LR_PARAM p)
   {
   lr_load_dll("orac1.dll");

   lrd_stmt(Csr1, "create table Test_1 (name char(15), id integer)\n", -1,
             1 /*Deferred*/, 1 /*Dflt Ora Ver*/, 0);
   lrd_exec(Csr1, 0, 0, 0, 0, 0);
   /* Call the insert_vals function to insert values into the table. */
   insert_vals();
   lrd_stmt(Csr1, "select * from Test_1\n", -1, 1 /*Deferred*/, 1 /*Dflt Ora Ve
   r*/, 0);
   lrd_bind_col(Csr1, 1, =;NAME_D11, 0, 0);
   lrd_bind_col(Csr1, 2, =;ID_D12, 0, 0);
   lrd_exec(Csr1, 0, 0, 0, 0, 0);
   lrd_fetch(Csr1, -4, 15, 0, PrintRow14, 0);
   ...
   ```

Hinweis: Sie können einen vollständigen Pfad für die DLL angeben. Wenn Sie keinen Pfad angeben, sucht **lr_load_library** mit der von der C++-Funktion verwendeten Standardsequenz nach der DLL (**LoadLibrary** auf Windows-Plattformen). Auf Linux-Plattformen können Sie die Umgebungsvariable **LD_LIBRARY_PATH** (oder die der Plattform entsprechende Variable) festlegen. Die Funktion **lr_load_dll** verwendet dieselben Suchregeln wie **dlopen**. Weitere Informationen finden Sie auf den Hauptseiten für **dlopen** oder ihre Entsprechung.

Globales Laden einer DLL

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie eine DLL global laden, um ihre Funktionen für Ihre gesamten Vuser-Skripts verfügbar zu machen. Nachdem die DLL geladen ist, können Sie jede in der DLL definierte Funktion aufrufen, ohne sie in Ihrem Skript deklarieren zu müssen.

Globales Laden einer DLL

1. Fügen Sie eine Liste der zu ladenden DLLs zum entsprechenden Abschnitt der Datei *mdrv.dat* hinzu, die sich im Ordner *dat* Ihrer Anwendung befindet.

   Verwenden Sie die folgende Syntax:

   *PLATFORM*_DLLS=*my_dll1*.dll, *my_dll2*.dll, ...

   und fügen Sie anstelle von *PLATTFORM* Ihre spezifische Plattform ein. Eine Liste der Plattformen finden Sie im ersten Abschnitt der Datei *mdrv.dat*.

   Um beispielsweise DLLs für Winsock-Vuser auf eine NT-Plattform zu laden, fügen Sie die folgende Anweisung zur Datei *mdrv.dat* hinzu:

   ```
   [WinSock]
   ExtPriorityType=protocol
   WINNT_EXT_LIBS=wsrun32.dll
   WIN95_EXT_LIBS=wsrun32.dll
   LINUX_EXT_LIBS=liblrs.so
   SOLARIS_EXT_LIBS=liblrs.so
   HPUX_EXT_LIBS=liblrs.sl
   AIX_EXT_LIBS=liblrs.so
   LibCfgFunc=winsock_exten_conf
   UtilityExt=lrun_api
   ExtMessageQueue=0
   ExtCmdLineOverwrite=-WinInet No
   ExtCmdLineConc=-UsingWinInet No
    WINNT_DLLS=user_dll1 .dll, user_dll2 .dll, ...
   ```

2. Rufen Sie die in der DLL definierte Funktion an der entsprechenden Stelle in Ihrem Skript auf.

Aufzeichnen von OLE-Servern

VuGen unterstützt das Aufzeichnen für OLE-Anwendungen zurzeit nicht. Dies sind Anwendungen, bei denen der tatsächliche Prozess nicht von den Standardprozess-Erstellungsroutinen gestartet wird, sondern vom OLE-Automatisierungssystem. Sie können jedoch ein Vuser-Skript für OLE-Anwendungen auf der Basis der folgenden Richtlinien erstellen.

Es gibt zwei Typen von OLE-Servern: ausführbare Server und DLL-Server.

DLL-Server

Wenn der Server der DLL-Server ist, wird er möglicherweise in den Anwendungsprozessbereich geladen und VuGen zeichnet den Aufruf zur LoadLibrary auf. In diesem Fall bemerken Sie möglicherweise nicht, dass es sich um eine OLE-Anwendung handelte.

Ausführbare Server

Wenn der Server der ausführbare Server ist, müssen Sie die ausführbare Datei in VuGen mit einer besonderen Methode aufrufen:

- Bestimmen Sie zunächst, welcher Prozess tatsächlich aufgezeichnet werden muss. In den meisten Fällen kennt der Kunde den Namen der ausführbaren Datei der Anwendung. Wenn der Kunde den Namen der Anwendung nicht kennt, rufen Sie sie auf und ermitteln Sie ihren Namen über den **NT Task-Manager**.

- Nachdem Sie den erforderlichen Prozess identifiziert haben, klicken Sie auf **Aufzeichnen** in VuGen. Wenn Sie aufgefordert werden, den Anwendungsnamen einzugeben, geben Sie die OLE-Anwendung gefolgt von dem Kennzeichen "/Automation" ein. Starten Sie anschließend den Benutzerprozess auf die übliche Weise (nicht über VuGen). VuGen zeichnet den laufenden OLE-Server auf und ruft keine andere Kopie davon auf. In den meisten Fällen sind diese Schritte ausreichend, um es VuGen zu ermöglichen, die Aktionen eines OLE-Servers aufzuzeichnen.

- Wenn Sie immer noch Schwierigkeiten mit dem Aufzeichnen haben, können Sie das Programm CmdLine verwenden, um die vollständige Befehlszeile eines nicht direkt gestarteten Prozesses zu ermitteln. (Das Programm ist in einem Knowledgebase-Artikel über die Customer Support-Website http://support.hp.com verfügbar.)

Verwenden von CmdLine

Im folgenden Beispiel wird CmdLine.exe verwendet, um die vollständige Befehlszeile für den Prozess *MyOleSrv.exe* zu bestimmen, der von einem anderen Prozess gestartet wird.

Bestimmen der zugehörigen vollständigen Befehlszeile

1. Benennen Sie *MyOleSrv.exe* in *MyOleSrv.orig.exe* um.

2. Stellen Sie die Datei *CmdLine.exe* in denselben Ordner wie die Anwendung und benennen Sie sie in *MyOleSrv.exe* um.

3. Starten Sie die Datei *MyOleSrv.exe*. Es wird ein Popup-Fenster mit einer Meldung angezeigt,

die die vollständige Befehlszeile der ursprünglichen Anwendung enthält (einschließlich zusätzlicher Informationen) und die Informationen werden in die Datei *c:\temp\CmdLine.txt* geschrieben.

4. Stellen Sie die alten Namen wieder her und starten Sie den OLE-Server *MyOleSrv.exe* mit den korrekten Befehlszeilenparametern über VuGen. Starten Sie die Benutzeranwendung mit der üblichen Methode - nicht über VuGen. In den meisten Fällen führt VuGen die Aufzeichnung korrekt durch.

Wenn Sie immer noch Schwierigkeiten beim Aufzeichnen haben, fahren mit den folgenden Schritten fort:

1. Benennen Sie den OLE-Server in *MyOleSrv.1.exe* und *CmdLine* in *MyOleSrv.exe* um.
2. Setzen Sie die Umgebungsvariablen *CmdStartNotepad* und *CmdNoPopup* auf *1*. Eine Liste der CmdLine-Umgebungsvariablen finden Sie unter ""CmdLine"-Umgebungsvariablen" unten.
3. Starten Sie die Anwendung (nicht über VuGen). Notepad wird mit der vollständigen Befehlszeile geöffnet. Prüfen Sie die Befehlszeilenargumente. Starten Sie die Anwendung mehrmals und vergleichen Sie die Befehlszeilenargumente. Wenn die Argumente bei jedem Aufrufen der Anwendung dieselben sind, können Sie die Umgebungsvariable *CmdStartNotepad* zurücksetzen. Lassen Sie sie andernfalls auf "1".
4. Rufen Sie in VuGen das Programm MyOleSrv.1.exe mit den Befehlszeilenparametern auf (verwenden Sie die Befehle zum Kopieren und Einfügen im Notepad-Fenster).
5. Starten Sie die Anwendung (nicht von VuGen aus).

"CmdLine"-Umgebungsvariablen

Sie können die Ausführung von *CmdLine* über die folgenden Umgebungsvariablen kontrollieren:

CmdNoPopup	Ist diese Variable festgelegt, wird das Popup-Fenster nicht angezeigt.
CmdOutFileName	Ist diese Variable festgelegt und nicht leer, versucht *CmdLine* diese Datei anstelle von *c:\temp\CmdLine.txt* zu erstellen.
CmdStartNotepad	Ist diese Variable festgelegt, wird die Ausgabedatei in Notepad angezeigt (optimale Verwendung mit *CmdNoPopup*).

Erstellen eines neuen Vuser-Typs

Mithilfe der folgenden Schritte können Sie einen neuen Vuser-Typ erstellen.

1. **Bearbeiten der Datei "mdrv.dat"**

   Bearbeiten Sie die Datei *mdrv.dat*, die sich im Ordner *M_LROOT\dat* befindet. Sie fügen einen Abschnitt für den neuen Vuser-Typ mit allen relevanten Parametern aus der folgenden Liste ein.

```
[<extension_name>]
ExtPriorityType=< {internal, protocol}>
WINNT_EXT_LIBS=<dll name for NT>
WIN95_EXT_LIBS=<dll name for 95>
SOLARIS_EXT_LIBS=<dll name for Solaris>
LINUX_EXT_LIBS=<dll name for Linux>
HPUX_EXT_LIBS=<dll name for HP>
AIX_EXT_LIBS=<dll name for IBM>
LibCfgFunc=<configuration function name>
UtilityExt=<other extensions list>
WINNT_DLLS=<dlls to load to the interpreter context, for NT>
WIN95_DLLS=<dlls to load to the interpreter context, for 95>
SOLARIS_DLLS=<dlls to load to the interpreter context, for Solaris>
LINUX_DLLS=<dlls to load to the interpreter context, for Linux>
HPUX_DLLS=<dlls to load to the interpreter context, for HP>
AIX_DLLS=<dlls to load to the interpreter context, for IBM>
ExtIncludeFiles=<extra include files. several files can be separated by a co
mma>
ExtCmdLineConc=<extra command line (if the attr exists concatenate value)>
ExtCmdLineOverwrite=<extra command line (if the attr exists overwrite value)>
CallActionByNameFunc=<interpreter exec_action function>
GetFuncAddress=<interpreter get_location function>
RunLogicInitFunc=<action_logic init function>
RunLogicRunFunc=<action_logic run function>
RunLogicEndFunc=<action_logic end function>
```

   Ein Oracle NCA Vuser-Typ wird beispielsweise durch Folgendes dargestellt:

```
[Oracle_NCA]
ExtPriorityType=protocol
WINNT_EXT_LIBS=ncarp11i.dll
WIN95_EXT_LIBS=ncarp11i.dll
LINUX_EXT_LIBS=liboranca11i.so
SOLARIS_EXT_LIBS=liboranca11i.so
HPUX_EXT_LIBS=liboranca11i.sl
AIX_EXT_LIBS=liboranca11i.so
LibCfgFunc=oracle_gui_configure
UtilityExt=lrun_api,HttpEngine
ExtCmdLineOverwrite=-WinInet No
ExtCmdLineConc=-UsingWinInet No
SecurityRequirementsFiles=oracle_nca.asl
SecurityMode=On
```

VuGen ist so konzipiert, dass es einen neuen Vuser-Typ ohne Codeänderungen verarbeiten kann. Sie müssen jedoch möglicherweise eine spezielle Ansicht hinzufügen.

Es wird kein generischer Treiber mit VuGen bereitgestellt, aber Sie können einen der vorhandenen Treiber anpassen. Um einen angepassten Treiber zu verwenden, ändern Sie die Datei *mdrv.dat*. Fügen Sie eine Zeile mit der Plattform und dem vorhandenen Treiber hinzu, und fügen Sie anschließend eine neue Zeile mit Ihrem angepassten Treibernamen im Format <platform>_DLLS=<my_replay.dll name> hinzu. Wenn Ihre SAP-Wiedergabe-DLL beispielsweise den Namen SAPPLAY32.DLL hat, fügen Sie die folgenden beiden Zeilen in den Abschnitt [sap] von *mdrv.dat* ein:

```
WINNT=sapdrv32.exe
WINNT_DLLS=sapplay32.dll
```

2. Hinzufügen einer CFG-Datei - optional

Sie können eine Konfigurationsdatei angeben, um die standardmäßigen Laufzeiteinstellungen für Ihr Protokoll festzulegen. Sie definieren diese in der Variablen LibCfgFunc in der Datei *mdrv.dat* oder Sie fügen eine Datei mit dem Namen *default.cfg* in den neuen Protokollunterordner unter Vorlagen ein. Nachfolgend finden Sie ein Beispiel für *default.cfg*.

```
[ThinkTime]
Options=NOTHINK
Factor=1
LimitFlag=0
Limit=1
[Iterations]
NumOfIterations=1
IterationPace=IterationASAP
StartEvery=60
RandomMin=60
RandomMax=90
[Log]
LogOptions=LogExtended
MsgClassData=0
MsgClassParameters=0
MsgClassFull=1
```

3. Einfügen einer LRP-Datei

Fügen Sie in den Ordner **dat/protocols** in eine *lrp*-Datei ein, die das Protokoll definiert. Diese Datei enthält die Konfigurationsinformationen für das Protokoll in den Abschnitten **Protokoll**, **Vorlage**, **VuGen** und **API**. Manche Protokolle enthalten möglicherweise zusätzliche Abschnitte, entsprechend den zusätzlichen Optionen der Laufzeiteinstellungen.

Der Abschnitt **Protokoll** enthält den Namen, die Kategorie, die Beschreibung und den Bitmapspeicherort für das Protokoll.

```
 [Protocol]
Name=WAP
CommonName=WAP
Category=Wireless
Description=Wireless Application Protocol - Dieses Protokoll dient der webba
sierten drahtlosen Kommunikation zwischen mobilen Geräten und Inhaltsanbiete
rn.
Icon=bitmaps\wap.bmp
Hidden=0
Single=1
Multi=0
```

Der Abschnitt **Vorlage** gibt den Namen der verschiedenen Abschnitte des Skripts und den Standardtestnamen an.

```
[Template]
vuser_init.c=init.c
vuser_end.c=end.c
Action1.c=action.c
Default.usp=test.usp
@@TestName@@.usr=wap.usr
default.cfg=default.cfg
```

Der Abschnitt **VuGen** enthält Informationen zu den Aufzeichnungs- und Wiedergabeengines, zusammen mit den erforderlichen DLLs und Laufzeitdateien.

Der Abschnitt **API** enthält Informationen zu den API-Skriptfunktionen des Protokolls.

Sie können eine der vorhandenen LRP-Dateien im Protokollordner als Basis für Ihr neues Protokoll verwenden.

4. ## Angeben einer Vorlage

Nach Hinzufügen einer *lrp*-Datei fügen Sie einen Unterordner zur Vorlage *M_LROOT/* hinzu, mit einem Namen, der dem in der *lrp*-Datei definierten Protokollnamen entspricht. Fügen Sie in diesen Unterordner eine *default.cfg*-Datei ein, die die Standardeinstellungen für die allgemeinen Einstellungen und die Laufzeiteinstellungen definiert.

Wenn Sie eine globale Kopfzeilendatei für die gesamten Protokolle Ihres Skripts verwenden möchten, fügen Sie eine Datei mit dem Namen *globals.h* hinzu. Diese Datei muss eine Include-Anweisung enthalten, die auf eine Kopfzeilendatei für das neue Protokoll verweist. Der Unterordner *template/http* enthält beispielsweise eine Datei mit dem Namen *globals.h*, die VuGen zur *as_web.h*-Datei im Verzeichnis *Include* leitet:

```
#include #as_web.h"
```

VuGen-Datei- und -Bibliotheksspeicherort

Die DAT-Dateien von VuGen enthalten Speicherortinformationen zu den Dateien des Skripts und den Bibliotheksdateien für bestimmte Protokolle.

Es werden zwei DAT-Dateien von VuGen verwendet: **vugen.dat** und **mdrv.dat**.

vugen.dat

Diese *vugen.dat*-Datei befindet sich im Ordner *M_LROOT\dat* und enthält allgemeine Informationen zu VuGen, die sowohl von VuGen als auch vom Controller verwendet werden müssen.

```
[Templates]
RelativeDirectory=template
```

Der Abschnitt Vorlagen gibt an, wo sich die Vorlagen für die VuGen-Protokolle befinden. Der Standardeintrag gibt an, dass sie sich im relativen Ordner *template* befinden. Jedes Protokoll besitzt einen Unterordner unter *template*, der die Vorlagendateien für dieses Protokoll enthält.

Der nächste Abschnitt ist **GlobalFiles**.

```
[GlobalFiles]
main.c=main.c
@@TestName@@.usr=test.usr
default.cfg=test.cfg
default.usp=test.usp
```

Der Abschnitt **GlobalFiles** enthält eine Liste von Dateien, die VuGen bei jedem Erstellen eines neuen Tests in den Testordner kopiert. Beispielsweise kopiert VuGen bei einem Test, der mit "user1" benannt ist, "main.c", "user1.usr" und "user1.cfg" in den Testordner.

Der Abschnitt ActionFiles enthält den Namen der Datei, die die vom Vuser auszuführenden Aktionen enthält und über die Iterationen ausgeführt werden müssen.

```
[ActionFiles]
@@actionFile@@=action.c
```

Neben den oben gezeigten Einstellungen enthält *vugen.dat* Einstellungen, die das Betriebssystem und andere Einstellungen in Verbindung mit der Kompilierung angeben.

mdrv.dat

Die Datei *mdrv.dat* enthält einen separaten Abschnitt für jedes Protokoll, das den Speicherort der Bibliotheksdateien und ausführbaren Treiberdateien definiert. Informationen zum Verwenden dieser Datei zum Erstellen eines neuen Protokolls finden Sie unter "Erstellen eines neuen Vuser-Typs" auf Seite 1077.

Speichern von Laufzeiteinstellungen in externen Dateien

Das Vuser-Verhalten bezieht sich auf die Elemente, die Sie in den Laufzeiteinstellungen festlegen können, z. B. Wartezeiten, Taktzeiten, Schleifeniterationen und Protokollierung.

Da VuGen das Vuser-Skript und das Vuser-Verhalten als zwei voneinander unabhängige Quellen erstellt, können Sie das Vuser-Verhalten konfigurieren, ohne direkt auf das Vuser-Skript zu verweisen. Über diese Funktion können Sie Konfigurationsänderungen an einem Vuser durchführen und mehrere Profile für dasselbe Vuser-Skript speichern.

VuGen speichert die Einstellungen für das Verhalten in der Standarddatei **Vuser.cfg**. Sie können mehrere Versionen dieser Datei für unterschiedliche Benutzerverhalten speichern und anschließend das Vuser-Skript ausführen, das die relevante *.cfg*-Datei referenziert.

Standardmäßig können Sie die Verhaltensdatei von VuGen aus nicht kontrollieren. VuGen verwendet automatisch die *.cfg* -Datei mit demselben Namen wie das Skript.

Um eine bestimmte Konfigurationsdatei aufzurufen, führen Sie den Vuser über die Befehlszeile aus, und fügen Sie folgende Zeichenfolge hinzu:

```
-cfg c:\tmp\<MyCustomConfigFile>.cfg
```

Weitere Informationen zu Befehlszeilenparametern finden Sie unter Command Line Parameters.

Hinweis: Das Linux-Dienstprogramm *run_db_vuser* unterstützt diese Option nicht.

Erstellen und Ausführen von Skripts unter Linux

Erstellen und Ausführen von Skripts unter Linux - Übersicht

Sie können VuGen in einer Linux-Umgebung auf folgende Weisen einsetzen:

- Sie können VuGen verwenden, um Vuser-Skripts zu erstellen, die auf Linux-Plattformen ausgeführt werden. In diesem Fall zeichnen Sie Ihre Anwendung in einer Windows-Umgebung auf und führen sie unter Linux aus - Aufzeichnen wird unter Linux nicht unterstützt.
- Benutzer, die nur in einer Linux-Umgebung arbeiten, können Vuser-Skripts programmieren. Skripts können in C oder C++ programmiert werden und müssen in einer dynamischen Bibliothek kompiliert werden.

Zum Erstellen eines Vuser-Skripts durch Programmieren können Sie eine Vuser-Vorlage als Basis für ein größeres Vuser-Skript verwenden. Die Vorlage stellt Folgendes bereit:

- korrekte Programmstruktur
- Vuser-API-Aufrufe
- Quellcode und Makefiles zum Erstellen einer dynamischen Bibliothek

Programmieren von Vuser-Aktionen

Die Vuser-Skriptdateien *test.c*, *test.usr* und *test.cfg* können für Ihren Vuser angepasst werden.

Sie programmieren die tatsächlichen Vuser-Aktionen in der Datei *test.c*. Diese Datei hat die vorgeschriebene Struktur für ein programmiertes Vuser-Skript. Das Vuser-Skript ist in drei Abschnitte gegliedert: **vuser_init**, **Actions** und **vuser_end**.

Beachten Sie, dass die Vorlage für Benutzer von C++ den Parameter **extern C** definiert. Diese Definition ist für alle C++-Benutzer erforderlich, um sicherzustellen, dass keine der exportierten Funktionen versehentlich geändert wird.

```
#include "lrun.h"
#if defined(__cplusplus) || defined(cplusplus) extern "C"
{
#endif
int LR_FUNC vuser_init(LR_PARAM p)
{
lr_message("vuser_init done\n");

return 0;
}
 int Actions(LR_PARAM p)
{
lr_message("Actions done\n");

return 0;
}
 int vuser_end(LR_PARAM p)
{
lr_message("vuser_end done\n");

return 0;
}
#if defined(__cplusplus) || defined(cplusplus)}
#endif
```

Sie programmieren Vuser-Aktionen direkt in das leere Skript vor der Funktion **lr_message** jedes Abschnitts.

Der Abschnitt *vuser_init* wird zuerst ausgeführt, d. h. während der Initialisierung. Fügen Sie in diesen Abschnitt die Verbindungsinformationen und das Anmeldeverfahren ein. Der Abschnitt *vuser_init* wird bei jedem Ausführen des Skripts nur einmal ausgeführt.

Der Abschnitt *Actions* wird nach der Initialisierung ausgeführt. Fügen Sie in diesen Abschnitt die tatsächlichen vom Vuser ausgeführten Vorgänge ein. Sie können festlegen, dass der Vuser den Abschnitt *Actions* (in der Datei *test.cfg*) wiederholen soll.

Der Abschnitt *vuser_end* wird zuletzt ausgeführt, nach allen Aktionen des Vusers. Fügen Sie in diesen Abschnitt die Verfahren zur Bereinigung und Abmeldung ein. Der Abschnitt *vuser_end* wird bei jedem Ausführen des Skripts nur einmal ausgeführt.

Hinweis: LoadRunner kontrolliert die Vuser durch Senden der Linux-Signale SIGHUP, SIGUSR1 und SIGUSR2. Verwenden Sie diese Signale nicht in Ihren Vuser-Programmen.

Erstellen einer Vorlage

VuGen umfasst ein Dienstprogramm, das eine Vorlage in Ihren Arbeitsordner kopiert. Dieses Dienstprogramm wird als *mkdbtest* bezeichnet und es befindet sich in *$M_LROOT/bin*. Das Dienstprogramm wird durch Eingabe des folgenden Befehls ausgeführt:

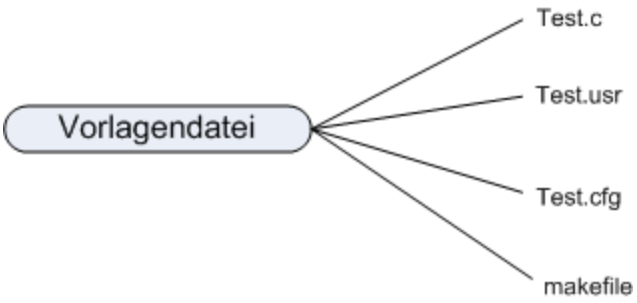
```
mkdbtest name
```

Bei Ausführen von *mkdbtest* wird ein Ordner mit dem Namen **name** erstellt, das die Vorlagendatei *name.c* enthält. Wenn Sie beispielsweise Folgendes eingeben:

```
mkdbtest test1
```

erstellt *mkdbtest* einen Ordner mit dem Namen *test1*, der das Vorlagenskript *test1.c.* enthält.

Bei Ausführen des Dienstprogramms *mkdbtest* wird ein Ordner erstellt, der die vier Dateien *test.c*, *test.usr*, *test.cfg* und *Makefile* enthält, wobei *test* der Testname ist, den Sie für `mkdbtest` angegeben haben.



Manuelles Konfigurieren von Laufzeiteinstellungen

Um Vuser-Laufzeiteinstellungen zu konfigurieren, ändern Sie die Dateien *default.cfg* und *default.usp*, die mit dem Skript erstellt wurden. Diese Laufzeiteinstellungen entsprechen den Laufzeiteinstellungen von VuGen. (Siehe "Laufzeiteinstellungen" auf Seite 330.) Die Datei *default.cfg* enthält die Einstellung für die Optionen *General*, *Think Time* und *Log*. Die Datei *default.usp* enthält die Einstellung für die Ausführungslogik und den Takt.

Allgemeine Optionen

Für Linux-Vuser-Skripts steht eine allgemeine Option zur Verfügung:

ContinueOnError legt fest, dass der Vuser bei einem Fehler fortfahren soll. Geben Sie **1** an, um diese Option zu aktivieren und **0**, um sie zu deaktivieren.

Im folgenden Beispiel fährt der Vuser bei einem Fehler fort.

```
[General]
ContinueOnError=1
```

Latenzzeitoptionen

Sie können die Latenzzeitoptionen festlegen, um zu kontrollieren, wie der Vuser die Latenzzeit während der Skriptausführung verwendet. Sie legen die Parameter **Options**, **Factor**, **LimitFlag** und **Limit** entsprechend dem folgenden Diagramm fest.

Option	Options	Factor	LimitFlag	Limit
Latenzzeit ignorieren	NOTHINK	n/v	n/v	n/v
Wie aufgezeichnet	RECORDED	1.000	n/v	n/v
Aufgezeichnete Latenzzeit multiplizieren mit	MULTIPLY	Zahl	n/v	n/v
Zufälligen Prozentsatz der Latenzzeit verwenden	RANDOM	Bereich	Niedrigster Prozentsatz	Höchster Prozentsatz
Latenzzeit begrenzen auf	RECORDED/ MULTIPLY	Zahl (für MULTIPLY)	1	Wert in Sekunden

Um die während der Ausführung verwendete Latenzzeit zu begrenzen, setzen Sie die Variable `LimitFlag` auf 1 und geben Sie die Latenzzeitbegrenzung in Sekunden an.

Im folgenden Beispiel geben die Einstellungen an, dass der Vuser die aufgezeichnete Latenzzeit durch einen zufälligen Prozentsatz von 50 bis 150 multiplizieren soll.

```
[ThinkTime]
Options=RANDOM
Factor=1
LimitFlag=0
Limit=0
ThinkTimeRandomLow=50
ThinkTimeRandomHigh=150
```

Protokolloptionen

Sie können die Protokolloptionen festlegen, um eine kurze oder eine ausführliche Protokolldatei für die Ausführung des Skripts zu erstellen.

```
[Log]
LogOptions=LogBrief
MsgClassData=0
```

```
MsgClassParameters=0
MsgClassFull=0
```

Sie legen die Parameter **LogOptions**, **MsGClassData**, **MsgClassParameters** und **MsgClassFull** entsprechend dem folgenden Diagramm fest:

Protokol-lierungstyp	LogOptions	MsgClassData	MsgClassParameters	MsgClassFull
Protokollierung deaktivieren	LogDisabled	n/v	n/v	n/v
Standard-protokoll	LogBrief	n/v	n/v	n/v
Parameter-substitution (ausschließlich)	LogExtended	0	1	0
Vom Server zurückgegebene Daten (ausschließlich)	LogExtended	1	0	0
Erweiterte Verfolgung (ausschließlich)	LogExtended	0	0	1
Alle	LogExtended	1	1	1

Im folgenden Beispiel teilen die Einstellungen dem Vuser mit, alle vom Server zurückgegebenen Daten und die für die Substitution verwendeten Parameter zu protokollieren.

```
[Log]
LogOptions=LogExtended
MsgClassData=1
MsgClassParameters=1
MsgClassFull=0
```

Iterationen und Ausführungslogik

Über die Iterationsoptionen können Sie festlegen, dass mehrere Iterationen ausgeführt werden sollen und der Takt zwischen den Iterationen kontrolliert werden soll. Außerdem können Sie die Reihenfolge der Aktionen und ihre Gewichtung manuell festlegen. Um die Ausführungslogik und die Iterationseigenschaften zu ändern, müssen Sie die Datei *default.usp* bearbeiten.

Um festzulegen, dass der Vuser mehrere Aktionen des Aktionsabschnitts ausführen soll, setzen Sie RunLogicNumOfIterations auf den entsprechenden Wert.

Um den Takt zwischen den Iterationen zu kontrollieren, legen Sie die Variable *RunLogicPaceType* und ihre zugehörigen Werte entsprechend dem folgenden Diagramm fest:

Takt	RunLogicPaceType	Zugehörige Variablen
Schnellstmöglich	ASAP	n/v
Festgelegte Verzögerung zwischen Iterationen	Const	RunLogicPaceConstTime
Zufällige Verzögerung zwischen Iterationen	Random	RunLogicRandomPaceMin, RunLogicRandomPaceMax
Feste Verzögerung nach jeder Iteration	ConstAfter	RunLogicPaceConstAfterTime
Zufällige Verzögerung nach jeder Iteration	After	RunLogicAfterPaceMin, RunLogicAfterPaceMax

Im folgenden Beispiel teilen die Einstellungen dem Vuser mit, vier Iterationen durchzuführen und zwischen Iterationen eine zufällige Anzahl von Sekunden lang zu warten. Der Bereich der zufälligen Zahl liegt zwischen 60 und 90 Sekunden.

```
[RunLogicRunRoot]
MercIniTreeFather=""
MercIniTreeSectionName="RunLogicRunRoot"
RunLogicRunMode="Random"
RunLogicActionOrder="Action,Action2,Action3"
RunLogicPaceType="Random"
RunLogicRandomPaceMax="90.000"
RunLogicPaceConstTime="40.000"
RunLogicObjectKind="Group"
RunLogicAfterPaceMin="50.000"
Name="Run"
RunLogicNumOfIterations="4"
RunLogicActionType="VuserRun"
RunLogicAfterPaceMax="70.000"
RunLogicRandomPaceMin="60.000"
MercIniTreeSons="Action,Action2,Action3"
RunLogicPaceConstAfterTime="30.000"
```

Manuelles Definieren von Transaktionen und Einfügen von Rendezvous-Punkten

Beim Programmieren eines Vuser-Skripts ohne VuGen müssen Sie die Vuser-Datei manuell konfigurieren, um Transaktionen und Rendezvous zu ermöglichen. Die Konfigurationseinstellungen sind in der Datei *test.usr* aufgeführt.

```
[General]
Type=any
DefaultCfg=Test.cfg
BinVuser=libtest.libsuffix
```

```
RunType=Binary
[Actions]
vuser_init=
Actions=
vuser_end=
[Transactions]
transaction1=
[Rendezvous]
Meeting=
```

Jede Transaktion und jedes Rendezvous muss in der Datei *usr* definiert werden. Fügen Sie den Transaktionsnamen zum Abschnitt **Transaktionen** hinzu (gefolgt von einem "="). Fügen Sie jeden Rendezvous-Namen zum Abschnitt **Rendezvous** hinzu (gefolgt von einem "="). Wenn die Abschnitte nicht vorhanden sind, fügen Sie sie zur Datei *usr* wie oben gezeigt hinzu.

Manuelles Kompilieren von Skripts unter Linux

Nachdem Sie die Vorlage geändert haben, kompilieren Sie sie mit der entsprechenden *Makefile* im Skriptordner. Der Compiler erstellt eine dynamische Bibliothek mit dem Namen **libtest.so**.

Sie können die *Makefile* ändern und zusätzliche Compilerkennzeichen und Bibliotheken durch Ändern der entsprechenden Abschnitte hinzufügen.

Wenn Sie mit einer allgemeinen Vorlage arbeiten, müssen Sie die Bibliotheken und Kopfzeilendateien Ihrer Anwendung einfügen. Wenn Ihre Anwendung beispielsweise eine Bibliothek mit dem Namen *testlib* verwendet, fügen Sie diese in den Abschnitt "LIBS" ein.

```
LIBS         = \
    -testlib \
    -lLrun50 \
    -lm
```

Nachdem Sie die *Makefile* geändert haben, geben Sie `make` über die Befehlszeile im Arbeitsordner ein, um die dynamischen Bibliotheksdateien für das Vuser-Skript zu erstellen.

Nach Erstellen eines Skripts können Sie seine Funktionalität über die Befehlszeile prüfen. Prüfen Sie, ob Ihr Skript mit dem Server kommuniziert und alle erforderlichen Aufgaben ausführt. Weitere Informationen finden Sie unter "Running a Vuser from the Linux Command Line".

Programmieren mit der XML API

Übersicht über das Programmieren mit der XML API

Die VuGen-Unterstützung für XML ermöglicht es Ihnen, dynamisch mit dem XML-Code zu arbeiten und die Werte während der Ausführung des Tests abzurufen. Führen Sie die folgenden Schritte aus, um ein effizientes XML-Skript zu erstellen:

- Zeichnen Sie ein Skript im gewünschten Protokoll auf, normalerweise **Web**, **Web Services** oder **Wireless**.
- Kopieren Sie die XML-Strukturen in Ihr Skript.
- Fügen Sie XML-Funktionen von der LR API hinzu, um dynamische Daten und die XML-Elementwerte abzurufen.

Die LR API verwendet XPath, die XML Path-Sprache zum Bearbeiten des Textes in einem XML-Dokument.

Über die Laufzeiteinstellungen können Sie festlegen, dass VuGen die Ausgabewerte von XML-Elementen im Ausführungsprotokollfenster anzeigt. VuGen zeigt die Zeilennummer, die Anzahl von Übereinstimmungen und den Wert an. Um die Anzeige der Werte zu ermöglichen, müssen Sie die Parametersubstitution aktivieren. Öffnen Sie in den Laufzeiteinstellungen den Knoten **Allgemein: Protokoll**, wählen Sie **Erweitertes Protokoll** und anschließend **Parametersubstitution** aus. Weitere Informationen finden Sie unter "Laufzeiteinstellungen" auf Seite 330.

Alle Vuser API XML-Funktionen geben die Anzahl der gefundenen Übereinstimmungen zurück. Bei einem Fehler wird Null zurückgegeben.

Verwenden von XML-Funktionen

Dieser Abschnitt enthält Beispiele für das Arbeiten mit Daten in einer XML-Struktur. Bestimmte Funktionen ermöglichen Ihnen das Abrufen von Informationen und andere das Schreiben von Informationen in eine XML-Struktur. Die Beispiele verwenden die folgende XML-Struktur, die die Namen und Durchwahlnummern von mehreren Mitarbeitern in der Organisation "Acme" enthält.

```
<acme_org>
    <accounting_dept>
         <employee type=’PT’>
              <name>John Smith</name>
              <extension>2145</extension>
         </employee>
    </accounting_dept>
    <engineering_dept>
         <employee type=’PT’>
              <name>Sue Jones</name>
              <extension>2375</extension>
         </employee>
    </engineering_dept>
</acme_org>
```

Lesen von Informationen aus einer XML-Struktur

Zum Lesen von Informationen aus einer XML-Struktur stehen die folgenden Funktionen zur Verfügung:

lr_xml_extract	Extrahiert XML-Zeichenfolgenfragmente aus einer XML-Zeichenfolge.
lr_xml_find	Führt eine Abfrage für eine XML-Zeichenfolge durch.
lr_xml_get_ values	Ruft Werte von XML-Elementen ab, die über eine Abfrage gefunden wurden.

Zum Abrufen eines spezifischen Werts über eine Abfrage, geben Sie die Tags der übergeordneten und untergeordneten Knoten in einem Pfadformat an.

Um beispielsweise einen Mitarbeiternamen in der Accounting-Abteilung zu finden, verwenden Sie die folgende Zeichenfolge:

```
lr_xml_get_values("XML={XML_Input_Param}",
"ValueParam=OutputParam",
"Query=/acme_org/accounting_dept/employee/name",
LAST);
```

Das Ausführunsprotokollfenster (mit aktivierter erweiterter Protokollierung) zeigt die Ausgabe dieser Funktion an:

Output.
Action.c(20): "lr_xml_get_values" was successful, 1 match processed
Action.c(25): Query result = **John Smith**

Schreiben in eine XML-Struktur

Zum Schreiben von Werten in eine XML-Struktur stehen die folgenden Funktionen zur Verfügung:

lr_xml_delete	Löscht Fragmente aus einer XML-Zeichenfolge.
lr_xml_insert	Fügt ein neues XML-Fragment in eine XML-Zeichenfolge ein.
lr_xml_replace	Ersetzt Fragmente einer XML-Zeichenfolge.
lr_xml_set_ values	Legt die Werte von XML-Elementen fest, die über eine Abfrage gefunden wurden.
lr_xml_ transform	Wendet XSL-Umwandlung (Extensible Stylesheet Language) auf XML-Daten an.

Die häufigste Funktion zum *Schreiben* ist **lr_xml_set_values**, mit der die Werte von angegebenen Elementen in einer XML-Zeichenfolge festgelegt werden. Das folgende Beispiel verwendet **lr_xml_ set_values** zum Ändern der Durchwahlnummern von zwei Elementen von *employee* in einer XML-Zeichenfolge.

Zunächst speichern wir die XML-Zeichenfolge in einem Parameter, der als *XML_Input_Param* bezeichnet wird. Wir möchten, dass die beiden Werte abgeglichen und ersetzt werden und bereiten

deshalb die zwei neuen Parameter *ExtensionParam_1* und *ExtensionParam_2* vor und setzen ihre Werte auf die beiden neuen Durchwahlnummern **1111** und **2222**.

lr_xml_set_values enthält das Argument **ValueName=ExtensionParam**, das die Werte von **ExtensionParam_1** und **ExtensionParam_2** aufgreift. Die aktuellen Durchwahlnummern der beiden Mitarbeiter werden durch die Werte dieser Parameter **1111** und **2222** ersetzt. Der Wert von **OutputParam** wird dann bewertet, um zu prüfen, ob die neuen Durchwahlnummern tatsächlich ersetzt wurden.

```
Action() {

     int i, NumOfValues;
     char buf[64];

     lr_save_string(xml_input, "XML_Input_Param"); // Save input as parameter
     lr_save_string("1111", "ExtensionParam_1");
     lr_save_string("2222", "ExtensionParam_2");

     lr_xml_set_values("XML={XML_Input_Param}",
          "ResultParam=NewXmlParam", "ValueParam=ExtensionParam",
          "SelectAll=yes", "Query=//extension", LAST);

     NumOfValues= lr_xml_get_values("XML={NewXmlParam}",
          "ValueParam=OutputParam", "Query=//extension",
          "SelectAll=yes", LAST);

     for (i = 0; i < NumOfValues; i++) {/* Print the multiple values of MultiPar
am */

          sprintf(buf, "Retrieved value %d : {OutputParam_%d}", i+1, i+1);
          lr_output_message(lr_eval_string(buf));
     }

     return 0;
}
Output:
Action.c(40): Retrieved value 1: 1111
Action.c(40): Retrieved value 2: 2222
```

Angeben von XML-Funktionsparametern

Die meisten XML API-Funktionen setzen voraus, dass Sie das **XML-Element** und eine **Abfrage** festlegen. Sie können auch angeben, ob Sie alle Ergebnisse oder nur ein Ergebnis abrufen wollen.

Definieren des XML-Elements

Zum Definieren des XML-Elements für die Abfrage können Sie eine literale Zeichenfolge des XML-Elements oder einen Parameter, der das XML-Element enthält, angeben. Das folgende Beispiel zeigt die XML-Zeichenfolge, definiert als literale Zeichenfolge:

"XML=<employee>JohnSmith</employee>"

Alternativ kann die **XML**-Zeichenfolge ein Parameter sein, der die XML-Daten enthält. Beispiel:

"XML={EmployeeNameParam}"

Abfragen einer XML-Struktur

Angenommen, Sie suchen einen Wert in einem XML-Tag, z. B. die Durchwahl eines Mitarbeiters. Sie formulieren eine Abfrage für den gewünschten Wert. Die Abfrage gibt den Speicherort des Elements und das abzurufende oder festzulegende Element an. Der Pfad, den Sie angeben, begrenzt den Umfang der Suche auf ein spezifisches Tag. Sie können auch nach allen Elementen eines spezifischen Typs unter allen Knoten unterhalb des Stammverzeichnisses suchen.

Verwenden Sie für einen spezifischen Pfad die folgende Abfrage: `"Query=/`*full_xml_path_name*`/`*element_name*`"`

Verwenden Sie für denselben Elementnamen unter allen Knoten die folgende Abfrage: `"Query=//`*element_name*`"`

In der VuGen-Implementierung von XML-Funktionen ist der Umfang einer Abfrage die gesamte XML-Struktur. Die Strukturinformationen werden an die Vuser API-Funktionen als Wert des *xml*-Arguments gesendet.

Abgleichen mehrerer Abfragen

Wenn Sie eine Abfrage an einem XML-Element durchführen, gibt VuGen standardmäßig nur die erste Übereinstimmung zurück. Um mehrere Werte über eine Abfrage abzurufen, müssen Sie in Ihren Funktionen das Attribut "`SelectAll=yes`" angeben. VuGen fügt das Suffix *_index* hinzu, um mehrere Parameter anzugeben. Wenn Sie beispielsweise einen Parameter über den Namen *EmployeeName* definiert haben, erstellt VuGen *EmployeeName_1*, *EmployeeName_2*, *EmployeeName_3* usw.

lr_xml_set_values("XML={XML_Input_Param}",
"ResultParam=NewXmlParam", "ValueParam=ExtensionParam",
"SelectAll=yes", "Query=//extension", LAST);

Mit Funktionen, die in einen Parameter *schreiben*, können die in den Parameter geschriebenen Werte dann bewertet werden. Beispielsweise ruft der folgende Code mehrere Übereinstimmungen einer Abfrage ab und druckt sie:

```
NumOfValues = lr_xml_get_values("Xml={XmlParam}", "Query=//name",
        "SelectAll=yes", "ValueParam=EmployeeName", LAST);
```

Für Funktionen, die aus Parametern *lesen*, müssen die Werte der Parameter vordefiniert sein. Der Parameter muss auch die Konvention *ParamName_IndexNumber* verwenden, z. B. *Param*_1, *Param*_2, *Param*_3 usw. Diese Sammlung von Parametern wird auch als Parametersatz bezeichnet.

Im folgenden Beispiel liest *lr_xml_set_values* Werte aus dem Parametersatz und verwendet diese Werte dann in der XPath-Abfrage. Der Parametersatz, der die Mitarbeiterdurchwahlen darstellt, wird als *ExtensionParam* bezeichnet. Er hat zwei Mitglieder: *ExtensionParam_1* und *ExtensionParam_2*. Die Funktion **lr_xml_set_values** führt eine Abfrage in der XML-Eingabezeichenfolge durch und setzt den Wert der ersten Übereinstimmung auf **1111** und den der zweiten Übereinstimmung auf **2222**.

```
lr_save_string("1111", "ExtensionParam_1");
lr_save_string("2222", "ExtensionParam_2");

lr_xml_set_values("XML={XML_Input_Param}",
    "ResultParam=NewXmlParam", "ValueParam=ExtensionParam",
    "SelectAll=yes", "Query=//extension", LAST);
```

XML-Attribute

VuGen bietet Unterstützung für Attribute. Sie können einen einfachen Ausdruck zum Bearbeiten von Attributen von XML-Elementen und Knoten verwenden, ebenso wie Sie die Elemente selbst bearbeiten können. Sie können das gewünschte Attribut oder nur Attribute mit spezifischen Werten ändern.

Im folgenden Beispiel löscht **lr_xml_delete** das erste Element **cubicle** mit dem Namensattribut.

```
lr_xml_delete(        "Xml={ParamXml}",
                         "Query=//cubicle/@name",
                        "ResultParam=Result",
                         LAST
    );
```

Im folgenden Beispiel löscht **lr_xml_delete** das erste Element **cubicle** mit dem Namensattribut **Paul**.

```
lr_xml_delete(        "Xml={ParamXml}",
         "Query=//cubicle/@name="Paul",
        "ResultParam=Result",
         LAST
    );
```

Strukturieren von XML-Skripts

Zunächst erstellen Sie ein neues Skript im gewünschten Protokoll. Sie können eine Sitzung in diesem Protokoll aufzeichnen oder Sie können die gesamte Sitzung ohne Aufzeichnung programmieren. Strukturieren Sie den Abschnitt **Actions** des Skripts wie folgt:

- XML-Eingabedeklaration
- Abschnitt **Actions**

Der XML-Eingabeabschnitt enthält die XML-Struktur, die Sie als Eingabevariable verwenden möchten. Sie definieren die XML-Struktur als eine Variable vom Typ **char**. Beispiel:

```
char *xml_input=
"<acme_org>"
    "<employee>"
        " <name>John Smith</name>"
        "<cubicle>227</cubicle>"
        "<extension>2145</extension>"
```

```
    "</employee>"
    "<employee>"
        "<name>Sue Jones</name>"
        "<cubicle>227</cubicle>"
        "<extension>2375</extension>"
    "</employee>"
"</acme_org>";
```

Der Abschnitt **Actions** enthält die Bewertung der Variablen und Abfragen für die Elementwerte. Im folgenden Beispiel wird die XML-Eingabezeichenfolge mit **lr_save_string** bewertet. Es erfolgt eine Abfrage der Eingabevariablen nach Mitarbeiternamen und Durchwahlnummern.

```
Action() {

    /* Save the input as a parameter.*/
    lr_save_string(xml_input, "XML_Input_Param");
    /* Query 1 - Retrieve an employee name from the specified element.*/
    lr_xml_get_values("XML={XML_Input_Param}",
        "ValueParam=OutputParam",
        "Query=/acme_org/employee/name", LAST);

    /* Query 2 - Retrieve an extension under any path below the root.*/
    lr_xml_get_values("XML={XML_Input_Param}",
        "ValueParam=OutputParam",
        "Query=//extension", LAST);

    return 0;
}
```

Erweitern einer aufgezeichneten Sitzung mit XML

Sie können ein XML-Skript vorbereiten, indem Sie eine Sitzung aufzeichnen und anschließend die relevanten XML- und Vuser API-Funktionen manuell hinzufügen.

Das folgende Beispiel zeigt, wie eine aufgezeichnete Sitzung mit Vuser API-Funktionen erweitert wurde. Beachten Sie, dass die einzige Funktion, die aufgezeichnet wurde, **web_submit_data** war (fett dargestellt).

Der erste Abschnitt enthält die XML-Eingabedeklaration der SOAPTemplate-Variablen für eine SOAP-Meldung:

```
#include "as_web.h"
// SOAP message
const char*        pSoapTemplate=
  "<soap:Envelope xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">"
      "<soap:Body>"
          "<SendMail xmlns=\"urn:EmailIPortTypeInft-IEmailService\"/>"
      "</soap:Body>"
  "</soap:Envelope>";
```

Der folgende Abschnitt stellt die Aktionen der Benutzer dar:

```
Action1()
{
// get response body
 web_reg_save_param("ParamXml", "LB=", "RB=", "Search=body", LAST);
// fetch weather by HTTP GET
web_submit_data("GetWeather","Action=http://glkev.net.innerhost.com/glkev_ws/
                WeatherFetcher.asmx/GetWeather",
               "Method=GET",
               "EncType=",
               "RecContentType=text/xml",
  "Referer=http://glkev.net.innerhost.com
                   /glkev_ws/WeatherFetcher.asmx?op=GetWeather",
         "Snapshot=t2.inf",
         "Mode=HTTP",
         ITEMDATA,
         "Name=zipCode", "Value=10010", ENDITEM,
         LAST);


// Get City value
  lr_xml_get_values("Xml={ParamXml}",
                  "Query=City",
                  "ValueParam=ParamCity",
               LAST
    );
 lr_output_message(lr_eval_string("***** City = {ParamCity} *****"));


// Get State value
 lr_xml_get_values("Xml={ParamXml}",
                  "Query=State",
                  "ValueParam=ParamState",
               LAST
    );
 lr_output_message(lr_eval_string("***** State ={ParamState}*****"));


// Get several values at once by using template

lr_xml_get_values_ex("Xml={ParamXml}",


"Template="
            "<Weather>"

               "<Time>{ParamTime}</Time>"
               "<Temperature>{ParamTemp}</Temperature>"
               "<Humidity>{ParamHumid}</Humidity>"
               "<Conditions>{ParamCond}</Conditions>"
            "</Weather>",
     LAST
    );
```

```
   lr_output_message(lr_eval_string(    "***** Time = {ParamTime},
                Temperature =  {ParamTemp}, "
                   "Humidity = {ParamHumid},
                Conditions = {ParamCond} *****"));


 // Generate readable forecast
    lr_save_string(lr_eval_string("\r\n\r\n*** Weather Forecast for {ParamCity},
             {ParamState} ***\r\n"
          "\tTime: {ParamTime}\r\n"
          "\tTemperature: {ParamTemp} deg. Fahrenheit\r\n"
          "\tHumidity: {ParamHumid}\r\n"
          "\t{ParamCond} conditions expected\r\n"
          "\r\n"),
          "ParamForecast"
    );
    // Save soap template into parameter
    lr_save_string(pSoapTemplate, "ParamSoap");


// Insert request body into SOAP template
    lr_xml_insert("Xml={ParamSoap}",
         "ResultParam=ParamRequest",
         "Query=Body/SendMail",
         "position=child",
         "XmlFragment="
             "<FromAddress>taurus@merc-int.com</FromAddress>"
             "<ToAddress>support@merc-int.com</ToAddress>"
             "<ASubject>Weather Forecast</ASubject>"
             "<MsgBody/>",
     LAST
    );
//
//"<soap:Envelope xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">"
//  "<soap:Body>"
//      "<SendMail xmlns=\"urn:EmailIPortTypeInft-IEmailService\"/>"
//         "<FromAddress>taurus@merc-int.com</FromAddress>"
//         "<ToAddress>support@merc-int.com</ToAddress>"
//         "<ASubject>Weather Forecast</ASubject>"
//         "<MsgBody/>"
//      "</SendMail>"
//   "</soap:Body>"
//"</soap:Envelope>";
//
    // Insert actual forecast text
    lr_xml_set_values("Xml={ParamRequest}",
                       "ResultParam=ParamRequest",
                       "Query=Body/SendMail/MsgBody",
                       "ValueParam=ParamForecast",
                      LAST);
```

```
// Add header for SOAP
 web_add_header("SOAPAction", "urn:EmailIPortTypeInft-IEmailService");
// Get response body
 web_reg_save_param("ParamXml", "LB=", "RB=", "Search=body", LAST);
// Send forecast to recipient, using SOAP request
 web_custom_request("web_custom_request",
   "URL=http://webservices.matlus.com/scripts/emailwebservice.dll/soap
        /IEmailservice",
        "Method=POST",
        "TargetFrame=",
         "Resource=0",
         "Referer=",
        "Body={ParamRequest}",
        LAST);
    // Verify that mail was sent
    lr_xml_find("Xml={ParamXml}",
                   "Query=Body/SendMailResponse/return",
                   "Value=0",
                LAST
    );
    return 0;
}
```

Verwenden von Ergebnisparametern

Einige der **lr_xml**-Funktionen geben einen Ergebnisparameter zurück, beispielsweise **ResultParam**. Dieser Parameter enthält die resultierenden XML-Daten nach Ausführung der Funktion. Die Ergebnisparameter sind über die Parameterliste im Dialogfeld **Parameter auswählen oder erstellen** verfügbar.

Beispielsweise enthält **ResultParam** für **lr_xml_insert** die gesamten XML-Daten, die aus dem Einfügen des neuen XML-Fragments resultieren.

Sie können die Ergebnisparameter als Eingabe für andere XML-bezogene Funktionen verwenden, wie z. B. Webdienstaufrufe. Bei der Wiedergabe zeichnet VuGen den Wert des Ergebnisparameters auf. In einem späteren Schritt können Sie diese Werte als ein Eingabeargument verwenden.

Die Funktionen, die Ergebnisparameter unterstützen, sind **lr_xml_insert**, **lr_xml_transform**, **lr_xml_replace**, **lr_xml_delete** und **lr_xml_set_values**.

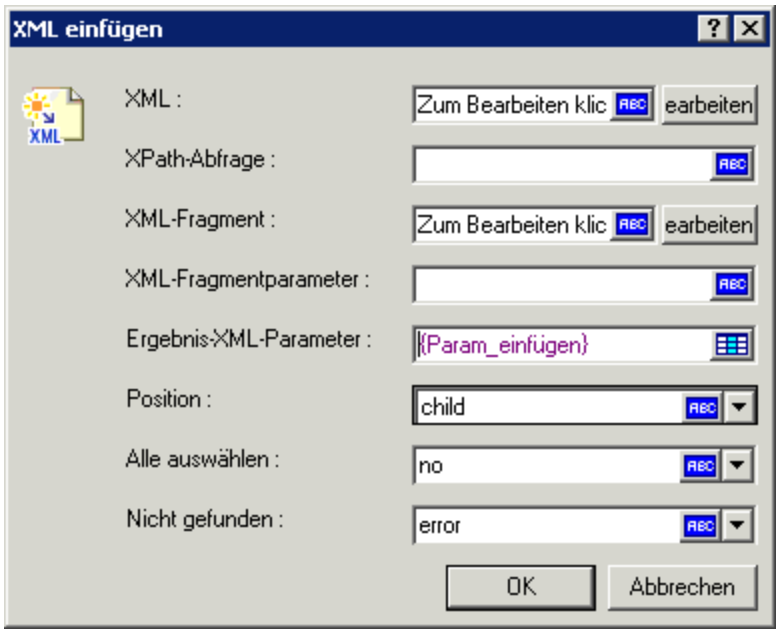
Die folgenden Funktionen speichern Werte in einem anderen Parameter als dem Ergebnisparameter **resultParam**: **lr_xml_get_values** speichert Werte in **ValueParam** und **lr_xml_extract** speichert Werte in **XMLFragmentParam**. Diese Werte sind auch für die Parametersubstitution verfügbar.

Verwenden des Ergebnisparameters als Eingabe

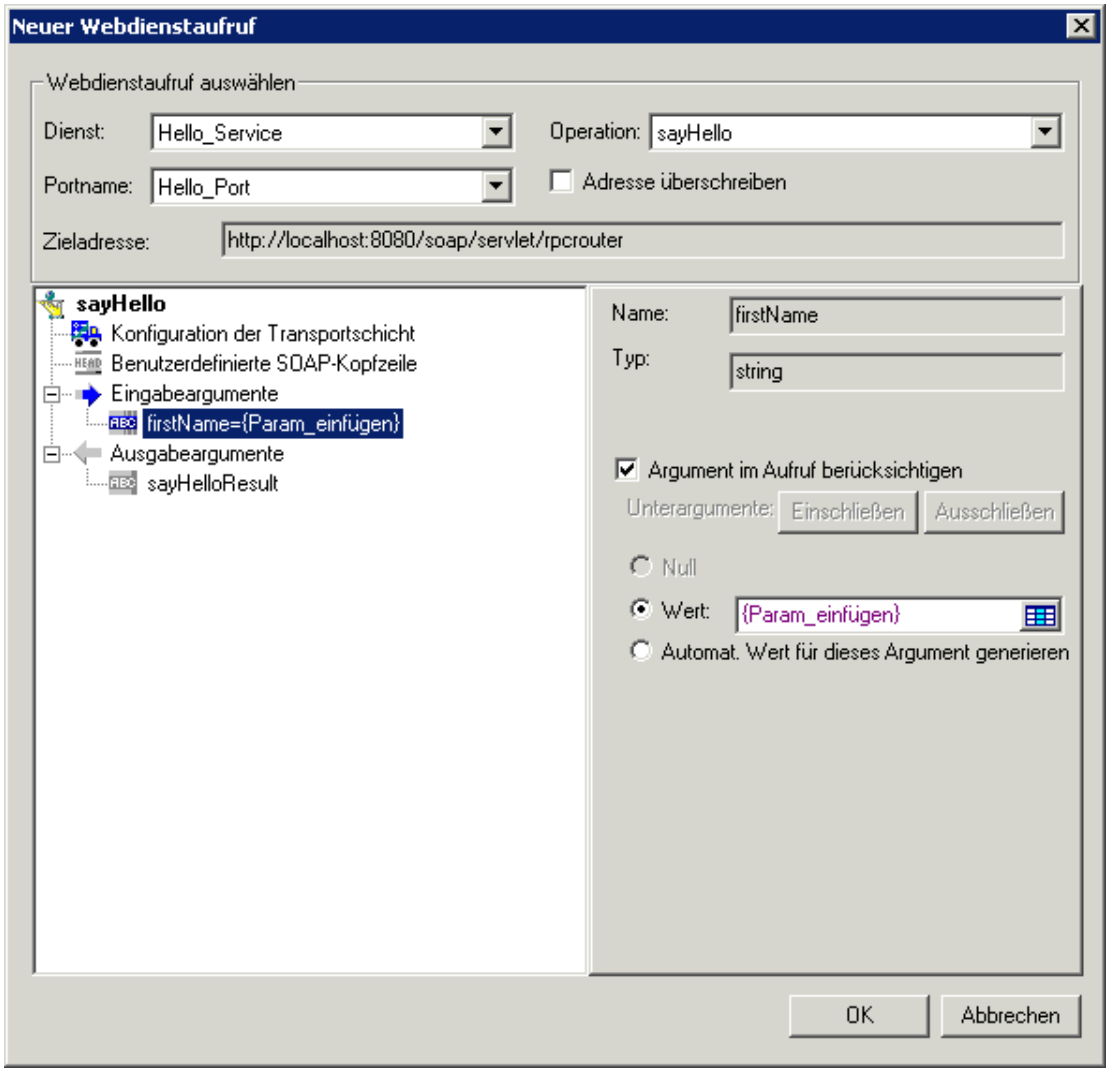
1. Doppelklicken Sie im **Schrittenavigator** auf einen XML-Schritt, um seine Eigenschaften

anzuzeigen.

2. Geben Sie im Feld **Ergebnis-XML-Parameter** einen Namen für den Ergebnis-XML-Parameter (oder **ValueParam** und **XMLFragmentParam**) an.



3. Referenzieren Sie den Parameternamen als Eingabeargument.



Weitere Informationen finden Sie unter "Dialogfeld "Neuer Webdienstaufruf"" auf Seite 950.

Fehlerbehebung und Einschränkungen - Programmierung

Framework 4,5 für .NET-Skripts

Problem: Die Ausführung eines in VuGen aufgezeichneten .NET-Skripts schlägt in Visual Studio fehl, da die zugehörige DLL nicht gefunden wird.

Schritte für die Behebung: Wenn das Skript mit einer .NET 4.5-Anwendung aufgezeichnet wurde, generieren Sie das Skript mit Framework Version 4.5 in Visual Studio neu.

Framework 4.5 und Visual Studio 2010

Problem: Kompilierung schlägt für alle C+ + -Projekte auf Computern mit Framework 4.5 fehl.

Schritte für die Behebung: Installieren Sie Visual Studio 2010 SP1.

Framework 3.5 für .NET-Skripts

Problem: In Visual Studio mit Framework 3.5 erstellte .NET-DLLs können möglicherweise nicht ausgeführt werden.

Schritte für die Behebung: Fügen Sie folgende Elemente zur Datei **<app>.config** hinzu: (Wenn keine solche Datei vorhanden ist, erstellen Sie eine.)

```
<configuration>

<startup>

    <supportedRuntime version="v4.0" />

   </startup>

</configuration>
```

Visual Studio Addin mit C Vuser-Skripts

Problem:C#-Skripts in Visual Studio 2010, Erstellung von .NET Assemblys, die vor der Ausführung im System registriert werden müssen. Registrierung der .NET-Assembly erfordert Administratorrechte.

Schritte für die Behebung: Es gibt 2 Möglichkeiten, um eine .NET-Assembly vor der Testausführung zu registrieren:

1. Führen Sie Visual Studio "Als Administrator" aus und geben Sie administrative Anmeldeinformationen ein, wenn Sie dazu aufgefordert werden. Vergewissern Sie sich in den Projekteinstellungen, dass die Option **Für COM-Interop registrieren** aktiviert ist (**Projekt > Eigenschaften > Build > Für COM-Interop registrieren**). Visual Studio registriert die Test-DLL bei jeder Generierung des Projekts automatisch als .NET-Assembly.

2. Führen Sie Visual Studio ohne Administratorrechte aus und registrieren Sie die Test-DLL nach der Generierung manuell. Führen Sie dazu die Windows-Konsole mit Administratorberechtigungen und danach den folgenden Befehl aus:
   %WINDIR%\Microsoft.NET\Framework\v4.0.30319\RegAsm.exe <TargetDLLwithPath**> /codebase** (ignorieren Sie die ausgegebene Warnung).
   Sie könnten beispielsweise <TargetDLLwithPath> durch "c:\users\qatest\documents\visual studio 2010\Projects\LoadRunnerUser1\LoadRunnerUser1\LoadRunnerUser1.dll" ersetzen. Sie müssen die DLL jedoch nicht nach jedem Build registrieren. Es reicht aus, sie einmal zu registrieren, nachdem Sie Änderungen an den in der DLL definierten COM-Schnittstellen vorgenommen haben.

Vergewissern Sie sich bei beiden Optionen, dass UAC aktiviert ist. Andernfalls werden bei der Ausführung von Visual Studio oder der Konsole "Als Administrator" keine administrativen Anmeldeinformationen abgefragt und die Anwendungen werden nur mit eingeschränkten Benutzerrechten ausgeführt.

Unterstützung für nicht-englische Sprachen

Übersicht über die Unterstützung für nicht-englische Sprachen

VuGen unterstützt mehrsprachige Umgebungen, die Ihnen bei der Erstellung und Ausführung von Skripts auf Computern, für die die jeweilige Landessprache konfiguriert wurde, die Verwendung anderer Sprachen als Englisch ermöglichen.

Beim Arbeiten mit anderen Sprachen als Englisch muss zunächst sichergestellt werden, dass VuGen die Codierung des Textes beim Aufzeichnen und Wiedergeben erkennt. Die Codierung gilt für alle vom Skript verwendeten Texte. Dies umfasst Text in HTTP-Kopfzeilen und HTML-Seiten für Web-Vuser, Daten in Parameterdateien und andere Texte.

Windows 2000 und höher ermöglicht es Ihnen, Dateien mit einer spezifischen Codierung direkt vom Editor aus zu speichern: ANSI, Unicode, Unicode Big Endian oder UTF-8.

Standardmäßig arbeitet VuGen mit der Codierung des lokalen Computers (ANSI). Manche Server, die mit anderen Sprachen als Englisch arbeiten, setzen die Verwendung der UTF-8-Codierung voraus. Um mit diesen Servern zu arbeiten, müssen Sie in den erweiterten Aufzeichnungsoptionen angeben, dass Ihr Skript UTF-8-Codierung erfordert.

Sprache in der Anforderungskopfzeile

Vor Ausführung eines Webskripts können Sie die Anforderungskopfzeile der Seite auf Ihre aktuelle Sprache festlegen. In den Internetprotokoll-Laufzeiteinstellungen setzen Sie die Sprache auf die *Accept-Language*-Kopfzeile. Diese Kopfzeile stellt dem Server eine Liste aller akzeptierten Sprachen bereit.

Wählen Sie zum Festlegen dieses Werts **Wiedergeben > Laufzeiteinstellungen > Internetprotokoll > Einstellungen > Erweitert > Optionen > Accept-Language-Anforderungskopfzeile** aus und legen Sie dann die gewünschte Sprache fest.

Informationen zur Benutzeroberfläche finden Sie unter "Internetprotokoll > Einstellungen (Knoten)" auf Seite 371.

Umwandeln des Codierungsformats einer Zeichenfolge

Über die Funktion **lr_convert_string_encoding** mit folgender Syntax können Sie eine Zeichenfolge manuell von einer Codierung in eine andere konvertieren (UTF-8, Unicode oder Codierung der Systemsprache des Computers):

```
lr_convert_string_encoding(char * sourceString, char * fromEncoding, char * toEn
coding, char * paramName)
```

Die Funktion speichert die Ergebniszeichenfolge (einschließlich ihrer abschließenden NULL) im dritten Argument *paramName*. Bei Erfolg wird 0 und bei einem Fehler wird -1 zurückgegeben.

Das Format für die Argumente **fromEncoding** und **toEncoding** ist:

LR_ENC_SYSTEM_LOCALE	NULL
LR_ENC_UTF8	"utf-8"
LR_ENC_UNICODE	"ucs-2"

Im folgenden Beispiel konvertiert **lr_convert_string_encoding** "Hello world" aus der Systemsprache in Unicode.

```
Action()
{
    int rc = 0;
    unsigned long converted_buffer_size_unicode = 0;
    char               *converted_buffer_unicode = NULL;
    rc = lr_convert_string_encoding("Hello world", NULL, LR_ENC_UNICODE, "string
InUnicode");
    if(rc < 0)
    {
        // error
    }
return 0;
}
```

Das Ausgabefenster im Wiedergabeprotokoll zeigt die folgenden Informationen an:

```
Output.
Starting action Action.
Action.c(7): Notify: Saving Parameter "stringInUnicode = H\x00e\x00l\x00l\x00o\x
00 \x00w\x00o\x00r\x00l\x00d\x00\\x00"
Ending action Action.
```

Das Ergebnis der Konvertierung wird im Argument *paramName* gespeichert.

Umwandeln des Codierungsformats von Parameterdateien

Die Parameterdatei enthält die Daten für Parameter, die im Skript definiert wurden. Diese im Skriptordner gespeicherte Datei hat die Erweiterung **.dat*. Beim Ausführen eines Skripts verwenden Vuser die Daten zum Ausführen von Aktionen mit unterschiedlichen Werten.

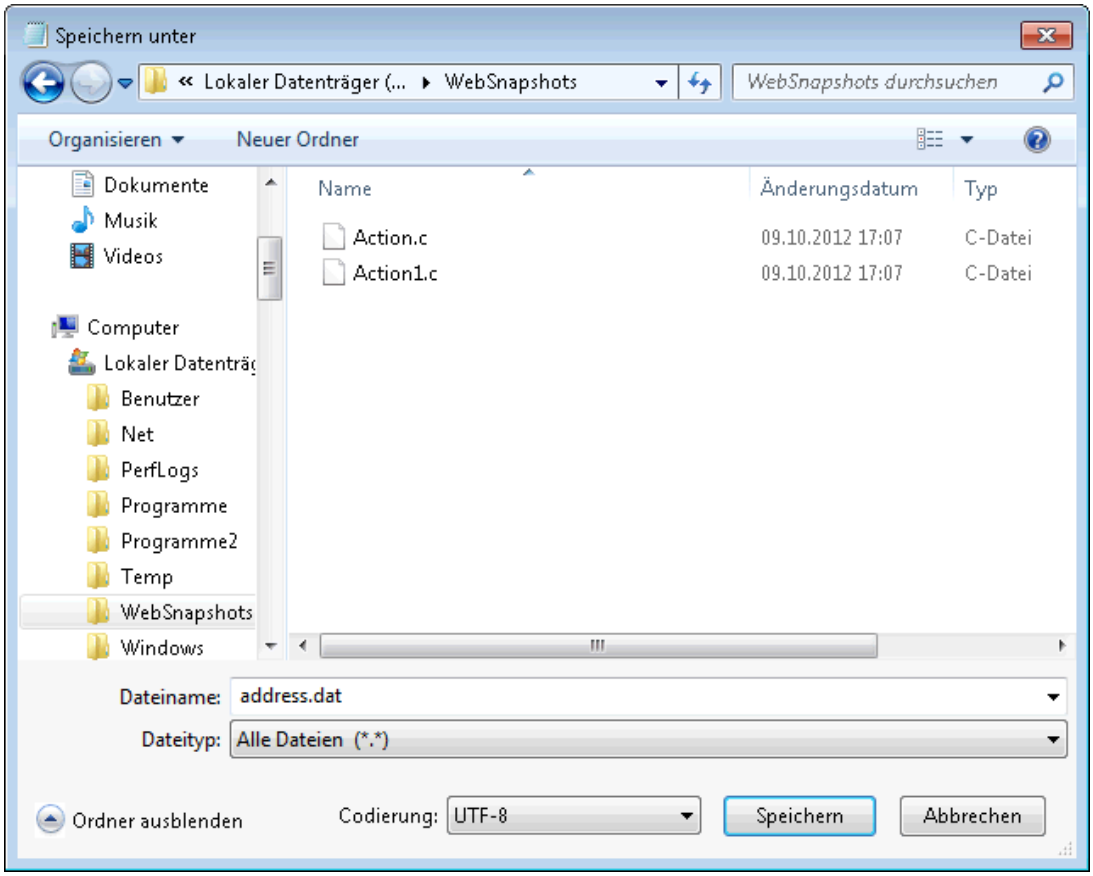
Standardmäßig speichert VuGen die Parameterdatei mit der Codierung Ihres Computers. Beim Arbeiten mit anderen Sprachen als Englisch müssen Sie die Parameterdatei jedoch möglicherweise

in UTF-8 konvertierten, wenn der Server die Zeichenfolge in UTF-8 erwartet. Sie können dies direkt über Editor ausführen, sofern Sie Windows 2000 oder höher verwenden.

Anwenden der UTF-8-Codierung auf eine Parameterdatei

1. Wählen Sie **Vuser > Parameterliste** und zeigen Sie die Parametereigenschaften an.
2. Suchen Sie die Parameterdatei im rechten Bereich im Feld **Dateipfad**.
3. Klicken Sie bei angezeigter Parametertabelle auf **Mit Editor bearbeiten**. Die Parameterdatei wird als csv-Datei im Editor geöffnet.
4. Wählen Sie im Feld **Dateityp** die Option *Alle Dateien* aus.

   Wählen Sie im Feld **Codierung** die Codierung *UTF-8* aus.



5. Klicken Sie auf **Speichern**. Editor fordert Sie auf, das Überschreiben der vorhandenen Parameterdatei zu bestätigen. Klicken Sie auf **Ja**.

   VuGen erkennt nun die Parameterdatei als UTF-8-Text, obwohl sie weiterhin mit regulären Zeichen angezeigt wird.

Aufzeichnen von Webseiten mit Fremdsprachen

Beim Arbeiten mit Web- oder anderen Internetprotokollen können Sie für die Aufzeichnung die Codierung des Webseitentextes angeben. Die Sprache der aufgezeichneten Site muss mit der Betriebssystemsprache übereinstimmen. Es ist nicht möglich, verschiedene Codierungen in einer Aufzeichnung zu kombinieren, wie z. B. **UTF-8** mit **ISO-8859-1** oder **shift_jis**.

Im Rahmen dieser Aufgabe wird beschrieben, wie Sie mit VuGen Webseiten aufzeichnen, die Fremdsprachen enthalten.

Automatisches Aufzeichnen von fremdsprachigen Webseiten

Um als eine nicht-englische Webseite erkannt zu werden, muss die Seite den Zeichensatz in der HTTP-Kopfzeile oder im HTML-Metatag angeben. Andernfalls erkennt VuGen die EUC-JP-Codierung nicht und die Website wird nicht korrekt aufgezeichnet. Damit VuGen Anforderungen in anderen Sprachen als Englisch als **EUC-JP** oder **UT-8** aufzeichnet, wählen Sie **Aufzeichnen > Aufzeichnungsoptionen > HTTP-Eigenschaften > Erweitert > Zeichensatz unterstützen** aus und aktivieren dann die entsprechende Option im Dialogfeld "Aufzeichnungsoptionen", Knoten **HTTP-Eigenschaften: Erweitert**. Informationen zur Benutzeroberfläche finden Sie unter "HTTP > Erweitert (Knoten)" auf Seite 202.

Beachten Sie, dass VuGen durch Auswahl der Option **EUC-JP** oder **UTF-8** in den Aufzeichnungsoptionen gezwungen wird, eine Webseite mit der ausgewählten Codierung aufzuzeichnen, auch wenn diese eine andere Codierung verwendet. Wenn beispielsweise eine nicht-EUC-codierte Webseite als EUC-JP aufgezeichnet wird, wird das Skript nicht korrekt abgespielt.

Manuelles Aufzeichnen von fremdsprachigen Webseiten

Mit der Funktion **web_sjis_to_euc_param** können Sie manuell vollständige Unterstützung für das Aufzeichnen und Wiedergeben von in EUC-JP codierten HTML-Seiten hinzufügen. Dadurch kann VuGen auch EUC-codierte japanische Zeichen in Vuser-Skripts korrekt anzeigen.

Wenn Sie **web_sjis_to_euc_param** verwenden, zeigt VuGen den Wert des Parameters im Ausführungsprotokoll mit EUC-JP-Codierung an. Wenn Sie beispielsweise die Funktion **web_find** wiedergeben, zeigt VuGen die codierten Werte an. Dazu gehören Zeichenfolgenwerte, die über die Funktion **web_sjis_to_euc_param** in EUC konvertiert wurden, oder Parametersubstitution, wenn sie über **Laufzeiteinstellungen > Protokoll > Erweitertes Protokoll** aktiviert wurde.

Fremdsprachen - Fehlerbeseitigung und Beschränkungen

Dieser Abschnitt enthält Optionen für die Fehlerbehebung und beschreibt die Einschränkungen im Umgang mit fremdsprachlichen Webseiten.

Skript/Szenarionamen

- Beim Aufzeichnen im COM-, FTP-, IMAP-, SMTP-, POP3- oder REAL-Modus ist die Länge des Skriptnamens auf 10 Multibytezeichen beschränkt (21 Byte).

- Skriptnamen und die zugehörigen Pfade dürfen nur Zeichen aus dem englischen Zeichensatz enthalten.
- Name und Pfad eines Szenarios dürfen keine Multibytezeichen enthalten.
- Beim Aufzeichnen im COM-, FTP-, IMAP-, SMTP-, POP3- oder REAL-Modus ist die Länge des Skriptnamens auf 10 Multibytezeichen beschränkt (21 Byte).
- Skriptnamen und die zugehörigen Pfade dürfen nur Zeichen aus dem englischen Zeichensatz enthalten.
- Argument- und Parameternamen sollten in Englisch angegeben werden.

Browserkonfiguration

Werden während der Aufzeichnung nicht-englische Zeichen im Skript als maskierte Hexadezimalzahlen angezeigt (beispielsweise wird die Zeichenfolge "" =;" zu "%DC%26"), können Sie dies korrigieren, indem Sie Ihren Browser so konfigurieren, dass er keine URLs in UTF-8-Codierung sendet. Wählen Sie im Internet Explorer **Extras > Internetoptionen** aus und klicken Sie auf die Registerkarte **Erweitert**. Deaktivieren Sie die Option **UTF-8-URLs senden** im Abschnitt **International**.

Weitere Informationen zu **web_sjis_to_euc_param** enthält die Function Reference von LoadRunner.

Protokollbeschränkungen

SMTP: Wenn Sie mit dem SMTP-Protokoll über MS Outlook oder Outlook Express arbeiten, wird der in einem Vuser-Skript aufgezeichnete japanische Texte nicht korrekt angezeigt. Das Skript wird jedoch korrekt aufgezeichnet und abgespielt.

ALM-Integration (Application Lifecycle Management)

Zum Öffnen eines in einem Application Lifecycle Management-Projekt gespeicherten Skripts über VuGen oder eines in einem ALM-Projekt gespeicherten Szenarios über den Controller fügen Sie eine neue Testreihe mit der Bezeichnung "Default" (in Englisch) zum ALM-Projekt hinzu.

ContentCheck in mehrsprachigen Umgebungen

- Diese Version unterstützt ContentCheck-Regeln in Französisch, Deutsch, Spanisch und Italienisch. Entsprechend der Spracheinstellungen des Systems muss die richtige Sprachdatei installiert werden.
- Die passende Sprachdatei kann auch aus dem Verzeichnis ..\lrunner\MSI\setup\international\<lang>\dat\LrwiAedInstallation.xml in das Verzeichnis <LoadRunner>\dat kopiert werden.

Sprachpakete

- **Deinstallieren von LoadRunner:** Beim Deinstallieren von LoadRunner unter einem chinesischen Betriebssystem tritt ein Fehler auf, wenn der Pfad der LoadRunner-Installation chinesische Zeichen enthält. **Problemumgehung:** xxxx

- **Aufzeichnungsfunktionen:** LoadRunner kann ein Vuser-Skript für bestimmte Protokolle nicht aufzeichnen, wenn LoadRunner unter einem chinesischen Betriebssystem installiert wurde und der Installationspfad chinesische Zeichen enthält. **Problemumgehung:** xxxx

- LoadRunner User Interface Pack. Bei der Installation des Sprachpakets wird möglicherweise eine Warnmeldung angezeigt, die Sie darauf hinweist, dass das Startprogramm von HP LoadRunner bereits verwendet wird.
  Problemumgehung: Klicken Sie auf **Fortsetzen**, um die Installation fortzusetzen.

- Das Sprachpaket für .NET Framework muss installiert werden, damit die lokalisierten Zeichenfolgen angezeigt werden.

- Tutorial-Skripts. Auch nach Installation des Sprachenpakets sind alle Sitzungen und Skripts aus dem Verzeichnis \HP\LoadRunner\tutorial immer noch in Englisch.

- LoadRunner User Interface Pack. Wenn Sie ein LoadRunner User Interface Pack verwenden, sollten Sie dieses installieren, bevor Sie LoadRunner zum ersten Mal ausführen.

- Menüs und Symbolleisten. Wenn Sie ein LoadRunner User Interface Pack installieren, nachdem Sie LoadRunner bereits schon einmal verwendet haben, werden die Menüs und Symbolleisten möglicherweise nicht in der lokalisierten Version angezeigt.
  Problemumgehung: Schließen Sie LoadRunner, und löschen Sie den folgenden Ordner aus der Registrierung: HKEY_CURRENT_USER\Software\<Verzeichnisname>, wobei **<Verzeichnisname>** für das Laufwerk steht, auf dem LoadRunner installiert wurde.
  Wenn LoadRunner beispielsweise auf Laufwerk C installiert wurde, so lautet der Eintrag in der Registrierung entsprechend: HKEY_CURRENT_USER\Software\C. Starten Sie LoadRunner neu.

- Berichtsvorlagen in Analysis. Wenn Sie ein LoadRunner User Interface Pack installieren, nachdem Sie LoadRunner bereits schon einmal verwendet haben, werden die Berichtsvorlagen in Analysis (**Berichte > Berichtsvorlagen**) möglicherweise nicht in der lokalisierten Version angezeigt.
  Problemumgehung: Schließen Sie LoadRunner, und kopieren Sie die Dateien aus <CD-Stammverzeichnis des LoadRunner User Interface Pack>\Reporting nach <LoadRunner-Installationsverzeichnis>\bin\dat\Reporting. Starten Sie LoadRunner neu.

Nicht-lokalisierte Versionen von LoadRunner unter Betriebssystemen in der jeweiligen Landessprache

- **Sprachunterstützung.** LoadRunner unterstützt nur Englisch und die jeweilige Sprache des Betriebssystems auf dem Computer. Wenn Sie beispielsweise ein japanisches Windows XP-Betriebssystem verwenden, können Sie mit LoadRunner in Japanisch und Englisch arbeiten.

- **Installationspfad.** Der Pfad, in dem sich die LoadRunner-Installationsdateien befinden, und der Pfad des LoadRunner-Installationsverzeichnisses dürfen nur englische Zeichen enthalten.

- **Diagnostics-Add-In.** Für die Verwendung des Diagnostics-Add-in mit Controller auf einem Computer mit einem nicht-englischen Betriebssystem muss das Hotfix "Diagnostics_9.0_8.0_

LR_Addin_QCCR1I52206" installiert werden. Wenden Sie sich an den HP Customer Support, wenn Sie weitere Unterstützung benötigen.

- ..**.NET Framework 3.5-Fehler.** Die Installation von LoadRunner auf einem Computer mit lokalisiertem Betriebssystem kann zu einem Fehler bei der Installation des .NET Framework 3.5 führen. Sie werden dann aufgefordert, die Installation zu beenden. Die Ursache ist, dass bei der .NET 3.5 Framework-Installation versucht wird, das Framework-Sprachpaket herunterzuladen, was zu einem Fehler führt.
  Problemumgehung: Beenden Sie die LoadRunner-Installation gemäß den Anweisungen des Installationsassistenten, und starten Sie sie dann erneut.

- **Onlinehilfe.** Die Suchfunktion kann möglicherweise nicht ordnungsgemäß für Begriffe mit chinesischen/japanischen Zeichen genutzt werden (ausgenommen Katakana in normaler Breite).
  Problemumgehung: Fügen Sie im Suchbegriff nach jedem Zeichen ein Leerzeichen halber Breite hinzu.

- **Onlinehilfe.** Installieren Sie die aktuelle JRE, um die Leistung der Onlinehilfe zu optimieren.

- **Japanische Zeichen in Web-Skripts (HTTP/HTML).** Wenn Sie die erweiterte Aufzeichnungsoption zur Angabe der Codierung der Anwendung verwenden und die Anwendung auf den einzelnen Seiten unterschiedliche Zeichencodierungen verwendet, ist es möglich, dass im Aufzeichnungsprotokoll oder im Skript ungültige japanische Zeichen angezeigt werden. Dies verursacht keine Fehler bei der Wiedergabe des Skripts.

- **Geschützte Leerzeichen** in Webprotokollen für fernöstliche Sprachen. Geschützte Leerzeichen (`\xA0' etc.) können in einigen fernöstlichen Zeichensätzen nicht dargestellt werden (in diesen Sprachen wird dieses Zeichen als führendes Byte gelesen). Diese geschützten Leerzeichen werden stattdessen bei der Generierung des Skriptcodes und beim Wiedergeben in normale Leerzeichen (` `, `\x20' etc.) umgewandelt. Dies kann zu Problemen bei der Wiedergabe führen, wie z. B. nicht übereinstimmende Längen durch das Löschen mehrerer normaler Leerzeichen.
  Problemumgehung: Fügen Sie dem Skript Leerzeichen hinzu, oder entfernen Sie Leerzeichen aus dem Skript, damit der Textvergleich erfolgreich durchgeführt werden kann, oder verwenden Sie reguläre Ausdrücke, um das Problem zu vermeiden.

- **Eigenständige Installationen.** Die Installationsoberflächen für die eigenständigen Installationen von VuGen und Analysis wurden nicht lokalisiert und stehen nur auf Englisch zur Verfügung.

- **Eigenschaften von Flex- oder AMF-Aufrufen.** Die Darstellung von Multibyte-Symbolen in den Eigenschaften von Flex- oder AMF-Aufrufen ist in der Skripttextansicht fehlerhaft.

- **rdp_type** Die Funktion **rdp_type** unterstützt keine Zeichen in der systemeigenen Sprache für Aufzeichnung und Wiedergabe.

- **Transaktionsnamen:** Koreanische Zeichen werden in Transaktionsnamen nicht unterstützt.

- **Wortvervollständigung.** Die Wortvervollständigung funktioniert nicht, wenn Windows für die Verwendung der Tastenkombination STRG+LEERTASTE konfiguriert ist. Dies ist der Fall bei Verwendung einer chinesischen Tastatur.
  Problemumgehung: Wählen Sie im Menü **Bearbeiten** den Befehl **Wort vervollständigen** aus. Versierte Benutzer können STRG+LEERTASTE für chinesische Tastaturen durch Festlegen folgender Registrierungsschlüssel deaktivieren:

  - [HKEY_CURRENT_USER\Control Panel\Input Method\Hot Keys\00000010]
    "Key Modifiers"=hex:00,c0,00,00
    "Target IME"=hex:00,00,00,00
    "Virtual Key"=hex:ff,00,00,00

  - [HKEY_CURRENT_USER\Control Panel\Input Method\Hot Keys\00000070]
    "Key Modifiers"=hex:00,c0,00,00
    "Target IME"=hex:00,00,00,00
    "Virtual Key"=hex:ff,00,00,00

- **ODBC-Protokoll und Oracle 2-Tier-Protokoll.** Wenn Sie in VuGen ein Skript mithilfe des ODBC-Protokolls oder des Oracle 2-Tier-Protokolls aufzeichnen, kann es zu einem Absturz kommen, wenn Sie die Aufzeichnung beenden, während die zu testende Anwendung noch geöffnet ist.
  Problemumgehung: Schließen Sie LoadRunner, und öffnen Sie die Datei **<LoadRunner-Installationsverzeichnis>\dat\protocols\options\script\general.opt** in einem Texteditor. Kommentieren Sie die folgende Zeile aus, indem Sie ein Semikolon am Anfang der Zeile einfügen: Option=DumpProcesses Auskommentiert sieht die Zeile dann folgendermaßen aus: ;Option=DumpProcesses

- **PDF-Berichte.** Bei PDF-Berichten, die in Analysis generiert werden, sind Zeichen, die nicht im englischen Zeichensatz vorkommen, möglicherweise unleserlich.
  Problemumgehung: Ändern Sie vor dem Generieren des PDF-Berichts die Schriftart in der verwendeten Berichtsvorlage.

  a. Wählen Sie **Berichte > Berichtsvorlagen** aus.

  b. Wählen Sie die gewünschte Vorlage aus.

  c. Wählen Sie im Abschnitt **Detaillierter Bericht** die Registerkarte **Format** aus.

  d. Wählen Sie für jedes Element der Benutzeroberfläche in der Liste eine Schriftart aus, welche die Sprache unterstützt, in welcher der Bericht abgefasst ist.

Weitere Komponenten

Sie können zusätzliche Komponenten installieren, die erweiterte Funktionen für die Arbeit mit LoadRunner bereitstellen. Sie finden diese Komponenten in den folgenden Speicherorten:

- Im Ordner **Additional Components** innerhalb des Stammordners der LoadRunner-Installations-DVD.
- **HP Software Support Online** (**Hilfe > HP Software Support-Website**). Suchen Sie nach der Anmeldung bei der Support-Website nach **LR 12.00 Additional Components**, und laden Sie die Datei, die alle zusätzlichen Komponenten enthält, herunter. Entpacken Sie die Datei und führen Sie die Installationsassistenten für die zusätzlichen Komponenten aus, die Sie installieren möchten.

Die folgende Tabelle unten zeigt, welche zusätzlichen Komponenten verfügbar sind und wo Sie die einzelnen Komponenten installieren sollten:

Ordner	Kompo-nente	Beschreibung	Instal-lieren auf...
Agent for Citrix Server	Setup-Datei	Installiert den Citrix Agent zum Verbessern der VuGen-Erkennung von Citrix-Clientobjekten während der Citrix-Protokollaufzeichnung und -wiedergabe. Installationsanweisungen finden Sie unter "Installieren des Citrix Server-Agenten" auf Seite 1112. Der Agent ermöglicht auch die Verwendung weiterer Citrix-API-Funktionen. Details finden Sie im LoadRunner Function Reference.	Citrix Server
Agent for Microsoft Terminal Server	Setup-Datei für MS Terminal Agent	Installiert ein Dienstprogramm, das den Aufzeichnungsmechanismus des RDP-Protokolls in VuGen verbessert. Installationsanweisungen finden Sie unter "Installieren des Microsoft Terminal Server-Agenten" auf Seite 1114.	RDP-Server
Assembly Crawler for Analysis API	Setup-Datei für Assembly Crawler Console	Installiert ein Befehlszeilendienstprogramm zur Erstellung einer .NET-Konfigurationsdatei für eine LoadRunner Analysis API-Anwendung. Weitere Informationen erhalten Sie, indem Sie über **Start > Dokumentation** (nicht verfügbar in der VuGen Standalone-Version) die **Analysis-API-Referenz** öffnen.	LoadRunner Analysis-Computer

Ordner	Kompo-nente	Beschreibung	Instal-lieren auf...
Host-ID-Generator	Host-ID-Generator-Tool, licidgenerator.exe	Öffnet das Host-ID-Generator-Dienstprogramm, das die Host-ID des Computers anzeigt. Dies ist beim Anfordern einer Lizenz nützlich. Weitere Informationen finden Sie unter LoadRunner License Utility.	LoadRunner Controller
HP Performance Validation SDK	Configuration Builder-Setup-Datei	Installiert den **Configuration Builder**, mit dem Sie ein benutzerdefiniertes Protokoll erstellen können. Weitere Informationen erhalten Sie, indem Sie nach der Installation über die LoadRunner-Gruppe den **Configuration Builder** öffnen und dann das Menü **Hilfe** aufrufen.	VuGen-Computer
IDE-Add-Ins	Add-In-Setup-Dateien für allgemeine Versionen von Visual Studio	Installiert Add-Ins für Visual Studio, mit denen Sie Vuser-Skripts in Ihrer Standardentwicklungsumgebung mithilfe der LoadRunner-API erstellen können. Diese Integration ermöglicht zum Testen der Funktionalität außerdem die Ausführung des Tests direkt in Visual Studio. Weitere Informationen finden Sie unter "Erstellen von Skripts in externen IDEs" auf Seite 1068.	Visual Studio-Computer mit VuGen
IDE-Add-Ins-Dev	Setup-Dateien für Entwickler-Add-Ins für Visual Studio 2010/2012 und Eclipse	Installiert Add-Ins für Visual Studio oder Eclipse, mit denen Sie NUnit- oder JUnit-Tests in Ihrer Standardentwicklungsumgebung mithilfe der LoadRunner-API erstellen können. Weitere Informationen finden Sie unter "Erstellen von Skripts in externen IDEs" auf Seite 1068.	Visual Studio- oder Eclipse-Computer mit VuGen
LRTCPDump	Ausführbare Datei über die Befehlszeile für Windows- und UNIX-Plattformen	Erstellt eine Trace-Datei mit Protokollen über den TCP-Verkehr im Netzwerk. Diese kann eine Alternative zur Wireshark verwendet werden. Weitere Informationen finden Sie unter "Erstellen eines Skripts durch Analysieren des Datenverkehrs" auf Seite 945.	Alle Computer mit WinPcap (siehe unten)

Ordner	Kompo-nente	Beschreibung	Instal-lieren auf...
Mobiler RemoteAgent	Ausführbare Dateien für mehrere Plattformen	Startet den Mongoose-Webserver zum Bereitstellen der Funktionalität für mobile Geräte. Weitere Informationen finden Sie unter "Aufzeichnen und Analysieren eines Skripts für mobile Anwendungen" auf Seite 774.	Mobiler AUT-Backend-Server
SAP-Tools	Ausführbare Dateien und Hilfedateien für SapSpy und VerifyScripting	• **SAPGUI Spy.** Bietet Unterstützung bei der Überprüfung der Hierarchie von GUI-Skriptobjekten in geöffneten **SAPGUI-Client für Windows**-Fenstern. • **SAPGUI Verify Scripting.** Überprüft, ob die SAPGUI Scripting API aktiviert ist. Weitere Informationen finden Sie unter "Konfigurieren der SAP-Umgebung" auf Seite 858.	VuGen-Computer mit SAPGUI-Client
Drittanbieter	Quelldateien	Der Ordner enthält den Quellcode für einige Softwarekomponenten von Drittanbietern, die in LoadRunner verwendet werden. In der Tabelle unten finden Sie Informationen zu einigen dieser Komponenten.	
Virtual Table Server	Setup-Datei	Diese Installation führt das VTS-Setup über einen Assistenten aus. Weitere Informationen finden Sie unter "Installation von VTS" auf Seite 1113.	Beliebiger Computer

Komponenten von Drittanbietern

Ordner	Kompo-nente	Beschreibung	Instal-lieren auf...
MQTester	ZIP-Datei mit der Setup-Datei für MQTester	Installiert den OpenText MQTester für HP LoadRunner. MQTester kann in LoadRunner integriert werden, sodass WebSphere MQ-basierte Systeme Last- und Stresstests unterzogen werden können.	LoadRunner- oder Lastgeneratorcomputer

Ordner	Kompo-nente	Beschreibung	Instal-lieren auf...
WinPcap	Setup-Datei	Installiert WinPcap, die Windows Packet Capture-Bibliotheken, mit denen Sie Netzwerkverkehr in einer Datei aufzeichnen können. Dies ist nützlich für die Erstellung eines Webdienste- oder Mobile HTTP-Vuser-Skripts auf der Grundlage des erfassten Datenverkehrs. Weitere Informationen finden Sie unter http://www.winpcap.org. Dies wird in Verbindung mit der zusätzlichen Komponente LRTCPDump verwendet.	Beliebiger Computer

Installieren des Citrix Server-Agenten

Die Installationsdatei für den Citrix Server-Agenten befindet sich auf der LoadRunner-Installations-DVD im Ordner **Additional Components\Agent for Citrix Server**.

Hinweis: Der Agent darf nur auf Citrix-Servern, nicht aber auf Lastgenerator-Computern installiert werden.

Wenn Sie eine neuere Version des Agenten installieren, müssen Sie darauf achten, zunächst die vorherige Version zu deinstallieren (siehe Anweisungen zum Entfernen weiter unten).

So installieren Sie den Citrix Server-Agenten:

1. Wenn auf Ihrem Server für die Installation von Software Administratorberechtigungen erforderlich sind, melden Sie sich als Administrator an.
2. Suchen Sie die Installationsdatei **Setup.exe** auf dem Installationsdatenträger des Produkts im Ordner **Additional Components\Agent for Citrix**.
3. Folgen Sie den Anweisungen des Installations-Assistenten.

Hinweis: Nach der Installation ist der Agent nur für Citrix-Sitzungen aktiv, die von LoadRunner aufgerufen wurden. Es können keine Citrix-Sitzungen ohne LoadRunner gestartet werden.

Wenn Sie den Agenten deaktivieren möchten, müssen Sie ihn deinstallieren.

So deinstallieren Sie den Citrix Server-Agenten:

1. Wenn auf Ihrem Server für die Entfernung von Software Administratorberechtigungen erforderlich sind, melden Sie sich als Administrator an.
2. Öffnen Sie auf dem Servercomputer in der Systemsteuerung die Option **Software**. Wählen Sie **HP Software Agent for Citrix Server** und klicken Sie auf **Ändern/Entfernen**.

Installation von VTS

Zwei Versionen von VTS sind verfügbar: 32-Bit und 64-Bit. Sie können 32-Bit-VTS sowohl auf 32-Bit als auch auf 64-Bit Betriebssystemen installieren; 64-Bit-VTS kann nur auf 64-Bit-Betriebssystemen installiert werden.

Hinweis: Installieren Sie die 64-Bit-VTS- und 32-Bit-VTS-Version nicht auf demselben Computer.

So installieren Sie VTS:

1. Führen Sie die entsprechende VTS-setup.exe-Datei aus [**Win32** oder **Win64**], die sich im Ordner **Additional Components\Virtual Table Server** auf dem LoadRunner-Installationsmedium befindet. Der VTS-Setup-Assistent wird gestartet und die Willkommenseite angezeigt.

2. Befolgen Sie die Online-Anweisungen zum Abschließen der VTS-Installation.

Hinweis: Am Ende der Installation wird ein VTS-Kontextmenü erstellt und auf dem Desktop hinzugefügt. Dieses Kontextmenü bietet Zugriff auf die VTS-Benutzeroberfläche auf dem lokalen Computer. Wenn Sie den Port, der für den Zugriff auf die VTS-Benutzeroberfläche verwendet wird, ändern, müssen Sie die **URL**-Eigenschaft des Kontextmenüs manuell aktualisieren. Weitere Informationen zum Ändern des Ports für den Zugriff auf die VTS-Benutzeroberfläche finden Sie unter **Konfigurieren von VTS** in der VTS-Online-Dokumentation.

Wenn Sie nicht auf die VTS-Benutzeroberfläche zugreifen können, stellen Sie sicher, dass der VTS-Dienst gestartet ist. Um den VTS-Dienst zu starten, öffnen Sie die Systemsteuerung und wählen Sie dann **Systeme** & **Sicherheit** > **Verwaltung** > **Dienste**. Klicken Sie mit der rechten Maustaste auf **VTS Service** und wählen Sie **Starten** aus.

Wenn Sie nicht auf die VTS-Benutzeroberfläche zugreifen können, stellen Sie sicher, dass der VTS-Dienst gestartet ist. Um den VTS-Dienst zu starten, öffnen Sie die Systemsteuerung und wählen Sie dann **Systeme & Sicherheit > Verwaltung > Dienste**. Klicken Sie mit der rechten Maustaste auf **VTS Service** und wählen Sie **Starten** aus.

Konfigurieren des VTS-Verwaltungsservers

Während der VTS-Installation, müssen Sie den Port festlegen, der für den Zugriff auf den VTS-Server für Verwaltungszwecke verwendet wird.

So konfigurieren Sie den VTS-Verwaltungsserver:

1. Beginnen Sie die VTS-Installation wie unter "Installing Additional Components" beschrieben. Während der VTS-Installation wird der Bildschirm **Configure VTS administration server** angezeigt.

2. Geben Sie im Feld **Admin UI server port** den Wert `4000` ein.

3. Klicken Sie auf **Next**, um die Installation fortzusetzen. Der Bildschirm zum Konfigurieren von VTS wird angezeigt.

4. Geben Sie den Speicherort der VTS-Datendatei an.

5. Stellen Sie sicher, dass das Kontrollkästchen **Start Virtual Table Server Automatically** aktiviert ist.

6. Klicken Sie auf **Next** und befolgen Sie die Anweisungen des Assistenten zum Abschluss der VTS-Installation.

Installieren des Microsoft Terminal Server-Agenten

Die Installationsdatei für den Microsoft Terminal Server-Agenten befindet sich auf dem Installationsdatenträger des Produkts im Ordner **Additional Components\Agent for Microsoft Terminal Server**.

Hinweis: Der Agent darf nur auf RDP-Servern, nicht aber auf Lastgenerator-Computern installiert werden.

Wenn Sie ein Upgrade des Agenten durchführen, vergewissern Sie sich vor der Installation, ob die Vorversion entfernt wurde (siehe nachfolgende Anweisungen zur Deinstallation).

So installieren Sie den Microsoft Terminal Server-Agenten:

1. Wenn auf Ihrem Server für die Installation von Software Administratorberechtigungen erforderlich sind, melden Sie sich als Administrator an.

2. Suchen Sie die Installationsdatei **Setup.exe** auf der LoadRunner-DVD im Ordner **Additional Components\Agent for Microsoft Terminal Server**.

3. Folgen Sie den Anweisungen des Installations-Assistenten.

Hinweis: Sie müssen die Aufzeichnungsoptionen festlegen, bevor Sie den Agenten verwenden und ein Vuser-Skript aufzeichnen können. Klicken Sie im Dialogfeld **Aufzeichnung starten** auf **Optionen**. Aktivieren Sie im Knoten **Erweiterte Codegenerierung** die Option **RDP-Agent verwenden**.

So deinstallieren Sie den Microsoft Terminal Server-Agenten:

1. Wenn auf Ihrem Server für die Entfernung von Software Administratorberechtigungen erforderlich sind, melden Sie sich als Administrator an.

2. Öffnen Sie auf dem Servercomputer in der Systemsteuerung die Option **Software**. Wählen Sie **HP Software Agent for Microsoft Terminal Server** und klicken Sie auf **Ändern/Entfernen**.

Fehlerbehebung und Einschränkungen – Zusätzliche Komponenten

In diesem Abschnitt werden die Fehlerbehebung und die für zusätzliche Komponenten geltenden Einschränkungen beschrieben.

Sichere Kanäle

- Sie können die Sicherheitseinstellungen auf Lastgeneratoren unter Linux, die die Verbindung mit Controller über rsh (Remote-Shell) herstellen, nicht mit dem Host Security Manager-Dienstprogramm aktualisieren.
- Sie können den Sicherheitsmodus eines Lastgenerators, der über eine Firewall verbunden ist, nicht mit dem Host Security Manager-Dienstprogramm aktivieren bzw. deaktivieren.
- Wenn der Lastgenerator über eine Firewall verbunden ist, ist die Kommunikation nur möglich, wenn die Sicherheitsmodi des Controllers und des Lastgenerators identisch sind.
- Wenn der Controller-Computer die Kommunikation über sichere Kanäle verwendet, darf der MI-Listener nicht auf demselben Computer wie der Controller installiert sein.

Lernprogramm zu HP LoadRunner

Das Lernprogramm zu HP LoadRunner können Sie ausdrucken und eigenständig absolvieren, um den Prozess von Lasttests zu durchlaufen und die LoadRunner-Testumgebung kennen zu lernen.

- Klicken Sie zum Aufrufen einer PDF-Version des Lernprogramms auf **Start > Alle Programme > HP Software > HP LoadRunner > Dokumentation > Lernprogramm > Lernprogramm – Druckversion**.

HP LoadRunner-Funktionsreferenz

Die HP LoadRunner-Funktionsreferenz beschreibt Funktionen, die in Vuser-Skripts in mehreren HP-Produkten verwendet werden können. Sie können mit unterstützten Protokollen in den Skripts verwendet werden, die in HP Virtual User Generator für die Verwendung in der Anwendungsverwaltung und in Lasttests verwaltet werden. Einige können in HP Service Test verwendet werden. Weitere Informationen zur Verwendung finden Sie in der Produktdokumentation.

Klicken Sie zum Aufrufen der Funktionsreferenz auf **Start > Alle Programme > HP Software > HP LoadRunner > Dokumentation > Funktionsreferenz**.

HP LoadRunner Data Format Extensions Developer Guide

HP Data Format Extensions Developer Guide unterstützt Entwickler beim Erstellen von DFE-DLLs, die es ermöglichen, HTTP-Rohdatenverkehr in ein strukturiertes und verwaltbares XML-Format zu konvertieren. Dies kann dann genauso verwaltet werden wie ein Skript, das auf einer Textwebseite basiert.

Klicken Sie zum Anzeigen dieses Handbuchs im PDF-Format auf **Start > Alle Programme > HP Software > HP LoadRunner > Dokumentation > Data Format Extensions Developer Guide** .