

HP Virtual User Generator

Windows オペレーティング・システム用

ソフトウェア・バージョン : 9.50

ユーザーズ・ガイド 第 I 巻 - VuGen の使用

製造部品番号 : T7182-99015

ドキュメント発行日 : 2009 年 1 月 (英語版)

ソフトウェア・リリース日 : 2009 年 1 月 (英語版)



利用条件

保証

HP の製品およびサービスの保証は、かかる製品およびサービスに付属する明示的な保証の声明において定められている保証に限ります。本ドキュメントの内容は、追加の保証を構成するものではありません。HP は、本ドキュメントに技術的な間違いまたは編集上の間違い、あるいは欠落があった場合でも責任を負わないものとします。

本ドキュメントに含まれる情報は、事前の予告なく変更されることがあります。

制限事項

本コンピュータ・ソフトウェアは、機密性があります。これらを所有、使用、または複製するには、HP からの有効なライセンスが必要です。FAR 12.211 および 12.212 に従って、商用コンピュータソフトウェア、コンピュータソフトウェアのドキュメント、および商用アイテムの技術データは、HP の標準商用ライセンス条件に基づいて米国政府にライセンスされています。

サードパーティ Web サイト

HP は、補足情報の検索に役立つ外部サードパーティ Web サイトへのリンクを提供します。サイトの内容と利用の可否は予告なしに変更される場合があります。HP は、サイトの内容または利用の可否について、いかなる表明も保証も行いません。

著作権

© 1992 - 2009 Hewlett-Packard Development Company, L.P.

商標

Adobe® および Acrobat® は、Adobe Systems Incorporated の商標です。

Intel®, Pentium® および Intel® Xeon™ は、米国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

Java™ は、Sun Microsystems, Inc. の米国商標です。

Microsoft®, Windows®, Windows NT® および Windows XP® は、Microsoft Corporation の米国登録商標です。

Oracle® は、カリフォルニア州レッドウッド市の Oracle Corporation の米国登録商標です。

Unix® は、The Open Group の登録商標です。

SlickEdit® は、SlickEdit Inc. の登録商標です。

文書の更新

本書のタイトル・ページには、次の識別情報が含まれています。

- ソフトウェアのバージョンを示すソフトウェア・バージョン番号
- ドキュメントが更新されるたびに更新されるドキュメント発行日
- 本バージョンのソフトウェアをリリースした日付を示す、ソフトウェア・リリース日付

最新のアップデートまたはドキュメントの最新版を使用していることを確認するには、<http://h20230.www2.hp.com/selfsolve/manuals> を参照します。

このサイトでは、HP Passport に登録してサインインする必要があります。HP Passport ID の登録は、以下の Web サイトにアクセスしてください。

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

または、HP Passport のログイン・ページの [**New users - please register**] リンクをクリックしてください。

適切な製品サポート・サービスに登録すると、更新情報や最新情報も入手できます。詳細については HP の営業担当にお問い合わせください。

サポート

HP ソフトウェアのサポート Web サイトは、次の場所にあります。

<http://www.hp.com/go/hpsoftwaresupport>

HP ソフトウェアのオンライン・サポートは、インタラクティブな技術サポート・ツールにアクセスするための効率的な手段を提供します。サポート・サイトを利用することで、次のようなことができるメリットがあります。

- 関心のある内容の技術情報の検索
- サポート・ケースおよび機能強化要求の提出および追跡
- ソフトウェア・パッチのダウンロード
- サポート契約の管理
- HP サポートの連絡先の表示
- 利用可能なサービスに関する情報の確認
- ほかのソフトウェア顧客との議論の開始
- ソフトウェアのトレーニングに関する調査と登録

ほとんどのサポート・エリアは、HP Passport ユーザとしての登録およびサインインが必要です。また多くは、サポート契約も必要です。アクセス・レベルの詳細情報については、**http://h20230.www2.hp.com/new_access_levels.jsp** を参照してください。

HP Passport ID の登録は、次の場所で行います。

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

目次

| | |
|------------------------------|----|
| 本書について..... | 13 |
| 本書の構成..... | 14 |
| 対象読者..... | 15 |
| LoadRunner オンライン・ドキュメント..... | 15 |
| その他のオンライン・リソース..... | 17 |
| ドキュメントのアップデート..... | 18 |

第 I 部 : VUGEN を使った作業

| | |
|---------------------------------------|----|
| 第 1 章 : Vuser スクリプトの開発 | 21 |
| Vuser の紹介..... | 22 |
| Vuser のタイプ..... | 24 |
| Vuser スクリプトの作成手順..... | 26 |
| HP LoadRunner ライセンス..... | 27 |
| LoadRunner と Service Test..... | 28 |
| 第 2 章 : VuGen の紹介 | 29 |
| VuGen について..... | 30 |
| VuGen の起動..... | 30 |
| コマンドのユーザ定義..... | 31 |
| VuGen 環境オプションについて..... | 34 |
| 環境オプションの設定..... | 36 |
| Vuser スクリプトの表示と変更..... | 36 |
| VuGen を使った Vuser スクリプトの実行..... | 51 |
| VuGen のコードについて..... | 52 |
| Vuser 関数に関するヘルプについて..... | 54 |
| 第 3 章 : プロトコル・アドバイザーの使用法 | 59 |
| プロトコル・アドバイザーについて..... | 59 |
| プロトコル・アドバイザーのワークフロー..... | 60 |
| 注意と制限事項..... | 63 |
| 第 4 章 : VuGen ワークフローの表示 | 65 |
| VuGen ワークフローの表示について..... | 65 |

| | |
|--------------------------------------|------------|
| タスク表示枠の表示 | 66 |
| ステップの記録 | 67 |
| スクリプトの検証 | 69 |
| スクリプトの拡張 | 70 |
| ロードの準備 | 76 |
| スクリプトの完了 | 77 |
| ワークフロー・ビューの復元 | 77 |
| 第 5 章：VuGen を使った記録 | 79 |
| VuGen を使った記録について | 80 |
| Vuser スクリプトのセクション | 80 |
| 仮想ユーザ・スクリプトの新規作成 | 82 |
| プロトコルの追加と削除 | 86 |
| 仮想ユーザ・カテゴリの選択 | 87 |
| ユーザ定義テンプレート | 88 |
| スクリプトの新規作成 | 90 |
| 既存のスクリプトを開く | 91 |
| アプリケーションの記録 | 92 |
| 記録セッションの終了と保存 | 96 |
| 記録ログの表示 | 97 |
| Zip ファイルの使用 | 99 |
| アクションのインポート | 100 |
| 認証情報の提供 | 101 |
| Vuser スクリプトの再生成 | 103 |
| 第 6 章：Vuser スクリプトの拡張 | 107 |
| Vuser スクリプトの拡張について | 108 |
| Vuser スクリプトへのトランザクションの挿入 | 110 |
| ランデブー・ポイントの挿入 (LoadRunner のみ) | 112 |
| Vuser スクリプトへのコメントの挿入 | 114 |
| Vuser 情報の取得 | 115 |
| 出力へのメッセージの送信 | 116 |
| 実行中の Vuser スクリプトのエラー処理 | 121 |
| Vuser スクリプトの同期化 | 122 |
| ユーザの思考遅延時間のエミュレート | 123 |
| コマンド・ライン引数の取り扱い | 124 |
| テキストの暗号化 | 125 |
| 手動でのパスワードのエンコーディング | 126 |
| スクリプトへのファイルの追加 | 127 |
| 第 7 章：ビジネス・プロセス・レポートの作成 | 129 |
| Word へのスクリプトのエクスポートについて | 129 |
| レポートの詳細の指定 | 130 |
| レポートの内容の指定 | 131 |

| | |
|---|-----|
| 第 8 章：ステートメントの相関 | 133 |
| ステートメントの相関について | 133 |
| C Vuser 用の相関関数の使用 | 135 |
| Java Vuser 用の相関関数の使用法..... | 137 |
| WDiff を使った Vuser スクリプトの比較 | 138 |
| 保存したパラメータの変更 | 140 |
| 第 9 章：スタンドアロン・モードでの Vuser スクリプトの実行 | 141 |
| スタンドアロン・モードでの Vuser スクリプトの実行について | 142 |
| VuGen での Vuser スクリプトの実行 | 142 |
| Vuser スクリプトの再生 | 146 |
| VuGen のデバッグ機能の使用 | 149 |
| Web Vuser スクリプトでの VuGen のデバッグ機能の使用 | 154 |
| VuGen ウィンドウを使った作業 | 155 |
| ファイル内の検索 | 156 |
| コマンド・プロンプトからの Vuser スクリプトの実行 | 158 |
| UNIX コマンド・ラインからの Vuser スクリプトの実行 | 159 |
| スクリプトのテストへの統合 | 161 |
| 第 10 章：テスト結果の表示 | 165 |
| テスト結果の表示について | 166 |
| [テスト結果] ウィンドウ | 167 |
| 結果の表示 | 170 |
| 結果ステップの検索 | 177 |
| 結果の印刷 | 178 |
| テスト結果のエクスポート | 181 |
| Quality Center への不具合の送信 | 182 |
| [テスト結果] ウィンドウから Quality Center への接続 | 183 |
| テスト結果の表示のカスタマイズ | 183 |
| Web サービス・レポートの表示 | 184 |
| レポートへのユーザ定義情報の送信 | 187 |
| 第 11 章：Quality Center を使ったスクリプト管理 | 189 |
| Quality Center を使ったスクリプト管理について | 189 |
| Quality Center の接続と切断 | 190 |
| Quality Center プロジェクトからスクリプトを開く | 194 |
| Quality Center プロジェクトへのスクリプトの保存 | 196 |
| スクリプトのバージョン管理 | 197 |
| 第 12 章：HP Performance Center によるスクリプトの管理 | 207 |
| HP Performance Center によるスクリプトの管理について | 208 |
| Performance Center での Vuser スクリプトの表示 | 208 |
| VuGen から Performance Center への接続 | 209 |
| Vuser スクリプトのアップロード | 210 |

| | |
|--------------------------|-----|
| Vuser スクリプトのダウンロード | 213 |
| ダウンロードしたスクリプトでの作業 | 216 |

第 II 部 : パラメータ

| | |
|---|-----|
| 第 13 章 : VuGen パラメータを使った作業 | 219 |
| VuGen パラメータの定義について | 220 |
| パラメータに関する制限 | 221 |
| パラメータの作成 | 223 |
| パラメータ・タイプについて | 226 |
| パラメータのプロパティの定義 | 228 |
| 既存のパラメータの使用 | 230 |
| パラメータ・リストの使用 | 233 |
| パラメータ化オプションの設定 | 235 |
| 第 14 章 : ファイル, テーブル, および XML パラメータ・タイプ | 239 |
| データ・ファイルまたはデータ・テーブルの選択または作成 | 240 |
| ファイル・タイプのパラメータのプロパティ設定 | 246 |
| テーブル・タイプのパラメータのプロパティ設定 | 248 |
| ファイル・タイプまたはテーブル・タイプのパラメータに データを割り当てる方法の選択 | 250 |
| XML パラメータのプロパティの設定 | 255 |
| 第 15 章 : パラメータのプロパティの設定 | 265 |
| パラメータのプロパティの設定について | 266 |
| 内部データ・パラメータ・タイプのプロパティの設定 | 267 |
| ユーザ定義関数のプロパティの設定 | 276 |
| パラメータ形式のカスタマイズ | 277 |
| 更新方法の選択 | 278 |
| ファイル・タイプのパラメータのシミュレーション | 279 |
| ファイル・パラメータのシミュレータの使用 | 281 |

第 III 部 : 記録オプション

| | |
|---|-----|
| 第 16 章 : 選択したプロトコルの記録オプション | 287 |
| EJB 記録オプション | 288 |
| RDP 記録オプション | 289 |
| Citrix 記録オプションについて | 294 |
| Citrix 記録オプションの設定 | 300 |
| データベース記録オプション | 301 |
| データベースの詳細記録オプション | 302 |
| AMF 記録モードの設定 | 305 |
| AMF コード生成オプション | 309 |
| Flex コード生成オプション | 310 |

| | |
|---------------------------------------|------------|
| Microsoft .NET 記録オプション | 312 |
| WCF 記録オプション | 318 |
| RTE 設定オプション | 320 |
| RTE 記録オプション | 321 |
| SAPGUI 記録オプション | 325 |
| SAP-Web 記録オプション | 328 |
| 相関記録オプションの設定 | 329 |
| Web, ワイヤレス, Oracle NCA 記録オプション | 331 |
| 第 17 章 : Click and Script 記録 | 339 |
| Click and Script による記録について | 339 |
| Web (HTTP/HTML) Vuser スクリプトの表示 | 340 |
| Click and Script 記録オプションの設定 | 341 |
| GUI プロパティの詳細設定 | 342 |
| Web イベント記録の定義 | 345 |
| 第 18 章 : Java 記録オプション | 357 |
| Java 記録オプションの設定について | 358 |
| Java 仮想マシン (JVM) 記録オプション | 359 |
| クラスパス記録オプションの設定 | 361 |
| レコーダ・オプション | 362 |
| シリアル化オプション | 364 |
| 相関オプション | 366 |
| ログ・オプション | 367 |
| CORBA オプション | 369 |
| 第 19 章 : スクリプト生成オプションの設定 | 371 |
| スクリプト生成オプションの設定について | 372 |
| スクリプト言語の選択 | 373 |
| 基本オプションの適用 | 374 |
| 相関オプションについて | 376 |
| 記録オプションの設定 | 377 |
| 第 20 章 : Web Vuser の記録オプションの設定 | 379 |
| 記録オプションの設定について | 379 |
| 記録レベルについて | 380 |
| 記録レベルの設定 | 393 |
| 第 21 章 : ポートの割り当て設定 | 395 |
| ポートの割り当て設定について | 396 |
| ポート割り当ての定義 | 396 |
| 新規サーバ・エントリの追加 | 399 |
| 高度なポート割り当てオプションの設定 | 402 |
| ポートの割り当て記録オプションの設定 | 404 |

第 IV 部：実行環境設定

| | |
|--|-----|
| 第 22 章：実行環境の設定 | 411 |
| 実行環境の設定について | 412 |
| 実行論理の設定（マルチ・アクション）..... | 413 |
| ペースの設定 | 418 |
| 実行環境のペースの設定（マルチ・アクション）..... | 419 |
| ペースの設定と実行論理オプションの設定 （シングル・アクション）..... | 420 |
| 実行環境設定のログの設定..... | 422 |
| 思考遅延時間の設定 | 427 |
| 実行環境の追加属性の設定..... | 429 |
| その他の設定 | 430 |
| VB 実行環境の設定..... | 435 |
| 第 23 章：ネットワーク実行環境の設定 | 437 |
| ネットワーク実行環境の設定について..... | 438 |
| ネットワーク速度の設定 | 438 |
| プロキシ・オプションの設定 | 439 |
| ブラウザのエミュレーション・プロパティの設定..... | 444 |
| インターネット・プリファレンスの設定..... | 449 |
| Web サイトのフィルタリング..... | 458 |
| デバッグ情報の取得 | 460 |
| HTML 圧縮の実行..... | 461 |
| Web ページのコンテンツ・チェック | 462 |
| 第 24 章：選択したプロトコルの実行環境の設定 | 469 |
| RDP の実行環境の設定 | 470 |
| Citrix の実行環境の設定..... | 475 |
| .NET 環境の実行環境の設定..... | 479 |
| Oracle NCA の実行環境の設定..... | 482 |
| SAPGUI の実行環境の設定 | 485 |
| Java および EJB の実行環境の設定..... | 489 |
| RTE の実行環境の設定 | 491 |
| WAP の実行環境の設定..... | 494 |
| MMS の実行環境の設定..... | 500 |
| Flex の実行環境の設定 | 502 |

第 V 部：上級ユーザのための情報

| | |
|--|-----|
| 第 25 章：Visual Studio による Vuser スクリプトの作成 | 505 |
| Visual Studio による Vuser スクリプトの作成について | 505 |
| Visual C による Vuser スクリプトの作成 | 507 |
| Visual Basic による Vuser スクリプトの作成..... | 508 |

| | |
|---|-----|
| 実行環境の設定とパラメータの設定 | 510 |
| 第 26 章 : XML API プログラミング | 511 |
| XML API プログラミングについて | 512 |
| XML ドキュメントについて | 513 |
| XML 関数の使用方法 | 514 |
| XML 関数のパラメータの指定 | 517 |
| XML 属性での作業 | 519 |
| XML スクリプトの作成 | 520 |
| 記録されたセッションの拡張 | 522 |
| 結果パラメータの使用 | 527 |
| 第 27 章 : VuGen のデバッグのヒント | 529 |
| 一般的なデバッグのヒント | 529 |
| C 関数を使用した追跡 | 530 |
| 付加的な C 言語のキーワードの追加 | 530 |
| 再生出力の検証 | 531 |
| データベース・アプリケーションのデバッグ | 531 |
| Oracle Applications を使った作業 | 533 |
| Oracle 2-Tier Vuser での一般的な問題の解決方法 | 534 |
| 2-tier データベースのスクリプト作成のヒント | 539 |
| PeopleSoft-Tuxedo スクリプトの実行 | 548 |
| 第 28 章 : 上級ユーザのために | 549 |
| 記録中に生成されるファイル | 550 |
| 再生中に生成されるファイル | 552 |
| UNIX コマンド・ラインからの Vuser の実行 | 554 |
| Vuser の動作の指定 | 555 |
| コマンド・ライン・パラメータ | 556 |
| OLE サーバの記録 | 556 |
| .dat ファイルの検証 | 559 |
| 新規 Vuser タイプの追加 | 561 |

第 VI 部 : 付録

| | |
|-------------------------------------|-----|
| 付録 A : 外部関数の呼び出し | 569 |
| DLL のローカル・ロード | 570 |
| DLL のグローバル・ロード | 572 |
| 付録 B : 多国語を使った作業 | 573 |
| 多国語を使った作業について | 574 |
| 手作業による文字列エンコーディングの変換 | 574 |
| パラメータ・ファイル内の文字列エンコーディングの変換 | 575 |
| Web 記録および再生のための文字列エンコーディングの設定 | 577 |
| Accept-Language ヘッダの言語の指定 | 580 |

| | |
|---|------------|
| プロトコルに関する制限 | 581 |
| Quality Center の統合 | 582 |
| 付録 C : UNIX プラットフォームでのスクリプトの プログラミング | 583 |
| UNIX プラットフォームで実行可能な Vuser スクリプトの プログラミングについて | 584 |
| テンプレートの生成 | 585 |
| Vuser のアクションのプログラミング | 586 |
| Vuser の実行環境設定 | 587 |
| トランザクションとランデブー・ポイントの定義 | 592 |
| スクリプトのコンパイル | 593 |
| 付録 D : キーボード・ショートカットの使用 | 595 |
| 索引 | 597 |

本書について

HP Virtual User Generator (VuGen) へようこそ。HP Virtual User Generator は、Vuser スクリプトを作成するための HP のツールです。**VuGen** では、典型的なビジネス・プロセスを実行しているユーザの操作を記録することによって、Vuser スクリプトを作成します。これらのスクリプトを使用することで、現実世界の状況をエミュレートできます。

VuGen で作成したスクリプトは、HP LoadRunner、HP Performance Center および HP Business Availability Center などのほかの HP 製品と組み合わせて使用できます。

HP LoadRunner は、パフォーマンス・テストのツールです。アプリケーション全体に負荷をかけて、クライアント、ネットワーク、サーバの潜在的なボトルネックを検出し、特定します。

HP Performance Center は、LoadRunner の機能をエンタープライズ・レベルで実装しています。

HP Business Availability Center は、実運用環境におけるビジネス・アプリケーションおよびシステムの管理と可用性を最適化するのに役立ちます。

本章の内容

- ▶ 本書の構成 (14 ページ)
- ▶ 対象読者 (15 ページ)
- ▶ LoadRunner オンライン・ドキュメント (15 ページ)
- ▶ その他のオンライン・リソース (17 ページ)
- ▶ ドキュメントのアップデート (18 ページ)

本書の構成

本書は、次の各部で構成されています。

第 I 部 VuGen を使った作業

Virtual User Generator のインタフェースおよびスクリプトの記録と再生について説明します。また、標準の実行環境の設定方法、データ・パラメータの使用方法、スクリプトのカスタマイズ方法も説明します。

第 II 部 パラメータ

パラメータの作成プロセスについて説明します。また、パラメータのタイプおよびプロパティの情報も提供します。

第 III 部 記録オプション

すべておよび個別のプロトコルに関連する記録オプションについて説明します。

第 IV 部 実行環境設定

すべておよび個別のプロトコルに関連する実行環境設定について説明します。

第 V 部 上級ユーザのための情報

デバッグのための一般的なヒント、VuGen によって生成されるファイルの説明、Visual C および Visual Basic でのスクリプトのプログラミング方法など、高度な使用方法について情報を提供します。

第 VI 部 付録

ほかの高度な話題について、技術的な概説や情報が含まれています。外部関数の呼び出し、UNIX 環境でのプログラミング、多国語での作業、キーボード・ショートカットについて説明しています。

注：『Virtual User Generator ユーザーズ・ガイド』のオンライン版は1冊、印刷版は『第1巻－VuGenの使用』および『第2巻－プロトコル』の2冊から成ります。

対象読者

本書は、次の利用者を対象としています。

- ▶ スクリプト開発者
- ▶ 機能テスト担当者
- ▶ 負荷テスト実施者

本書では、読者をご利用のエンタープライズ・アプリケーションについてある程度の知識を持っていることを前提とします。

LoadRunner オンライン・ドキュメント

LoadRunner には、製品の使用方法を説明するドキュメント・セットが含まれています。[ヘルプ] メニューからおよび PDF 形式でドキュメントを利用できます。PDF は Adobe Reader を使って閲覧および印刷できます。Adobe Reader は、Adobe の Web サイト (<http://www.adobe.com/jp/>) からダウンロードできます。

ドキュメントへのアクセス

次のようにしてドキュメントにアクセスすることができます。

- ▶ [スタート] メニューから、[スタート] > [LoadRunner] > [Documentation] をクリックして、参照するドキュメントを選択します。
- ▶ [ヘルプ] メニューから [Documentation Library] をクリックして、統合されたヘルプを開きます。

スタートアップ用のドキュメント

- ▶ 『最初にお読みください』では、LoadRunner に関する最新のお知らせと情報が含まれます。「最初にお読みください」には、[スタート] メニューからアクセスします。
- ▶ 『HP LoadRunner クイック・スタート』では、LoadRunner の概要を、簡潔に順を追って説明し、その使用法を紹介します。[スタート] メニューからクイック・スタートにアクセスするには、[Start] > [LoadRunner] > [Quick Start] をクリックします。

- ▶ 『**HP LoadRunner チュートリアル**』は、自分のペースで学習できる、印刷可能なガイドです。負荷テストのプロセスについて説明しており、LoadRunner テスト環境の理解を深めることができます。[スタート] メニューから『チュートリアル』にアクセスするには、[スタート] > [LoadRunner] > [Tutorial] をクリックします。

LoadRunner のガイド

- ▶ 『**Mercury Virtual User Generator ユーザーズ・ガイド**』: VuGen を使用してスクリプトを作成する方法について説明します。印刷版は『第1巻 – VuGen の使用』および『第2巻 – プロトコル』の2冊から成ります。必要に応じ、このマニュアルと併せてオンラインの『**HP LoadRunner Online Function Reference**』(英語版)をお読みください。
- ▶ 『**Mercury LoadRunner Controller ユーザーズ・ガイド**』: Windows 環境で LoadRunner Controller を使って LoadRunner シナリオを作成し実行する方法について説明します。
- ▶ 『**HP LoadRunner Monitor Reference**』(英語版): サーバ・モニタ環境のセットアップ方法と、LoadRunner モニタを設定してシナリオ実行中に生成されたデータの監視を行う方法について説明します。
- ▶ 『**HP LoadRunner Analysis ユーザーズ・ガイド**』: シナリオの実行後に LoadRunner Analysis のグラフとレポートを使用してシステムのパフォーマンスを分析する方法について説明します。
- ▶ 『**HP LoadRunner インストール・ガイド**』: LoadRunner のインストール方法および LoadRunner のサンプルを含むその他の LoadRunner コンポーネントのインストール方法について説明します。

LoadRunner リファレンス

- ▶ 『**LoadRunner Function Reference**』(英語版): Vuser スクリプトの作成時に使用する LoadRunner の関数をすべて、その使用例と共に参照できます。
- ▶ 『**Analysis API Reference**』(英語版): このアナリシス API セットは、Analysis セッションを無人で作成するため、あるいは Controller 下でのテスト実行の結果からのデータの抽出をカスタムで行うために使用できます。このリファレンスには、[Analysis Help] メニューからアクセスします。

- ▶ 『**LoadRunner Controller Automation COM and Monitor Automation Reference**』(英語版) : LoadRunner Controller を実行し、コントローラのユーザ・インタフェースで使用できるほとんどのアクションを実行するためのプログラムを書くことのできるインタフェースです。< **LoadRunner インストール・ディレクトリ**> /bin ディレクトリから、このリファレンス (**automation.chm**) にアクセスできます。
- ▶ 『**Error Codes and Troubleshooting**』(英語版) : Controller の接続性および Web プロトコル・エラーの分かりやすい説明およびトラブルシューティングのヒントを提供します。また、Winsock, SAPGUI および Citrix プロトコルの一般的なトラブルシューティングのヒントも提供します。

その他のオンライン・リソース

その他の情報については、次のいずれかのリソースを参照してください。

トラブルシューティング & ナレッジベース : HP ソフトウェア・サポート Web サイトのトラブルシューティングのページにアクセスします。このサイトからはセルフ・ソルブ技術情報を検索できます。[ヘルプ] > [トラブルシューティング & ナレッジベース] を選択します。この Web サイトの URL は、<http://h20230.www2.hp.com/troubleshooting.jsp> です。

HP ソフトウェア・サポート : HP ソフトウェアのサポート Web サイトにアクセスします。このサイトでは、セルフ・ソルブ技術情報を閲覧できます。また、ユーザ・ディスカッション・フォーラムへの投稿や検索、サポート依頼の送信、パッチや更新されたドキュメントのダウンロードなども行えます。[ヘルプ] > [HP ソフトウェア サポート] を選択します。この Web サイトの URL は、www.hp.com/go/hpsoftwaresupport です。

ほとんどのサポート・ページでは、HP Passport ユーザとして登録してログインすることを求められます。また、多くはサポート契約が必要です。

アクセス・レベルの詳細については、http://h20230.www2.hp.com/new_access_levels.jsp を参照してください。

HP Passport ユーザ ID の登録は、次の場所で行います。

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

HP ソフトウェア Web サイトからは、HP ソフトウェア Web サイトにアクセスします。このサイトでは、HP ソフトウェア製品に関する最新情報を提供します。新しいソフトウェアのリリース、セミナー、展示会、カスタマー・サポートなどの情報も含まれています。[ヘルプ] > [HP ソフトウェア Web サイト] を選択します。この Web サイトの URL は、<http://welcome.hp.com/country/jp/ja/prodserv/software.html> です。

ドキュメントのアップデート

HP ソフトウェアの製品ドキュメントは、常に更新されています。

最新の更新情報の確認や、使用しているドキュメントが最新版であるかどうかの確認を行うには、HP ソフトウェアの製品ドキュメント Web サイトを参照してください (<http://h20230.www2.hp.com/selfsolve/manuals>)。

第 1 部

VuGen を使った作業

第1章

Vuser スクリプトの開発

環境のテストや監視を実行する際は、システムにおけるユーザの振る舞いを正確にエミュレートする必要があります。HP のテスト・ツールは、複数のユーザが同時に作業している環境や、複数のユーザがシステムに同時にアクセスしている環境をエミュレートします。

そのような環境をエミュレートするために、実際のユーザの代わりとして **Vuser** を使用します。Vuser が実行する操作は **Vuser スクリプト** によって表現されます。Vuser スクリプトを作成するための中心的なツールとなるのが **Virtual User Generator**, **VuGen** です。

本章の内容

- ▶ Vuser の紹介 (22 ページ)
- ▶ Vuser のタイプ (24 ページ)
- ▶ Vuser スクリプトの作成手順 (26 ページ)
- ▶ HP LoadRunner ライセンス (27 ページ)
- ▶ LoadRunner と Service Test (28 ページ)

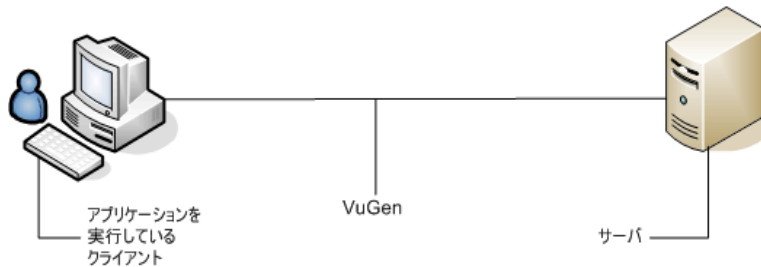
注 : **Virtual User Generator** ガイドのオンライン版は 1 巻から構成されていますが、印刷版は『**第 1 巻 – VuGen の使用**』および『**第 2 巻 – プロトコル**』の 2 巻から構成されています。

Vuser の紹介

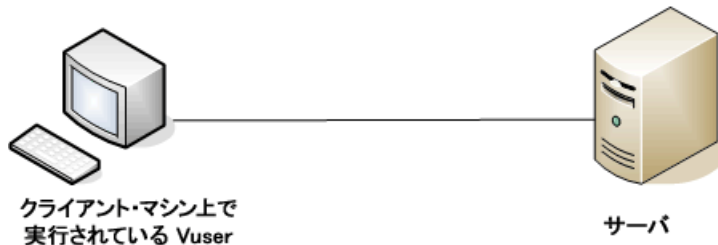
Vuser は実際のユーザのアクションを、一般的なアプリケーションの標準的なビジネス・プロセスを実行することでエミュレートします。Vuser が記録セッション中に実行する操作は **Vuser スクリプト** によって表現されます。

HP のツールの中で、Vuser スクリプトを作成するための中心的なツールとなるのが **Virtual User Generator, VuGen** です。VuGen では、クライアント・アプリケーションで典型的なビジネス・プロセスを実行しているユーザの操作を記録することによって、Vuser スクリプトを作成します。VuGen は記録セッション中に実行された操作を記録しますが、記録されるのはクライアントとサーバの間のやり取りだけです。アプリケーションの API 関数からサーバへの呼び出しを手作業でプログラミングする代わりに、VuGen は、実際に起こった状況を正確にモデル化しエミュレートする関数を、自動的に生成します。

記録時には、VuGen がデータベースのクライアント側を監視し、ユーザとサーバとの間で送受信されるすべての要求を追跡します。



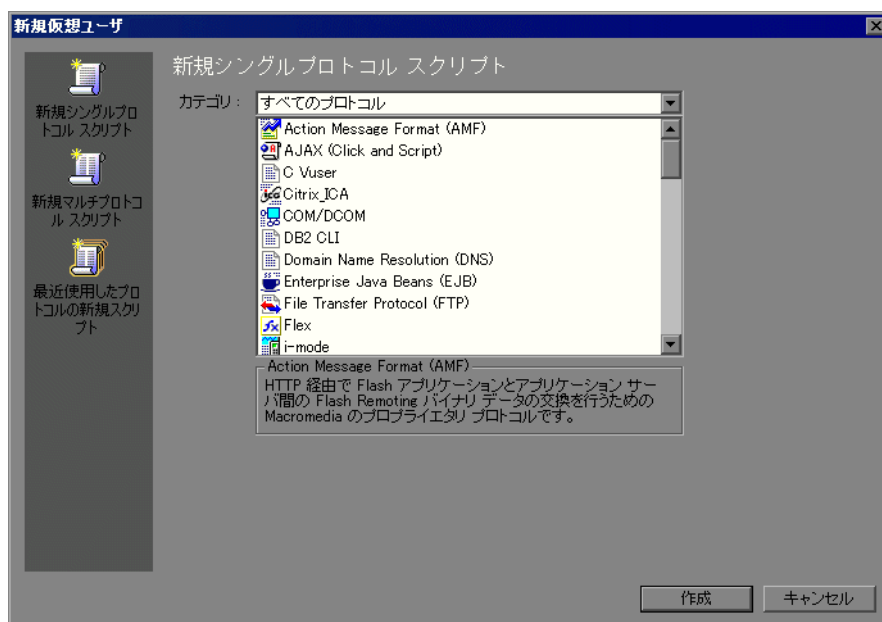
再生時には、Vuser スクリプトがサーバ API への呼び出しを実行して、サーバと直接やり取りします。Vuser からサーバと直接通信するときは、システム・リソースがクライアント・インタフェースで必要になりません。このため、1 台のワークステーションで多数の Vuser を同時に実行でき、数台のテスト用マシンだけで大きなサーバ負荷をエミュレートすることが可能になります。



また、Vuser スクリプトはクライアント・ソフトウェアに依存しないため、クライアント・ソフトウェアのユーザ・インタフェースが完成していなくても、Vuser を使ってサーバのパフォーマンスを検査できます。

VuGen を使うと、スクリプトをスタンドアロン・テストとして実行できます。スクリプトを VuGen から実行することで、デバッグの際に Vuser の振る舞いを調べ、どのような機能拡張が必要なのかを判断できるので便利です。

VuGen ではさまざまなタイプの Vuser を記録でき、それぞれのタイプが特定の負荷テスト環境またはトポロジに対応しています。VuGen で新しいテストを開くと、サポートされているすべてのプロトコルの一覧が表示されます。



Vuser の実行中は、システムの応答に関する情報を収集します。その後で、これらの情報を Analysis ツールを使って表示できます。たとえば、銀行の ATM から現金を引き出す 100 の Vuser を同時に実行して、そのときのサーバの動作を観察することができます。

Vuser のタイプ

VuGen は、システムのエミュレートを可能にするさまざまな Vuser テクノロジを備えています。それぞれのテクノロジは特定のアーキテクチャに対応しており、アーキテクチャごとに専用の Vuser スクリプトが作成されます。たとえば、Web Vuser スクリプトは、Web ブラウザを操作しているユーザをエミュレートする場合に使用します。FTP Vuser は、FTP セッションのエミュレートに使用します。さまざまな Vuser テクノロジを単独で使用したり、組み合わせて使用したりすることによって、効果的な負荷テスト、または Business Process Monitor プロファイルを作成できます。

Vuser のタイプは次のカテゴリに分類されます。

- ▶ **[すべてのプロトコル]** : サポートされている全プロトコルのアルファベット順リスト。
- ▶ **[アプリケーションの導入ソリューション]** : Citrix および Microsoft Remote Desktop Protocol (RDP) プロトコル用。
- ▶ **[クライアント/サーバ]** : DB2 CLI, DNS, Informix, Microsoft .NET, MS SQL, ODBC, Oracle (2-Tier), Sybase CTlib, Sybase DBlib, Windows Sockets プロトコル用。
- ▶ **[ユーザ定義]** : C テンプレート, Java テンプレート, Javascript, VB Script, VBNet, および VB テンプレート・タイプ・スクリプト用。
- ▶ **[分散コンポーネント]** : COM/DCOM, および Microsoft .NET プロトコル用。
- ▶ **[e ビジネス]** : AJAX (Click and Script), AMF, Flex, FTP, LDAP, Microsoft .NET, Web (Click and Script), Web (HTTP/HTML), および Web サービス・プロトコル用。
- ▶ **[ERP/CRM]** : Oracle NCA, Oracle Web Applications 11i, Peoplesoft Enterprise, Peoplesoft-Tuxedo, SAP-Web, SAPGUI, SAP (Click and Script), および Siebel Web プロトコル用。
- ▶ **[Enterprise Java Beans]** : EJB テスト用。
- ▶ **[Java]** : Java Record/Replay プロトコル用。
- ▶ **[レガシ]** : ターミナル・エミュレーション (RTE) 用。
- ▶ **[メール・サービス]** : Internet Messaging (IMAP), MS Exchange (MAPI), Post Office Protocol (POP3), Simple Mail Protocol (SMTP) プロトコル用。
- ▶ **[ミドルウェア]** : Tuxedo プロトコル用。

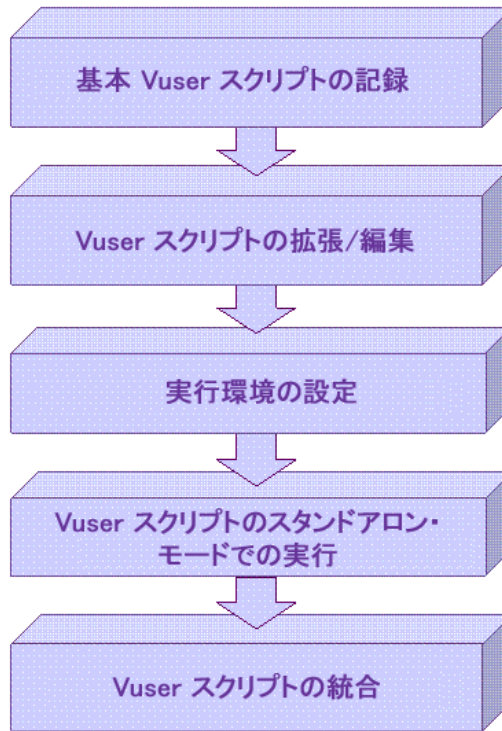
- ▶ **[ストリーミング]** : MediaPlayer (MMS) と RealPlayer プロトコル用。
- ▶ **[ワイヤレス]** : マルチメディア・メッセージング・サービス (MM) および WAP プロトコル用。

サポートされているプロトコルをアルファベット順に並べたリストを見るには、**[ファイル]** > **[新規作成]** を選択し、**[カテゴリ]** リスト・ボックスで **[すべてのプロトコル]** を選択します。

注 : さまざまなプロトコルを実行するには、グローバル・ライセンスか希望のプロトコルのライセンスを取得する必要があります。詳細については、LoadRunner Launcher (**[スタート]** > **[すべてのプログラム]** > **[LoadRunner]** > **[LoadRunner]**) で **[設定]** > **[LoadRunner ライセンス]** を選択します。

Vuser スクリプトの作成手順

次の図は、Vuser スクリプトの作成プロセスの概略です。



Vuser スクリプトの作成プロセスは次のとおりです。

- 1 VuGen を使って基本となるスクリプトを記録します。Windows ベースの GUI アプリケーションをテストする場合、またはアプレットや Flash などの複合的な Web 環境をテストする場合は、HP の GUI ベースのツールである WinRunner や QuickTest Professional を使用しなければならないことがあります。
- 2 制御フロー・ステートメントやその他の LoadRunner API 関数をスクリプトに追加して、基本となるスクリプトを拡張します。
- 3 実行環境を設定します。実効環境の設定には、反復、ログ、タイミングの情報などがあり、これらの設定によってスクリプト実行中における Vuser の振る舞いが決まります。
- 4 スクリプトの機能を確認し、スタンドアロン・モードで実行します。

- 5 スクリプトが正常に機能することを確認した後、スクリプトを自分の環境 (LoadRunner シナリオ, Performance Center 負荷テスト, または Business Process Monitor プロファイル) に統合します。詳細については、**HP LoadRunner Controller**, **HP Performance Center**, または **HP Business Availability Center** を参照してください。

HP LoadRunner ライセンス

LoadRunner の購入時に、設定とニーズに合うようにユーザ定義ライセンスを受け取ります。この情報には、LoadRunner Launcher を使っていつでもアクセスできます。ライセンス・キー情報を確認するには、**[スタート]** > **[プログラム]** > **[LoadRunner]** を選択し、LoadRunner Launcher を起動します。Launcher の **[設定]** メニューから **[LoadRunner ライセンス]** を選択します。

Service Test のライセンスを購入した場合は、ライセンス・マネージャからインストールします (**[スタート]** > **[プログラム]** > **[LoadRunner]** > **[Service Test]** > **[Service Test License Manager]**)。ライセンスには、シート・ライセンス (ホスト・ロック) とコンカレント・ライセンス (サイト・ライセンス) があります。詳細については、『**HP LoadRunner インストール・ガイド**』を参照してください。

HP Service Test は、スタンドアロン製品としても、HP LoadRunner の Virtual User Generator, VuGen の拡張としても利用できます。Service Test には、VuGen のすべての機能に加えて、SOA および Web サービスのテストの領域の機能が追加されています。

HP Service Test は機能テスト・ツールであるため、VuGen より機能的なテスト機能が提供されます。これに加えて、Service Test には、WSDL, XML, および SOAP を使って作業できる複数のユーティリティが含まれています。

HP Service Test で作成したスクリプトはすべて、HP LoadRunner および HP Performance Center で実行できます。この機能は、Service Test と LoadRunner のバージョンが同じである場合のみ利用できます。

LoadRunner と Service Test

次の機能は、HP Service Test (スタンドアロン) または LoadRunner と Service Test でのみ利用できます。

機能テスト・ユーティリティ

- ▶ **XML 妥当性検証** : XML が正しく書式化されていて、スキーマに準拠し、期待値が使用されているかどうかを確認します。
- ▶ **チェックポイント** : 応答を期待値と比較できます。
- ▶ **ネガティブ・テスト** : SOAP 障害を期待する応答として定義する機能で、エラー・ケースをテストできるようにします。
- ▶ **WS-I 妥当性確認** : WSDL および SOAP が WS-I に準拠しているかどうかを確認する機能です。
- ▶ **テスト生成ウィザード** : アスペクト・ベースのテストを作成するのに役立つウィザードです。

Quality Center との統合

- ▶ **BPT** : Business Process Testing。Service Test では、Quality Center の BPT に対するコンポーネントを作成し、ビジネス・シナリオ内で実行できます。
- ▶ **QC からのリモート実行** : Service Test スクリプトを、Quality Center から直接起動できます。パラメータおよび実行環境の設定、テストの実行、および Quality Center 内からの結果の検査を行えます。

その他のツール

- ▶ **サービス・エミュレーション** : クライアントとサーバの両方をシミュレートするエミュレーション・サービスを作成するツールです。
- ▶ **コマンド・ライン呼び出し** : コマンドラインからスクリプトを呼び出す機能です。

第 2 章

VuGen の紹介

仮想ユーザ・ジェネレータ（VuGen）を使って、さまざまなタイプのアプリケーションと通信プロトコルに対応した Vuser スクリプトを作成できます。

本章の内容

- ▶ VuGen について（30 ページ）
- ▶ VuGen の起動（30 ページ）
- ▶ コマンドのユーザ定義（31 ページ）
- ▶ VuGen 環境オプションについて（34 ページ）
- ▶ 環境オプションの設定（36 ページ）
- ▶ Vuser スクリプトの表示と変更（36 ページ）
- ▶ VuGen を使った Vuser スクリプトの実行（51 ページ）
- ▶ VuGen のコードについて（52 ページ）
- ▶ Vuser 関数に関するヘルプについて（54 ページ）

VuGen について

Virtual User Generator は、「VuGen」とも呼ばれ、Vuser スクリプトの作成に使用される主要ツールです。

VuGen では、Vuser スクリプトの記録だけでなく実行も可能です。VuGen でのスクリプト実行はデバッグに役立ちます。スクリプトを実行することによって、Vuser スクリプトが大規模なテストの一部としてどのように動作するかを確認できます。

Vuser スクリプトの記録時に、記録セッション中に実行されたアクションを定義するさまざまな関数が VuGen によって生成されます。VuGen では、これらの関数が VuGen エディタに挿入されることによって、基本となる Vuser スクリプトが作成されます。

VuGen で Vuser スクリプトを記録できるのは Windows プラットフォームのみです。しかし、記録した Vuser スクリプトは、Windows および UNIX の両プラットフォームで実行できます。

VuGen の起動

VuGen は、HP Business Availability Center、HP LoadRunner、および HP Performance Center など、さまざまなアプリケーションから実行できます。

VuGen を起動するには、[スタート] > [プログラム] > <アプリケーション名 > > [Applications] > [Virtual User Generator] を選択します。Virtual User Generator のスタート・ページが開きます。

スタート・ページからは、次のアクションを利用できます。

| アクション | 手順 |
|------------------------|---|
| 既存のスクリプトを開く | [既存のスクリプトを開く] をクリックします。 |
| zip アーカイブから既存のスクリプトを開く | [ファイル] > [ZIP 操作] > [Zip ファイルからインポート] を選択します。 |
| 新しいスクリプトを作成する | [新規スクリプト] をクリックします。 |

コマンドのユーザ定義

VuGen の外観をユーザ定義するには、次のいずれかのタブを使用して [カスタマイズ] ダイアログ・ボックスからツールバーやショートカットを設定します。

- ▶ コマンド・タブ
- ▶ ツールバー・タブ
- ▶ ツール・タブ
- ▶ キーボード・タブ
- ▶ オプション・タブ

コマンド・タブ

[カテゴリ] セクション（左側の表示枠）でツールバーを選択します。ツールバーのボタンが右側の表示枠の **[コマンド]** に表示されます。コマンドをクリックするとその説明が表示されます。必要な項目を、右側の表示枠から開いている目的のツールバーにドラッグします。

ツールバー・タブ

[ツールバー] タブでは、どのツールバーを表示するかを指定できます。表示するツールバーごとにチェック・ボックスを選択します。使用可能なツールバーは、**[標準]**、**[編集]**、**[表示]**、**[記録]**、**[仮想ユーザ]**、**[デバッグ ツール]**、**[ツール]**、**[ツリー ビュー]**、**[詳細]** です。

さらに、このタブにある次のボタンを使用すると、ツールバーの設定を変更できます。




- ▶ **[リセット]**：左側の表示枠で選択したツールバーの標準のツールバー表示設定に戻ります。
- ▶ **[すべてリセット]**：すべてのツールバーについて標準のツールバー表示設定に戻ります。標準設定に戻る前に、VuGen は警告を表示します。
- ▶ **[新規作成]**：表示されているツールバーのリストにユーザ定義ツールバーを追加します。このツールバーを、VuGen ウィンドウの必要な位置にドラッグします。その後、**[コマンド]** タブを使用してアイコンをツールバーに追加します。
- ▶ **[名前の変更]**：選択したツールバーの名前を変更できます。このボタンは、ユーザ定義ツールバーを選択したときのみ有効になります。

- ▶ **[削除]**：選択したツールバーを削除します。このボタンは、ユーザ定義ツールバーを選択したときのみ有効になります。
- ▶ **[テキストラベル表示]**：選択したツールバーのテキスト・ラベルを表示します。たとえば、[記録] ツールバーでこのオプションを有効にすると、[停止] ボタンに「停止」という文字が表示され、[実行] ボタンに「実行」という文字が表示されます。

ツール・タブ

[ツール] タブでは、VuGen ツールバーにユーザ定義コマンドを追加できます。通常、コマンドは、実行可能ファイル、バッチ・ファイル、または **.com** ファイルです。また、VuGen の中から特定のドキュメントや PDF ファイルを開くように指定することもできます。1 つまたは複数のツールを [ツール] タブの [ツール] リストで定義できます。これらのツールは [コマンド] タブの [ツール] メニューの下に一覧表示されます。必要なコマンドを [ツール] メニューから選択し、それを必要なツールバーにドラッグします。たとえば、Windows の電卓 **calc.exe** を VuGen から開くことができるように指定できます。

メニュー内容

-  ▶ **[新規作成]**：(一番左のボタン) 項目を [メニュー内容] リストに追加します。[メニュー内容] リストに新規項目が作成され、カーソルが行の先頭に移動します。コマンドの名前を入力します。
-  ▶ **[削除]**：(左から2番目のボタン) 選択した項目を [メニュー内容] リストから削除します。
-  ▶ **[項目を上に移動 / 項目を下に移動]**：選択した項目を上または下に移動します。

項目詳細

- ▶ **[コマンド]**：実行可能ファイルまたはバッチ・ファイルの名前を入力します。または、フィールドの右にある [参照] ボタンをクリックして、対象ファイルを選択します。実行可能でないファイルを指定する場合は、ファイル拡張子がマシンにインストールされているプログラムに関連付けられていることを確かめます。
- ▶ **[引数]**：指定したファイルと一緒に実行される実行時の引数を指定します。
- ▶ **[初期ディレクトリ]**：ユーザ定義ツールを実行する初期ディレクトリです。

キーボード・タブ

[**キーボード**] タブでは、キーボード・ショートカットをメニュー・コマンドに割り当てることができます。ショートカットは、標準コマンドとユーザ定義コマンドの両方に割り当てることができます。

キーボード・ショートカットを割り当てするには、次の手順を実行します。

- 1 [**カテゴリ**] ボックスでメニュー・カテゴリを選択します。
- 2 選択したカテゴリの中にあるコマンドを [**コマンド**] ボックスから選択します。
- 3 [**新規ショートカット キーを押す**] ボックスの中をクリックし、ショートカット・キー、またはキーの組み合わせを押します。キー、またはキーの組み合わせがボックスに表示されます。
- 4 [**割り当て**] をクリックします。
- 5 割り当てを削除するには、コマンドを選択し、割り当てられているショートカットを表示します。削除するショートカットを [**現在のキー設定**] ボックスで選択し、**[削除]** をクリックします。
- 6 すべて標準の割り当てに戻すには、**[すべてリセット]** をクリックします。これを実行すると、ユーザ定義のキーボード割り当てもすべて削除されます。

オプション・タブ

ツールバー

すべてのツールバーに適用されるいくつかの表示オプションを設定できます。

- ▶ [**ツールバーにヒントを表示する**] : カーソルをツールバー・ボタンの上に移動したときにヒントを表示します (標準設定では有効)。
 - ▶ [**ヒントにショートカット キーを表示する**] : ツールヒントにキーボード・ショートカットを表示します (標準設定では無効)。
- ▶ [**大きいアイコン**] : ツールバーに大きなボタンを表示します (標準設定では無効)。

メニューとツールバーの個人用設定

メニューやツールバーでのコマンドの表示方法をパーソナライズできます。

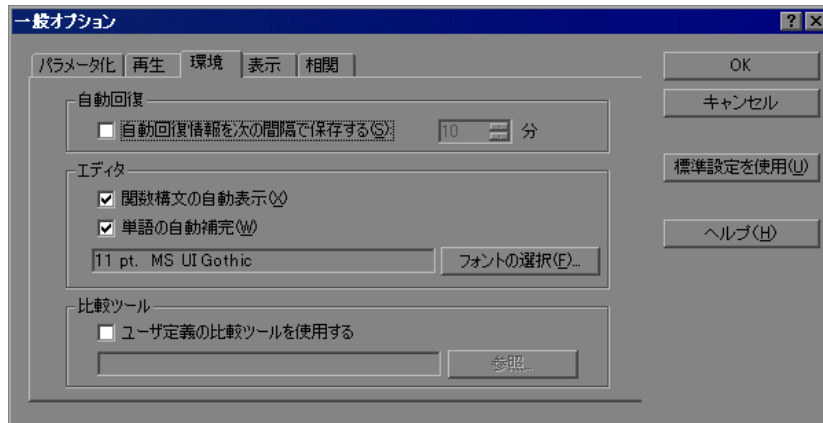
- ▶ **[最近使用したコマンドからメニューに表示する]**：最近使用したコマンドがメニューの先頭に表示されます（標準設定で有効）。
- ▶ **[少し時間が経過したら、すべてのメニューを表示する]**：展開可能なメニューの場合、少し時間をおいてからメニュー全体を表示します。

標準設定への復元

- ▶ **[使用履歴データをリセットする]**：ユーザ定義コマンドをすべて削除し、メニューとツールバーに表示されるコマンドのセットを標準設定に戻します。明示的なカスタマイズは元に戻しません。

VuGen 環境オプションについて

VuGen 作業環境の自動回復、およびエディタの設定をカスタマイズできます。
[ツール] > [一般オプション] > [環境] タブで次のオプションを設定します。



自動回復

自動回復オプションを使用することにより、クラッシュまたは停電が発生したときにスクリプトの設定を復元できます。自動回復オプションを使用するには、**[自動回復情報を次の間隔で保存する]** チェック・ボックスを選択し、保存の実行間隔を分単位で指定します。

エディタ

エディタのオプションで、単語の自動補完や関数構文の自動表示を行う VuGen の IntelliSense 機能を設定できます。

- ▶ **[関数構文の自動表示]**：関数の開始括弧を入力すると、関数の構文、引数、プロトタイプが自動的に表示されます。この関数をグローバルに有効にするにはチェック・ボックスを選択します。このボックスをチェックしなくても、エディタで開始括弧を入力した後に、Ctrl キーと Shift キーを押しながらスペース・キーを押すか、**[編集]** > **[関数の構文を表示]** > を選択すれば、この関数を手作業で有効にできます。
- ▶ **[単語の自動補完]**：関数の最初のアンダーバーを入力すると、関数の一覧が表示されます。関数全部を手作業で入力しなくても正確な関数を選ぶことができます。この関数をグローバルに有効にするにはチェック・ボックスを選択します。このボックスをチェックしなくても、Ctrl キーを押しながらスペース・キーを押すか、エディタで入力しながら **[編集]** > **[単語入力候補]** > を選択すれば、この関数を手作業で有効にできます。
- ▶ **[フォントの選択]**：エディタに表示されるフォントを設定するには、**[フォントの選択]** をクリックします。**[Font Configuration]** ダイアログ・ボックスが表示されます。使用するフォント、スタイル、サイズを選択し、**[OK]** をクリックします。使用できるフォントは、固定幅のフォント (Courier, Lucida Console, FixedSys など) だけです。

標準設定を使用

標準設定では、**[関数構文の自動表示]** や **[単語の自動補完]** は VuGen 全体で有効になっています。自動復元は、10 秒に設定されています。これらの設定を手作業で変更した場合は、**[標準設定を使用]** をクリックすると標準設定値を復元できます。

比較ツール

Vuser スクリプトは、比較ツールを使用すると横に並べて表示して比較できます。

Vuser スクリプトを比較するには、次の手順を実行します。

- 1 比較対象の最初の Vuser スクリプトを開きます。
- 2 **[ツール]** > **[スクリプトと比較]** を選択します。
- 3 2 つ目の Vuser スクリプトを選択します。Vuser スクリプトが、新しいウィンドウに横に並んで表示されます。差異は黄色で強調表示されます。

環境オプションの設定

環境関連オプションを設定するには、次の手順を実行します。

- 1 [ツール] > [一般オプション] を選択して、[環境] タブをクリックします。
- 2 現在のスクリプトの自動回復に関する情報を保存するには、[自動回復情報を次の間隔で保存する] オプションを選択して、設定を保存する間隔を分単位で指定します。
- 3 エディタに表示されるフォントを設定するには、[フォントの選択] をクリックします。[Font Configuration] ダイアログ・ボックスが表示されます。使用するフォント、スタイル、サイズを選択し、[OK] をクリックします。使用できるフォントは、固定幅のフォント（Courier, Lucida Console, FixedSys など）だけです。
- 4 WDiff 以外の比較ツールを使用するには、[ユーザ定義の比較ツールを使用する] を選択し、使用するツールを指定または参照します。指定する実行可能ファイルが有効な比較ツールで、パスが正しく指定されていることを確認してください。比較ツールでない実行可能ファイルを指定すると、予期しない結果が生じます。
- 5 設定を承認するには [OK] をクリックします。[一般オプション] ダイアログ・ボックスが閉じます。

Vuser スクリプトの表示と変更

VuGen には、スクリプトの内容を調べるためのビューが複数用意されています。1 つ目はテキスト形式のスクリプト・ビュー、2 つ目はアイコン形式のツリー・ビュー、3 つ目はアイコン形式のサムネイル・ビューです。

スクリプト・ビューとツリー・ビューは、ほとんどの Vuser のタイプで利用できます。多くのプロトコルでは、サムネイル・ビューもサポートされています。

スクリプト・ビューでのコードの表示

スクリプト・ビューでは、スクリプトに記録または挿入された実際の API 関数を表示できます。上級ユーザは、このビューで C や Vuser API の関数、あるいはフロー制御ステートメントを追加することで、スクリプトを編集できます。

スクリプト・ビューを表示するには、次の手順を実行します。

VuGen のメイン・メニューから [表示] > [スクリプト ビュー] を選択するか、[スクリプト] ツールバー・ボタンをクリックします。スクリプトがテキスト形式のスクリプト・ビューで表示されます。すでにスクリプト・ビューが表示されている場合は、このメニュー項目は選択できません。



- ☞ ☐ スクリプトの左側のマージンにあるプラスとマイナスの記号をクリックして、関数を展開したり折りたたんだりできます。これによって、スクリプトの表示を整理したり見やすくしたりできます。

スクリプト・ビューで作業を行うときは、[挿入] > [新規ステップ] コマンドを使用してスクリプトにステップを追加できます。または、単語の補完機能や関数構文の表示機能を使用して、関数を手作業で入力することもできます。詳細については、54 ページ「Vuser 関数に関するヘルプについて」を参照してください。

注：スクリプト・ビューの表示中に Vuser スクリプトを変更すると、それに応じて Vuser スクリプトのツリー・ビューも変更されます。テキスト形式のスクリプトに行われた変更を解釈できなかった場合は、スクリプト・ビューがツリー・ビューやサムネイル・ビューに変換されません。

ツリー・ビューでのスクリプトの表示

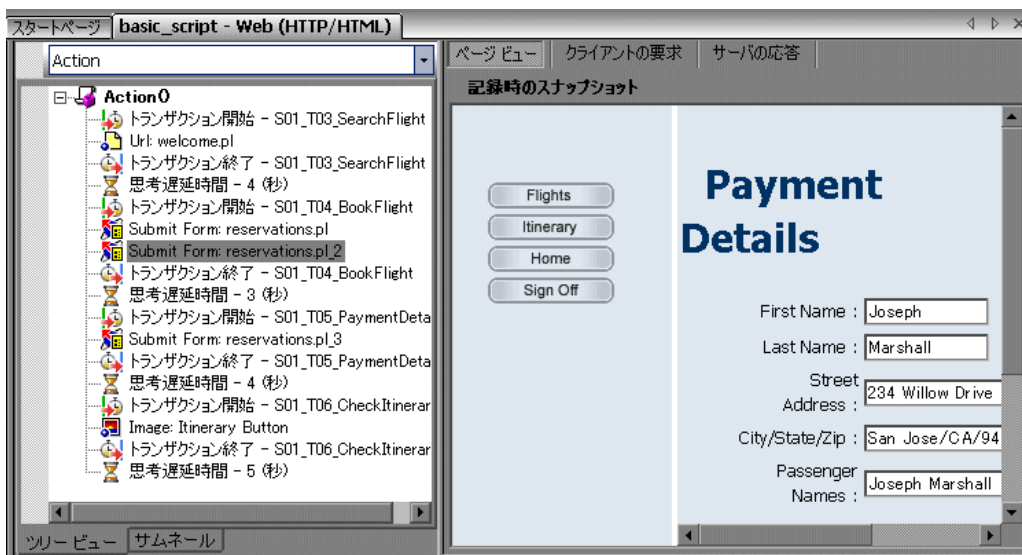
VuGen のツリー・ビューでは、Vuser スクリプトがアイコン形式で表示され、各ステップが異なるアイコンで表されます。

ツリー・ビューを表示するには、次の手順を実行します。

VuGen のメイン・メニューから、[表示] > [ツリー ビュー] を選択するか、[ツリーを表示] ボタンをクリックします。

Vuser スクリプトの Actions セクションがアイコン形式のツリー・ビューで表示されます。異なるセクションを表示するには、そのセクションをツリーの上にあるドロップダウン・リストで選択します。

ツリー・ビューがすでに選択されている場合、このメニュー項目は無効になっています。



ツリー・ビュー内では、ステップをドラッグして望みの位置に移動できます。また、ツリー階層の中で、既存のステップの間にステップを追加することもできます。

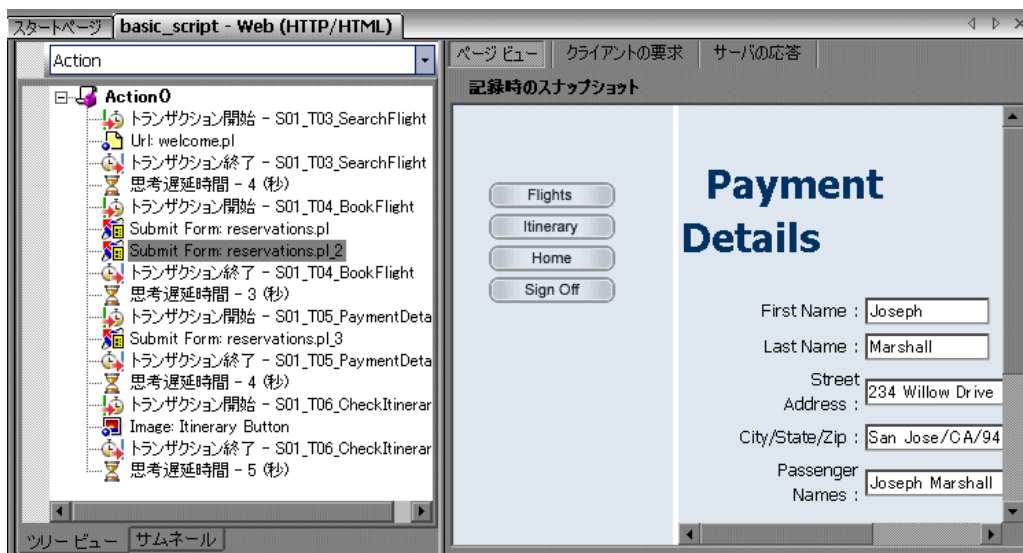
ツリー・ビューにステップを挿入するには、次の手順を実行します。

- 1 ステップをクリックします。
- 2 右クリック・メニューから [前に挿入] または [後に挿入] を選択します。
[ステップの追加] ダイアログ・ボックスが開きます。
- 3 ステップを選択して [OK] をクリックします。[プロパティ] ダイアログ・ボックスが開きます。
- 4 プロパティを指定して [OK] をクリックします。現在のステップの前または後ろにステップが挿入されます。

注：親ノードの後にステップを挿入すると、親ノードから子ノードが分離されます。

スナップショットについて

スナップショットは、現在のステップを視覚的に表したものです。ツリー・ビューで作業しているときは、選択したステップのスナップショットが VuGen の右側の表示枠に表示されます。スナップショットに、ステップ実行後のクライアント・ウィンドウが表示されます。



VuGen は、記録中に基本となるスナップショットをキャプチャし、再生中にも新たなスナップショットをキャプチャします。記録時のスナップショットと再生時のスナップショットを比較することによって、スクリプトを実行するために相関させる必要がある動的な値を見つけます。詳細については、記録後の相関に関する項を参照してください。

次のツールバー・ボタンを使用して、さまざまなスナップショット・ウィンドウを表示または非表示にすることができます。



記録時のスナップショットをフル・サイズで表示します。



記録時と再生時のスナップショットを分割したウィンドウで表示します。



再生時のスナップショットをフル・サイズで表示します。

スナップショットの表示と非表示を切り替えるには、次の手順を実行します。

- 1 ツリー・ビューが表示されていることを確認します。表示されていない場合は、ツリー・ビューに切り替えます ([表示] > [ツリー ビュー])。
- 2 [表示] > [スナップショット] > [スナップショットを表示] を選択します。クライアント・ウィンドウのスナップショットが表示されます。スナップショットがすでに表示されている場合は、非表示になります。
- 3 [表示] > [スナップショット] の展開したメニューを使用して、記録時、および再生時のスナップショットを表示できます。また、ショートカット・ツールバー・ボタンを使用して必要なスナップショットを表示することもできます。

スクリプトを再生するたびに、VuGen はスクリプトの結果ディレクトリ (Iteration1, Iteration2, など) に新たな再生時のスナップショットを保存します。

標準では、VuGen は記録時のスナップショットと、最初に再生されたときのスナップショットを比較します。別のスナップショットを選択して比較することもできます。特定の再生時のスナップショットを選択するには、[表示] > [スナップショット] > [反復の選択] を選択します。結果セットを選択して、[OK] をクリックします。

複数のスナップショット

Microsoft Remote Desktop Protocol (RDP) などの複数のプロトコルにおいて、単一のステップに対して複数のスナップショットを表示できます。この状態が発生するのは、再生時に不一致が生じ、新しい画像をステップに付加することを選択した場合です。詳細については、RDP プロトコルの節を参照してください。

スナップショットのトラブルシューティング

スナップショットのないステップが見つかった場合は、次のガイドラインに従ってスナップショットが生成されない原因を特定します。すべてのステップがスナップショットに関連付けられているわけではありません。スナップショットがあるのは、画面操作があるステップかブラウザ・ウィンドウの内容を表示しているステップ (Web の場合) のみです。

いくつかのプロトコルでは、記録オプションにより、記録中にスナップショットのキャプチャを無効にできます。

選択したステップの**記録時**のスナップショットがない場合は、次のような原因が考えられます。

- ▶ スクリプトが **VuGen 6.02** 以前のバージョンで記録された。
- ▶ スナップショットが特定の種類のステップについて生成されていない。
- ▶ インポートされたアクションに、スナップショットが含まれていない。

選択したステップの**再生時**のスナップショットがない場合は、次のような原因が考えられます。

- ▶ スクリプトが **VuGen 6.02** 以前のバージョンで記録された。
- ▶ インポートされたアクションに、スナップショットが含まれていない。
- ▶ **Vuser** ファイルが読み取り専用のディレクトリに格納されているため、**VuGen** が再生時のスナップショットを保存できなかった。
- ▶ ステップがリソースへのナビゲーションを表している。

スナップショット・ファイル

スクリプトを再生するたびに、VuGen は拡張子 **.inf** を付けてスクリプト・ディレクトリにスナップショットを保存します。再生時のスナップショットは、結果セットごとにスクリプトの結果ディレクトリ (**Iteration1**, **Iteration2**, など) に格納されます。

Web Vuser のスナップショット・ファイルの名前を調べるには、スクリプト・ビューに切り替えます ([表示] > [スクリプト ビュー])。次の例では、スナップショット情報が **t1.inf** であることがわかります。

```
web_url("WebTours",
        "URL=http://localhost/WebTours/",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=",
        "Snapshot=t1.inf",
        "Mode=HTML",
        LAST);
```

Citrix Vuser スクリプトでは、スナップショットが **.png** ファイルとしてスクリプトのディレクトリに保存されます。スナップショット・ファイルの名前を調べるには、スクリプト・ビューで関数の引数を調べます。

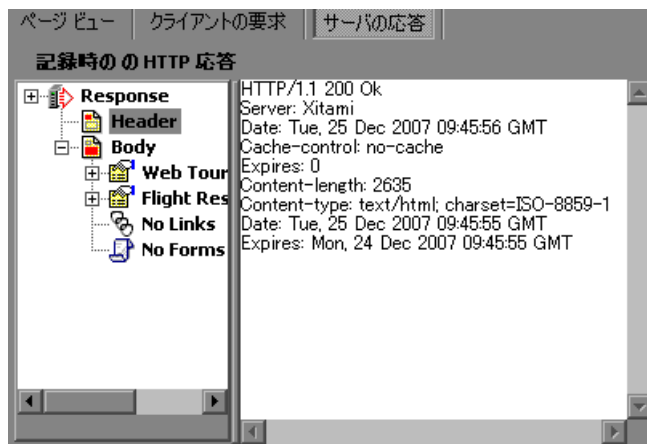
```
ctrx_sync_on_window("ICA Administrator Toolbar", ACTIVATE, 768, 0, 33,
                    573, "snapshot12", CTRX_LAST);
```

Vuser のスナップショット・タブ

Web Vuser のスナップショット・ウィンドウには、次のタブがあります。

- ▶ **[ページ ビュー]** : ブラウザに表示されるとおりに、スナップショットを HTML で表示します。このボタンは、記録時および再生時の両方のスナップショットに対して使用できます。このビューを使って、正しいスナップショットを表示しているかどうかを確認できます。ただし、このビューでは、関連させる必要のある値はわかりません。
- ▶ **[サーバの応答]** : スナップショットのサーバ応答 HTML コードを表示します。このタブは、記録時および再生時の両方のスナップショットに対して使用できます。[HTML] ビューでは、左の表示枠にスクリプトのツリー構造と、ドキュメントのコンポーネントであるヘッダと本体、およびそのタイトル、リンク、フォームなどの詳細も表示されます。

右側の表示枠のソースの中から要素の HTML テキストを見つけるには、[サーバの応答] の左側の表示枠で要素を選択し、右クリック・メニューから [要素の検索] を選択します。



- ▶ [クライアントの要求] : スナップショットのクライアント要求 HTML コードを表示します。このタブは、記録時および再生時の両方のスナップショットに対して使用できます。[HTML] ビューでは、左の表示枠にスクリプトのツリー構造と、ドキュメントのコンポーネントであるヘッダと本体、およびそのサブコンポーネントの詳細が表示されます。



スクリプトのサムネイルの表示

Web, SAPGUI, Citrix など一部の Vuser タイプでは、スナップショットをサムネイルで表示できます。サムネイルは、ツリー・ビューに表示するか、トランザクション・エディタを使って表示します。

ツリー・ビューでのサムネイルの表示

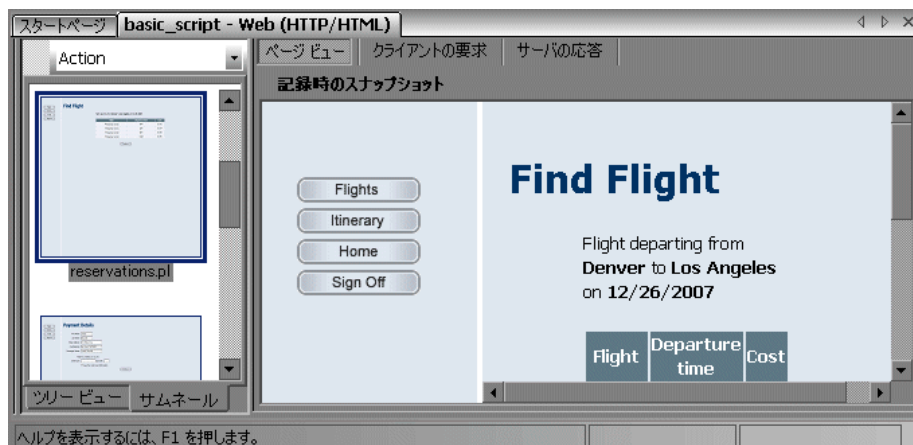
ツリー・ビューでは、ツリー・ビュー・ウィンドウの下部に **[サムネイル]** タブが表示されます。

標準設定では、スクリプト内の主要なステップのみがサムネイル・ビューに表示されます。すべてのサムネイルを表示するには、**[表示] > [すべてのサムネイルを表示]** を選択します。スクリプト内の全ステップのサムネイルが表示されます。

注：反復が複数の場合、最後の反復の再生のサムネイルが表示されます。特定の反復のサムネイルを表示するには、**[表示] > [スナップショット] > [反復の選択]** を選択し、希望の反復を選びます。

ツリー・ビューでサムネイルを表示するには、次の手順を実行します。

- 1 左側の表示枠の下部にある **[サムネイル]** タブをクリックします。
- 2 目的のサムネイル画像をクリックすると、右側の表示枠にサムネイルのスナップショットが開きます。



- 3 大きな画像を表示するには、サムネイル画像をダブルクリックします。別のウィンドウが開き、サムネイルが大きく表示されます。

アクションの表示モードの設定

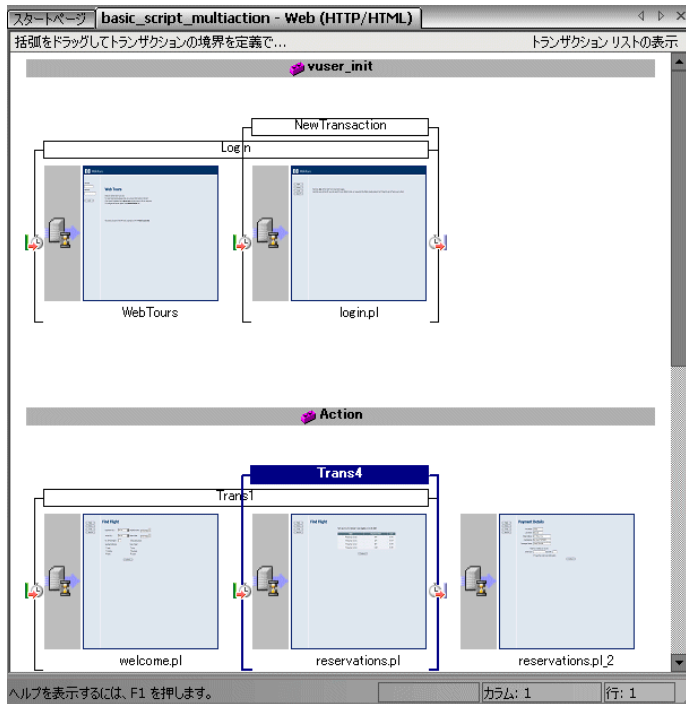
アクションを基準にしてスクリプトを表示する方法を指定できます（**[表示]** > **[アクション]**）。標準設定では、スクリプト・ビューの場合にのみ、左側の表示枠にスクリプト・アクションが表示されます。標準設定（**[自動的に開く]**）をそのまま使用することも、ツリー・ビューやスクリプト・ビューでアクション表示枠を開くこともできます。

- ▶ **[自動的に開く]**：スクリプト・ビューの場合のみ、左側の表示枠にスクリプト・アクションが表示されます。これは、標準設定です。アクション表示枠を非表示にするには、選択をクリアします。
- ▶ **[ツリーモードで開く]**：ツリー・ビューにアクションを表示します。アクション、ステップ、スナップショットの3つの表示枠が開きます（スナップショット・ビューが有効な場合）。各表示枠の幅を調整するには、境界線を任意の方向にドラッグします。アクション表示枠を非表示にするには、選択をクリアします。

- ▶ **[スクリプト モードで開く]** : スクリプト・ビューにアクションを表示します。アクションとスクリプト・ビューの2つの表示枠が開きます。アクション表示枠の幅を調整するには、境界線を任意の方向にドラッグします。アクション表示枠を非表示にするには、選択をクリアします。

ワークフロー・ウィザードでのサムネイルの表示

トランザクション・エディタを使ってスナップショットを表示できます。このビューでは、サムネイルがアクション順に並べ替えられ、スクリプトの全ステップのサムネイルが横並びで表示されます。



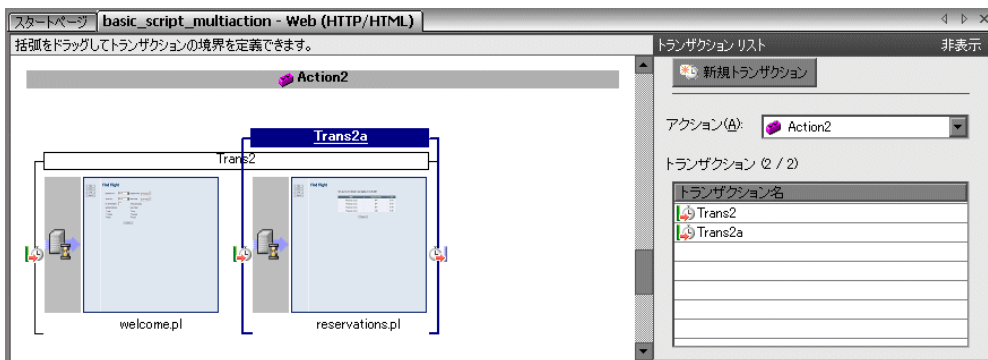
トランザクション・エディタにサムネイルを表示するには、次の手順を実行します。

- 1 ツールバーの [**タスク**] をクリックして、タスク・リスト表示枠を開きます。
- 2 [**拡張**] > [**トランザクション**] リンクをクリックします。中央の表示枠と右側の表示枠にトランザクション・エディタが開きます。

より包括的に表示するには、[**タスク**] をクリックしてタスク・リストを非表示にします。

- 3 右側の表示枠で、表示するアクションを選択します。選択したアクションが表示されます。

次の例では、「**Action2**」が選択されています。



サムネイルを使った作業

VuGen では、サムネイル名の変更、サムネイルへの注釈の追加、および大きなサイズでのサムネイルの表示ができます。

サムネイルを大きなサイズで表示するには、次の手順を実行します。

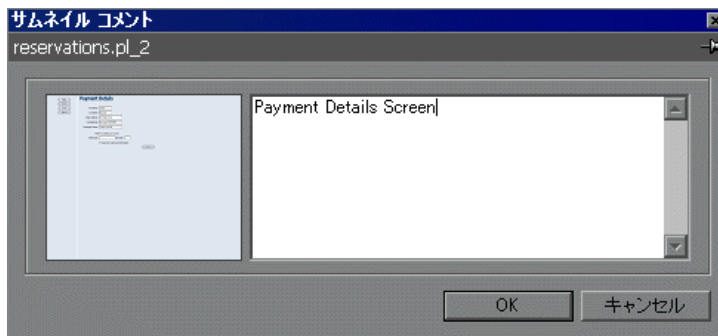
右クリックして表示されるメニューから **[画像表示を拡大]** を選択するか、**Alt** キーを押しながら **F6** キーを押します。別のウィンドウが開き、サムネイルが大きく表示されます。

サムネイルの名前を変更するには、次の手順を実行します。

- 1 サムネイルを選択し、右クリックして表示されるメニューから **[名前の変更]** を選択するか、**F2** キーを押します。
- 2 必要なテキストを入力します。

サムネイルに注釈を追加するには、次の手順を実行します。

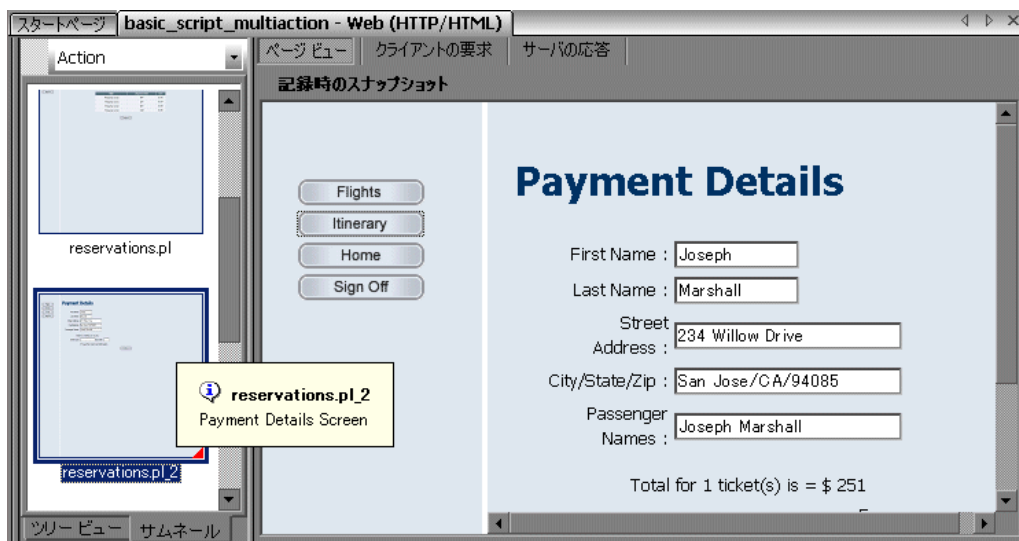
- 1 サムネイルを選択し、右クリックして表示されるメニューから **[コメント]** を選択するか、**Alt** キーを押しながら **F2** キーを押します。**[サムネイル コメント]** ダイアログ・ボックスが開きます。



- 2 **[サムネイル コメント]** ダイアログ・ボックスの右側の表示枠にテキストを入力します。
- 3 **[OK]** をクリックして注釈を保存し、ダイアログ・ボックスを閉じます。

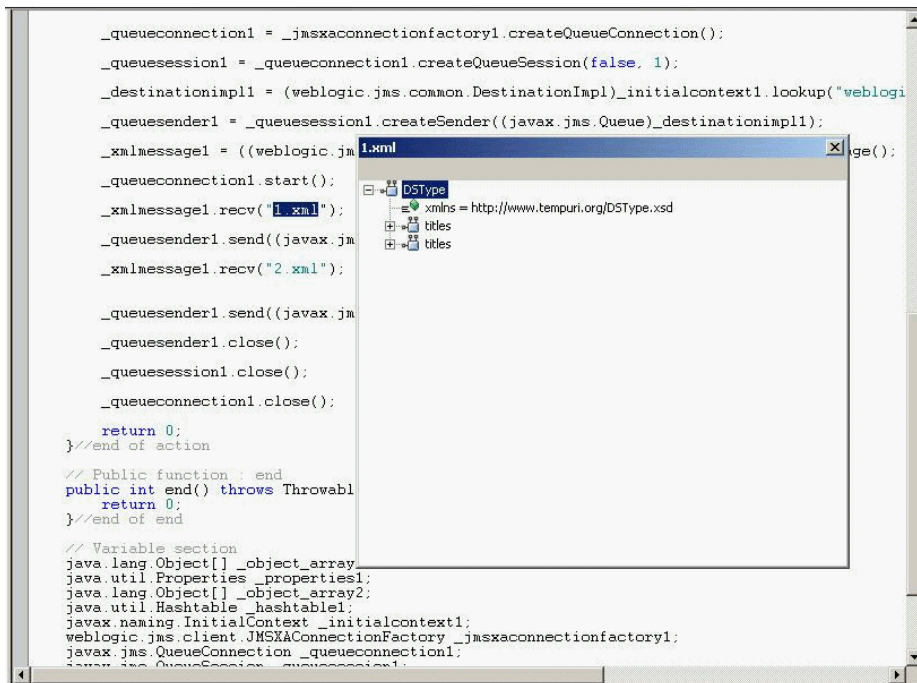
[サムネイル コメント] ダイアログ・ボックスを開いたままにして、テキストの追加やほかのスナップショットでの作業を行うには、左上角にある **[常に表示]** を選択します。**[OK]** ボタンが **[適用]** に変わります。

サムネイルに注釈を挿入すると、右下隅に注釈があることを示す赤いマークが付きます。サムネイルの上にマウスを移動すると、注釈テキストのポップアップが表示されます。



XML ビューアの使用

JMS などの XML ファイルを使用するプロトコルについては、VuGen では、編集ウィンドウから XML 構造を直接表示できます。ビューアに XML 要素が表示され、それぞれのノードを折りたたんだり展開したりできます。



XML ビューアにファイルを表示するには、次の手順を実行します。

- 1 ツリー・ビューで XML ファイルを含むステップを選択し、[XML の表示] を選択します。
- 2 スクリプト・ビューで XML ファイル名を右クリックし、[XML の表示] を選択します。

VuGen を使った Vuser スクリプトの実行

Vuser スクリプトを使ってテストを実行したりアプリケーションを監視したりするには、スクリプトを LoadRunner のシナリオ、または Business Process Monitor のプロファイルに組み込む必要があります。これを行う前に、VuGen でスクリプトを実行してその動作を検証します。詳細については、第 9 章「スタンドアロン・モードでの Vuser スクリプトの実行」を参照してください。

スクリプトの再生が成功したら、LoadRunner のシナリオ、Performance Center の負荷テスト、または Business Process Monitor のプロファイルなどの環境にスクリプトを組み込むことができます。詳細については、**HP LoadRunner Controller**、**HP Performance Center**、または **HP Business Availability Center** を参照してください。

Vuser スクリプトを実行する前に、実行環境の設定を変更できます。これらの設定の中には、Vuser の反復実行回数や、スクリプトの実行時に Vuser に適用される反復のペースおよび思考遅延時間も含まれます。実行環境の設定の詳細については、第 22 章「実行環境の設定」を参照してください。

Vuser スクリプトを実行すると、スクリプトがインタプリタによって解釈および実行されます。スクリプトをコンパイルする必要はありません。スクリプトに変更を加えたときに、スクリプトに何らかの構文エラーが持ち込まれた場合には、そのエラーはインタプリタによって検出されます。スクリプトからインタプリタが認識して実行できる外部関数を呼び出すこともできます。詳細については、付録 29「外部関数の呼び出し」を参照してください。

上級ユーザは、記録されたスクリプトをコンパイルして実行プログラムを作成することもできます。詳細については、第 6 章「Vuser スクリプトの拡張」を参照してください。

VuGen のコードについて

Vuser スクリプトを記録すると、VuGen は Vuser 関数を生成してスクリプトに挿入します。Vuser 関数には、次の2つのタイプがあります。

- ▶ 一般 Vuser 関数
- ▶ プロトコル固有 Vuser 関数

一般 Vuser 関数とプロトコル固有の関数によって LoadRunner API が構成されます。Vuser は、この API によってサーバと直接通信できるようになります。VuGen では新しいスクリプトを作成するときに、サポートしている全プロトコルのリストが表示されます。全 Vuser 関数の構文については、『**Online Function Reference**』（英語版）（[ヘルプ] > [関数リファレンス]）を参照してください。

一般 Vuser 関数

一般 Vuser 関数は、関数名に **lr** というプレフィックスが付いているので LR 関数とも呼ばれます。LR 関数はどのタイプの Vuser スクリプトでも使えます。LR 関数を使って次のようなことができます。

- ▶ Vuser, Vuser グループ, およびホストに関する実行情報の取得。
- ▶ Vuser スクリプトへのトランザクションと同期ポイントの追加。たとえば, **lr_start_transaction** (Java では **lr.start_transaction**) 関数はトランザクションの開始を, **lr_end_transaction** (Java では **lr.end_transaction**) 関数はトランザクションの終了を示します。詳細については、第6章「Vuser スクリプトの拡張」を参照してください。
- ▶ エラーや警告を示すメッセージの出力への送信。

LR 関数のリストは、54 ページ「Vuser 関数に関するヘルプについて」を参照してください。詳細については、『**Online Function Reference**』（英語版）（[ヘルプ] > [関数リファレンス]）を参照してください。

プロトコル固有 Vuser 関数

Vuser 関数に加えて、VuGen は記録中にプロトコル固有の関数を生成して、Vuser スクリプトに挿入します。

プロトコル固有の関数は、記録対象となっている Vuser のタイプに固有のもので、VuGen は、たとえばデータベース・スクリプトには LRD 関数を、Tuxedo スクリプトには LRT 関数を、Windows Sockets スクリプトには LRS 関数を挿入します。

標準設定では、VuGen の自動スクリプト・ジェネレータは、ほとんどのプロトコルで C 言語の Vuser スクリプトを生成し、Java タイプ・プロトコルには Java の Vuser スクリプトを生成します。VuGen に、Visual Basic や Javascript でコードを生成させることができます。詳細については、第 19 章「スクリプト生成オプションの設定」を参照してください。

フロー制御や構文も含め、言語のすべての標準規則がスクリプトに追加できます。ほかのプログラミング言語と同様、スクリプトにコメントや条件ステートメントを追加することもできます。

次の Web Vuser スクリプト（抜粋）は、VuGen によって記録および生成され、スクリプトに挿入される関数の例です。

```
#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");
    ...
}
```

Vuser スクリプトで C 言語の関数を使用する方法の詳細については、第6章「Vuser スクリプトの拡張」を参照してください。Java スクリプトを変更する方法の詳細については、『第2巻 – プロトコル』の「Java Vuser スクリプトのプログラミング」を参照してください。

注： C 言語で記述された Vuser スクリプトの実行に使用される C インタプリタは、ANSI C 言語だけをサポートします。Microsoft 社による ANSI C への拡張はサポートしていません。

Vuser 関数に関するヘルプについて

任意のスクリプトに API Vuser 関数を追加して、機能を拡張できます。VuGen では、記録中に Vuser 関数が生成されます。必要に応じて、記録後にスクリプトに手作業で追加の関数を挿入できます。一般的な拡張の詳細については、第6章「Vuser スクリプトの拡張」を参照してください。

VuGen の API 関数に関する情報を得るには、次のように複数の方法があります。

- ▶ オンライン関数リファレンス
- ▶ 単語の補完
- ▶ 関数構文の表示
- ▶ ヘッダ・ファイル

さらに、標準の検索機能を使用して（[編集] > [検索]）スクリプト内の関数を見つけたり、または 156 ページの「ファイル内の検索」機能を使用して、スクリプト内のファイルをすべて検索したりできます。

オンライン関数リファレンス

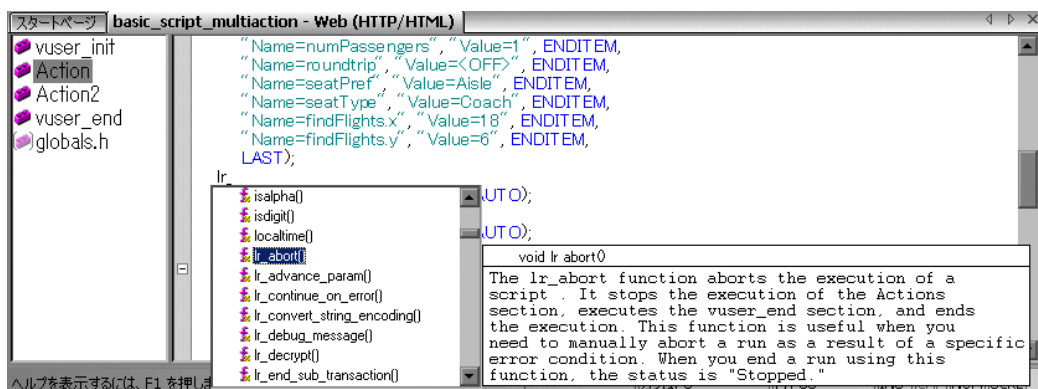
『**Online Function Reference**』（英語版）には、すべての VuGen 関数についての詳しい構文情報や関数の例が含まれています。関数の使用例も含まれています。関数名の検索が可能のほか、カテゴリまたはアルファベット順のリストから参照できます。

『**Online Function Reference**』(英語版)を開くには、VuGen インタフェースから [ヘルプ] > [関数リファレンス] を選択します。次に、プロトコルおよび必要なカテゴリを選択します。

スクリプトにすでに含まれている特定の関数についての情報を取得するには、VuGen エディタのカーソルをその関数に移動して、F1 キーを押します。

単語の補完

VuGen エディタには、IntelliSense 拡張機能 I の一部として**単語の補完**機能が組み込まれています。関数のタイプ入力を始めたときに最初のアンダーバーを入力すると、その関数のプレフィックスに一致する使用可能なすべての関数の構文と説明を示すリスト・ボックスが開きます。

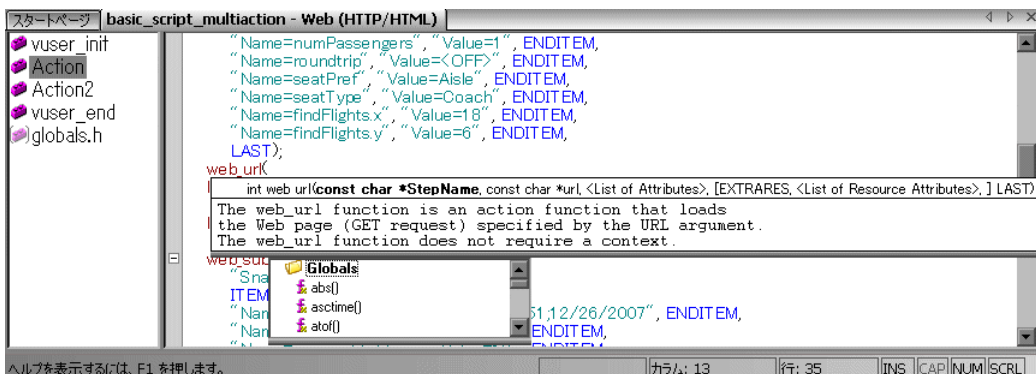


表示された関数を使用するには、その項目を選択するか、目的の項目までスクロールしてから選択します。選択した関数がカーソルの位置に挿入されます。リスト・ボックスを閉じるには、Esc キーを押します。

標準設定では、単語の補完機能は VuGen 全体で有効になっています。単語の補完機能を無効にするには、[ツール] > [一般オプション] から [環境] タブを選択します。[単語の自動補完] オプションの横にあるチェック・ボックスをクリアします。単語の補完機能を VuGen 全体で無効にしている場合でも、Ctrl キーを押しながらスペース・キーを同時に押すか、エディタで入力しながら [編集] > [単語入力候補] を選択すれば、自動補完のリスト・ボックスを表示できます。

関数構文の表示

VuGen の IntelliSense には、このほかにも**関数構文の自動表示**機能があります。関数の開始括弧を入力すると、関数の構文、引数、プロトタイプ、および簡単な説明が自動的に表示されます。



標準設定では、**関数構文の自動表示**機能が VuGen 全体で有効になっています。この機能を無効にするには、[ツール] > [一般オプション] から [環境] タブを選択します。[関数構文の自動表示] チェック・ボックスをクリアします。

[関数構文の自動表示] オプションを VuGen 全体で無効にしている場合でも、エディタ画面で開始括弧を入力した後に、Ctrl キーと Shift キーを押しながらスペース・キーを押すか、[編集] > [関数の構文を表示] を選択すれば、構文を表示させることができます。

ヘッダ・ファイル

Java 以外の関数のプロトタイプはすべてライブラリ・ヘッダ・ファイルに含まれています。ヘッダ・ファイルは、製品のインストール先ディレクトリの下**の include** ディレクトリにあります。ヘッダ・ファイルには、詳しい構文情報と戻り値が含まれています。また、定数の定義、適用範囲、その他の詳細情報が含まれており、関数リファレンスにはない情報もあります。

ほとんどの場合、ヘッダ・ファイル名はプロトコルのプレフィックスに対応しています。たとえば、**lrd** というプレフィックスで始まるデータベース関数のリストは、**lrd.h** ファイルにあります。

次の表に、使用頻度の高いプロトコルと、対応するヘッダ・ファイルを示します。

| プロトコル | ファイル |
|-------------------------|--------------------|
| AJAX (Click and Script) | web_ajax.h |
| Citrix | ctrxfuncs.h |
| COM/DCOM | lrc.h |
| データベース | lrd.h |
| FTP | mic_ftp.h |
| 一般的な C 関数 | 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 |
| SAPGUI | as_sapgui.h |
| SAP (Click and Script) | sap_api.h |
| Siebel | lrsiebel.h |
| SMTP | mic_smtp.h |
| ターミナル・エミュレータ | lrrte.h |
| WAP | as_wap.h |
| Web (HTML&HTTP) | as_web.h |
| Web (Click and Script) | web_api.h |
| Web サービス | wsoap.h |
| Windows Sockets | lrs.h |

第 3 章

プロトコル・アドバイザーの使用方法

VuGen では、Vuser スクリプトを作成するために使用できるさまざまなプロトコルが提供されます。プロトコル・アドバイザーは、ビジネス・プロセスを分析し、スクリプトを記録するためにプロトコルを選択するためのガイダンスを提供します。

本章の内容

- ▶ プロトコル・アドバイザーについて (59 ページ)
- ▶ プロトコル・アドバイザーのワークフロー (60 ページ)
- ▶ 注意と制限事項 (63 ページ)

プロトコル・アドバイザーについて

プロトコル・アドバイザーは、Vuser スクリプトを記録するために適切なプロトコルを決定するのに役立ちます。プロトコル・アドバイザーは、アプリケーションで異なるプロトコルの要素をスキャンし、検出したプロトコルのリストを表示します。これらのプロトコルは、アプリケーションに適したプロトコルを検出するガイドラインおよび起点として使用します。

ほとんどの場合、プロトコルの組み合わせとともに、複数のプロトコルが提案されて表示されます。次の項では、提案されたプロトコルのリストを使用する方法に関するガイドラインを提供します。

プロトコル・アドバイザーのワークフロー

次の手順は、プロトコル・アドバイザーを使用して、アプリケーションを記録するために最適なプロトコルを検出するための一般的なワークフローを示します。

1 プロトコル・アドバイザーを実行します。



- a スタート・ページで [ファイル] > [プロトコル アドバイザ] > [アナライザ アプリケーション] を選択するか、[プロトコル アドバイザ] ボタンをクリックします。[プロトコル アドバイザ] ダイアログ・ボックスが開きます。
- b 次のようにフィールドに入力します。
 - ▶ [アプリケーション タイプ]: アプリケーションの種類を選択します。
 - ▶ [分析するプログラム]: 分析対象のプログラムを選択します。
 - ▶ [URL アドレス]: このフィールドはインターネット・アプリケーションのみを対象としています。開始する URL アドレスを指定します。
 - ▶ [プログラム引数]: このフィールドは Win32 アプリケーションのみを対象としています。前述で指定した実行ファイルのコマンド・ライン引数を指定します。たとえば、**plus32.exe** にプログラム引数 **peter@neptune** を指定した場合、**plus32.exe** を起動すると、ユーザ **Peter** がサーバ **Neptune** に接続されます。
 - ▶ [作業ディレクトリ]: 作業ディレクトリを指定する必要があるアプリケーションの場合は、ここで指定します。必要となる情報は、Vuser スクリプトのタイプによって異なります。
- c [OK] をクリックします。分析の進行状況を示すボックスが表示されます。
- d アプリケーションで一般的なビジネス・プロセスを実行します。包括的な結果が得られるように、さまざまなビジネス・プロセスを実行してください。
- e [分析を停止] をクリックして分析を終了します。

[プロトコル アドバイザ結果] ページに結果が表示されます。

2 結果を保存します。

[ファイル] > [プロトコル アドバイザ] > [結果の保存] を選択します。名前を入力してディレクトリを選択します。

3 使用するプロトコルを選択します。

次の優先順位に従ってプロトコルを選択することをお勧めします。

- ▶ 複数のプロトコル/プロトコルの組み合わせ
- ▶ 最上位レベルのアプリケーション・プロトコル (61 ページの手順6 の図を参照)。
- ▶ リストで一番上に表示されているプロトコル

4 Vuser スクリプトを新規作成します。

選択したプロトコルに関する項に従って新しい Vuser スクリプトを作成します。

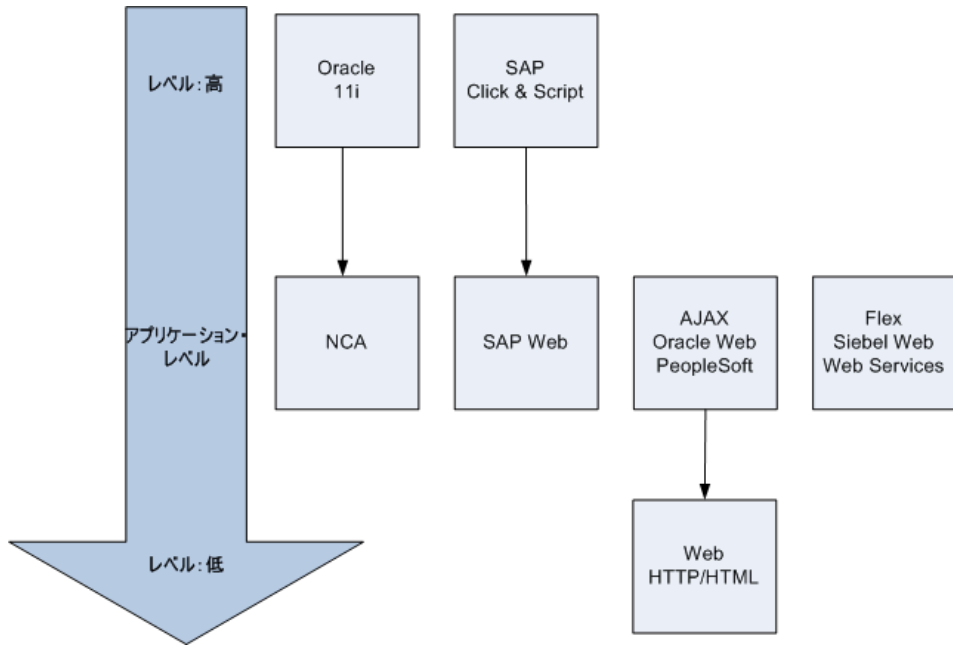
5 再生を拡張, デバッグ, 検証します。

再生に成功するまでスクリプトを拡張してデバッグします。Vuser スクリプトを拡張およびデバッグした後に, 再生に成功しない場合は, 次の手順に進みます。

6 スクリプトを再生できない場合は, 別のプロトコルを選択します。

- ▶ **Web ベースでないすべてのプロトコル** : [プロトコルアドバイザー結果] ページに戻り, リストで次のプロトコルを選択して手順2 から続行します。ほかにプロトコルがない場合は, HP のカスタマ・サポートに連絡してください。
- ▶ **Web ベースのプロトコル** : 次の図は, アプリケーション・レベルに基づいて配置した Web ベースのプロトコルの例を示します。矢印は, 新しいプロトコルを試す順序を示しています。たとえば, 最初に使用したプロトコルが Oracle Web アプリケーション 11i だった場合, このプロトコルの再生に失敗したら, 次に Oracle NCA プロトコルを試します。

第3章・プロトコル・アドバイザの使用法



注意と制限事項

次の注意と制限事項について検討してください。

- ▶ この機能は、LoadRunner でサポートされているプロトコルのみを検出します。
- ▶ 一部の Web プロトコルは URL に基づいて検出されます。たとえば、URL に SAP のようなキーワードが含まれている場合があります。このため、異なる URL または同じ URL を使う異なるアプリケーションを使用する場合は、結果が異なる場合があります。
- ▶ 次のプロトコルは頻繁に検出されますが、必ずしも使用に適していません。ほかにプロトコルが検出されなかった場合以外は使用しないでください。
 - ▶ COM/DCOM
 - ▶ Java
 - ▶ .NET
 - ▶ WinSocket
 - ▶ LDAP

第4章

VuGen ワークフローの表示

VuGen のワークフロー画面を使って、負荷テストや監視に使用できる Vuser スクリプトを作成できます。

本章の内容

- ▶ VuGen ワークフローの表示について (65 ページ)
- ▶ タスク表示枠の表示 (66 ページ)
- ▶ ステップの記録 (67 ページ)
- ▶ スクリプトの検証 (69 ページ)
- ▶ スクリプトの拡張 (70 ページ)
- ▶ ロードの準備 (76 ページ)
- ▶ スクリプトの完了 (77 ページ)
- ▶ ワークフロー・ビューの復元 (77 ページ)

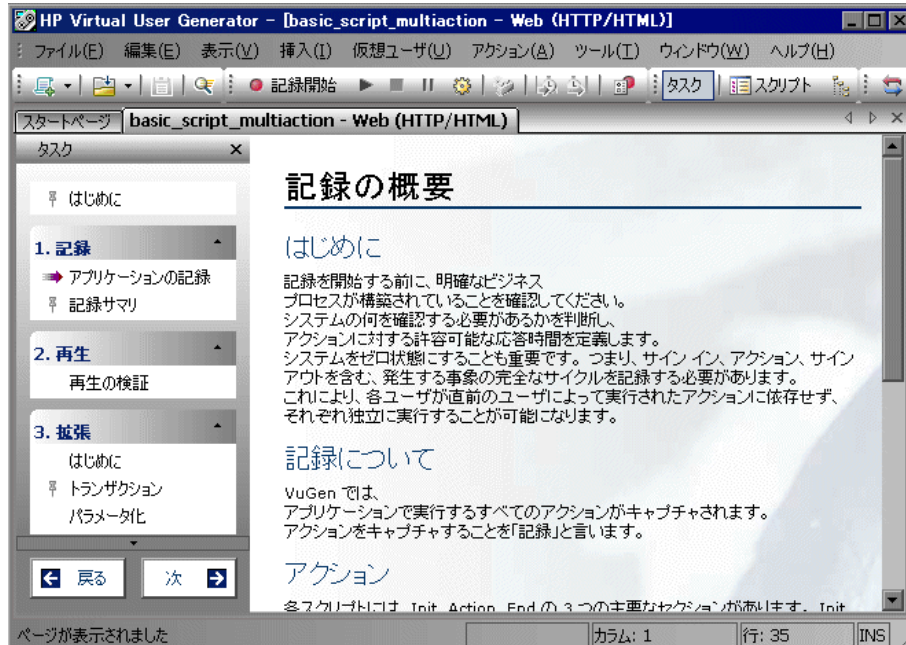
VuGen ワークフローの表示について

VuGen のワークフロー画面には、スクリプト作成における各工程が示されます。

スクリプト・ビューやツリー・ビューでも作業できます。次回 VuGen を起動してスクリプトを開くと、VuGen の終了時に開いていたビューが開きます。タスク表示枠でタスクをクリックすると、ウィザード・ビューに戻ります。

タスク表示枠の表示

VuGen の左側の表示枠には機能スクリプトの作成に必要なタスクの一覧が表示されます。一覧でタスクをクリックすると、ウィザードでそのステップが開きます。現在のタスクは矢印で示されます。



タスクのリストは、5つのパートに分かれています。[記録] (C Vuser および Web サービス仮想ユーザでのスクリプト作成), [再生], [拡張], [ロードの準備], [完了] です。

Web, Web Services, 記録不可能なプロトコル (ユーザ定義 C Vuser) では、最初のタスクが若干異なります。

タスクの多くにはサブ・タスクがあります。次の表にサブ・タスクを表示します。

| タスク | サブ・タスク |
|----------|--|
| 記録 | アプリケーションの記録, 記録サマリ (記録可能なプロトコル) |
| スクリプトの作成 | スクリプトの作成, 作成サマリ (Web Services, C の場合) |
| 再生 | 再生の検証 |
| 拡張 | はじめに, トランザクション, パラメータ化, コンテンツ・チェック (Web Vuser 用) |
| ロードの準備 | はじめに, 反復, 同時実行ユーザ |

ステップの記録

[記録] セクション (Web Services, C, 記録不能なプロトコルは除外) には, [アプリケーションの記録] と [記録サマリ] の2つのステップがあります。

アプリケーションの記録

このウィザード・ステップでは記録プロセスを概説します。次のセクションが含まれます。

- ▶ [はじめに]
- ▶ [記録について]
- ▶ [アクション]
- ▶ 記録オプション
- ▶ [記録開始]

記録サマリ

このウィザード・ステップでは, プロトコル情報とセッションが記録されるアクションを含む記録のサマリが提供されます。

このステップでは、記録されたスナップショットのサムネイルも提供されます。

The screenshot displays the 'Recording Summary' window in VuGen. The window title is 'basic_script_multiaction - Web (HTTP/HTML)'. On the left, a 'タスク' (Tasks) pane shows a list of steps: 1. 記録 (Recording), 2. 再生 (Playback), 3. 拡張 (Extension), 4. ロードの準備 (Load Preparation), and 5. 完了 (Completed). The '記録' step is expanded, showing sub-tasks like 'アプリケーションの記録' and '記録サマリ'. The main area is titled '記録サマリ' (Recording Summary) and is divided into sections: 'プロトコル' (Protocols), 'アクション' (Actions), and 'トランザクション' (Transactions). Each section contains a table of recorded items with their status. The '記録' sidebar on the right shows three thumbnails of captured screenshots labeled 'login.pl', 'welcome.pl', and another 'welcome.pl'.

記録サマリ

プロトコル
記録セッション中に次のプロトコルが検出されました:

| 選択されたプロトコル | データが生成されました |
|-----------------|-------------|
| Web (HTTP/HTML) | ✓ |

アクション
記録セッション中に次のアクションが作成されました:

| スクリプト アクション | データが生成されました |
|-------------|-------------|
| user_init | ✓ |
| Action | ✓ |
| Action2 | ✓ |
| user_end | ✓ |

トランザクション
次のトランザクションがスクリプトに挿入されました:

- Login
- NewTransaction
- Trans1
- Trans4
- Trans2
- Trans2a

記録情報の詳細については、[記録ログを開いてください。](#)

スクリプトの検証

[再生] セクションには、[再生の検証] ステップが含まれます。スクリプトを再生すると、[完了] ステップの下の [一般] セクションにある [再生のサマリ] をクリックして、いつでも再生サマリを表示できます。

再生の検証

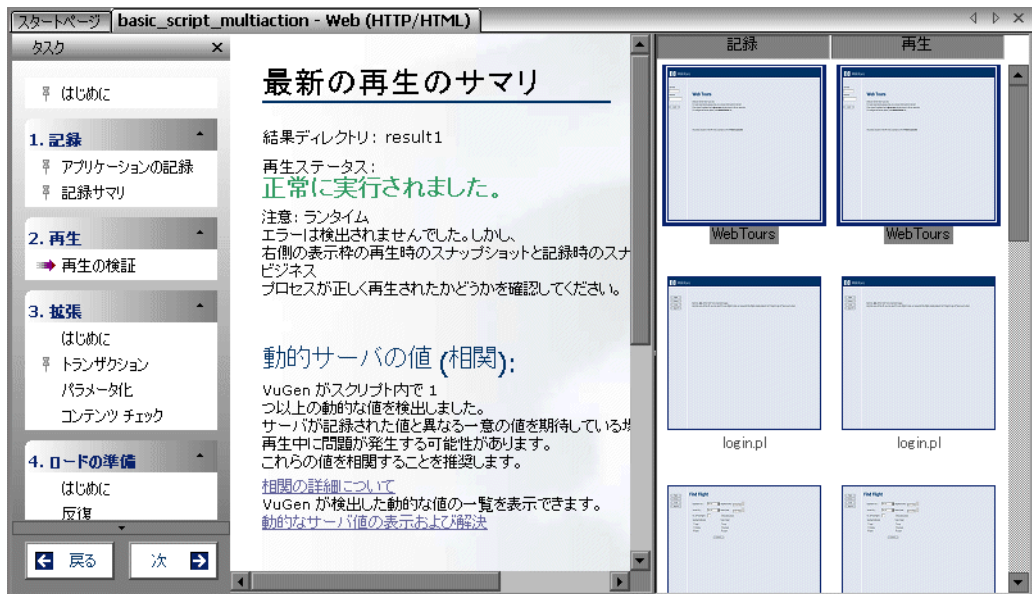
このウィザード・ステップでは検証を概説します。次のセクションが含まれます。

- ▶ [再生について]
- ▶ [実行環境の設定]
- ▶ [再生の前に] (再生中の検索対象)

再生のサマリ

このウィザード・ステップでは、再生のサマリが提供されます。エラーを一覧表示し、再生ログにエラーへのリンクをはります。

[再生のサマリ] では、[記録と再生] のスナップショットのサムネイルも表示されます。スナップショットを目で見て比較し、不一致を調べることができます。



注：反復が複数の場合、[再生サマリ] ウィンドウには最後の反復の再生のサムネイルが表示されます。特定の反復のサムネイルを表示するには、[表示] > [ツリー ビュー] を選択してツリー・ビューに切り替えます。[表示] > [スナップショット] > [反復の選択] を選択し、希望の反復を選びます。

スクリプトの拡張

[拡張] セクションには3つの主要なステップがあります。

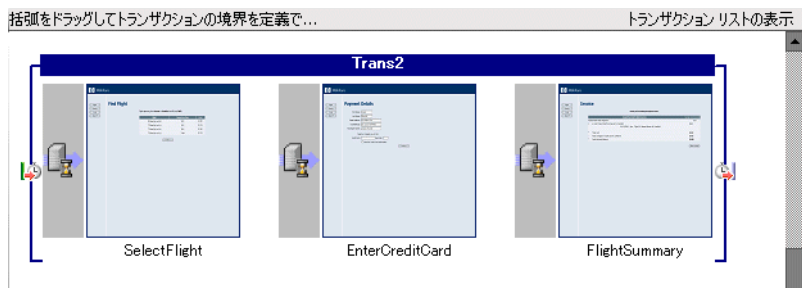
- ▶ トランザクション
- ▶ パラメータ化
- ▶ コンテンツ・チェック

トランザクション

VuGen では、**トランザクション・エディタ**を使用して、スクリプトのサムネイル・ビューから直接トランザクションを追加および管理できます。

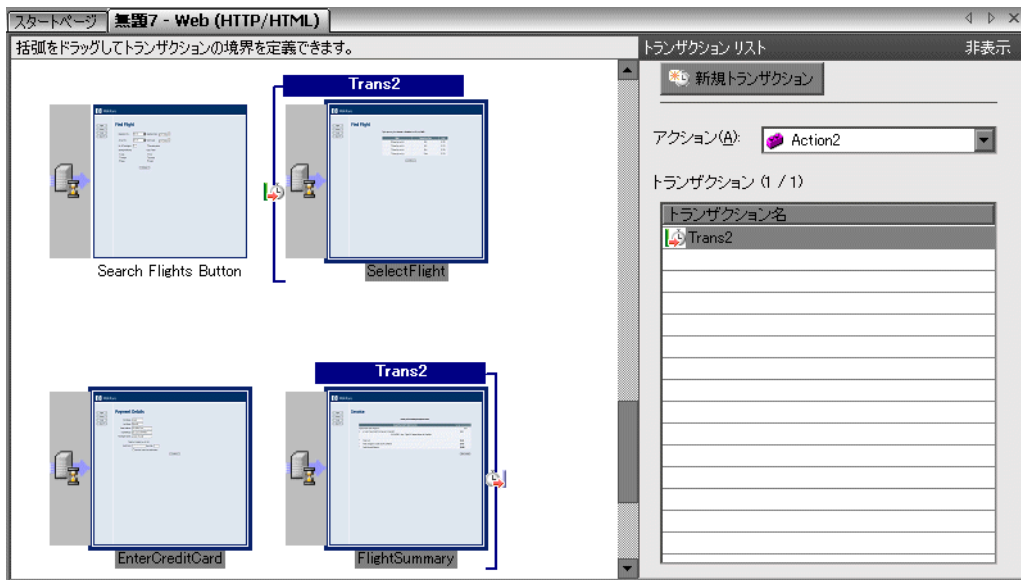
標準設定では、スクリプトの主要なステップのサムネイルのみが表示されます。スクリプト内のすべてのステップのサムネイルを表示するには、[表示] > [すべてのサムネイルを表示] を選択します。

次の例では、**Trans2** は **Select Flight**, **Enter Credit Card**, **Flight Summary** という3つのステップを測定しています。



トランザクション・リストでの作業

トランザクション・エディタの右側の表示枠にあるトランザクション・リストに、スクリプト内のトランザクションのリストが表示されます。スクリプト内のトランザクションのリスト全体を表示することもできますし、特定のアクションのトランザクションのみを表示することもできます。



すべてのトランザクションを表示するには、[アクション] ボックスで [全て] を選択します。特定のアクションにおけるトランザクションを表示するには、[アクション] ボックスでアクション名を選択します。



トランザクションのリストの表示 / 非表示を切り替えるには、VuGen ウィンドウの右上隅にある [非表示] ボタンまたは [トランザクション リストの表示] ボタンをクリックします。

標準設定では、トランザクション・リストには、スクリプトの主要なステップに対するサーバの応答を測定する、エラーの生じないトランザクションのみが表示されます。次の項目は表示されません。

- ▶ 主要でないステップ
- ▶ クライアント側のトランザクション
- ▶ エラーの生じたトランザクション

したがって、トランザクション・リストの上部に次のようなキャプションが表示される場合があります。トランザクション (2/4)。

主要でないトランザクションやクライアント側のトランザクションといった非表示のトランザクションを表示するには、トランザクション・リスト下部の**[非表示のトランザクションを表示]**の隣のボタンをクリックします。非表示のトランザクションが灰色で表示されます。非表示にするには、再度ボタンをクリックします。

エラーの生じたトランザクションはサーバ・ステップを測定しないトランザクションか、使用できない名前のついたトランザクションです。エラーの生じたトランザクションを表示するには、**[エラーのあるトランザクションを表示]**ボタンをクリックします。VuGenにより、エラーの生じたトランザクションが赤で表示されます。非表示にするには、再度ボタンをクリックします。

主要でないステップのトランザクションを表示するには、すべてのサムネイルを表示する必要があります。**[表示] > [すべてのサムネイルを表示]**を選択します。トランザクション・エディタに、スクリプト内のすべてのステップのサムネイルとトランザクションが表示されます。

トランザクション・エディタでのトランザクションの定義

トランザクションの開始と終了のサムネイルに印を付けて、トランザクション・エディタ内のトランザクションを定義します。

トランザクションを定義するには、次の手順を実行します。

- 1 [タスク] リストで**[トランザクション]**をクリックしてトランザクション・エディタを開きます。
- 2 サムネイル領域（中央の表示枠）で、トランザクションとして印を付けるステップまでスクロール・ダウンします。

ヒント：より多くのサムネイルを表示するには、ツールバーの**[タスク]**ボタンをクリックして、[タスク] リストを非表示にします。

- 3 単一のステップをトランザクションとして印を付けるには、サムネイルをクリックして右クリック・メニューから**[新規シングル ステップ トランザクション]**を選択します。新しいトランザクションの名前を入力するダイアログ・ボックスが表示されます。後からトランザクションを拡張するには、トランザクションの括弧をドラッグしてトランザクションに追加のステップを含めます。

- 4 トランザクションとして複数のステップに印を付けるには、サムネイル領域をクリックして右クリック・メニューから **[新規トランザクション]** を選択するか、右側の表示枠の上部にある **[新規トランザクション]** ボタンをクリックします。
- 5 VuGen により、次のようにサムネイルの上部のステータス領域に手順が示されます。

[ステップ 1/3 新規トランザクションの開始位置を選択します。]

トランザクションの最初のステップのサムネイルをクリックします。

[ステップ 2/3 新規トランザクションの終了位置を選択します。]

トランザクションの最後のステップのサムネイルをクリックします。

[ステップ 3/3 新規トランザクションの名前を指定します。]

トランザクションの最初のステップの上部の大括弧内に、直接トランザクション名を入力します。

トランザクションを完了するには、Enter キーを押します。

前述の手順の最中にトランザクションを終了するには、Esc キーを押します。

- 6 トランザクションの開始点を変更するには、トランザクションの左大括弧を新しい場所にドラッグします。トランザクションの終了点を変更するには、トランザクションの右大括弧を新しい場所にドラッグします。
 - 7 トランザクションを追加するには、前述のステップを繰り返します。
- ▶ トランザクション名を変更するには、右側の表示枠でタイトルを選択して、右クリック・メニューで **[名前変更]** を選びます。新しい名前を入力してください。
 - ▶ トランザクションを削除するには、右側の表示枠でタイトルを選択して、右クリック・メニューから **[削除]** を選びます。

トランザクション・エディタのガイドライン

トランザクション・エディタでトランザクションを作成および定義するには、これらのガイドラインに従います。

- ▶ トランザクションは単一のアクション内で開始し終了しなければなりません。トランザクションは複数のアクションにまたがって拡張できません。

- ▶ トランザクション名は、複数のアクションにおいてでも、スクリプト内で一意でなくてはなりません。
- ▶ トランザクションの追加、名前の変更、削除には、右クリック・メニューを使用します。
- ▶ 既存のトランザクション内でトランザクションを作成できます。これらは「入れ子」のトランザクションと呼ばれます。

注：トランザクションを入れ子にする場合は、2番目のトランザクションを1番目のトランザクションより前に閉じるか、同時に閉じる必要があります。そうしないと、正しく分析されません。

パラメータ化

[パラメータ化] 画面には、スクリプト内のパラメータ化された値の概要が提供されます。パラメータ化のステップが示されます。

- ▶ パラメータ化する引数を検索します。
- ▶ パラメータに名前を付けます。
- ▶ パラメータのタイプを選択します。
- ▶ パラメータのタイプのプロパティを定義します。
- ▶ 引数をパラメータに置換します。

スクリプト内で引数をパラメータ化したら、[**拡張**] ステップに戻ることも、スクリプトを再生することもできます。

コンテンツ・チェック

[コンテンツ チェック] ウィザードのステップでは、再生した Web ページの特定のテキストまたはコンテンツをチェックできます。

[**テキスト チェック**] を使用して、再生中にテキスト文字列を検索できます。

[**コンテンツ チェック**] を使用して、サーバによって返されるテキストが厳密にわからない場合にも、VuGen が定義済みの規則を使って自動的にサーバの文字列を検索することができます。

ロードの準備

ワークフロー・ウィザードの4番目のステップは、システムにおいて負荷テストの実行し、マシンの応答と処理能力を検証するために重要です。このステップには次の2つの主要なステップがあります。

- ▶ 反復
- ▶ 同時ユーザ

反復

このウィザードのステップは、反復について説明し、[実行環境の設定]を開いて値を設定できるようにします。

反復回数を設定するには、次の手順を実行します。

- 1 [実行環境設定] (F4) を開きます。
- 2 [実行論理] ノードを選択します。
- 3 反復回数を指定します。

同時ユーザ

このステップでは、LoadRunner Controller を使用してシナリオを作成するプロセスが示されます。

シナリオでは、同時に実行するユーザの数を指定して、複数ユーザを稼動しているシステムの振る舞いを観察できます。

スクリプトの完了

ワークフロー・ウィザードの最終ステップは **[完了]** です。

[完了] には、次のセクションが含まれます。

- ▶ **[シナリオの作成]** : LoadRunner Controller を使用してシステムを対象に負荷テストを実行します。
- ▶ **[Performance Center へのアップロード]** : Performance Center サーバのインストール先からテストを実行します。
- ▶ **[Quality Center へのアップロード]** : テスト・レポジトリにテストを追加します。
- ▶ **[ビジネス・プロセス・レポートを作成]** : VuGen スクリプトのサマリを含む Microsoft Word ドキュメントを作成します。

ワークフロー・ビューの復元

VuGen にワークフロー・ビューが表示されなくなった場合は、簡単に表示できません。

- ▶ **[タスク]** 表示枠が左側表示枠に表示されていることを確認してください。表示されていない場合は、ツールバーの **[タスク]** ボタンをクリックします。
- ▶ 最上部の **[はじめに]** リンクをクリックします。右側表示枠に1つ目のワークフロー画面が表示されます。

ヒント : 再生が終了するたびにワークフロー・ビューを表示するように VuGen に指示するには、**[再生]** タブ (**[ツール]** > **[一般オプション]**) を開き、**[再生後]** セクションで **[再生のサマリ]** を選択します。

第 5 章

VuGen を使った記録

VuGen はクライアント・アプリケーションとサーバの間の通信を記録することによって、Vuser スクリプトを作成します。

本章の内容

- ▶ VuGen を使った記録について (80 ページ)
- ▶ Vuser スクリプトのセクション (80 ページ)
- ▶ 仮想ユーザ・スクリプトの新規作成 (82 ページ)
- ▶ プロトコルの追加と削除 (86 ページ)
- ▶ 仮想ユーザ・カテゴリの選択 (87 ページ)
- ▶ ユーザ定義テンプレート (88 ページ)
- ▶ スクリプトの新規作成 (90 ページ)
- ▶ 既存のスクリプトを開く (91 ページ)
- ▶ アプリケーションの記録 (92 ページ)
- ▶ 記録セッションの終了と保存 (96 ページ)
- ▶ 記録ログの表示 (97 ページ)
- ▶ Zip ファイルの使用 (99 ページ)
- ▶ アクションのインポート (100 ページ)
- ▶ 認証情報の提供 (101 ページ)
- ▶ Vuser スクリプトの再生成 (103 ページ)

VuGen を使った記録について

VuGen は、クライアント・アプリケーションでのアクションを記録することによって、Vuser スクリプトを作成します。記録されたスクリプトを実行すると、Vuser はクライアントとサーバ間のユーザ操作をエミュレートします。

作成する各 Vuser スクリプトには、少なくとも次の3つのセクションがあります。vuser_init, 1つ以上の Actions, および vuser_end です。VuGen で記録中に、記録対象の関数を挿入する対象となるスクリプトのセクションを選択できます。一般的には、サーバへのログインを vuser_init セクションに、クライアントの動作を Actions セクションに、ログオフの手順を vuser_end セクションに記録します。

テストを作成した後、テストを Zip アーカイブに保存し、電子メールの添付ファイルとして送信できます。

記録中、スクリプトにトランザクション、コメント、ランデブー・ポイントを挿入できます。詳細については、第6章「Vuser スクリプトの拡張」を参照してください。

Vuser スクリプトのセクション

各 Vuser スクリプトには、少なくとも次の3つのセクションがあります。vuser_init, 1つまたは複数の Actions, および vuser_end です。記録前または記録中に、VuGen によって記録される関数の挿入先となるスクリプトのセクションを選択できます。次の表に、各セクションに何が記録され、各セクションがどのタイミングで呼び出されるかを示します。

| スクリプトのセクション | 何を記録するときに使われるか | 実行のタイミング |
|-------------|----------------|-----------------------|
| vuser_init | サーバへのログイン | Vuser を初期化 (ロード) するとき |
| Actions | クライアントの動作 | Vuser が「実行」状態のとき |
| vuser_end | ログオフの手順 | Vuser を終了または停止するとき |

Vuser スクリプトの反復を複数回実行するときは、スクリプトの **Actions** セクションだけが繰り返されます。**vuser_init** セクションと **vuser_end** セクションは繰り返されません。反復回数の設定の詳細については、第22章「実行環境の設定」を参照してください。

各スクリプト・セクションの内容の表示と編集には、VuGen スクリプト・エディタを使用します。一度に表示できるのは、1つのセクションの内容だけです。セクションを表示するには、左側の表示枠でセクションの名前を選択して強調表示します。

Java クラスを使用する Vuser スクリプトの場合には、すべてのコードを **Actions** クラスに置きます。**Actions** クラスには、**init**、**action**、**end** の3つのメソッドが含まれています。これらのメソッドは、ほかのプロトコルの場合に作成されるスクリプトの各セクションに対応します。つまり、初期化ルーチンは **init** メソッドに、クライアントの動作は **action** メソッドに、ログオフの手順は **end** メソッドに、それぞれ挿入します。詳細については、『第2巻－プロトコル』の「Java Vuser スクリプトのプログラミング」を参照してください。

```
public class Actions{
    public int init() {
        return 0;}
    public int action() {
        return 0;}
    public int end() {
        return 0;}
}
```

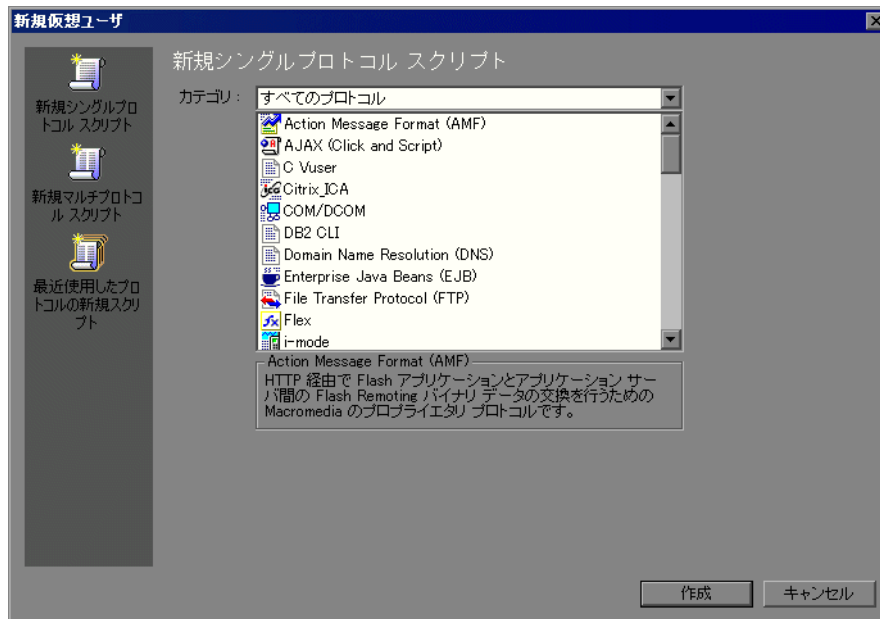
注： Oracle DB のトランザクション・ブレークダウンは、**vuser_init** セクションで記録されたアクションには使用できません。

仮想ユーザ・スクリプトの新規作成

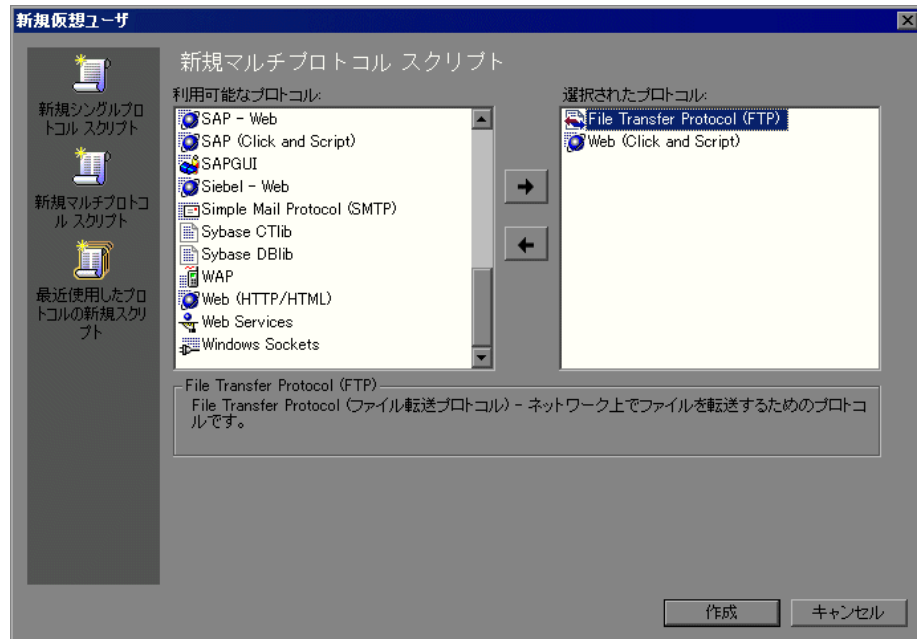
VuGen では、シングル・プロトコル・モードまたはマルチ・プロトコル・モードで記録することによって、スクリプトを新規作成できます。また、SOA（サービス指向アーキテクチャ）環境でスクリプトを作成するためのソリューションも提供されます。

[**新規作成**] をクリックすると、いつでも [新規仮想ユーザ] ダイアログ・ボックスが開きます。このダイアログ・ボックスには次のものへのショートカットがあります。

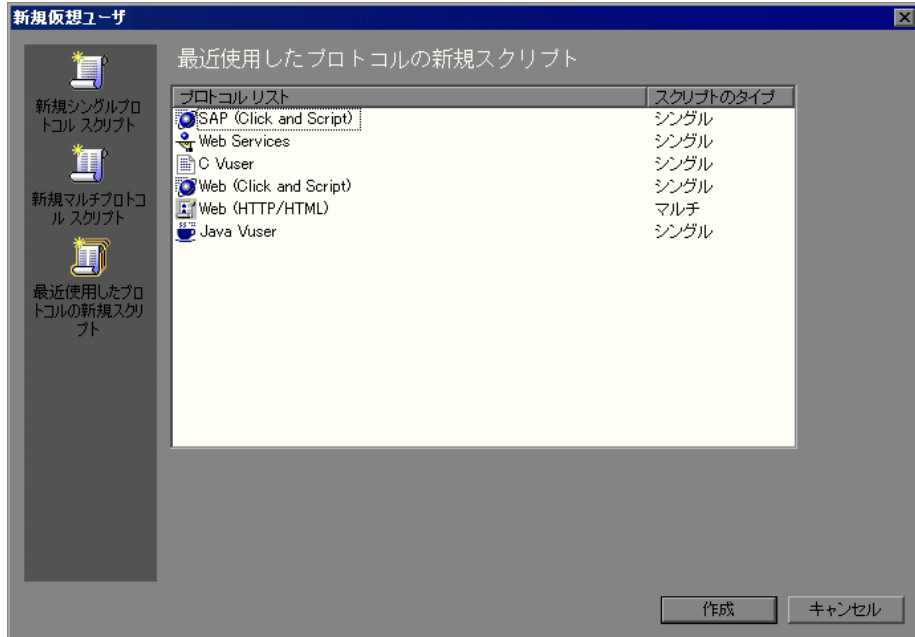
[**新規シングル プロトコル スクリプト**] : シングル・プロトコルの Vuser スクリプトを作成します。[カテゴリ] リスト（87 ページ「仮想ユーザ・カテゴリの選択」を参照）からカテゴリを選択し、そのカテゴリのプロトコルのリストでプロトコルを選択します。



[**新規マルチ プロトコル スクリプト**] : マルチ・プロトコルの Vuser スクリプトを作成します。VuGen には使用可能なすべてのプロトコルが表示され、どのプロトコルを記録するかを指定できます。マルチ・プロトコルのスクリプトを作成するには、プロトコルを選択し、右矢印をクリックしてプロトコルを [選択されたプロトコル] セクションに移動します。



[最近使用したプロトコルの新規スクリプト]：Vuser スクリプトの新規作成に使用された最近のプロトコルを一覧表示し、それらがシングル・プロトコルかマルチ・プロトコルかを示します。リストからプロトコルを選択して **[OK]** をクリックすると、そのプロトコルに対応するスクリプトが新規作成されます。



シングル・プロトコル・モードで記録する場合、VuGen は指定したプロトコルだけを記録します。マルチ・プロトコル・モードで記録する場合、VuGen はアクションを複数のプロトコルで記録します。マルチ・プロトコル・スクリプトは、次のプロトコルをサポートします。COM, FTP, IMAP, Oracle NCA, POP3, RealPlayer, Window Sockets (raw), SMTP, および Web。

Vuser タイプ間のもう 1 つの違いは、マルチ・アクションのサポートです。ほとんどのプロトコルは、複数の action セクションをサポートします。現在、次のプロトコルがマルチ・アクションをサポートしています。Oracle NCA, Web, RTE, General (C Vuser), WAP, i モード, VoiceXML

大部分の Vuser タイプでは、記録するたびに新しい Vuser スクリプトが作成されます。既存のスクリプトに記録することはできません。しかし、Java, Web, WAP, i モード, Oracle NCA, または RTE の Vuser スクリプトを記録する場合は、既存のスクリプトに記録することもできます。

VuGen はさまざまなプロトコルをサポートしているため、以降で説明する記録手順のいくつかは、特定のプロトコルにのみ適用されます。

すべての Java 言語 Vuser (CORBA, RMI, Jacada, および EJB) については、『第2巻 – プロトコル』の「Java プロトコル – 記録」を参照してください。

SOA (サービス指向アーキテクチャ) システムでは、デプロイメントの前にアプリケーションとサービスの安定性をテストすることは重要です。

LoadRunner VuGen を使用すると、基本的な Web サービス・スクリプトを作成できます。**HP LoadRunner with Service Test** および HP の SOA テスト・ツールには、SOA 環境に対する包括的なテスト・ソリューションを作成するのに役立つ追加機構が含まれています。**Service Test** の詳細については、HP の担当窓口にお問い合わせください。

プロトコルの追加と削除

マルチ・プロトコル・セッションを記録する前に、VuGen では、記録セッション中にコードを生成するプロトコルのリストを修正できます。スクリプトの作成時に特定のプロトコルを指定した場合は、プロトコルの記録オプションを使用してそれらを有効または無効にできます。

記録オプションを開くには、[ツール] > [記録オプション] を選択するか、または Ctrl キーを押しながら F7 キーを押します。[一般: プロトコル] ノードを選択します。

次回の記録セッションで記録するプロトコルについて、それらの横にあるチェック・ボックスを選択します。次回の記録セッションで記録しないプロトコルについて、それらの横にあるチェック・ボックスをクリアします。



AMF および Web プロトコルのプロトコル・オプションの設定の詳細については、305 ページ「AMF 記録モードの設定」を参照してください。

仮想ユーザ・カテゴリの選択

Vuser のタイプは次のカテゴリに分類されます。

- ▶ **[すべてのプロトコル]** : サポートされている全プロトコルのアルファベット順リスト。
- ▶ **[アプリケーションの導入ソリューション]** : Citrix および Microsoft Remote Desktop Protocol (RDP) プロトコル用。
- ▶ **[クライアント / サーバ]** : For DB2 CLI, DNS, Informix, Microsoft .NET, MS SQL, ODBC, Oracle 2-Tier, DB2 CLI, Sybase Ctlib, Sybase Dblib, および Windows Sockets プロトコル用。
- ▶ **[ユーザ定義]** : C テンプレート, Visual Basic テンプレート, Java テンプレート, VBNet, Javascript, および VBscript タイプ・スクリプト用。
- ▶ **[分散コンポーネント]** : COM/DCOM, および Microsoft .NET プロトコル用。
- ▶ **[e ビジネス]** : AMF, AJAX (Click and Script), Flex, FTP, LDAP, Microsoft .NET, Web (Click and Script), Web (HTTP/HTML), および Web サービス /SOA プロトコル用。
- ▶ **[ERP/CRM]** Oracle NCA, Oracle Web Applications 11i, Peoplesoft Enterprise, Peoplesoft-Tuxedo, SAP-Web, SAPGUI, SAP (Click and Script), および Siebel Web プロトコル用。
- ▶ **[Enterprise Java Beans]** : EJB テスト・プロトコル用。
- ▶ **[Java]** : CORBA, RMI-Java, および JMS など, Java タイプのプロトコルの記録と再生用。
- ▶ **[レガシ]** : ターミナル・エミュレーション (RTE) 用。
- ▶ **[メール サービス]** : Internet Messaging (IMAP), MS Exchange (MAPI), POP3, SMTP プロトコル用。
- ▶ **[ミドルウェア]** : Tuxedo プロトコル用。
- ▶ **[ストリーミング]** : Real および Media Player (MMS) プロトコル用。
- ▶ **[ワイヤレス]** : マルチメディア・メッセージング・サービス (MMS) および WAP プロトコル用。

ユーザ定義テンプレート

ユーザ定義テンプレートにより、特有の構成のスクリプトをテンプレートとして保存できます。作成した後は、スクリプトを作成する際のプロトタイプとしてこのテンプレートを使用できます。

テンプレートでは、次のファイルとデータがサポートされています。

- ▶ 実行環境の設定
- ▶ パラメータ
- ▶ その他のファイル
- ▶ アクション
- ▶ スナップショット

[記録] および [一般] オプションは一般的な設定であり、特定のスクリプトに関連していないため、サポートされていません。

テンプレートの保存と作成

テンプレートとしてスクリプトを保存することも、または保存されているテンプレートからスクリプトを開くこともできます。

スクリプトをテンプレートとして保存するには、次の手順を実行します。

- 1 VuGen でスクリプトを開きます。
- 2 [ファイル] > [ユーザ定義テンプレート] > [テンプレートとして保存] を選択します。[スクリプトをテンプレートとして保存する] ダイアログ・ボックスが開きます。すべての既存のテンプレート・フォルダにアイコンがあります。
- 3 [ファイル名] 領域に保存するテンプレートの名前を入力し、[保存] をクリックします。スクリプトは指定した場所にテンプレートとして保存され、リストに追加されます。

テンプレートからスクリプトを開くには、次の手順を実行します。

- 1 [ファイル] > [ユーザ定義テンプレート] > [スクリプトをテンプレートから作成する] を選択します。[スクリプトをテンプレートから作成する] ダイアログ・ボックスが開きます。すべての既存のテンプレート・フォルダにアイコンがあります。

- 2 スクリプトの原型として使用するテンプレートを選択し、**[開く]** をクリックします。テンプレートに基づく新しいスクリプトが、VuGen で開きます。

注意と制限事項

- ▶ 特定のプロトコルに対するスクリプトを設定し、そのスクリプトをテンプレートとして保存すると、そのテンプレートに基づくほかのスクリプトは、そのプロトコルでのみ機能します。たとえば、Web (Click and Script) プロトコルに対するスクリプトを設定し、そのスクリプトからテンプレートを作成した場合、そのテンプレートは Web (Click and Script) でしか使用できません。
- ▶ テンプレートを作成した後は、そのテンプレートを VuGen で直接編集することはできません。変更を加えるには、テンプレートを開いて別の名前を付けて再び保存するか、既存のテンプレートを上書きします。
- ▶ テンプレートから元のスクリプトを再生成すると、手作業で加えた変更はすべて失われます。

スクリプトの新規作成

本項では、VuGen を起動してスクリプトを新規作成する方法について説明します。

新しい Vuser スクリプトを作成するには、次の手順を実行します。

- 1 [スタート] > [プログラム] > <アプリケーション名> > [Applications] > [Virtual User Generator] を選択して VuGen を起動します。起動画面が開きます。
- 2 シングル・プロトコル・スクリプトを作成するには、[カテゴリ] リストからプロトコルを1つ選択します。
- 3 1つの記録セッションで2つ以上のプロトコルを記録できるマルチ・プロトコル・スクリプトを作成するには、左側の表示枠の [新規マルチプロトコル スクリプト] ボタンをクリックして [新規マルチプロトコル スクリプト] ウィンドウを表示します。

使用するプロトコルを [利用可能なプロトコル] リストから選択します。右方向の矢印をクリックして、選択したプロトコルを [選択されたプロトコル] リストに移動します。使用するすべてのプロトコルについて、この手順を繰り返します。

注：特定の Oracle NCA アプリケーションを記録する場合は、**Web プロトコル**ではなく **Oracle NCA** を選択します。詳細については、『第2巻 – プロトコル』の「Oracle NCA プロトコル」を参照してください。

- 4 [作成] をクリックしてダイアログ・ボックスを閉じ、Vuser スクリプトの生成を開始します。

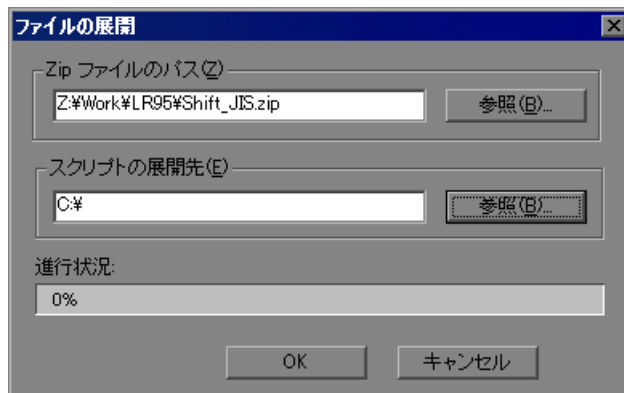
ヒント：あるいは、[ユーザ定義テンプレート] からスクリプトを開くこともできます。詳細については、88 ページ「ユーザ定義テンプレート」を参照してください。

既存のスクリプトを開く

ローカル・マシンまたはネットワーク上にすでにスクリプトがある場合は、そのスクリプトを変更して追加のアクションを記録できます。

既存のスクリプトを開くには、次の手順を実行します。

- 1 ローカル・マシンまたはネットワーク・ドライブに格納されているスクリプトを開くには、**[ファイル]** > **[開く]** を選択します。
- 2 Quality Center リポジトリからファイルを開く方法については（LoadRunner の場合のみ）、194 ページ「Quality Center プロジェクトからスクリプトを開く」を参照してください。
- 3 圧縮 Zip ファイルに格納されているスクリプトを開くには、**[ファイル]** > **[ZIP 操作]** > **[Zip ファイルからインポート]** を選択します。Zip ファイルを選択すると、圧縮解除されたファイルを格納する場所を指定するよう求められます。



- 4 Zip ファイルから作業を行い、その間スクリプト・ファイルの展開や保存をしないようにするには、**[ファイル]** > **[ZIP 操作]** > **[Zip ファイルで作業]** を選択します。スクリプトを変更して保存すると、変更内容が Zip ファイルに直接格納されます。

アプリケーションの記録

ほとんどの Vuser スクリプト・タイプでは、新しいスクリプトの作成を開始すると、自動的に [記録開始] ダイアログ・ボックスが表示されます。

記録を開始するには、次の手順を実行します。



- 1 [記録開始] ダイアログ・ボックスが開かない場合は、[記録開始] ボタンをクリックします。[記録開始] ダイアログ・ボックスが開きます。このダイアログ・ボックスはプロトコルごとに多少異なります。
- 2 ほとんどのクライアント/サーバ・プロトコルでは、次のダイアログ・ボックスが表示されます。

記録するアプリケーション、作業ディレクトリ（任意）、およびアクションを入力します。必要があれば、[オプション] をクリックして記録オプションを設定します。

- 3 インターネット以外のアプリケーションの場合は、アプリケーションの種類を選択します。アプリケーションの種類には、Win32 アプリケーションおよびインターネット・アプリケーションがあります。たとえば、Web および Oracle NCA スクリプトはインターネット・アプリケーションを記録し、Windows Sockets Vuser は Win32 アプリケーションを記録します。Citrix ICA Vuser の場合は、VuGen によって Citrix クライアントが自動的に記録されるため、[アクション内に記録] ボックスにアクションを指定するだけで済みます。

- 4 インターネット・ベースのアプリケーションについて、適切な情報を入力します。

記録開始

アプリケーションタイプ(A): インターネット アプリケーション

記録するプログラム(P): Microsoft Internet Explorer

URL アドレス(U):

作業ディレクトリ(W): C:\Program Files\HP\LoadRunner\bin

アクション内に記録(R): vuser_init

アプリケーションの起動処理を記録する(A)

オプション(O)...

OK

キャンセル

- ▶ **[記録するプログラム]**：記録するブラウザまたはインターネット・アプリケーションを選択します。
- ▶ **[URL アドレス]**：開始する URL アドレスを指定します。
- ▶ **[作業ディレクトリ]**：作業ディレクトリを指定する必要があるアプリケーションの場合は、ここで指定します。必要となる情報は、Vuser スクリプトのタイプによって異なります。

- 5 Win32 アプリケーションについて、適切な情報を入力します。

記録開始

アプリケーションタイプ(A): Win32 アプリケーション

記録するプログラム(P):

プログラム引数(U):

作業ディレクトリ(W):

アクション内に記録(R): Action

アプリケーションの起動処理を記録する(A)

オプション(O)...

OK

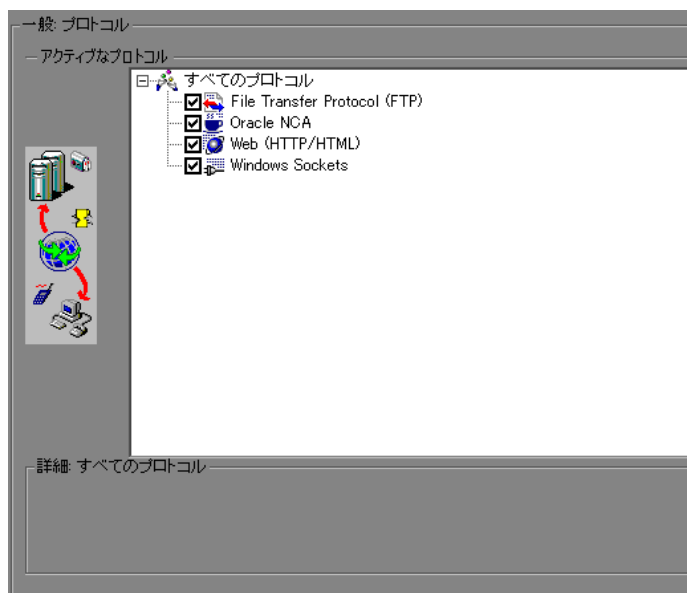
キャンセル

- ▶ **[記録するプログラム]**：記録する Win 32 アプリケーションを入力します。
 - ▶ **[プログラム引数]**：前述で指定した実行ファイルのコマンド・ライン引数を指定します。たとえば、**plus32.exe** にコマンド・ライン・オプション **peter@neptune** を指定した場合、**plus32.exe** を起動すると、ユーザ **Peter** がサーバ **Neptune** に接続されます。
 - ▶ **[作業ディレクトリ]**：作業ディレクトリを指定する必要があるアプリケーションの場合は、ここで指定します。
- 6 **[アクション内に記録]** ボックスで、記録を挿入するセクションを選択します。最初を選択できるセクションは **vuser_init**、**Action**、および **vuser_end** です。マルチ・アクション（Oracle NCA、Web、RTE、C Users、WAP、i-Mode、VoiceXML）をサポートするシングル・プロトコル Vuser スクリプトの場合は、**[アクション]** > **[新規アクションを作成]** を選択して新しいセクションを追加し、新しいアクションの名前を指定できます。



- 7 アプリケーションの起動を記録するには、**[アプリケーションの起動処理を記録する]** を選択します（Java タイプの Vuser スクリプトには適用されません）。アプリケーションの起動を記録しない場合は、このチェック・ボックスをクリックします。次の場合は、起動を記録しないことをお勧めします。
- ▶ マルチ・アクションの場合。起動が必要なアクションは1つだけです。
 - ▶ アプリケーションで特定の位置まで移動し、その位置から記録を開始する場合。
 - ▶ 既存のスクリプトに記録する場合。
- 8 **[オプション]** または **[記録オプション]** ボタンをクリックして **[記録オプション]** ダイアログ・ボックスを開き、記録オプションを設定します。使用可能なオプションは記録するプロトコルによって異なります。詳細については、対応する各章を参照してください。

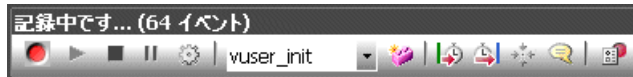
- 9 コード生成のための言語を選択し、スクリプトに関するオプションを設定するには、[スクリプト] ノードをクリックします。詳細については、第19章「スクリプト生成オプションの設定」を参照してください。
- 10 ポート情報を指定するには、[ポート マッピング] ノードをクリックします。これは、非標準ポートのSSLアプリケーションを記録する場合に有用です。ポートのリストを確認します。使用するポートがリストにない場合は、[ポート マッピング] オプションを使用して情報を指定できます。詳細については、第21章「ポートの割り当て設定」を参照してください。
- 11 マルチ・プロトコル記録のみ：記録するプロトコルのリストを変更するには、[プロトコル] ノードをクリックします。ノードを展開し、必要なプロトコルを選択します。



これで、記録開始の準備が整います。

- 12 [OK] をクリックしてダイアログ・ボックスを閉じ、記録を開始します。
- 13 [アプリケーションの起動処理を記録する] チェック・ボックスがクリアしてある場合、[記録の一時停止] ダイアログ・ボックスが表示されます。操作を進めて、記録を開始したい位置に到達したら、[記録] をクリックします。記録を行わない場合は、[中止] をクリックします。

- 14 VuGen によってアプリケーションが起動され、記録ツールバーが開きます。





アプリケーション内で、標準的な操作を実行します。操作と同時に、VuGen によって Vuser スクリプトが選択されたアクション・セクションに挿入されます。記録中にセクションを切り替えるには、フローティング・ツールバーを使います。

使用しているアプリケーションまたはサーバで認証を行う必要がある場合は、ユーザ名とパスワードを入力するよう VuGen から求められます。認証の詳細については、該当する項を参照してください。

記録セッションの終了と保存

一般的なビジネス・プロセスを記録したら、ビジネス・プロセスの終了ステップを実行し、Vuser スクリプトを保存して、セッションの記録を完了します。

記録を終了するには、次の手順を実行します。

- 1 フローティング・ツールバーで **vuser_end** セクションに切り替えて、ログオフまたはクリーンアップ処理を実行します。
- 2 記録ツールバーの **記録停止** ボタンをクリックします。VuGen エディタに、記録されたすべてのステップが表示されます（スクリプト・ビューで作業を始めている場合は、記録された関数が表示されます）。
- 3 **上書き保存** ボタンをクリックして、記録されたセッションを保存します。 [テストを保存] ダイアログ・ボックスが表示されます（新規 Vuser スクリプトの場合のみ）。スクリプト名を指定します。**注**：スクリプトに **init**, **run**, **end** という名前は付けしないでください。これらの名前は VuGen によって使用されます。

- 4 スクリプトのディレクトリ全体を Zip ファイルとして保存するには、[ファイル] > [ZIP 操作] > [Zip ファイルにエクスポート] を選択します。

保存するファイルを指定します。実行可能ファイルだけを保存するには、[ファイルの圧縮] セクションで [実行可能ファイル] を選択します。標準設定では、すべてのファイルがアーカイブに保存されます。

圧縮率を [最高 (最低速)], [標準], [高速], [超高速], または [なし] から選択します。圧縮率が高いほど、VuGen によるアーカイブ作成に時間がかかります。

[OK] をクリックします。

ヒント: ユーザ定義テンプレートとしてスクリプトを保存できます。詳細については、88 ページ「ユーザ定義テンプレート」を参照してください。

- 5 Zip ファイルを作成して電子メールの添付ファイルとして送信するには、[ファイル] > [ZIP 操作] > [Zip して電子メールで送信] を選択します。

[OK] をクリックします。電子メールの作成フォームが表示されます。

電子メール・アドレスを入力してメールを送信します。

記録ログの表示

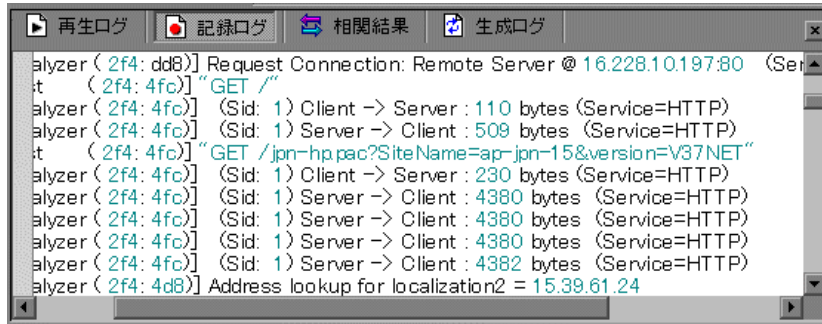
記録終了後に、`vuser_init`、`Actions`、および `vuser_end` セクションの内容を VuGen スクリプト・エディタに表示できます。アクションを表示するには、左側の表示枠でアクション名を選択します。

記録中、VuGen は一連の設定ファイル、データ・ファイル、ソースコード・ファイルを作成します。これらのファイルには、Vuser の実行時の情報および設定情報が含まれます。VuGen は、これらのファイルをスクリプトと一緒に保存します。スクリプト・フォルダを開くには、スクリプト・ビュー ([表示] > [スクリプト ビュー]) に切り替えて、右クリック・メニューから [スクリプト ディレクトリを開く] を選択します。

ログを下部のウィンドウに表示することで、記録とスクリプト生成に関する情報を確認できます。出力ウィンドウを開くには、[表示] > [出力ウィンドウ] を選択し、[記録ログ] タブまたは [生成ログ] タブを選択します。

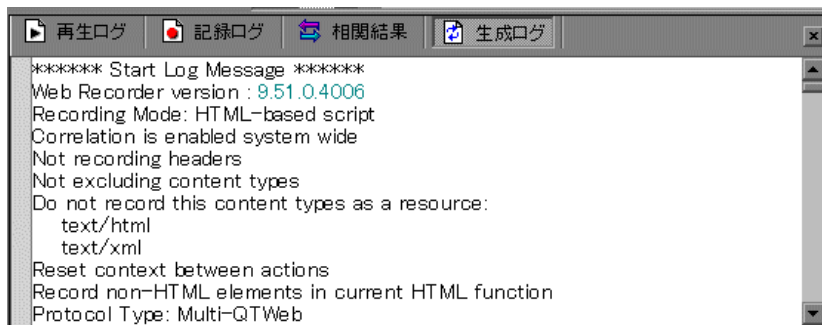
記録ログ

記録中に発行されたメッセージのログを表示するには、**[記録ログ]** タブを選択します。**[記録オプション]** ダイアログ・ボックスの **[詳細]** タブで、このログの詳細レベルを設定できます。



生成ログ

コード生成に使用されたスクリプトの設定の概要を表示するには、**[生成ログ]** タブを選択します。ここでは、レコーダのバージョンや記録オプションの値その他の追加情報が表示されます。



Zip ファイルの使用

VuGen では、Zip ファイルでの作業をいくつかの方法で実行できます。Zip ファイルでの作業には、ディスク容量が節約され、スクリプトが移動しやすくなる、などの利点があります。多数のファイルをマシン間でコピーせずに、1 つの Zip ファイルをコピーするだけで済みます。

Zip ファイルからのインポート

圧縮 Zip ファイルに格納されているスクリプトを開くには、[ファイル] > [ZIP 操作] > [Zip ファイルからインポート] を選択します。Zip ファイルを選択すると、圧縮解除されたファイルを格納する場所を指定するよう求められます。

Zip ファイルからの作業

Zip ファイルから作業を行い、その間スクリプト・ファイルの展開や保存をしないようにするには、[ファイル] > [ZIP 操作] > [Zip ファイルで作業] を選択します。スクリプトを変更して保存すると、変更内容が Zip ファイルに直接格納されます。

Zip ファイルへのエクスポート

スクリプトのディレクトリ全体を Zip ファイルとして保存するには、[ファイル] > [ZIP 操作] > [Zip ファイルにエクスポート] を選択します。

すべてのファイルを保存するか、実行可能ファイルだけを保存するかを指定できます。標準設定では、すべてのファイルがアーカイブに保存されます。実行可能ファイルだけを保存するには、[実行可能ファイル] オプションを選択します。

また、**圧縮率**として、[最高]、[標準]、[高速]、[超高速]、または [なし] を選択することもできます。圧縮率が高いほど、VuGen によるアーカイブ作成に時間がかかります。したがって、[最高 (最低速)] の圧縮オプションが最も低速になります。

Zip の作成と電子メール送信

Zip ファイルを作成して電子メールの添付ファイルとして送信するには、[ファイル] > [ZIP 操作] > [Zip して電子メールで送信] を選択します。[ファイルの圧縮] ダイアログ・ボックスで [OK] をクリックすると、設定に従ってファイルが圧縮され、Zip ファイルを添付ファイルとして持つ電子メール作成フォームが開きます。

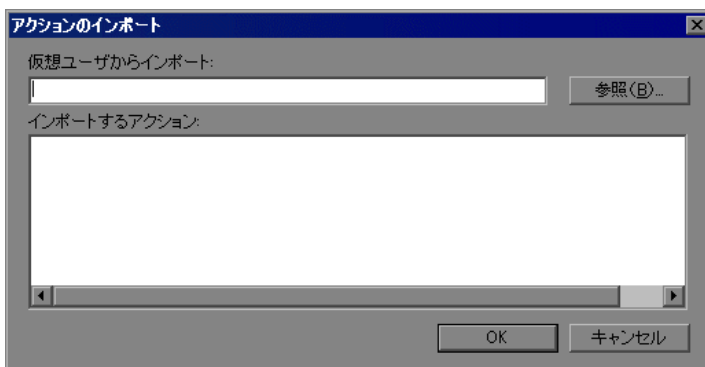
アクションのインポート

複数のアクションをサポートする Vuser タイプの場合、別の Vuser スクリプトから現在のスクリプトにアクションをインポートできます。インポートできるのは、同じタイプの Vuser のアクションだけです。インポートされたアクションに関連付けられているパラメータは、スクリプトに組み込まれます。次のオプションが使用できます。

- ▶ **[仮想ユーザからインポート]**：インポート元の Vuser スクリプトを入力または参照します。
- ▶ **[インポートするアクション]**：インポートするアクションを選択します。

現在のスクリプトへアクションをインポートするには、次の手順を実行します。

- 1 **[アクション]** > **[アクションを仮想ユーザにインポート]** を選択するか、[タスク] 表示枠を右クリックして **[アクションを仮想ユーザにインポート]** を右クリック・メニューから選択します。[アクションのインポート] ダイアログ・ボックスが表示されます。



- 2 **[参照]** をクリックして Vuser スクリプトを選択します。[インポートするアクション] セクションに、スクリプトのアクションのリストが表示されます。
- 3 アクションを強調表示して **[OK]** をクリックします。スクリプトにアクションが表示されます。

- 4 アクションの順序を並べ替えるには、最初にアクションの順序の並べ替えを有効にしておく必要があります。アクションを右クリックして「**アクションの順序を並べ替え**」を選択します。アクションをドラッグして順序を並べ替えます。アクションを VuGen の左側の表示枠で並べ替えても、それらの実行順序には影響はありません。実行順序を変更するには、実行環境の設定の「ペースの設定」ノードを使用します。詳細については、第 22 章「実行環境の設定」を参照してください。

認証情報の提供

次の項の内容は、マルチ・プロトコルによる記録を対象としています。

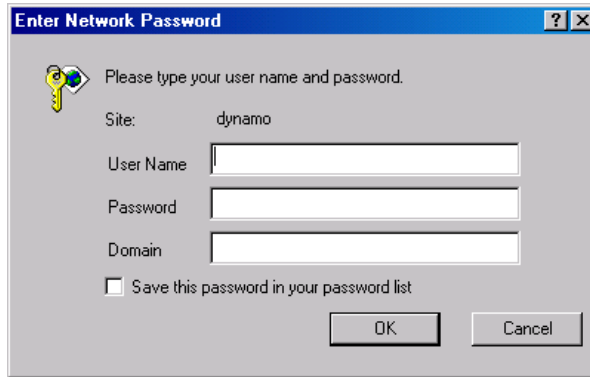
NTLM 認証を使用する Web セッションを記録する際、サーバでユーザ名やパスワードなどの詳細情報を入力しなければならない場合があります。

最初に、IE (Internet Explorer) は、現在のユーザの NT 認証情報の使用を試みます。

- ▶ IE からこの情報を使用してログインに成功し、スクリプトを記録した場合は、記録の最後にパスワードを入力するよう VuGen から求められます。ユーザ名とドメイン情報は VuGen が自動的に取得します。必要ならば、「Web レコーダ NTLM 認証」ダイアログ・ボックスでユーザ名を編集することもできます。



- ▶ IE から現在のユーザの情報を使用してログインできない場合は、標準のブラウザ認証ダイアログ・ボックスを使用してユーザ名とパスワードを入力するよう IE から求められます。



web_set_user 関数の生成

NTLM 認証を実行すると、VuGen によって **web_set_user** 関数がスクリプトに追加されます。

- ▶ 認証が成功した場合は、ユーザ名、暗号化されたパスワード、およびホストが設定された **web_set_user** 関数が、VuGen によって生成されます。

```
web_set_user("domain1¥¥dashwood",  
            lr_decrypt("4042e3e7c8bbbcfde0f737f91f"),  
            "sussex:8080");
```

- ▶ [Web レコーダ NTLM 認証] ダイアログ・ボックスで情報を入力せずにキャンセルした場合も、手作業で編集できるように、VuGen によって **web_set_user** 関数が生成されます。

```
web_set_user("domain1¥¥dashwood",  
            "NTLM パスワードをここに入力",  
            "sussex:8080");
```

注：パスワードを手作業で入力した場合は、スクリプト内にそのまま出現するため、セキュリティ上問題があります。パスワードを暗号化するには、パスワードを右クリックして [文字列を暗号化] を選択します。VuGen によって文字列が暗号化され、再生時にパスワードの復号に使用する **lr_decrypt** 関数が生成されます。文字列の暗号化の詳細については、125 ページ「テキストの暗号化」を参照してください。

Vuser スクリプトの再生成

スクリプトを記録した後で、トランザクション、ランデブー・ポイント、メッセージ、コメントを追加してスクリプトを拡張できます。詳細については、第6章「Vuser スクリプトの拡張」を参照してください。

さらに、スクリプトのパラメータ化や、変数の相関も可能です。詳細については、第13章「VuGen パラメータを使った作業」を参照してください。

最初に記録したスクリプトに戻す必要がある場合は、スクリプトを再生成します。この機能は、デバッグや、破損したスクリプトの修復に非常に役立ちます。スクリプトを再生成すると、記録されたアクションに手作業で追加した拡張機能はすべて削除されます。スクリプトにパラメータを追加した場合は、VuGenによって元の値に戻されます。ただし、パラメータ・リストは削除されないため、それまでに作成したパラメータは再挿入できます。再生成で復元されるのは記録されたアクションだけです。手作業で追加されたアクションは復元されません。

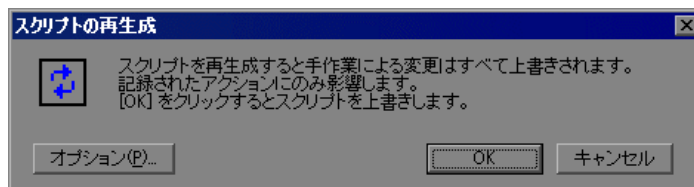
次のボタンが「スクリプトの再生成」ダイアログ・ボックスから使用できます。

[OK]：元の記録ログから Vuser スクリプトを再生成します。再生成では、スクリプトで手動実行したすべての相関とパラメータ化が削除されます。

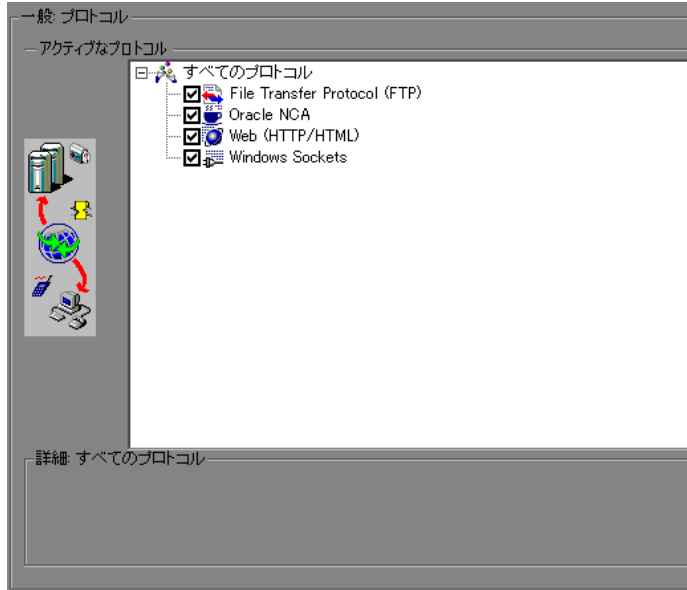
[オプション]：マルチ・プロトコル・スクリプトを処理する場合は、再生成するプロトコルを指定できます。再生成をカスタマイズするには、[仮想ユーザの再生成] ダイアログ・ボックスで **[オプション]** ボタンをクリックし、[オプションの再生成] を開きます。**[プロトコル]** ノードを選択し、再生成するプロトコルとそのままにするプロトコルを指定します。再生成するプロトコルのチェック・ボックスを選択します。再生成しないプロトコルのチェック・ボックスはクリアします。

マルチ・プロトコル Vuser スクリプトを再生成するには、次の手順を実行します。

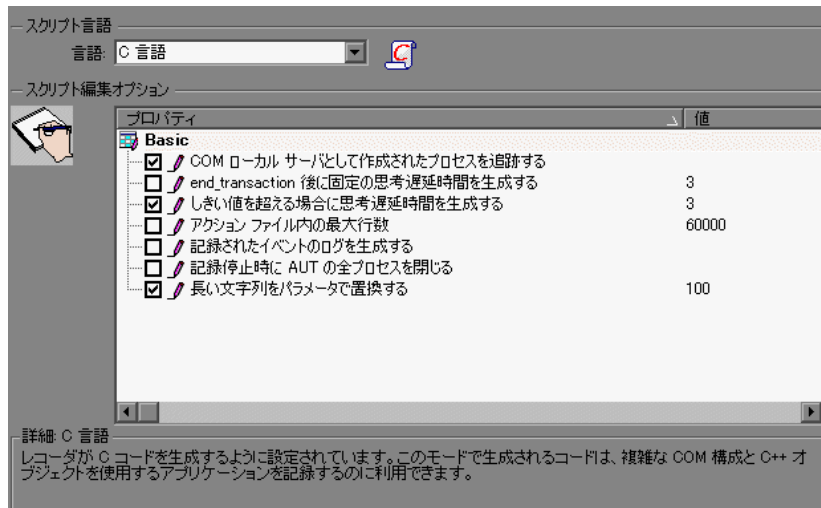
- 1 **[ツール]** > **[スクリプトの再生成]** を選択します。手作業で行ったすべての変更が上書きされることを示す警告が表示されます。



- 2 [オプション] をクリックして [オプションの再生成] ダイアログ・ボックスを開きます。
- 3 [一般: プロトコル] ノードを選択します。再生成するプロトコルとそのままにするプロトコルを指定します。再生成するプロトコルのチェック・ボックスを選択します。変更しないプロトコルのチェック・ボックスはクリアします。



- 4 スクリプト・オプションを変更するには、[一般:スクリプト] ノードを選択し、適切なチェック・ボックスを選択またはクリアします。



第 6 章

Vuser スクリプトの拡張

記録中または記録後に、一般 Vuser 関数、プロトコル固有の Vuser 関数、および標準 ANSI C 関数を追加して、Vuser スクリプトを拡張できます。

本章の内容

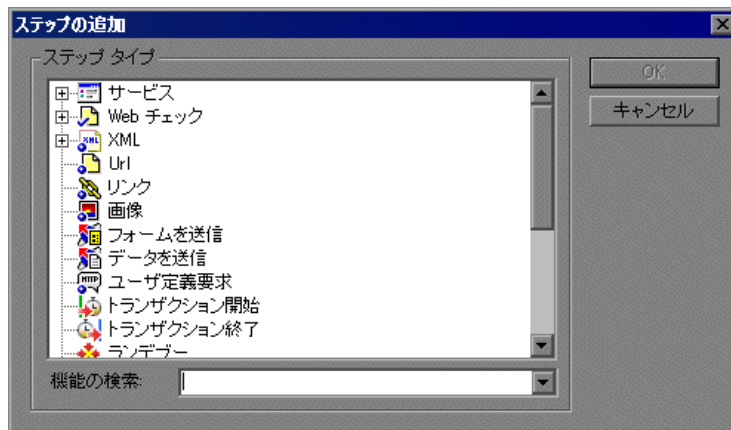
- ▶ Vuser スクリプトの拡張について (108 ページ)
- ▶ Vuser スクリプトへのトランザクションの挿入 (110 ページ)
- ▶ ランデブー・ポイントの挿入 (LoadRunner のみ) (112 ページ)
- ▶ Vuser スクリプトへのコメントの挿入 (114 ページ)
- ▶ Vuser 情報の取得 (115 ページ)
- ▶ 出力へのメッセージの送信 (116 ページ)
- ▶ 実行中の Vuser スクリプトのエラー処理 (121 ページ)
- ▶ Vuser スクリプトの同期化 (122 ページ)
- ▶ ユーザの思考遅延時間のエミュレート (123 ページ)
- ▶ コマンド・ライン引数の取り扱い (124 ページ)
- ▶ テキストの暗号化 (125 ページ)
- ▶ 手動でのパスワードのエンコーディング (126 ページ)
- ▶ スクリプトへのファイルの追加 (127 ページ)

Vuser スクリプトの拡張について

Vuser スクリプトの記録中または記録後に、ステップ（または関数）を手作業で追加することによって機能を拡張できます。

スクリプトに新規ステップを追加するには、次の手順を実行します。

- 1 希望の場所にカーソルを置きます。
- 2 [挿入] > [新規ステップ] を選択します。[ステップの追加] ダイアログ・ボックスが開き、現在のプロトコルに関連するステップが表示されます。



- 3 ステップを選択して [OK] をクリックします。カーソルのある位置にステップ（[スクリプトビュー] では関数）が挿入されます。

[ステップの追加] ダイアログ・ボックスでは、次の種類の関数が使用できます。

- ▶ 一般 Vuser 関数
- ▶ プロトコル固有の Vuser 関数
- ▶ 標準 ANSI C 関数

一般 Vuser 関数

一般 Vuser 関数は、Vuser スクリプトの機能を大幅に強化します。たとえば、一般 Vuser 関数を使って、サーバ・パフォーマンスの測定、サーバ負荷の制御、デバッグ・コードの追加、テストの Vuser についてのランタイム情報の取得などができます。

一般 Vuser 関数は任意のタイプの Vuser スクリプトで使用できます。一般 Vuser 関数には、必ず **LR** というプレフィックスが付いています。VuGen は、スクリプト記録中にいくつかの一般 Vuser 関数を生成し、Vuser スクリプトに挿入します。自動生成されなかった、ほかの関数を使用するには、VuGen のメイン・ウィンドウで **[挿入]** > **[新規ステップ]** を選び関数を選択します。

本章では、最もよく使う一般 Vuser 関数だけを対象に使い方を説明します。Vuser 関数の詳細については、『**Online Function Reference**』（英語版）（**[ヘルプ]** > **[関数リファレンス]**）を参照してください。

プロトコル固有の Vuser 関数

Vuser スクリプトを拡張する関数ライブラリがいくつかあります。各ライブラリは、Vuser のタイプに固有のものです。たとえば、Windows Sockets Vuser スクリプトでは **LRS** 関数を使用し、Tuxedo Vuser スクリプトでは **LRT** 関数を使用します。プロトコル固有の Vuser 関数の詳細については、『**Online Function Reference**』（英語版）（**[ヘルプ]** > **[関数リファレンス]**）を参照してください。

標準 ANSI C 関数

標準 ANSI C 関数を追加して Vuser スクリプトを拡張できます。ANSI C 関数を使って、Vuser スクリプトにコメント文、フロー制御ステートメント、条件文などを追加できます。標準 ANSI C 関数は任意のタイプの Vuser スクリプトに追加できます。詳細については、『**第 2 巻 – プロトコル**』の「VuGen エディタでのスクリプトのプログラミング」を参照してください。

Vuser スクリプトへのトランザクションの挿入

トランザクションを定義することによって、サーバのパフォーマンスを評価できます。各トランザクションは、サーバが特定の Vuser 要求に応答するまでにかかる時間を測定します。これらの要求は、1つのクエリに対する応答を待つような簡単な処理の場合や、いくつかのクエリを発行してレポートを作成するという複雑な処理の場合があります。

トランザクションを測定するために、タスクの開始とタスクの終了を示す Vuser 関数を挿入します。スクリプトには、任意の数のトランザクションを異なる名前でも挿入することができます。

LoadRunner に対して、コントローラは各トランザクションの実行に要する時間を測定します。テスト実行の後、アナリシスのグラフとレポートを使用して、トランザクションごとのサーバ・パフォーマンスを分析します。

トランザクションは記録中または記録後に作成できます。記録後にトランザクションを追加するには、トランザクション・エディタを使用し、トランザクションのステップをグラフィックで示します。詳細については、70 ページ「トランザクション」を参照してください。または、**[挿入]** メニューを使用して、**トランザクション開始**および**トランザクション終了**マーカを追加します。

ここでは、記録中にトランザクションを作成する方法について説明します。

トランザクションの開始を示す方法

スクリプトを作成する前に、測定の対象となるビジネス・プロセスを決めておきます。次に、それぞれのビジネス・プロセスまたはサブプロセスをトランザクションとして指定します。

トランザクションの開始を示すには、次の手順を実行します。



- 1 Vuser スクリプトの記録中に、記録ツールバーの **[トランザクション開始]** ボタンをクリックします。[トランザクション開始] ダイアログ・ボックスが開きます。



- 2 トランザクションの名前を [トランザクション名] ボックスに入力します。トランザクション名の最初の文字は、文字または数字である必要があります、その後の文字には文字または数字を使用できます。

注：.,:#/¥"<は使用しないでください。

[OK] をクリックしてトランザクション名を確定します。VuGen によって `lr_start_transaction` ステートメントが Vuser スクリプトに挿入されます。たとえば、次の関数は `trans1` トランザクションの開始を示します。

```
lr_start_transaction("trans1");
```

トランザクションの終了を示す方法

ビジネス・プロセスの終了を、トランザクション終了ステートメントで示すことができます。

トランザクションの終了を示すには、次の手順を実行します。



- 1 スクリプトの記録中に、記録ツールバーの [トランザクション終了] ボタンをクリックします。[トランザクション終了] ダイアログ・ボックスが開きます。



- 2 矢印をクリックして、開始されているトランザクションのリストを表示します。終了するトランザクションを選択します。

[OK] をクリックしてトランザクション名を確定します。VuGen によって `lr_end_transaction` ステートメントが Vuser スクリプトに挿入されます。たとえば、次の関数は `trans1` トランザクションの終了を示します。

```
lr_end_transaction("trans1", LR_AUTO);
```

注：トランザクションにトランザクションが含まれる入れ子のトランザクションを作成できます。トランザクションを入れ子にする場合は、含まれる側のトランザクションを含む側のトランザクションをよりも前に閉じなければなりません。そうしないとトランザクションが正しく解析されません。

ランデブー・ポイントの挿入（LoadRunner のみ）

負荷テストを実行する際には、システムに高いユーザ負荷がかかっている状態をエミュレートする必要があります。そのためには、複数の Vuser を同期させ、まったく同じ瞬間にタスクを実行させます。複数の Vuser を同時に動作させるには、「ランデブー・ポイント」を作成します。ある Vuser がランデブー・ポイントに到達すると、ほかのすべての Vuser がランデブー・ポイントに到着するまで、その Vuser は待機させられます。指定した数の Vuser が到着すると、Vuser が解放されます。

Vuser スクリプトにランデブー・ポイントを挿入することによって、待機場所を指定します。Vuser は、スクリプトを実行してランデブー・ポイントに到達すると、スクリプトの実行を一時停止し、コントローラからの再開許可を待ちます。Vuser は、ランデブー・ポイントから解放されると、スクリプト内の次のタスクを実行します。

注：ランデブー・ポイントは **Action** セクションでのみ有効です。**init** または **end** セクションでは無効になります。

ランデブー・ポイントを挿入するには、次の手順を実行します。



- 1 Vuser スクリプトの記録中に、記録ツールバーの [**ランデブー**] ボタンをクリックします。[ランデブー] ダイアログ・ボックスが開きます。



- 2 ランデブー・ポイントの名前を [**ランデブー名**] ボックスに入力します。

注：ランデブー・ポイントの名前では、大文字と小文字は区別されません。たとえば、Vuser は Rendezvous1 と rendezvous1 を同じポイントとして認識します。

[OK] をクリックします。VuGen によって lr_rendezvous ステートメントが Vuser スクリプトに挿入されます。たとえば、次の関数は rendezvous1 という名前のランデブー・ポイントを定義します。

```
lr_rendezvous("rendezvous1");
```

- 3 記録セッションの後にスクリプトにランデブー・ポイントを挿入するには、VuGen のメニューから [**挿入**] > [**ランデブー**] を選択します。

Vuser スクリプトへのコメントの挿入

VuGen では Vuser の実行アクション間にコメントを挿入できます。コメントを挿入して、動作内容を説明したり、特定の操作に関する情報を記入したりできます。たとえば、データベースの動作を記録している場合は、「これは最初のクエリです」のように最初のクエリであることを示すコメントを挿入できます。

コメントを挿入するには、次の手順を実行します。



- 1 スクリプトの記録中に、記録ツールバーの [コメント] ボタンをクリックします。[コメントの挿入] ダイアログ・ボックスが開きます。



- 2 テキスト・ボックスにコメントを入力します。
- 3 [OK] をクリックするとダイアログ・ボックスが閉じ、コメントが挿入されます。テキストはスクリプトの現在位置に、コメント記号で囲まれた状態で挿入されます。次のスクリプトの抜粋は、Vuser スクリプトに挿入されたコメントを示します。

```
/*  
* これは最初のクエリです  
*/
```

注：記録セッションを完了した後でスクリプトにコメントを挿入するには、VuGen のメニューから [挿入] > [コメント] を選択します。

Vuser 情報の取得

次の関数を Vuser スクリプトに追加して、Vuser 情報を取得できます。

| 関数 | 説明 |
|--------------------------------|---|
| lr_get_attrib_string | コマンド・ライン・パラメータ文字列を返します。 |
| lr_get_host_name | Vuser スクリプトを実行しているマシンの名前を返します。 |
| lr_get_master_host_name | コントローラを実行しているマシンの名前を返します。HP Business Availability Center は適用対象外です。 |
| lr_whoami | スクリプトを実行している Vuser の名前を返します。HP Business Availability Center は適用対象外です。 |

次の例では、**lr_get_host_name** 関数を使用して、Vuser を実行しているコンピュータの名前を取得しています。

```
my_host = lr_get_host_name( );
```

前述の関数の詳細については、『**Online Function Reference**』（英語版）（[ヘルプ](#)） > [関数リファレンス](#)）を参照してください。

出力へのメッセージの送信

Vuser スクリプトの中でメッセージ・タイプの関数を使用すると、カスタマイズしたエラー・メッセージや通知メッセージを出力やログ・ファイル、およびテスト・レポートのサマリに送信できます。たとえば、クライアント・アプリケーションの現在のステータスを示すメッセージを挿入することができます。LoadRunner Controller は、これらのメッセージを [出力] ウィンドウに表示します。また、これらのメッセージをファイルに保存することもできます。

HP Business Availability Center で作業をしているときは、メッセージ・タイプの関数を使用して、エラー・メッセージや通知メッセージを Web サイトや Business Process Monitor ログ・ファイルに送信できます。たとえば、Web ベース・アプリケーションの現在の状態を表示するメッセージを挿入できます。

注： トランザクションの中からメッセージを送信することは避けてください。トランザクションの実行時間が長くなり、トランザクションの結果が不正確になることがあります。

次のメッセージ関数を Vuser スクリプトの中で使用できます。

| 関数 | 説明 |
|-----------------------------|--|
| lr_debug_message | デバッグ・メッセージを [出力] ウィンドウまたは Business Process Monitor ログ・ファイルに送ります。 |
| lr_error_message | エラー・メッセージを [出力] ウィンドウ、テスト結果レポート、または Business Process Monitor ログ・ファイルに送ります。 |
| lr_get_debug_message | 現在のメッセージ・クラスを取得します。 |
| lr_log_message | 出力メッセージを Vuser スクリプトのディレクトリにあるログ・ファイル output.txt に直接送信します。この関数は、出力メッセージによって TCP/IP のトラフィックが妨げられるのを防ぐので便利です。 |

| 関数 | 説明 |
|--------------------------------|--|
| lr_output_message | メッセージを [出力] ウィンドウ、テスト結果レポート、または Business Process Monitor ログ・ファイルに送ります。 |
| lr_set_debug_message | 出力メッセージのメッセージ・クラスを設定します。 |
| lr_vuser_status_message | メッセージを [Controller] の Vuser ステータス領域に送ります。HP Business Availability Center は適用対象外です。 |
| lr_message | メッセージを、Vuser ログと、[出力] ウィンドウまたは Business Process Monitor ログ・ファイルに送ります。 |

lr_message 関数、**lr_output_message** 関数、**lr_log_message** 関数の振る舞いは、実行環境の設定の [ログ] の設定のスクリプトのデバッグ・レベルの影響を受けません。これらの関数は常にメッセージを送信します。

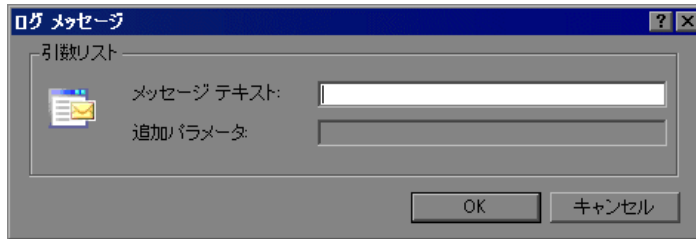
lr_output_message および **lr_error_message** 関数を使用すると、[テスト結果] サマリ・レポートに、意味のあるメッセージを送信することもできます。詳細については、第 10 章「テスト結果の表示」を参照してください。

ログ・メッセージ

VuGen を使って `lr_log_message` 関数を生成し、Vuser スクリプトに挿入できます。たとえば、データベースを対象としたアクションを記録しているときに、最初のクエリを示すために「これは最初のクエリです」というメッセージを挿入できます。

`lr_log_message` 関数を挿入するには、次の手順を実行します。

- 1 [挿入] > [ログ メッセージ] を選択します。[ログ メッセージ] ダイアログ・ボックスが開きます。



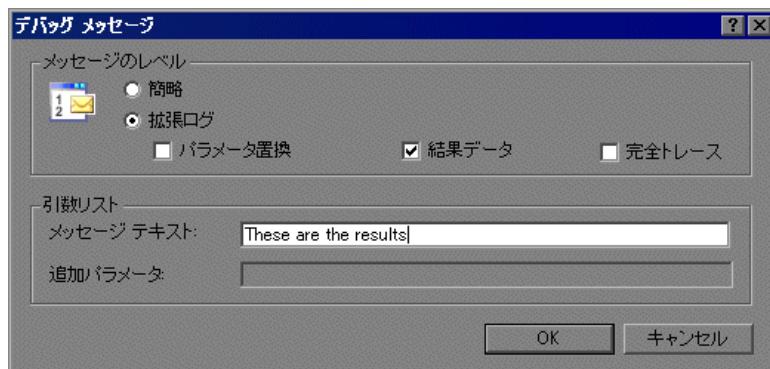
- 2 [メッセージ テキスト] ボックスにメッセージを入力します。
- 3 [OK] をクリックするとメッセージが挿入され、ダイアログ・ボックスが閉じます。`lr_log_message` 関数がスクリプトの現在の位置に挿入されます。

デバッグ・メッセージ

VuGen のユーザ・インタフェースを使って、デバッグ・メッセージまたはエラー・メッセージを追加できます。デバッグ・メッセージの場合には、テキスト・メッセージのレベルを指定できます。対象のメッセージは、指定したレベルがメッセージ・クラスのレベルに一致する場合にだけ発行されます。メッセージ・クラスは、`lr_set_debug_message` を使用して設定します。

デバッグ関数を挿入するには、次の手順を実行します。

- 1 **[挿入]** > **[新規ステップ]** を選択します。[ステップの追加] ダイアログ・ボックスが開きます。
- 2 **[デバッグ メッセージ]** ステップを選択して **[OK]** をクリックします。[デバッグ メッセージ] ダイアログ・ボックスが開きます。



- 3 メッセージのレベルとして、**[簡略]** または **[拡張ログ]** を選択します。[拡張ログ] を選択した場合には、ログに記録する情報の種類を、**[パラメータ置換]**、**[結果データ]**、**[完全トレース]** の中から指定します。
- 4 **[メッセージ テキスト]** ボックスにメッセージを入力します。
- 5 **[OK]** をクリックするとメッセージが挿入され、ダイアログ・ボックスが閉じます。lr_debug_message 関数がスクリプトの現在の位置に挿入されます。

エラーおよび出力メッセージ

Web, Winsock, および Oracle NCA など、スクリプトのツリー・ビューの表示が可能なプロトコルの場合、VuGen の中でエラーまたは出力メッセージを追加できます。この関数の一般的な使用法としては、条件文を挿入して、エラー状態が検出されたときにメッセージを発行するという方法があります。

エラーまたは出力メッセージ関数を挿入するには、次の手順を実行します。

- 1 [挿入] > [新規ステップ] を選択します。[ステップの追加] ダイアログ・ボックスが開きます。
- 2 [エラー メッセージ] または [出力メッセージ] ステップを選択して [OK] をクリックします。[エラー メッセージ] または [出力メッセージ] ダイアログ・ボックスが開きます。



- 3 [メッセージ テキスト] ボックスにメッセージを入力します。
- 4 [OK] をクリックするとメッセージが挿入され、ダイアログ・ボックスが閉じます。`lr_error_message` 関数または `lr_output_message` 関数がスクリプトの現在の位置に挿入されます。

メッセージ関数の詳細については、『[Online Function Reference](#)』（英語版）（[ヘルプ] > [関数リファレンス]）を参照してください。

実行中の Vuser スクリプトのエラー処理

スクリプト実行時の Vuser によるエラー処理の方法を指定することができます。標準設定では、Vuser によってエラーが検出されると、スクリプトの実行が停止されます。次のいずれかの方法を使って、エラーが発生したときに次の回復を継続するように Vuser を指定できます。

- ▶ 実行環境の設定を使用する：[エラーでも処理を継続する] 実行環境の設定を指定できます。実行環境の設定で [エラーでも処理を継続する] を設定すると、Vuser スクリプト全体に適用されます。スクリプトの一部分で、実行環境の設定の [エラーでも処理を継続する] をオーバーライドするには、**lr_continue_on_error** 関数を使用します。詳細については、430 ページ「エラー処理」を参照してください。
- ▶ **lr_continue_on_error** 関数を使用する：**lr_continue_on_error** 関数を使用して、Vuser スクリプトの特定のセグメントでのエラー処理を制御できます。対象セグメントを指定するには、セグメントを **lr_continue_on_error(1);** と **lr_continue_on_error(0);** ステートメントで囲みます。新しいエラー設定は、囲まれた部分の Vuser スクリプト・セグメントに適用されます。詳細については、以降の説明を参照してください。

たとえば、実行環境の設定で [エラーでも処理を継続する] を有効にしている、Vuser が次に示すスクリプトのセグメントの再生中にエラーに遭遇した場合には、Vuser はスクリプトの実行を継続します。

```
web_link("EBOOKS",
  "Text=EBOOKS",
  "Snapshot=t2.inf",
  LAST);

web_link("Find Rocket eBooks",

  "Text=Find Rocket eBooks",
  "Snapshot=t3.inf",
  LAST);
```

特定のセグメントを対象に、Vuser にエラーでもスクリプトの実行を継続させるには、適切な `lr_continue_on_error` ステートメントで対象セグメントを囲みます。

```
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);
```

Vuser スクリプトの同期化

同期化関数を追加して、Vuser スクリプトの実行をアプリケーションの出力と同期させることができます。同期化は、RTE Vuser スクリプトにのみ適用されます。

次の同期化関数を使用できます。

| 関数 | 説明 |
|---------------------------------|---|
| TE_wait_cursor | カーソルがターミナル・ウィンドウ内の指定した位置に出現するまで待ちます。 |
| TE_wait_silent | クライアント・アプリケーションが指定した秒数の間無動作になるまで待ちます。 |
| TE_wait_sync | システムが X-SYSTEM または入力抑制モードから制御が戻るまで待ちます。 |
| TE_wait_text | 文字列が指定した位置に出現するまで待ちます。 |
| TE_wait_sync_transaction | システムが最新の X SYSTEM モードを維持していた時間を記録します。 |

RTE Vuser スクリプトにおける同期化関数の使用法の詳細については、『第 2 巻 - プロトコル』の「RTE - 同期」を参照してください。

ユーザの思考遅延時間のエミュレート

連続する操作の合間のユーザの待ち時間を「**思考遅延時間**」といいます。Vuser は、`lr_think_time` 関数を使ってユーザの思考遅延時間をエミュレートできます。Vuser スクリプトを記録すると、VuGen によって実際の思考遅延時間が記録され、適切な `lr_think_time` ステートメントが Vuser スクリプトに挿入されます。記録された `lr_think_time` ステートメントは後で編集できます。また、手作業でも `lr_think_time` ステートメントを Vuser スクリプトに追加できます。

思考遅延時間ステートメントを手作業で追加するには、次の手順を実行します。

- 1 希望の場所にカーソルを置きます。
- 2 **[挿入]** > **[ステップの追加]** を選択します。[ステップの追加] ダイアログ・ボックスが開きます。
- 3 **[思考遅延時間]** を選択し、「**OK**」をクリックします。[思考遅延時間] ダイアログ・ボックスが開きます。



- 4 思考遅延時間を秒単位で指定し、「**OK**」をクリックします。

注： Java Vuser スクリプトを記録する場合、`lr_think_time` ステートメントは Vuser スクリプトに挿入されません。

思考遅延時間の設定を使って、Vuser スクリプトを実行するときの `lr_think_time` ステートメントの処理方法を設定できます。思考遅延時間の設定にアクセスするには、VuGen のメイン・メニューから **[仮想ユーザ]** > **[実行環境の設定]** を選択して **[思考遅延時間]** ノードをクリックします。詳細については、『**Online Function Reference**』（英語版）（**[ヘルプ]** > **[関数リファレンス]**）を参照してください。

コマンド・ライン引数の取り扱い

スクリプトを実行するときにコマンド・ライン引数を指定することで、実行時に Vuser スクリプトに値を渡すことができます。[実行環境設定] ダイアログ・ボックスの中でコマンド・ライン引数を指定できます。詳細については、429 ページ「実行環境の追加属性の設定」を参照してください。

コマンド・ライン引数を読み取って、値を Vuser スクリプトに渡すことができる関数として、次の3つがあります。

| 関数 | 説明 |
|-----------------------------------|--------------------------|
| <code>lr_get_attrib_double</code> | double float 型の引数を取得します。 |
| <code>lr_get_attrib_long</code> | long int 型の引数を取得します。 |
| <code>lr_get_attrib_string</code> | 文字列を取得します。 |

コマンド・ラインの形式は次の2つのどちらかを使用します。スクリプト名の後ろに、引数とその値を2つ1組で指定します。

```
スクリプト名 -引数 引数値 -引数 引数値
```

```
スクリプト名 -引数 引数値 -引数 引数値
```

たとえば次の例は、pc4 という Load Generator で script1 を5回繰り返すためのコマンド・ライン文字列を示します。

```
script1 -host pc4 -loop 5
```

コマンド・ライン解析関数またコマンド・ラインでの引数の使い方の詳細については、『[Online Function Reference](#)』（英語版）（[ヘルプ](#) > [関数リファレンス](#)）を参照してください。

テキストの暗号化

スクリプト内のテキストを暗号化して、パスワードなどの機密性の高いテキスト文字列を保護できます。暗号化は、ユーザ・インタフェースから自動的に行うことができます。また、プログラムを使って手作業でもできます。暗号化した文字列は、スクリプト中に暗号化された文字列として表れます。VuGen は 32 ビットの暗号化をサポートしています。

スクリプトで暗号化された文字列を使用するには、`lr_decrypt` 関数を使用して復号する必要があります。

```
lr_start_transaction(lr_decrypt("3c29f4486a595750"));
```

文字列は、いつでも復号して元の値に戻せます。

文字列を暗号化するには、次の手順を実行します。

- 1 ツリー・ビューを表示できるプロトコルの場合は、スクリプト・ビューでスクリプトを表示します。[表示] > [スクリプト ビュー] を選択します。
- 2 暗号化するテキストを選択します。
- 3 右クリック・メニューから [文字列を暗号化 (対象文字列)] を選択します。

暗号化された文字列を元に戻すには、次の手順を実行します。

- 1 ツリー・ビューを表示できるプロトコルの場合は、スクリプト・ビューでスクリプトを表示します。[表示] > [スクリプト ビュー] を選択します。
- 2 元に戻す文字列を選択します。
- 3 右クリック・メニューから [暗号化文字列を復元 (対象文字列)] を選択します。

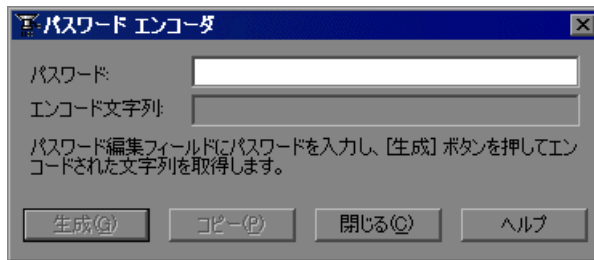
`lr_decrypt` 関数の詳細については、『[Online Function Reference](#)』（英語版）（[ヘルプ] > [関数リファレンス]）を参照してください。

手動でのパスワードのエンコーディング

パスワードを暗号化し、その結果生成された文字列をスクリプト内の引数またはパラメータ値として使用できます。たとえば、ユーザがパスワードを入力しなければならないフォームが Web サイトにあるとします。異なるパスワードにサイトがどのように応答するかをテストしたいが、同時にパスワードの安全性も保護したいとします。[パスワード エンコーダ] を使えばパスワードを暗号化し、テーブルに値を安全な形式で入力できます。

パスワードを暗号化するには、次の手順を実行します。

- 1 Windows メニューの [スタート] > [プログラム] > [LoadRunner] > [Tools] > [パスワード エンコーダ] を選択します。[パスワード エンコーダ] ダイアログ・ボックスが開きます。



- 2 [パスワード] ボックスにパスワードを入力します。
- 3 [生成] をクリックします。[パスワード エンコーダ] によってパスワードが暗号化され、暗号化された値が [エンコード文字列] フィールドに表示されます。
- 4 [コピー] ボタンを使用して、暗号化された値をコピーし、データ・テーブルに貼り付けます。
- 5 暗号化したいパスワードごとに、この手順を繰り返します。
- 6 [閉じる] をクリックして、[パスワード エンコーダ] を閉じます。

スクリプトへのファイルの追加

ファイルをスクリプト・ディレクトリに追加し、スクリプトの実行時に使用できるようにすることが可能です。ファイルがテキスト・ベースの場合、VuGenのエディタに表示して編集できます。

ファイルを追加するには、次の手順を実行します。

- 1 スクリプトを開きます。
- 2 [ファイル] > [ファイルをスクリプトに追加] を選択するか、[タスク] 表示枠を右クリックして右クリック・メニューで [ファイルをスクリプトに追加] を選択します。
- 3 ファイルを見つけて [開く] をクリックします。VuGen によって、選択したファイルがスクリプト・ディレクトリおよび VuGen エディタに追加されます。
- 4 内容を表示するには、左側表示枠でファイルを選択します。



第 7 章

ビジネス・プロセス・レポートの作成

VuGen では、スクリプトの情報を Microsoft Word ドキュメントにエクスポートすることによって、ビジネス・プロセス・レポートを作成できます。

本章の内容

- ▶ Word へのスクリプトのエクスポートについて (129 ページ)
- ▶ レポートの詳細の指定 (130 ページ)
- ▶ レポートの内容の指定 (131 ページ)

次の情報は、AJAX (Click and Script)、Citrix_ICA、Oracle NCA、Oracle Web Applications 11i、PeopleSoft Enterprise、RDP、SAP (Click and Script)、SAPGUI、SAP - Web、Web (Click and Script)、Web (HTTP/HTML)、および Web サービス・プロトコルに適用されます。

Word へのスクリプトのエクスポートについて

スクリプト作成の最終段階で、ビジネス・プロセスについて記述されたレポートを作成できます。VuGen によって、スクリプトの情報が Microsoft Word ドキュメントにエクスポートされます。

事前に作成したテンプレートまたは VuGen に用意されているテンプレートを使用して、テスト実行に関するサマリ情報が記載されたレポートを作成できます。

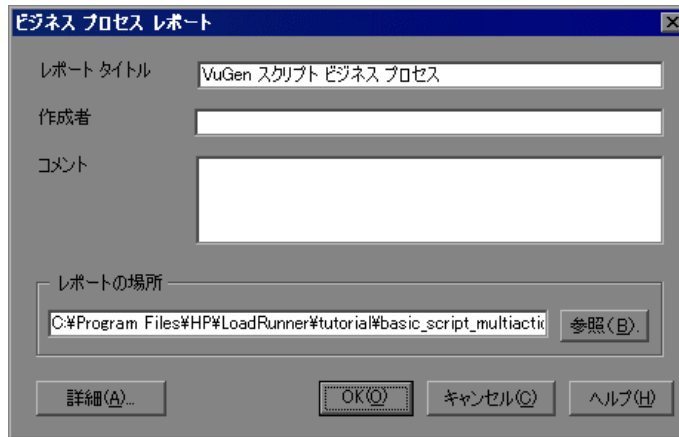
VuGen では、レポートに含める情報のタイプを指定することによって、レポートの内容をカスタマイズできます。

レポートの詳細の指定

レポートの内容を指定する前に、レポートの名前と簡単な説明を指定します。適切なコメントを入力し、レポートを格納する場所を指定します。

ビジネス・プロセス・レポートを作成するには、次の手順を実行します。

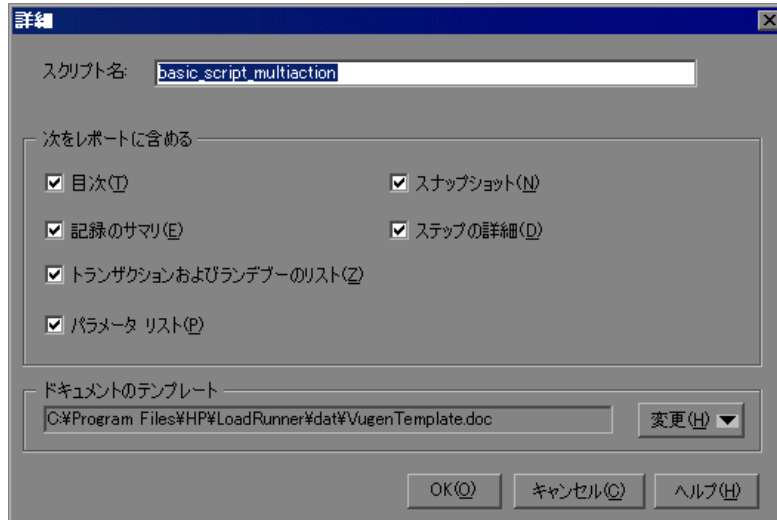
- 1 [ファイル] > [ビジネス プロセス レポートを作成] を選択します。[ビジネス プロセス レポート] ダイアログ・ボックスが開きます。



- 2 [レポートタイトル] ボックスにタイトルを指定します。
- 3 [作成者] ボックスに名前を入力します。
- 4 [コメント] ボックスにコメントを入力します。
- 5 標準設定のレポートの場所およびファイル名を受け入れます。または、必要なパスを参照します。標準設定の場所は、スクリプトが保存されている場所です。また、標準設定のレポート名は <スクリプト名> **_Business_Processes.doc** です。
- 6 [詳細] をクリックし、レポートの内容を指定します。内容の選択の詳細については、131 ページ「レポートの内容の指定」を参照してください。
- 7 レポートを生成するには、[OK] をクリックします。

レポートの内容の指定

ビジネス・プロセス・レポートの内容を選択できます。標準設定では、すべてのオプションが有効になっています。標準設定では、.usr ファイル名は「**スクリプト名**」ボックスに表示されます。



次の内容オプションのうち、1つ以上を選択できます。

- ▶ **[目次]**：レポート上のほかのすべての内容のページ番号を示す目次。特定のオプションを無効にすると、そのオプションに対応する項目は目次に表示されません。
- ▶ **[記録のサマリ]**：[タスク] リストで [作成サマリ] リンクをクリックすると表示される、記録セッションのサマリ。
- ▶ **[トランザクションおよびランデブーのリスト]**：スクリプトに定義されているすべてのトランザクションおよびランデブーの包括的なリスト。
- ▶ **[パラメータ リスト]**：スクリプトに対して定義されているすべてのパラメータのリスト。このリストは、[パラメータ リスト] ダイアログ・ボックス（**[仮想ユーザ]** > **[パラメータ リスト]**）に表示されるパラメータに対応します。
- ▶ **スナップショット**：ステップの名前と説明の横に表示される、記録されたステップの実際のスナップショット。

注： Oracle NCA および Web サービス・レポートには、スナップショットは含まれていません。

- ▶ **ステップの詳細：** ツリー・ビューに表示される各ステップの簡単な説明。
- ▶ **ドキュメントのテンプレート：** レポートに使用するテンプレートのパスおよびファイル名。標準設定のテンプレートは、LoadRunner の **dat** フォルダにあります。

レポート・テンプレートを変更するには、**[変更]** を選択し、.doc 拡張子を持つ新しいテンプレートを指定します。新しいテンプレートを作成する場合は、新しいテンプレートの基礎として既存のテンプレートを使用することをお勧めします。これにより、必要なブックマークおよびスタイルが、新しいテンプレートでも維持されるようになります。

第 8 章

ステートメントの相関

ステートメントを相関させることで、Vuser スクリプトを最適化できます。VuGen の相関クエリ機能によって、ステートメントの結果を別のステートメントへの入力項目として使うことで、ステートメントを相互に結び付けることができます。

本章の内容

- ▶ ステートメントの相関について (133 ページ)
- ▶ C Vuser 用の相関関数の使用 (135 ページ)
- ▶ Java Vuser 用の相関関数の使用法 (137 ページ)
- ▶ WDiff を使った Vuser スクリプトの比較 (138 ページ)
- ▶ 保存したパラメータの変更 (140 ページ)

ステートメントの相関について

ステートメントを相関させる主な目的を以下に示します。

コードを簡素化または最適化するため

たとえば、相互に依存するクエリを連続して実行する場合、コードが非常に長くなってしまふことがあります。コードのサイズを小さくするためにクエリをネストできますが、精度が損なわれたり、コードが複雑化してわかりにくくなったりします。ステートメントを相関させることにより、ネストさせずにクエリを連結できます。

動的データの生成のため

多くのアプリケーションや Web サイトは現在の日時に基づいてセッションを識別します。スクリプトを再生しようとする時、現在の時刻が記録された時刻と異なるので失敗します。データを相関させると、動的なデータを保持し、これをシナリオ実行の全体を通して使用できます。

固有のデータ・レコードに対応するため

アプリケーション（たとえばデータベース）によっては、一意の値を使用する必要があります。記録時に一意だった値も、スクリプト実行時にはもう一意ではありません。たとえば、新しい銀行口座を開設するプロセスを記録したとします。各新規口座にはユーザの知らない一意の口座番号が割り当てられます。この番号は記録時に、一意のキーでなければならないという制約のあるテーブルに挿入されます。記録どおりにスクリプトを実行しようとする時、新しい一意の番号ではなく、記録された番号で口座を作成しようとする時、その結果、口座番号がすでに存在するという理由でエラーとなります。

スクリプト実行時にエラーが発生した場合は、スクリプトの中でエラーが発生した場所を調べます。多くの場合、相関クエリによって、あるステートメントの結果を別のステートメントの入力として使用できるようにすれば問題を解決できます。

スクリプトの相関の主な手順は次のとおりです。

1 相関させる値を特定します。

データベース `Vuser` スクリプトの場合、`VuGen` の助けを借りて相関対象を決めることができます。実行ログでエラー・メッセージをダブルクリックして、スクリプト内の問題が生じているステートメントに移動し、相関の候補となる値を探します。

あるいは、`VuGen` に付属の `WDiff` ユーティリティを使って、スクリプト内の不一致を調べることができます。詳細については、138 ページ「`WDiff` を使った `Vuser` スクリプトの比較」を参照してください。

2 結果を保存します。

適切な関数を使って、クエリの値を変数に保存します。相関関数はプロトコルごとに異なります。相関関数の名前には通常 `web_reg_save_param` や `lrs_save_param` といった `save_param` 文字列が含まれます。相関の方法の詳細については、各プロトコルを説明している章を参照してください。データベースや Web など、いくつかのプロトコルには、`VuGen` によってスクリプトに関数が自動的に追加されます。

3 保存した値を参照します。

クエリまたはステートメント内の定数を保存された変数で置換します。

いくつかのプロトコルには、組み込み式の自動または部分的に自動の相関機能があります。

- ▶ Java 言語 Vuser については、『第2巻 – プロトコル』の「Java – 相関」を参照してください。
- ▶ データベース Vuser については、『第2巻 – プロトコル』の「データベース – スクリプトの相関」を参照してください。
- ▶ Web Vuser については、『第2巻 – プロトコル』の「Web (HTTP/HTML) の相関ルール」を参照してください。
- ▶ COM Vuser については、『第2巻 – プロトコル』の「COM の概要と相関」を参照してください。

C Vuser 用の相関関数の使用

固有の相関関数を持たないプロトコルのステートメントを相関させるには、C Vuser 相関関数を使うことができます。これらの関数はすべての C 言語に基づく Vuser に対して使用できます。これらの関数を使って、文字列をパラメータに保存し、必要に応じて取り出せます。Java Vuser 用の類似関数については、137 ページ「Java Vuser 用の相関関数の使用法」を参照してください。

| | |
|-----------------------|--------------------------------------|
| lr_eval_string | 指定されたパラメータに一致するパラメータをすべて現在の値で置き換えます。 |
| lr_save_string | NULL で終わる文字列をパラメータに保存します。 |
| lr_save_var | 可変長文字列をパラメータに保存します。 |

これらの関数の構文の詳細については、『**Online Function Reference**』（英語版）を参照してください。

lr_eval_string の使用法

次の例では、lr_eval_string を使ってパラメータ row_cnt を現在の値で置換しています。この値を、lr_output_message を使って出力ウィンドウに送っています。

```
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>"));
```

lr_save_string の使用法

NULL で終了する文字列をパラメータに保存するには、lr_save_string を使います。可変長文字列を保存するには、lr_save_var を使い、保存する文字列の長さを指定します。

次の例では、lr_save_string を使用してパラメータ emp_id に 777 という値を保存しています。このパラメータを後で別のクエリや処理で使用することができます。

```
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 は戻り値 "777" を示す */
lr_save_string("777", "emp_id");
```


Java Vuser 用の相関関数の使用法

Java Vuser のステートメントを相関させるには、Java Vuser の相関関数を使用します。これらの関数はすべての Java タイプの Vuser に対して使用できます。これらの関数を使って文字列をパラメータに保存し、必要に応じて取り出せます。

| | |
|-----------------------|---------------------------|
| lr.eval_string | パラメータを現在の値で置き換えます。 |
| lr.eval_data | パラメータをバイト値で置き換えます。 |
| lr.eval_int | パラメータを整数値で置き換えます。 |
| lr.eval_string | パラメータを文字列で置き換えます。 |
| lr.save_data | バイトをパラメータとして保存します。 |
| lr.save_int | 整数をパラメータとして保存します。 |
| lr.save_string | NULL で終わる文字列をパラメータに保存します。 |

CORBA または RMI セッションを記録するときに、VuGen は内部で相関を実行します。詳細については、『第2巻 – プロトコル』の「Java – 相関」を参照してください。

Java 文字列関数の使用法

Java Vuser スクリプトをプログラミングするときに、Java Vuser 文字列関数を使用してスクリプトを相関させることができます。

次の例では、lr.eval_int を使って、変数 ID_num をその値で置き換えています。ID_num は、スクリプトの中で先に定義されているものとします。

```
lr.message(" Track Stock :" + lr.eval_int(ID_num) );
```

次の例では、lr.save_string を使って、John Doe をパラメータ Student に保存しています。その後、このパラメータを出力メッセージの中で使っています。

```
lr.save_string("John Doe", "Student");
// ...
lr.message("Get report card for " + lr.eval_string(" < Student > "));
classroom.getReportCard
```

WDiff を使った Vuser スクリプトの比較

相関させる値を判断するのに役立つツールとして、**WDiff** があります。このツールを使用することにより、記録したスクリプトと結果を比較し、相関させるべき値を判断することができます。

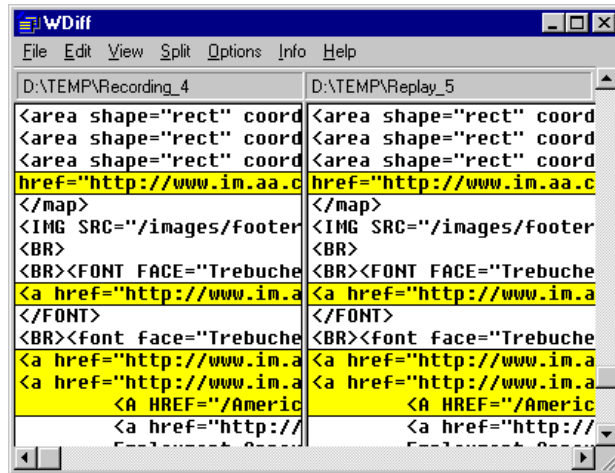
ほかのプロトコルを使って作業をする場合は、実行ログを参照し、スクリプトが失敗した場所を特定し、その後 **WDiff** ユーティリティを使って相関させる値を調べます。

WDiff ユーティリティを効果的に使うには、同じ操作を2度記録し、スクリプト（または Tuxedo, WinSock, Jolt の場合にはデータ・ファイル）を比較します。WDiff に、違いが黄色で示されます。ただし、すべての相違部分が相関すべき値というわけではありません。たとえば、実行の時刻を示す受信バッファは相関を必要とはしません。

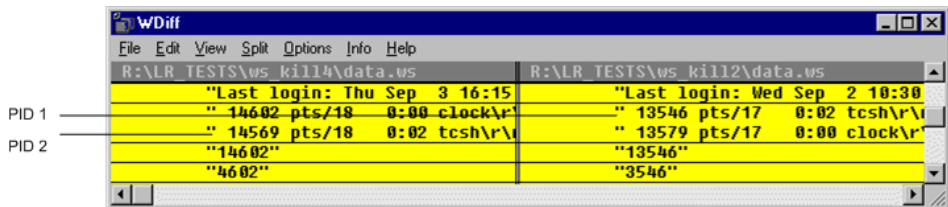
WDiff を使って相関を検索するには、次の手順を実行します。

- 1 スクリプトを記録し、保存します。
- 2 新しいスクリプトを作成し、まったく同じ操作を記録します。スクリプトを保存します。
- 3 比較したい部分（**Actions** や **data.ws** など）を選択します。
- 4 **[ツール]** > **[仮想ユーザと比較]** を選択します。**[テストを開く]** ダイアログ・ボックスが開きます。

- 5 比較する Vuser スクリプト（現在の VuGen ウィンドウに表示されているもの以外のスクリプト）を指定し，[OK] をクリックします。WDiff が開き，スクリプト間の相違が黄色で強調表示されます。



- 6 相違だけを表示するには，[WDiff] ウィンドウをダブルクリックします。



- 7 どの値を相関させるかを判断します。

上の例では，WDiff は 2 つの Winsock Vuser スクリプトの **data.ws** を比較しています。この場合，相関すべき値は 2 つの記録の間で違いのある **clock** プロセスの PID です。

注： WDiff が標準設定のユーティリティですが，ユーザ定義の比較ツールを指定することもできます。詳細については，35 ページ「比較ツール」を参照してください。

相関の以降の作業については、該当する項を参照してください。

- ▶ Java 言語 Vuser については、『第2巻 – プロトコル』の「Java – 相関」を参照してください。
- ▶ データベース Vuser については、『第2巻 – プロトコル』の「データベース – スクリプトの相関」を参照してください。
- ▶ Web Vuser については、『第2巻 – プロトコル』の「Web (HTTP/HTML) の相関ルール」を参照してください。
- ▶ COM Vuser については、『第2巻 – プロトコル』の「COM の概要と相関」を参照してください。
- ▶ Tuxedo Vuser については、『第2巻 – プロトコル』の「Tuxedo プロトコル」を参照してください。
- ▶ WinSock Vuser については、『第2巻 – プロトコル』の「Windows Sockets (WinSock) プロトコル」を参照してください。

保存したパラメータの変更

パラメータに値を保存したら、実際にスクリプトの中で使う前に、パラメータを変更する必要がある場合があります。パラメータを使って算術演算を行う場合は、C 関数の `atoi` または `atol` を使って値を文字列から整数に変更する必要があります。値を整数に変換したら、スクリプトの中で新しい変数を使用するには、その整数をもう一度文字列に戻す必要があります。

次の WinSock の例では、オフセット 67 の位置にあるデータをパラメータ `param1` に保存しています。次に、`atol` を使って文字列を `long int` 型に変換します。`param1` の値に 1 を加算した後で、`sprintf` を使って値を文字列に戻し、新しい文字列 `new_param1` として保存します。パラメータの値は `lr_output_message` を使って表示しています。この新しい値は、以降でスクリプトの中で使用することができます。

```
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"));
```

第 9 章

スタンドアロン・モードでの Vuser スクリプトの実行

Vuser スクリプトを作成し、その実行環境を設定したら、スクリプトをスタンドアロン・モードで実行してテストします。

本章の内容

- ▶ スタンドアロン・モードでの Vuser スクリプトの実行について (142 ページ)
- ▶ VuGen での Vuser スクリプトの実行 (142 ページ)
- ▶ Vuser スクリプトの再生 (146 ページ)
- ▶ VuGen のデバッグ機能の使用 (149 ページ)
- ▶ Web Vuser スクリプトでの VuGen のデバッグ機能の使用 (154 ページ)
- ▶ VuGen ウィンドウを使った作業 (155 ページ)
- ▶ ファイル内の検索 (156 ページ)
- ▶ UNIX コマンド・ラインからの Vuser スクリプトの実行 (159 ページ)
- ▶ スクリプトのテストへの統合 (161 ページ)

スタンドアロン・モードでの Vuser スクリプトの実行について

スクリプトを作成したら、スクリプトを VuGen インタフェースから直接、スタンドアロン・モードで実行し、その機能を確認めます。UNIX ベースのスクリプトについては、UNIX のコマンド・ラインから実行します。GUI の Vuser をスタンドアロン・モードで実行するには、WinRunner を使用します。

基本的な機能を確認するためにスタンド・アロンのテストとしてスクリプトを実行します。

スタンドアロンで正常に実行できたら、スクリプトを自分の環境 (LoadRunner シナリオ、Performance Center 負荷テスト、または Business Process Monitor プロファイル) に統合します。詳細については、**HP LoadRunner Controller**、**HP Performance Center**、または **HP Business Availability Center** を参照してください。

スタンドアロン・モードでスクリプトを実行したら、次を使ってスクリプトを拡張してカスタマイズできます。

- ▶ Vuser 関数 (第6章「Vuser スクリプトの拡張」または『オンライン関数リファレンス』を参照)。
- ▶ パラメータ (第13章「VuGen パラメータを使った作業」を参照)
- ▶ スクリプトのクエリの相関 (第8章「ステートメントの相関」を参照)

前述のステップは任意であり、すべてのスクリプトに適用できるわけではありません。

VuGen での Vuser スクリプトの実行

Vuser スクリプトを作成したら、VuGen を使ってスクリプトを実行し、正しく実行できることを確認します。再生のためのオプションをいくつか設定できます。

注： VuGen で Vuser スクリプトを実行できるのは Windows プラットフォームだけです。UNIX ベースの Vuser スクリプトを実行するには、554 ページ「UNIX コマンド・ラインからの Vuser の実行」を参照してください。

再生オプションの設定

Vuser スクリプトは表示実行モードまたは非表示実行モードで実行できます。表示実行モードで実行すると、VuGen によって Vuser スクリプト内の現在実行されている行が強調表示されます。各ステップの様子がさらによく見えるように、このモードの遅延を設定することができます。非表示実行モードで実行すると、VuGen によって Vuser スクリプトが実行されますが、現在実行されている行は強調表示されません。

- ▶ **[表示実行遅延]**：コマンドを実行する遅延間隔をミリ秒単位で指定します。標準の遅延時間は、0 です。
- ▶ **[Action のセクションのみで関数を表示実行する]**：Action セクションの内容だけ (init および end セクションの内容を除く) を表示実行モードで実行します。
- ▶ **[結果ディレクトリの指定を求める]**：VuGen からスクリプトを実行する前に結果ディレクトリを指定するよう求められます。このオプションを選択していない場合は、VuGen によって **result1** という名前が自動的にディレクトリに付けられます。スクリプトの以降の実行結果は、別の結果ファイルを指定しない限り、前回の結果ファイルに自動的に上書きされます。結果は必ずスクリプトのサブディレクトリに格納されます。
- ▶ **[再生後に表示]**：再生後の VuGen の動作を指示します。
 - ▶ **[再生の前のビュー]**：再生前のビューに戻します。
 - ▶ **[再生のサマリ]**：ワークフロー・ウィザードの [再生のサマリ] ウィンドウに直接移動します。
 - ▶ **[テスト結果]**：[テスト結果] を開きます (再生後に [表示] > [テスト結果] を選択しても開くことができます)。

標準設定値を使用

[標準設定を使用] をクリックすると、元の値にリセットされます。

[表示実行遅延] は 0 ミリ秒になります。

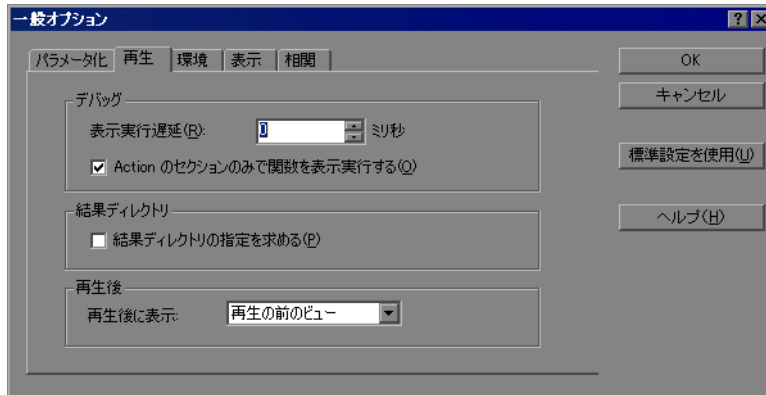
[Action のセクションのみで関数を表示実行する] は有効になります。

[結果ディレクトリの指定を求める] は無効になります。

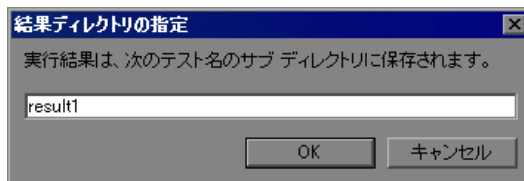
[再生後に表示] は [再生の前のビュー] に設定されます。

表示実行モードを有効にして、そのプロパティを設定するには、次の手順を実行します。

- 1 [表示] > [実行のアニメーション化] を選択して、表示実行モードで実行します。VuGen によって、[実行のアニメーション化] メニュー・オプションの横にチェック・マークが付けられ、表示実行モードが有効になります。
- 2 表示実行に対して遅延を設定するには、[ツール] > [一般オプション] を選択します。[一般オプション] ダイアログ・ボックスが開きます。



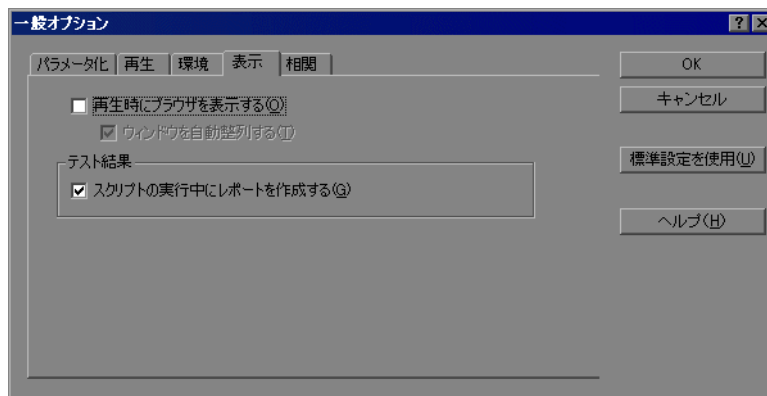
- 3 [再生] タブを選択します。
- 4 [表示実行遅延] ボックスに、コマンド間の遅延時間をミリ秒単位で指定し、[OK] をクリックします。
- 5 Actions セクションの内容のみを表示実行するには、[Actions のセクションのみで関数を表示実行する] を選択します。
- 6 VuGen からスクリプトを実行する前に結果ディレクトリを指定できるようにするには、[結果ディレクトリの指定を求める] を選択します。実行コマンドをクリックすると、[結果ディレクトリの指定] ダイアログ・ボックスが開きます。



- 7 実行結果を格納するディレクトリの名前を入力するか、標準の名前をそのまま使用して [OK] をクリックします。

表示オプションの設定

Web Vuser スクリプトを実行しているときは、[表示] オプション ([ツール] > [一般オプション]) を設定できます。[表示] オプションによって、VuGen で実行時ビューアを表示するか、スクリプト実行中に VuGen でレポートを生成するかなどを指定します。



- ▶ **[再生時にブラウザを表示する]**：実行時ビューアを有効にします。スクリプトの実行終了時に実行時ビューアを最小化するには、**[ウィンドウを自動整列する]** オプションを選択します。
- ▶ **[スクリプトの実行中にレポートを作成する]**：結果サマリ・レポートを生成するように Vuser に指示します。スクリプトの実行後にレポートを開くには、**[表示] > [テスト結果]** を選択します。

標準設定では、**[再生時にブラウザを表示する]** オプションは無効で、**[スクリプトの実行中にレポートを作成する]** オプションは有効になっています。これらの値を元に戻すには、**[標準設定を使用]** をクリックします。

これらのオプションを使用したデバッグ方法の詳細については、154 ページ「Web Vuser スクリプトでの VuGen のデバッグ機能の使用」を参照してください。

[表示] オプションを設定するには、次の手順を実行します。

- 1 VuGen のメニューから [ツール] > [一般オプション] を選択します。[一般オプション] ダイアログ・ボックスが開きます。[表示] タブを選択します。
- 2 実行時ビューアを有効にするには、[再生時にブラウザを表示する] を選択します。スクリプトの実行終了時に実行時ビューアを最小化するには、[ウィンドウを自動整理する] を選択します。
- 3 Vuser に結果サマリ・レポートを生成させるには、[スクリプトの実行中にレポートを作成する] を選択します。スクリプトの実行後にレポートを開くには、[表示] > [テスト結果] を選択します。
- 4 設定を承認するには [OK] をクリックします。[一般オプション] ダイアログ・ボックスが閉じます。

Vuser スクリプトの再生

スクリプトをテスト環境や実運用環境に統合する前に、スクリプトを VuGen から実行して正しく機能することを確認します。VuGen は、再生を監視し、既存の問題または問題の可能性を特定するためのツールをいくつか備えています。次のツールがあります。

- ▶ 再生ログの表示
- ▶ [実行時データ] タブ
- ▶ ステップ実行コマンド
- ▶ ブレークポイント
- ▶ ブックマーク
- ▶ [移動] コマンド

VuGen でスクリプトを再生するには、次の手順を実行します。

- 1 [仮想ユーザ] > [実行] を選択します。

VuGen のメイン・ウィンドウの下部に出力ウィンドウが開いた後、あるいは、すでに開いている場合には内容がクリアされた後に、Vuser スクリプトの実行が開始されます。ツリー・ビューでは、Vuser スクリプトはスクリプト内の 1 番目のアイコンから実行されます。スクリプト・ビューでは、Vuser スクリプトはスクリプトの 1 行目から実行されます。

- 2 Vuser のすべてのアクションのログのほか、警告やエラーを見るには、[出力] ウィンドウの **[再生ログ]** タブをクリックします。詳細については、147 ページ「再生ログの表示」を参照してください。
- 3 実行時データおよびパラメータのサマリをそれらの使用中に表示するには、[出力] ウィンドウの **[実行時データ]** タブをクリックします。詳細については、148 ページ「[実行時データ] タブ」を参照してください。
- 4 スクリプトの実行中または実行後に、[出力] ウィンドウを非表示にするには、**[表示]** > **[出力ウィンドウ]** を選択します。[出力] ウィンドウが閉じられ、**[表示]** メニューの **[出力ウィンドウ]** の横のチェック・マークが消えます。
- 5 実行中の Vuser スクリプトを中断するには、**[仮想ユーザ]** > **[一時停止]** を選択して、スクリプトの実行を一時停止します。または、**[仮想ユーザ]** > **[停止]** を選択して、スクリプトの実行を終了します。

再生ログの表示

[出力] ウィンドウの [再生ログ] には、実行時の Vuser のアクションを表すメッセージが表示されます。この情報は、スクリプトがシナリオまたはプロファイルの中で実行されたときに、どのように実行されるかを示します。

スクリプトの実行が完了したら、スクリプトをエラーなく実行できたかどうかを確認するために、再生ログのメッセージを調べます。

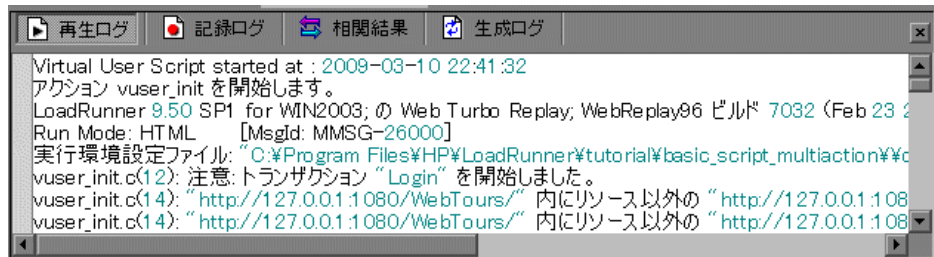
[再生ログ] では、テキストがさまざまに色分けされています。

- ▶ **黒**：標準の出力メッセージ
- ▶ **赤**：標準のエラー・メッセージ
- ▶ **緑**：引用符に囲まれて表示されるリテラル文字列（URL など）
- ▶ **青**：トランザクション情報（開始ステータス、終了ステータス、持続時間）
- ▶ **橙**：反復の始まりと終わり

アクション名で始まる行をダブルクリックすると、出力ログを生成したスクリプト内の対応するステップにカーソルが移動します。

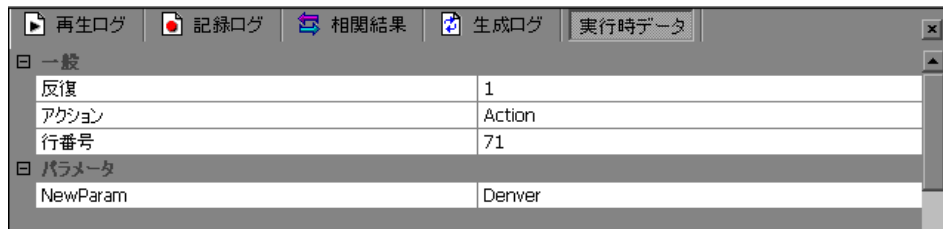
出力ウィンドウの表示と非表示の詳細については、146 ページ「Vuser スクリプトの再生」を参照してください。

次の例は、Web Vuser スクリプトの実行によって生成された再生ログのメッセージを示します。



【実行時データ】タブ

再生中に最新になったスクリプト情報を、[実行時データ] タブを使用して追跡することができます。



再生中に、右端の **【実行時データ】** タブをクリックします。このタブには展開または折りたたみが可能な2つのセクションがあります。

- ▶ **【一般】** : [一般] セクションには、現在の反復数、現在再生されているステップのアクション名、スクリプト内での行番号 (スクリプト・ビュー) が表示されます。
- ▶ **【パラメータ】** : [パラメータ] セクションには、スクリプトに定義されているすべてのパラメータと、選択した更新方法 (順次、一意など) に基づくそれらのパラメータの置換値が表示されます。これらの情報はパラメータがスクリプトの中で使用されていない場合にも表示されます。詳細については、第13章「VuGen パラメータを使った作業」を参照してください。

テストの実行後には [実行時データ] タブにはアクセスできません。このタブには、テストの再生中に変化するデータのみが表示されるからです。

VuGen のデバッグ機能の使用

VuGen には、Vuser スクリプトのデバッグに役立つ、ステップ実行コマンドとブレイクポイントの2つのオプションがあります。これらのオプションは、VBScript Vuser および VB Vuser タイプの Vuser では使用できません。

また VuGen には、Web Vuser スクリプトのデバッグに役立つ追加機能があります。詳細については、154 ページ「Web Vuser スクリプトでの VuGen のデバッグ機能の使用」を参照してください。

デバッグ・ツールバーを表示するには、次の手順を実行します。

ツールバー領域を右クリックして、**[デバッグ]** を選択します。デバッグ・ツールバーがツールバー領域に表示されます。



ステップ実行コマンド

ステップ実行コマンドは、スクリプトを1回に1行ずつ実行します。これによりスクリプトの実行を追うことができます。

ステップ実行コマンドでスクリプトを実行するには、次の手順を実行します。



- 1 **[仮想ユーザ]** > **[ステップごとに実行]** を選択するか、デバッグ・ツールバーの **[ステップ]** ボタンをクリックします。

VuGen によってスクリプトの最初の行が実行されます。

- 2 スクリプトの実行が完了するまで **[ステップ]** ボタンをクリックして、スクリプトの実行を続けます。

ブレイクポイント

ブレイクポイントは、スクリプトの特定の場所で実行を一時停止します。これにより、スクリプトの実行中に、あらかじめ定義しておいたポイントで、スクリプトによるアプリケーションへの影響を調査できます。ブレイクポイントを管理するには、ブレイクポイント・マネージャを使用します。

ブレークポイントを設定するには、次の手順を実行します。

- 1 スクリプト内で実行を停止する行にカーソルを移動します。



- 2 [挿入] > [ブレークポイントの設定 / 解除] を選択するか、デバッグ・ツールバーの [ブレークポイント] ボタンをクリックします。または、キーボードの F9 キーを押します。ブレークポイントの記号 (●) がスクリプトの左余白に表示されます。



- 3 ブレークポイントを無効にするには、ブレークポイントの記号のある行にカーソルを合わせて、デバッグ・ツールバーの [ブレークポイントの有効化 / 無効化] ボタンをクリックします。ブレークポイント記号の中の白い点が表示され、そのブレークポイントが無効であることを示します。あるブレークポイントが無効にすると、その次のブレークポイントでスクリプトの実行が一時停止します。ブレークポイントを有効にするにはもう一度ボタンをクリックします。

ブレークポイントを削除するには、ブレークポイントの記号のある行にカーソルを合わせて、[ブレークポイント] ボタンをクリックするか、F9 キーを押します。

ブレークポイントを設定したスクリプトを実行するには、次の手順を実行します。

- 1 通常どおり、スクリプトの実行を開始します。

VuGen がブレークポイントに到達すると、スクリプトの実行が停止されます。ブレークポイントまでのスクリプト実行の影響を調査して、必要な変更を加え、そのブレークポイントからスクリプトの実行を再開します。

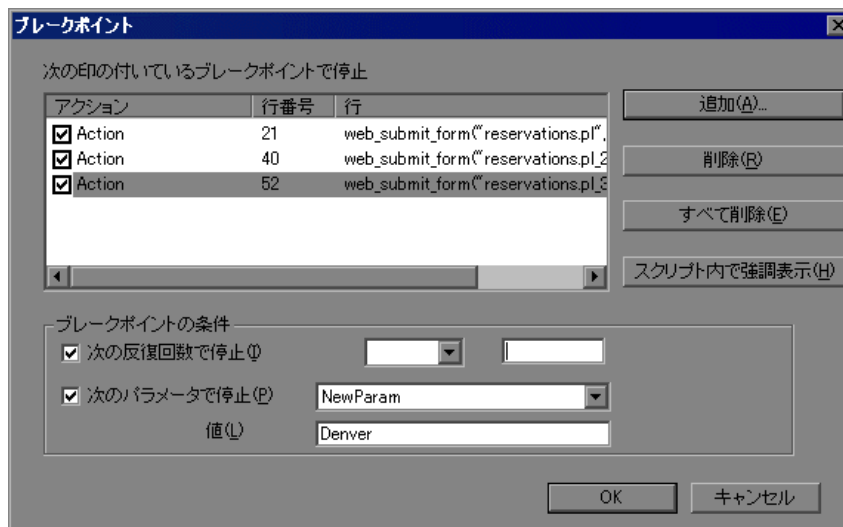
- 2 実行を再開するには、[仮想ユーザ] > [実行] を選択します。

再開すると、次のブレークポイントに到達するか、スクリプトが終了するまで、スクリプトの実行が続けられます。

ブレークポイント・マネージャ

ブレークポイント・マネージャを使用してブレークポイントを表示および管理することができます。ブレークポイント・マネージャから、スクリプト内のすべてのブレークポイントを操作できます。

ブレークポイント・マネージャを開くには、[編集] > [ブレークポイント] を選択します。



スクリプト内のブレークポイント位置へ移動するには、次の手順を実行します。

- 1 リストからブレークポイントを選択します。
- 2 [スクリプト内で強調表示] をクリックします。スクリプト内の行が強調表示されます。

一度に強調表示できるブレークポイントは1つだけです。

ブレイクポイントの管理

ブレイクポイント・マネージャから、ブレイクポイントの追加、削除、無効化、または条件の設定を実行できます。

ブレイクポイントを追加するには、次の手順を実行します。

- 1 **[追加]** をクリックします。[ブレイクポイントの追加] ダイアログ・ボックスが開きます。
- 2 **[アクション]** を選択し、ブレイクポイントを追加する **[行番号]** を指定します。
- 3 **[OK]** をクリックします。ブレイクポイントがブレイクポイントのリストに追加されます。

ブレイクポイントを削除するには、次の手順を実行します。

- 1 ブレイクポイントを1つ削除するには、ブレイクポイントを選択して **[削除]** をクリックします。
- 2 ブレイクポイントをすべて削除するには、**[すべて削除]** をクリックします。

ブレイクポイントを有効化または無効化するには、次の手順を実行します。

- 1 ブレイクポイントを有効にするには、[アクション] カラムで、アクションのチェック・ボックスを選択します。
- 2 ブレイクポイントを無効にするには、[アクション] カラムで、アクションのチェック・ボックスをクリアします。

ブレイクポイント・マネージャから、特定の条件のもとで実行を一時停止するためのブレイクポイントを設定できます。

ブレイクポイントの条件を設定するには、次の手順を実行します。

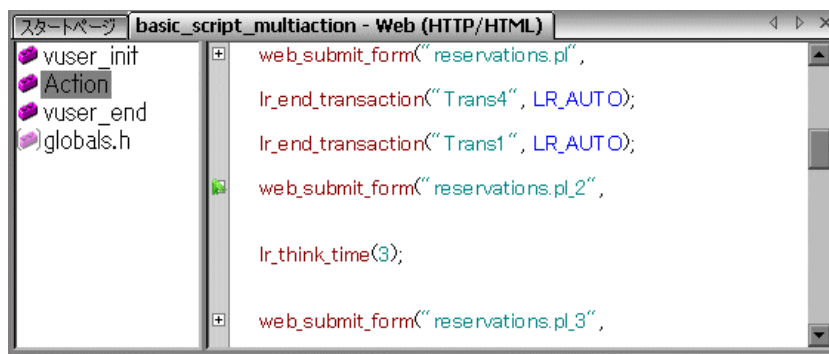
- 1 特定の反復回数後にスクリプトを一時停止するには、**[次の反復回数で停止]** を選択して、必要な回数を入力します。
- 2 パラメータ **X** が特定の値になったときにスクリプトを一時停止するには、**[次のパラメータで停止]** を選択して、必要な値を入力します。パラメータの詳細については、第13章「VuGen パラメータを使った作業」を参照してください。

ブックマーク

スクリプト・ビューで作業をしているときに、スクリプト内のさまざまな位置にブックマークを置くことができます。ブックマーク間を移動することで、コードの分析やデバッグを実行できます。

ブックマークを作成するには、次の手順を実行します。

- 1 必要な位置にカーソルを置いて **Ctrl + F2** キーを押します。アイコンがスクリプトの左余白に置かれます。



- 2 ブックマークを削除するには、必要な位置にカーソルを置いて **Ctrl + F2** キーを押します。ブックマーク・アイコンが左余白から削除されます。
- 3 ブックマーク間を移動するには、次の手順を実行します。

次のブックマークに移動するには、**F2** キーを押します。

前のブックマークに移動するには、**Shift + F2** キーを押します。

[編集] > [ブックマーク] メニュー項目を通じて、ブックマークの作成や、ブックマーク間の移動が可能です。

注：ブックマーク間の移動は現在のアクションの中でのみ実行できます。別のアクションのブックマークに移動するには、そのアクションを左側の表示枠で選択した後、**F2** キーを押します。

[移動] コマンド

ブックマークを使用せずにスクリプト内を移動するには、[移動] コマンドを使用できます。[編集] > [指定行へ移動] を選択し、スクリプトの行番号を指定します。この移動方法はツリー・ビューでもサポートされています。

特定のステップまたは関数の再生ログ・メッセージを調べるには、VuGen で該当するステップを選択し、[編集] > [再生ログのステップに移動] を選択します。[出力] ウィンドウの [再生ログ] タブ内の対応するステップの位置に、カーソルが置かれます。

Web Vuser スクリプトでの VuGen のデバッグ機能の使用

VuGen には、Web Vuser スクリプトのデバッグに役立つ付加的なツールとして、実行時ビューア（オンライン・ブラウザ）と結果サマリ・レポートがあります。

- ▶ Web Vuser スクリプトの実行時に実行時ビューアを表示するように VuGen を設定することができます。実行時ビューアは、VuGen 向けに開発されました。Vuser スクリプトの記録に使用するブラウザとは無関係です。実行時ビューアには、Vuser によってアクセスされる各 Web ページが表示されます。これにより Vuser が正しい Web ページにアクセスしているかどうかを確認できるので、Web Vuser スクリプトのデバッグ時に役立ちます。
- ▶ Web Vuser に対して、スクリプトの実行時に結果サマリ・レポートを生成させるかどうかを指定できます。結果サマリ・レポートとは、Web Vuser スクリプト内の各ステップの成功や失敗についてまとめたもので、各ステップで返された Web ページも確認できます。結果サマリ・レポートを使った作業の詳細については、[表示] > [テスト結果] を選択し、F1 キーを押して、オンライン・ヘルプを参照してください。

前述の [表示] オプションの設定の詳細については、145 ページ「表示オプションの設定」を参照してください。

注：Vuser に結果サマリ・レポートを生成させると、トランザクションの処理時間が増える場合があります。Vuser から結果サマリ・レポートを生成できるのは、VuGen から実行した場合だけです。スクリプトを Controller, または Business Process Monitor から実行した場合は、Vuser からレポートは生成されません。

VuGen ウィンドウを使った作業

次の機能を実行することで、VuGen のウィンドウを表示して配置を変更し、スクリプトの関連データを表示することができます。

- ▶ **[出力]** ウィンドウの表示 / 非表示 : VuGen スクリプト・エディタの下の出力ウィンドウの表示 / 非表示を切り替えるには、**[表示]** > **[出力ウィンドウ]** を選択します。出力ウィンドウには、プロトコルに応じていくつかのタブが表示されます。最も一般的なタブは、**[再生ログ]**、**[記録ログ]**、**[生成ログ]**、および**[相関結果]**です。詳細については、147 ページ「再生ログの表示」を参照してください。
- ▶ **すべてのサムネイルの表示** : スクリプトのすべてのステップをサムネイルとして表示するには、**[表示]** > **[すべてのサムネイルを表示]** を選択します。主要なステップのみについてサムネイルを表示するには、このオプションをクリアします。詳細については、44 ページ「スクリプトのサムネイルの表示」を参照してください。
- ▶ **グリッドの表示** : データ・グリッドをサポートしているプロトコル（データベース、COM、および Microsoft .NET）について、データのグリッド表示を有効にするには、**[表示]** > **[データ グリッド]** を選択します。グリッドがスクリプトの中に表示されます。
- ▶ **ウィンドウの操作** : 開いているスクリプトをすべて閉じるには、**[ウィンドウ]** > **[すべて閉じる]** を選択します。必要に応じて、保存していないスクリプトを保存するよう VuGen から求められます。

ファイル内の検索

ファイル内の検索を使用すると、現在開いているスクリプトのすべてのファイルで、任意の文字列または式を検索できます。

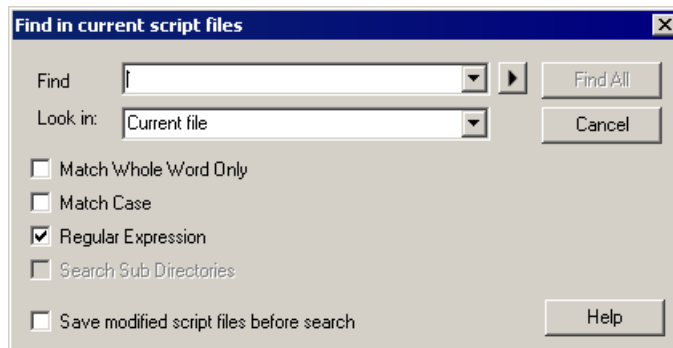
たとえば、再生ログにエラーがあるために **VuGen** がスクリプトの再生に失敗した場合、スクリプト内のすべてのファイルを同時に検索して失敗の原因と考えられる特定の値を探することができます。

Vugen によって、個別の結果ウィンドウに一致箇所がすべて表示されます。ウィンドウで任意の行をダブルクリックすると、関連するファイルを開くことができます。

注： VuGen には、通常の**検索**機能も用意されています。この機能を使用すると、一度に1つのファイル内でのみ値を検索できます。スクリプト内で値が強調表示され、F3 キーを押すと次の一致箇所に移動できます。

ファイル内の検索を実行するには、次の手順を実行します。

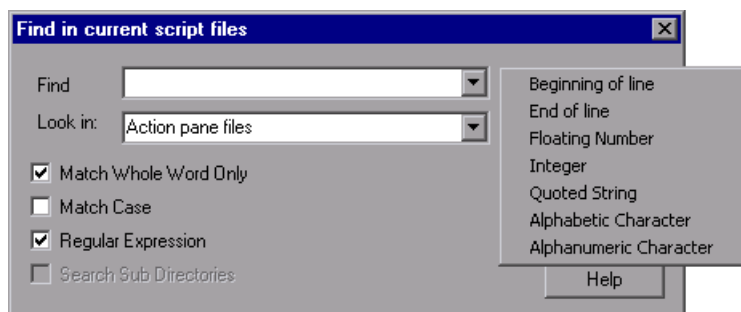
- 1 VuGen でスクリプトを開きます。
- 2 **[編集]** > **[現在のスクリプト ファイルで検索]** を選択します。[Find in current script files] ダイアログ・ボックスが開きます。



3 **[Find]** ボックスに、検索する文字列を入力します。以前に入力した検索対象の文字列がドロップダウン・リストに 10 個表示され、この中から選択できます。

また、正規表現を使って検索することもできます。正規表現とは、単一の文字または 1 組の文字から構成される検索文字列のことで、VuGen によってスクリプト内においてパターンとして解釈されます。VuGen は、このパターンに一致する値をスクリプト内で検索します。

[Regular Expression] を選択すると、**[Find]** ボックスの横にある矢印ボタンがアクティブになり、手作業で入力する代わりに 7 つの一般的な正規表現から選択できます。



4 **[Look in]** ボックスで、検索するスクリプトのディレクトリを指定します。また、VuGen によってドロップダウン・リストで提供される 2 つのオプションから選択することもできます。

- **[Current file]** : 右側の表示枠に現在表示されているファイルを検索します。
- **[Action pane files]** : 左側の表示枠に表示されているすべてのファイルを検索します。

提供されるディレクトリを使わずに、スクリプトのディレクトリを指定する場合は、**[Search Sub Directories]** オプションがアクティブになります。このオプションを選択すると、現在の Vuser フォルダのサブディレクトリも検索対象にできます。

5 **[Match Whole Word Only]** および **[Match Case]** から必要な設定を選択します。

6 **[Find All]** をクリックします。[検索結果] タブが開きます。

注： 検索を実施する前にスクリプトを保存することをお勧めします。結果に変更を加えると、元のスクリプトに戻れなくなる場合があります。ファイル内の検索では、最後に保存したバージョンのスクリプトのみが検索対象となります。スクリプトにすでに変更を加えている場合は、**[Save modified script files before search]** オプションが表示されます。

コマンド・プロンプトからの Vuser スクリプトの実行

VuGen のユーザ・インタフェースを使わずに、コマンド・プロンプトや Windows の [ファイル名を指定して実行] ダイアログ・ボックスから Vuser スクリプトをテストできます。

DOS コマンド・ラインや [ファイル名を指定して実行] ダイアログ・ボックスからスクリプトを実行するには、次の手順を実行します。

- 1 [スタート] > [プログラム] > [アクセサリ] > [コマンド プロンプト] を選択すると、[コマンド プロンプト] ウィンドウを開きます。または、[スタート] > [ファイル名を指定して実行] を選択して、[ファイル名を指定して実行] ダイアログ・ボックスを開きます。
- 2 次のとおりに入力して、**Enter** キーを押します。

```
<インストール・ディレクトリ> /bin/mdrv.exe -usr <スクリプト名> -vugen_win_0
```

スクリプト名 は、**.usr** スクリプト・ファイルのフル・パスです。たとえば、**c:%temp%mytest%mytest.usr** などになります。

mdrv プログラムによって、ユーザ・インタフェースなしでスクリプトの1つのインスタンスが実行されます。出力ファイルで実行時の情報を確認します。

UNIX コマンド・ラインからの Vuser スクリプトの実行

VuGen を使って UNIX ベースの Vuser を作成するときには、記録されたスクリプトを UNIX プラットフォームで実行できることを確認する必要があります。スクリプトを正しく実行できることを次の手順で確認します。

1 記録されたスクリプトを VuGen からテストします。

記録されたスクリプトを VuGen から実行して、Windows ベースのシステムで正しく実行できることを確認します。

2 Vuser スクリプトのファイルを UNIX ドライブにコピーします。

ファイルをローカルの UNIX ドライブに転送します。

3 verify_generator を使用して、UNIX マシン上での Vuser の設定を確かめます。

詳細については、159 ページ「verify_generator によるテスト」を参照してください。

4 UNIX コマンド・ラインからスクリプトをテストします。

Vuser スクリプトのディレクトリから、run_db_vuser シェル・スクリプトを使って、スクリプトをスタンドアロン・モードで実行します。

```
run_db_vuser.sh <スクリプト名> .usr
```

verify_generator によるテスト

検証ユーティリティは、通信パラメータと、すべての種類の Vuser の互換性について、ローカル・ホストを検査します。

ユーティリティは、Vuser 環境で次の項目を検査します。

- ▶ 最低 128 のファイル記述子を持っていること。
- ▶ .rhost ファイルの権限 (-rw-r--r--) が正しいこと。
- ▶ rsh を使ったホスト通信が可能なこと。通信が不可能な場合は、.rhosts 内のホスト名を検査します。
- ▶ M_LROOT が定義されていること。
- ▶ .cshrc が正しい M_LROOT を定義していること。
- ▶ ホーム・ディレクトリに .cshrc が存在すること。

- ▶ 現在のユーザが `.cshrc` のオーナーであること。
- ▶ LoadRunner が `$M_LROOT` にインストールされていること
- ▶ 実行可能ファイルが実行可能な権限を持っていること。
- ▶ `PATH` に `$M_LROOT/bin` と `/usr/bin` が含まれていること。
- ▶ `rstatd` デーモンが存在し、起動していること。

すべての Vuser を 1 台のホストで実行する場合は、次のように入力します。

```
verify_generator
```

`verify_generator` は、設定が正しければ「OK」を返します。「Failed」が返された場合は、正しく設定し直す方法を示します。

この検証ユーティリティの詳細については、次を入力してください。

```
verify_generator [-v]
```

コマンド・ライン・オプション : `run_db_vuser` シェル・スクリプト

`run_db_vuser` シェル・スクリプトには、以下のコマンド・ライン・オプションがあります。

--help

使用可能なオプションを表示します（このオプションの先頭には、ダッシュを 2 つ付ける必要があります）。

-cpp_only

スクリプトを対象に `cpp` のみ（プリプロセス）を実行します。

-cci_only

スクリプトを対象に `cci` のみ（プリコンパイル）を実行して、拡張子が `.ci` のファイルを作成します。`cpp` が成功しないと、`cci` は実行できません。

-driver <ドライバのパス>

指定されたドライバ・プログラムを使用します。各データベースに固有のドライバ・プログラムが、`/bin` ディレクトリにあります。たとえば、`/bin` ディレクトリにある `CtLib` のドライバは `mdrv` です。このオプションを使って、外部ドライバを指定することができます。

-exec_only

Vuser の .ci ファイルを実行します。このオプションは、有効な .ci ファイルが存在するときのみ使用できます。

-ci < ci ファイル名 >

指定された .ci ファイルを実行します。

-out < 出力パス >

指定されたディレクトリに結果を格納します。

標準設定では、`run_db_vuser.sh` は冗長モードで **cpp**, **cci**, **execute** を実行します。`run_db_vuser.sh` は < **VuGen のインストール先** > /bin ディレクトリにあるドライバを使用して、結果を Vuser スクリプト・ディレクトリ内の出力ファイルに保存します。必ず **.usr** ファイルを指定する必要があります。カレント・ディレクトリがスクリプト・ディレクトリでない場合は、**.usr** ファイルのフル・パスを指定します。

たとえば、次のコマンド・ラインは Vuser スクリプト `test1` を実行し、出力ファイルをディレクトリ `results1` に格納します。結果ディレクトリは、自動的に作成されないの、既存のディレクトリでなければなりません。

```
run_db_vuser.sh -out /u/joe/results1 test1 usr
```

スクリプトのテストへの統合

スタンドアロン・モードでのスクリプトの実行が成功し、スクリプトが正常に機能することを確認したら、そのスクリプトを自分の環境（LoadRunner シナリオ、Performance Center 負荷テスト、または Business Process Monitor プロファイル）に統合します。

テストを統合するときは、次の情報を指定します。

- ▶ 統合するスクリプト
- ▶ スクリプトを実行する Vuser
- ▶ スクリプトが実行される Load Generator
- ▶ スケジュール設定

詳細については、**HP LoadRunner Controller**、**HP Performance Center**、または**HP Business Availability Center** を参照してください。

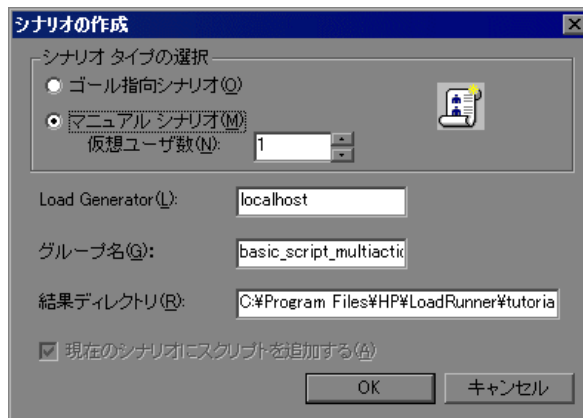
VuGen を使用しての LoadRunner シナリオの作成

注： 次の項の内容は、LoadRunner のみを対象としています。スクリプトをビジネス・プロセス・プロファイルに統合する方法の詳細については、『**HP Business Availability Center**』のドキュメントを参照してください。

通常は、LoadRunner Controller でシナリオを作成します。現在のスクリプトを使用して、基本的なシナリオを VuGen で作成することもできます。

VuGen でシナリオを作成するには、次の手順を実行します。

- 1 [ツール] > [Controller のシナリオを作成] を選択します。[シナリオの作成] ダイアログ・ボックスが開きます。



- 2 ゴール指向シナリオか、マニュアル・シナリオかを選択します。
ゴール指向シナリオを選択した場合、指定した目的に応じたシナリオが LoadRunner によって自動的に作成されます。それに対しマニュアル・シナリオでは実行する Vuser 数を指定します。
- 3 マニュアル・シナリオの場合は、スクリプトを実行する Vuser の数を入力します。

- 4 Vuser の実行に使用するマシンの名前を [Load Generator] ボックスに入力します。
- 5 マニュアル・シナリオの場合、共通の特性を持つユーザをグループに編成します。それらの Vuser のグループ名を [グループ名] ボックスで指定します。
- 6 ゴール指向シナリオの場合、[スクリプト名] を指定します。
- 7 結果を格納する場所を、[結果ディレクトリ] ボックスに入力します。
- 8 Controller ですでにシナリオを開いていて、このシナリオにスクリプトを追加する場合は、[現在のシナリオにスクリプトを追加する] チェック・ボックスを選択します。チェック・ボックスをクリアすると、LoadRunner によって、指定した数の Vuser を含んだ新しいシナリオが作成されます。
- 9 [OK] をクリックします。VuGen によって、Vuser のビューに Controller が表示されます。
- 10 共有ネットワーク・ドライブにスクリプトを保存するように Controller を設定した場合、パス変換を行う必要があるかもしれません。

詳細については、『HP LoadRunner Controller ユーザーズ・ガイド』を参照してください。

第 10 章

テスト結果の表示

Vuser スクリプトの実行結果をまとめたレポートは、スクリプトのデバッグ作業に利用できます。VuGen は Vuser スクリプトの実行中にレポートを生成し、スクリプトの実行が完了したらレポートを表示します。

本章の内容

- ▶ テスト結果の表示について (166 ページ)
- ▶ [テスト結果] ウィンドウ (167 ページ)
- ▶ 結果の表示 (170 ページ)
- ▶ 結果ステップの検索 (177 ページ)
- ▶ 結果の印刷 (178 ページ)
- ▶ テスト結果のエクスポート (181 ページ)
- ▶ Quality Center への不具合の送信 (182 ページ)
- ▶ [テスト結果] ウィンドウから Quality Center への接続 (183 ページ)
- ▶ テスト結果の表示のカスタマイズ (183 ページ)
- ▶ Web サービス・レポートの表示 (184 ページ)
- ▶ レポートへのユーザ定義情報の送信 (187 ページ)

テスト結果の表示について

コンポーネントの実行が終わったら、[テスト結果] ウィンドウに、次を含むテスト実行のあらゆる側面が表示されます。

- ▶ 高レベルの結果概要レポート（テストの成功 / 失敗のステータス）
- ▶ すべてのテスト実行に使用されたデータ
- ▶ アプリケーション・エラーの発生場所を正確に示す、ステップの展開可能なツリー
- ▶ エラーが発生したスクリプト内の具体的な場所
- ▶ 特定のステップにおけるアプリケーションの状態の静止画像
- ▶ 特定のステップまたはテスト全体のアプリケーションの状態のムービー・クリップ
- ▶ テストの各段階で、成功または失敗した各ステップとチェックポイントの詳細な説明

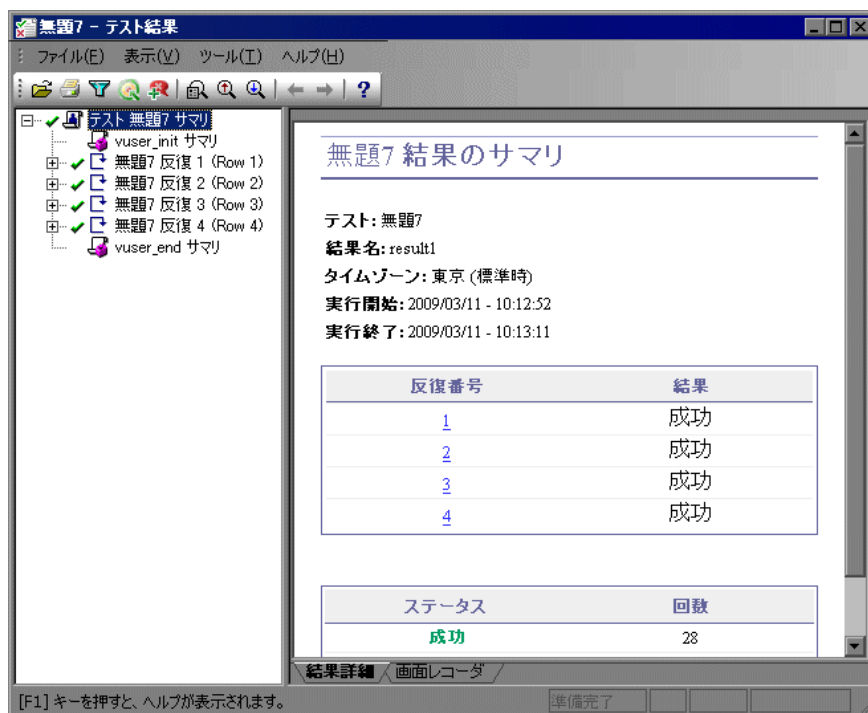
Web サービス Vuser のレポートには、操作ごとのブレイクダウン、チェックポイント結果、および HTTP トラフィックのビューなど、複数の拡張が含まれています。詳細については、184 ページ「Web サービス・レポートの表示」を参照してください。

注：[テスト結果] ウィンドウには、300 レベルまでの結果をツリー階層に表示できます。結果に 300 レベル以上の入れ子が含まれている場合は、**results.xml** ファイルを手作業で開くことによって、レポート全体を表示できます。

[テスト結果] ウィンドウ

実行セッションが終わったら、[テスト結果] ウィンドウで結果を参照します。

- ▶ 左側の表示枠には、結果をグラフィカルに示す「レポート・ツリー」が表示されます。レポート・ツリーでは、成功したステップが緑色のチェック印で、失敗したステップが赤色の「X」印で示されます。
- ▶ 右側の表示枠には「レポートの詳細」が表示されます。ここには、スクリプト実行全般についてのサマリーや、レポート・ツリーで選択された分岐についての補助的な情報が表示されます。







[テスト結果] ウィンドウには、次の主要な要素があります。

- ▶ **テスト結果タイトル・バー**：テストの名前が表示されます。
- ▶ **メニュー・バー**：利用可能なコマンドのメニューが表示されます。
- ▶ **実行結果ツールバー**：テスト結果を表示するボタンが含まれています（このツールバーを表示するには、[表示] > [テスト結果ツールバー] を選択）。詳細については、169 ページ「テスト結果ツールバー」を参照してください。
- ▶ **実行結果ツリー**：実行結果ツリーにテスト結果が視覚的に表示されます。実行結果ツリーは、[テスト結果] ウィンドウの左側の表示枠に表示されます。詳細については、168 ページ「実行結果ツリー」を参照してください。
- ▶ **[結果詳細] タブ**：実行結果ツリーで選択したノードの詳細が表示されます。[結果詳細] タブは、[テスト結果] ウィンドウの右側の表示枠に表示されません。詳細については、169 ページ「実行結果詳細」を参照してください。
- ▶ **[画面レコーダ] タブ**：テスト結果に関連付けられている記録済みムービーが表示されます。[画面レコーダ] タブは [テスト結果] ウィンドウの右側の表示枠に表示されます。この機能は、HP QuickTest Professional で作成されたスクリプトでサポートされています。
- ▶ **ステータス・バー**：現在選択されているコマンドのステータスが表示されます（このステータス・バーを表示するには、[表示] > [ステータス バー] を選択）。

[テスト結果] ウィンドウの外観は変更できます。詳細については、169 ページ「[テスト結果] ウィンドウの外観の変更」を参照してください。

実行結果ツリー

[テスト結果] ウィンドウの左側の表示枠には、**実行結果ツリー**が表示されません。テスト結果ツリーには、テスト結果が次のように図示されます。

-  ▶ 成功したステップを示します。
-  ▶ 失敗したステップを示します。
-  ▶ 警告を示します。そのステップは成功しなかったが、テストが失敗する原因にはならなかったことを意味します。
-  ▶ 予期せず失敗したステップを示します。

ツリーに表示される詳細のレベルを変更するには、実行結果ツリーの分岐を折りたたむか、展開します。

実行結果詳細

標準設定では、[テスト結果] ウィンドウが開くと、ウィンドウの右側の表示枠にある **[結果詳細]** タブに、実行したスクリプトのサマリが表示されます。

[テスト結果] ウィンドウの右側の表示枠には、**[結果詳細]** および **[画面レコーダ]** というタブがあります。実行結果ツリーの最上位ノードを選択すると、[結果詳細] タブに、テスト結果のサマリが表示されます。ツリー内の分岐またはステップを選択すると、[結果詳細] タブに、該当するステップの詳細が表示されます。また、[結果詳細] タブには、強調したステップに関するアプリケーションの静止画像を表示することもできます。

[画面レコーダ] タブには、テスト結果に関連付けられているムービーが表示されます。テスト結果に関連付けられているムービーがない場合は、[画面レコーダ] タブに、「**結果に関連付けられたムービーはありません。**」というメッセージが表示されます。

[結果詳細] タブで結果ツリーの最上位ノードを選択すると、スクリプト名、結果の名前、実行の開始日時と終了日時、反復回数、および反復に成功したか失敗したかが示されます。Web サービス・スクリプトのチェックポイントでは、得られる結果は、**[成功]** または **[失敗]** です。反復にチェックポイントが含まれていない場合は、得られる結果は **[完了]** または **[失敗]** です。

テスト結果ツールバー

テスト結果ツールバーには、テスト結果を表示するためのボタンがあります。



[テスト結果] ウィンドウの外観の変更

標準設定では、[テスト結果] ウィンドウは、Microsoft Office 2003 テーマを使用する QuickTest ウィンドウと同じルック・アンド・フィールを採用しています。必要に応じて、[テスト結果] ウィンドウのルック・アンド・フィールを変更できます。

[テスト結果] ウィンドウの外観を変更するには、次の手順を実行します。

[テスト結果] ウィンドウで [表示] > [ウィンドウのテーマ] を選択し、利用可能なテーマのリストからウィンドウの外観を選択します。たとえば、Microsoft Office 2000 や Microsoft Windows XP のテーマを適用できます。

注：[テスト結果] ウィンドウに Microsoft Windows XP のテーマを適用するには、ご利用のコンピュータが Windows XP のテーマを使用するように設定されている必要があります。

結果の表示

標準設定では、実行が終了すると、VuGen によってテスト結果が生成されて自動的に表示されます。

VuGen によって結果が生成されないようにするには、[ツール] > [一般オプション] を選択し、[表示] タブを選択して [スクリプトの実行中にレポートを作成する] オプションをクリアします。

スクリプトの実行後に結果を開くかどうかを指定するには、[ツール] > [一般オプション] を選択し、[再生] タブを選択して [再生後] セクションでビューを選択します。

[表示] オプションを設定する方法の詳細については、『Virtual User Generator ユーザーズ・ガイド』の「スタンドアロン・モードでの Vuser スクリプトの実行」を参照してください。

また、現在のスクリプトの以前の実行結果や、ほかのスクリプトの結果を表示することもできます。また、画面で結果をプレビューして標準設定の Windows プリンタで印刷したり、HTML ファイルにエクスポートしたりできます。

詳細については、下記を参照してください。

- ▶ 166 ページ「テスト結果の表示について」
- ▶ 167 ページ「[テスト結果] ウィンドウ」
- ▶ 170 ページ「結果の表示」

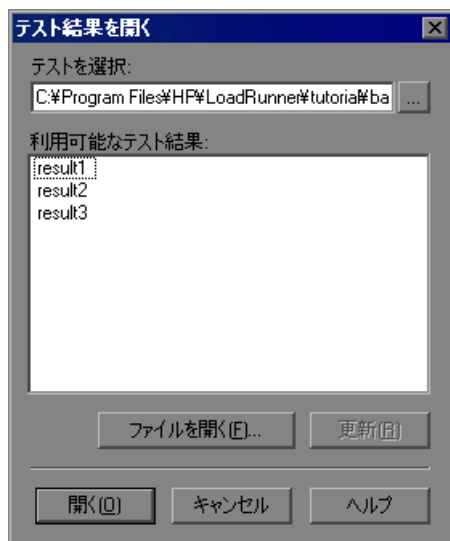
- ▶ 177 ページ「結果ステップの検索」
- ▶ 178 ページ「結果の印刷」
- ▶ 179 ページ「テスト結果のプレビュー」
- ▶ 181 ページ「テスト結果のエクスポート」

特定の実行を表示するためのテスト結果の表示

現在のスクリプトに関して保存されている結果を表示できます。また、ほかのスクリプトに関して保存されている結果も表示できます。

テスト結果の特定のセットを選択するには、次の手順を実行します。

- 1 [テスト結果] ウィンドウ内で [ファイル] > [開く] を選択します。
- 2 結果をすべて表示するには、親スクリプトのパスを指定します。



- 3 結果セットを選択し、[開く] をクリックします。

ヒント: 新しいパスを指定した後に結果リストを更新するには、[更新] をクリックします。

ファイル・システムでの結果の検索

標準設定では、ファイル・システムに保存された結果は、スクリプト・フォルダに保存されます。Quality Center に接続している場合は結果も格納され、スクリプトごと、または結果ファイルごとに、ファイル・システムで結果を検索できます。

ファイル・システム内でスクリプトごとに結果を検索するには、次の手順を実行します。

- 1 [テスト結果を開く] ダイアログ・ボックスで、スクリプトの結果ファイルが含まれているフォルダのパスを入力するか、参照ボタンをクリックして [テスト結果を開く] ダイアログ・ボックスを開きます。
- 2 結果を表示するスクリプトを見つけて強調表示し、[開く] をクリックします。
- 3 [テスト結果を開く] ダイアログ・ボックスで、表示する結果セットを強調表示し、[開く] をクリックします。[テスト結果] ウィンドウに選択した結果が表示されます。

ファイル・システム内で結果ファイルごとに結果を検索するには、次の手順を実行します。

- 1 [テスト結果を開く] ダイアログ・ボックスで [ファイルを開く] ボタンをクリックして [結果ファイルの選択] ダイアログ・ボックスを開きます。
- 2 結果ファイルが格納されているフォルダを参照します。
- 3 表示するテスト結果 (.xml) ファイルを選択して強調表示してから、[開く] をクリックします。[テスト結果] ウィンドウに選択した結果が表示されます。

注：

- ▶ 標準設定では、テストの結果ファイルは <スクリプト> ¥ <結果名> の形式で保存されます。
 - ▶ 以前のバージョンの結果ファイルは、.qtp という拡張子付きで保存されません。標準設定では、[結果ファイルの選択] ダイアログ・ボックスでは拡張子が .xml の結果ファイルのみが表示されます。[結果ファイルの選択] ダイアログ・ボックスで拡張子が .qtp の結果ファイルが表示されるようにするには、[ファイルの種類] ボックスで「Test Results (*.qtp)」を選択します。
-

Quality Center で保存した結果の検索

スクリプトが Quality Center に保存されている場合、結果も Quality Center に保存されます。テスト結果の保存場所は変更できません。

Quality Center に保存されているテスト結果を検索するには、次の手順を実行します。



- 1 [テスト結果] ウィンドウで、[ツール] > [Quality Center への接続] を選択するか、または [Quality Center への接続] ボタンをクリックして Quality Center プロジェクトに接続します。
- 2 [テスト結果を開く] ダイアログ・ボックスで、QuickTest テストの結果ファイルが含まれているフォルダのパスを入力するか、参照ボタンをクリックして [Quality Center プロジェクトからテストを開く] ダイアログ・ボックスを開きます。
- 3 [Test Type] リストで [VuGen スクリプト] を選択します。
- 4 テスト結果を表示するスクリプトを見つけて強調表示し、[OK] をクリックします。
- 5 [テスト結果を開く] ダイアログ・ボックスで、表示するテスト結果セットを強調表示し、[開く] をクリックします。[テスト結果] ウィンドウに選択したテスト結果が表示されます。

Quality Center での作業の詳細については、第11章「Quality Center を使ったスクリプト管理」を参照してください。

[テスト結果] ツリーでの作業

次の手順では、[テスト結果] ツリーのノード内で作業する方法について説明します。

- 1 ツリーに表示される詳細のレベルを選択するには、実行結果ツリーの分岐を折りたたむか、展開します。
 - ▶ 分岐を閉じるには、その分岐を選択して分岐のアイコンの左にある折りたたみ (-) 記号をクリックするか、キーボードのテンキー部分にあるマイナス・キー (-) を押します。結果ツリーでは、その分岐の詳細が消えて、折りたたみ記号が展開記号 (+) に変わります。



- ▶ 実行結果ツリーのすべての分岐を折りたたむには、[表示] > [すべて折りたたみ] を選択するか、または分岐を右クリックして [すべて折りたたみ] を選択します。
- ▶ 分岐を展開するには、その分岐を選択して分岐アイコンの左にある展開 (+) 記号をクリックするか、キーボードのテンキー部分にあるプラス・キー (+) を押します。ツリーの分岐の詳細が表示され、展開 (+) 記号が折りたたみ (-) 記号に変わります。

[テスト結果] ウィンドウを初めて開いたときには、ツリーは1レベルずつ展開します。ツリーが以前に展開されていた場合は、その前の状態に戻ります。

- ▶ 1つの分岐と、その分岐の下にあるすべての分岐を展開するには、その分岐を選択し、キーボードのテンキーにあるアスタリスク・キー (*) を押します。
- ▶ 実行結果ツリーの分岐をすべて展開するには、[表示] > [すべて展開] を選択するか、分岐を右クリックして [すべて開く] を選択するか、またはツリーの最上位レベルを選択してキーボードのテンキーにあるアスタリスク・キー (*) を押します。

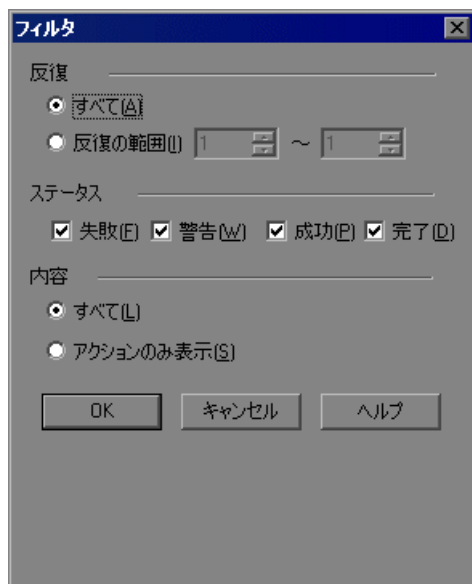
- 2 個々の反復、アクション、またはステップの結果を表示できます。実行結果ツリーでステップを選択すると、[テスト結果] ウィンドウの右側に、選択したステップの詳細が表示されます。[オプション] ダイアログ・ボックスの [実行] タブの設定に基づいて、[テスト結果] ウィンドウの右側は2つの表示枠に分割され、下部表示枠には選択したステップの静止画像（または場合によっては、ほかのデータ）が表示されます。

結果は、次のいずれかの種類になります。

- ▶ チェックポイントを含む反復、アクション、およびステップは、[テスト結果] ウィンドウの右側に「**成功**」または「**失敗**」と表示され、ツリー・ウィンドウではアイコンで見分けることができます。
- ▶ チェックポイントは含まれないが、実行が成功した反復、アクション、ステップは、[テスト結果] ウィンドウの右側に「**完了**」と表示されます。
- ▶ 成功しなかったが、スクリプトの実行停止には至らなかったステップは、[テスト結果] ウィンドウの右側に「**警告**」と表示され、 や  アイコンで示されます。



- 3 [テスト結果] ウィンドウに表示される情報を絞り込むには、[フィルタ] ボタンをクリックするか、[表示] > [フィルタ] を選択します。[フィルタ] ダイアログ・ボックスが開きます。



標準のフィルタ・オプションは、上の画像のように表示されます。[フィルタ] ダイアログ・ボックスには、次のオプションが含まれます。

[反復] 領域：

- ▶ [すべて]：すべての反復のテスト結果を表示します。
- ▶ [反復の範囲 X ~ Y]：指定した範囲のテスト反復のテスト結果を表示します。

[ステータス] 領域：

- ▶ [失敗]：失敗したステップの結果を表示します。
- ▶ [警告]：ステータスが**警告**のステップ（成功はしなかったがスクリプトが失敗する原因にはならなかったステップ）に関する結果を表示します。
- ▶ [成功]：成功したステップの結果を表示します。
- ▶ [完了]：ステータスが**完了**のステップ（ステップの実行に成功したが、成功、失敗、警告のステータスを受け取らなかったステップ）に関する結果を表示します。

[内容] 領域：

- ▶ **[すべて]**：テストのすべてのノードからすべてのステップを表示します。
- ▶ **[アクションの表示]**：テスト内のアクション・ノードを表示します（アクション・ノードの特定のステップではありません）。



4 テスト結果内で特定のステップを検索するには、**[検索]** ボタンをクリックするか、**[ツール]** > **[検索]** を選択します。詳細については、「結果ステップの検索」を参照してください。



5 実行結果ツリー内で以前選択したノード間を移動するには、**[前のノードに移動]** または **[次のノードに移動]** ボタンをクリックします。



6 ほかにの実行セッションの結果を表示するには、**[開く]** ボタンをクリックするか、**[ファイル]** > **[開く]** を選択します。詳細については、171 ページ「特定の実行を表示するためのテスト結果の表示」を参照してください。



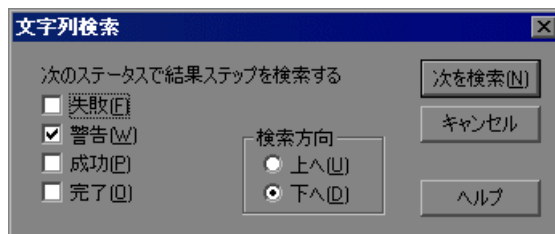
7 結果を印刷するには、**[印刷]** ボタンをクリックするか、**[ファイル]** > **[印刷]** を選択します。詳細については、178 ページ「結果の印刷」を参照してください（実行結果は、印刷する前に表示できます。詳細については、179 ページ「テスト結果のプレビュー」を参照してください）。

注：Quality Center がインストールされている場合は、検出された不具合を Quality Center プロジェクトに追加できます。詳細については、182 ページ「Quality Center への不具合の送信」を参照してください。

- 8 実行結果を HTML ファイルにエクスポートするには、**[ファイル]** > **[HTML ファイルへエクスポート]** を選択します。詳細については、181 ページ「テスト結果のエクスポート」を参照してください。
- 9 **[ファイル]** > **[終了]** を選択し、テスト結果ウィンドウを閉じます。

結果ステップの検索

[検索] ダイアログ・ボックスでは、エラーや警告など特定のステップを、テスト結果から検索することができます。たとえば、**成功**と**完了**など、ステータスの組み合わせを選択して検索できます。



次のオプションがあります。

| オプション | 説明 |
|-------|---|
| 失敗 | テスト結果で、失敗したステップを検索します。 |
| 警告 | 警告が発行されたステップを検索します。 |
| 成功 | テスト結果で、成功したステップを検索します。 |
| 完了 | 実行を完了したステップを検索します。 |
| 検索方向 | テスト結果のステップを先頭（上）に向かって検索するか、末尾（下）に向かって検索するかを指定します。 |

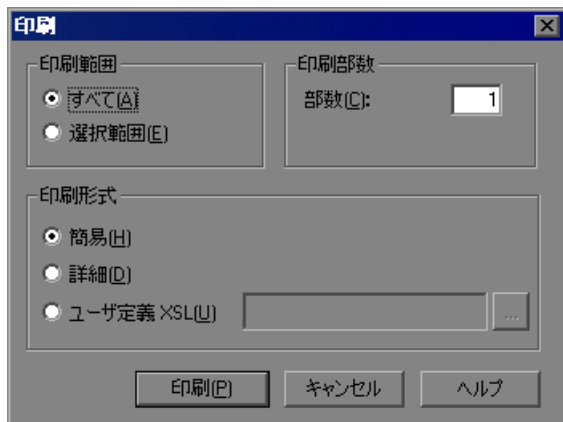
結果の印刷

[テスト結果] ウィンドウから、結果を印刷できます。印刷するレポートのタイプを選択できます。また、ユーザ定義のレポートの作成や印刷も行えます。

結果を印刷するには、次の手順を実行します。



- 1 **[印刷]** ボタンをクリックするか、**[ファイル]** > **[印刷]** を選択します。印刷ダイアログ・ボックスが開きます。



- 2 **[印刷範囲]** オプションを選択します。
 - ▶ **[すべて]** : スクリプト全体の結果を印刷します。
 - ▶ **[選択範囲]** : 実行結果ツリーで選択した分岐の結果を印刷します。
- 3 **[印刷部数]** で印刷する結果の部数を指定します。
- 4 **[印刷形式]** オプションを選択します。
 - ▶ **[簡易]** : 実行結果ツリーの各項目のサマリ行 (使用可能な場合) を印刷します。このオプションは、手順2で**[すべて]** を選択した場合にのみ使用できます。
 - ▶ **[詳細]** : 手順2での選択に基づいて、実行結果ツリーの各項目または選択した分岐に対して利用できるすべての情報を印刷します。
 - ▶ **[ユーザ定義 XSL]** : ユーザ定義の **.xsl** ファイルを参照したり選択したりできます。印刷するレポートに含める情報やその表示形式を指定するユーザ定義の **.xsl** ファイルを作成できます。詳細については、183 ページ「テスト結果の表示のカスタマイズ」を参照してください。

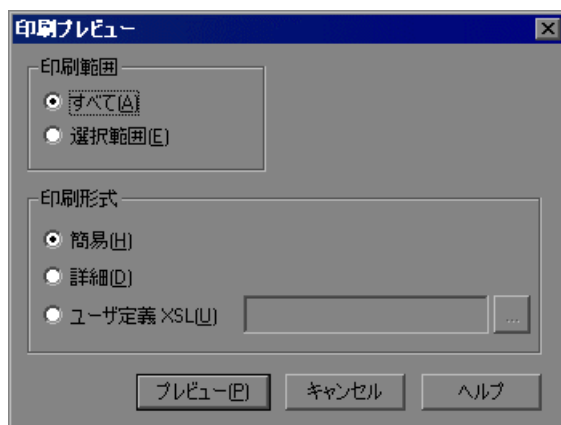
- 5 [印刷] をクリックして、選択した結果情報を標準の Windows プリンタに出力します。

テスト結果のプレビュー

結果は、印刷する前に画面に表示できます。表示する情報の形式や範囲を選択できます。また、情報をユーザ定義形式で表示できます。

結果をプレビューするには、次の手順を実行します。

- 1 [ファイル] > [印刷プレビュー] を選択します。[印刷プレビュー] ダイアログ・ボックスが開きます。



- 2 [印刷範囲] オプションを選択します。
 - ▶ [すべて] : スクリプト全体の結果をプレビューします。
 - ▶ [選択範囲] : 結果ツリーで選択した分岐の結果情報をプレビューします。
- 3 [印刷形式] オプションを選択します。
 - ▶ [簡易] : 結果ツリーの各項目のサマリ行 (使用可能な場合) をプレビューします。このオプションは、手順2で [すべて] を選択した場合にのみ使用できます。
 - ▶ [詳細] : 手順2での選択に基づいて、結果ツリーの各項目または選択した分岐に対して利用できるすべての情報をプレビューします。

- ▶ **[ユーザ定義 XSL]** : ユーザ定義の **.xsl** ファイルを参照したり選択したりできます。プレビューに含める情報やその表示形式を指定するユーザ定義の **.xsl** ファイルを作成できます。詳細については、183 ページ「テスト結果の表示のカスタマイズ」を参照してください。

4 **[プレビュー]** をクリックし、画面に結果のプレビューを表示します。



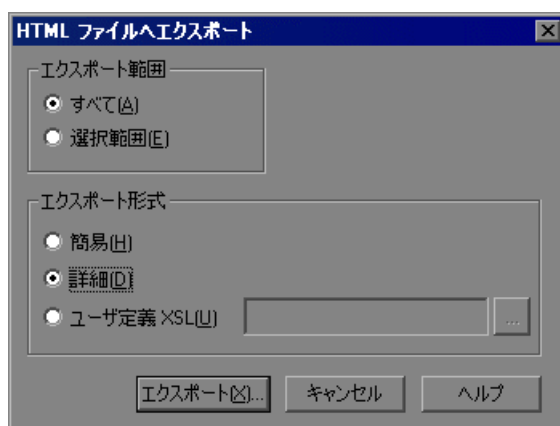
ヒント : プレビューに表示されない情報がある場合は (たとえば、チェックポイント名が長すぎてディスプレイに表示されないなど), **[印刷プレビュー]** ウィンドウの **[ページの設定]** ボタンをクリックして、ページの向きを **[縦]** から **[横]** に変更します。

テスト結果のエクスポート

結果はHTMLファイルにエクスポートできます。テスト結果ビューア環境を利用できない場合でも、これにより結果を表示できます。たとえば、結果を含むファイルを電子メールでサードパーティに送信できます。エクスポートするレポートのタイプを選択できます。また、ユーザ定義のレポートの作成やエクスポートも行えます。

結果をエクスポートするには、次の手順を実行します。

- 1 [ファイル] > [HTML ファイルへエクスポート] を選択します。[HTML ファイルへエクスポート] ダイアログ・ボックスが開きます。





- 2 [エクスポート範囲] オプションを選択します。
 - ▶ [すべて] : スクリプト全体の結果をエクスポートします。
 - ▶ [選択範囲] : 結果ツリーで選択した分岐の結果情報をエクスポートします。
- 3 [エクスポート形式] オプションを選択します。
 - ▶ [簡易] : 結果ツリーの各項目のサマリ行（使用可能な場合）をエクスポートします。このオプションは、手順2で [すべて] を選択した場合にのみ使用できます。
 - ▶ [詳細] : 手順2での選択に基づいて、結果ツリーの各項目または選択した分岐に対して利用できるすべての情報をエクスポートします。

- ▶ **[ユーザ定義 XSL]** : ユーザ定義の **.xsl** ファイルを参照したり選択したりできます。エクスポートするレポートに含める情報やその表示形式を指定するユーザ定義の **.xsl** ファイルを作成できます。詳細については、183 ページ「テスト結果の表示のカスタマイズ」を参照してください。
- 4 **[エクスポート]** をクリックします。**[名前を付けて保存]** ダイアログ・ボックスが開き、必要に応じて、標準設定の保存先フォルダを変更し、ファイルの名前を変更できます。標準設定では、ファイルは結果フォルダに保存されます。
- 5 **[保存]** をクリックして HTML ファイルを保存し、ダイアログ・ボックスを閉じます。

Quality Center への不具合の送信

結果の表示中は、検出された不具合を、[テスト結果] ウィンドウから Quality Center プロジェクトに直接送信できます。

不具合を Quality Center に直接送信するには、次の手順を実行します。

- 1 Quality Center クライアントがコンピュータにインストールされていることを確認します（ブラウザに Quality Center サーバの URL を入力し、ログイン画面が表示されることを確認します）。
- 2  **[ツール]** > **[Quality Center への接続]** を選択するか、**[Quality Center への接続]** ボタンをクリックして Quality Center プロジェクトに接続します。プロジェクトへの接続の詳細については、190 ページ「Quality Center の接続と切断」を参照してください。
- 3  **[ツール]** > **[不具合の追加]** を選択するか、**[不具合の追加]** ボタンをクリックして、指定した Quality Center プロジェクトの **[不具合の追加]** ダイアログ・ボックスを開きます。**[不具合の追加]** ダイアログ・ボックスが表示されます。
- 4 必要に応じて不具合の情報を変更できます。説明に含まれる基本的な情報は次のとおりです。

| | |
|---------------------------|--|
| Operating system : | Windows 2000 |
| Test path : | [QualityCenter] Components\YEX\ComponentWithDefect |

- 5 **[送信]** をクリックし、Quality Center プロジェクトに不具合情報を追加します。
- 6 **[閉じる]** をクリックし、**[不具合の追加]** ダイアログ・ボックスを閉じます。

[テスト結果] ウィンドウから Quality Center への接続

[テスト結果] ウィンドウから Quality Center へ手作業で不具合を送信するには、Quality Center に接続する必要があります。

接続プロセスには 2 つの段階があります。まず、ローカルまたはリモートの Quality Center サーバに接続します。このサーバによって、テスト結果と Quality Center プロジェクトの間の接続が処理されます。

次に、ログインしてアクセスするプロジェクトを選択します。プロジェクトには、テストと、テスト対象アプリケーションの実行セッション情報が格納されます。Quality Center プロジェクトはパスワードで保護されているため、ユーザ名とパスワードを指定する必要があります。

Quality Center プロジェクトへの接続の詳細については、190 ページ「Quality Center の接続と切断」を参照してください。

テスト結果の表示のカスタマイズ

各結果セットは、単一の **.xml** ファイル (**results.xml**) に保存されます。この **.xml** ファイルには、ディスプレイのそれぞれのテスト結果ノードに関する情報が保存されます。ノードの情報は、[テスト結果] ウィンドウの右上の表示枠に表示される **.htm** ファイルを動的に作成するのに使用されます。

実行結果ツリーの各ノードは **results.xml** ファイルの要素です。また、テスト結果に表示される異なるタイプの情報を表す要素もあります。**.xml** ファイルからテスト結果情報を取得し、XSL を使用して必要な情報をカスタマイズされた形式で表示できます ([テスト結果] ウィンドウから印刷する場合でも、テスト結果を独自にカスタマイズした結果ビューアで表示する場合でも、またはテスト結果を HTML ファイルにエクスポートする場合でも可能)。

XSL には、どのテスト結果情報を表示するか、それをどこでどのように表示、印刷、またはエクスポートするかを示すツールがあります。XSL エディタを使用すると、結果フォルダ内の **.css** ファイルおよび **.xsl** ファイルを変更し、レポートの外観を変更できます (フォント、色など)。

たとえば、**results.xml** ファイルで、ある要素タグにはアクション名が、もう 1 つの要素タグには実行が開始される時間についての情報が含まれているとします。XSL を使用すれば、カスタマイズしたエディタに、アクション名をページの特定の位置に緑色の太字で表示し、時間の情報はまったく表示しないように指示できます。

Web サービス・レポートの表示

この項では、Web サービス・スクリプトに特有のレポート情報について説明します。

Web サービスのテスト結果には、再生中に呼び出されたサービスの操作のリストが表示されます。操作を選択すると、レポートには、サービス、操作、ツールキット、テスト・アスペクト、および WSDL に関する情報が表示されます。

The screenshot shows the '無題8 - テスト結果' (Untitled8 - Test Results) window. The left pane displays a tree view of test steps, with 'Add_101' selected. The right pane shows the 'Web Service Call Properties' report for this step.

ステップ名: Add_101
ステップ 成功

| オブジェクト | 詳細 | 結果 | 時間 |
|---------|---------------------|----|-----------------------|
| Add_101 | Web サービス呼び出しが成功しました | 成功 | 2009/03/11 - 11:21:33 |

Web Service Call Properties

| | |
|----------------|--|
| Service Name | Calc |
| Port Name | CalcSoapPort |
| Operation Name | Add |
| WSDL location | C:/Documents and Settings/Administrator/デスクトップ/Calc.wsdl |
| Toolkit | Net |
| Testing Aspect | ポジティブ テスト |

At the bottom of the window, there are tabs for '結果詳細' (Results Details) and '画面レコーダ' (Screen Recorder), and a status bar with the text '[F1] キーを押すと、ヘルプが表示されます。' (Press [F1] key to display help.)

VuGen がスクリプトを解釈できない場合、またはほかのタイプのエラーが発生した場合は、問題を示すメッセージがレポートの右側の表示枠に表示されます。

操作のノードをさらに展開すると、レポートには、要求と応答に対する実際の SOAP トラフィックが表示されます。

The screenshot shows a software interface for test results. On the left is a tree view of test steps, including 'user_init サマリ', '無題8 回復 1 (Row 1)', 'Action サマリ', 'Add_101', 'Multiply_101', 'Divide_101', 'Subtract_101', and 'user_end サマリ'. The 'HTTP Traffic' node is selected. On the right, the 'ステップ名: HTTP Traffic' section shows 'ステップ 完了' and a table with the following data:

| オブジェクト | 詳細 | 結果 | 時間 |
|--------------|------------|----|-----------------------|
| HTTP Traffic | HTTPトラフィック | 完了 | 2009/03/11 - 11:21:33 |

Below this is a 'Web Service Call HTTP Snapshot' section. It displays the 'Request' details, including the 'HTTP Header' and 'SOAP' body. The 'HTTP Header' shows a POST request to 'MSSoapSamples30/Calc/Service/Rpc/IsapiCpp/' with 'Content-Type: text/xml; charset=utf-8'. The 'SOAP' body contains XML elements like '<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">' and '<wsa:Action>http://tempuri.org/Calc/action/Calc'.

チェックポイントの結果

[結果] ウィンドウにはチェックポイントの結果も表示されます。チェックポイントの設定については、『第2巻－プロトコル』の第7章「Web サービス再生の準備」を参照してください。

チェックポイントが失敗した場合は、レポートに失敗の原因を記載したサマリが表示されます。また、期待値と実際の結果、および引数ツリーも提供されます。

第10章・テスト結果の表示

チェックポイントの詳細を表示するには、左側の表示枠の操作で該当するステップを展開し、**チェックポイント・ノード**をクリックします。

The screenshot shows the 'WS_Check - テスト結果' application window. The left sidebar displays a tree view of test steps, with 'Checkpoint_GetDicList' selected. The main area shows the details for this step, including a table of results and a 'Checkpoints Details' section.

ステップ名: Checkpoint_GetDicList

ステップ 成功

| オブジェクト | 詳細 | 結果 | 時間 |
|-----------------------|-----------------------|----|----------------------|
| Checkpoint_GetDicList | チェックポイントのチェックが成功しました! | 成功 | 2009/03/27 - 2:16:12 |

Checkpoints Details:

| Result | XPath | Evaluation Style | Expected Values | Actual Result |
|--------|---|------------------|--|--|
| ✓ | GetDicListResult [1]/DicInfo[1]/DicID [1] | 完全一致 | "8a68bb8a-16ee-4b51-afaa-74c277bb600a" | "8a68bb8a-16ee-4b51-afaa-74c277bb600a" |
| ✓ | GetDicListResult [1]/DicInfo [1]/StartItemID[1] | 完全一致 | "NOTICE" | "NOTICE" |

結果詳細 / 画面レコーダ

[F1] キーを押すと、ヘルプが表示されます。 準備完了

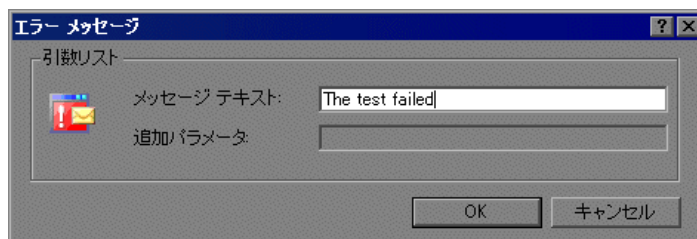
レポートへのユーザ定義情報の送信

Web サービス Vuser のために、レポートに自動的に送信される情報に加えて、メッセージ関数 `lr_output_message` または `lr_error_message` を使用すると、レポートに情報を送信できます。

次のメッセージ関数を Vuser スクリプトの中で使用できます。

エラーまたは出力メッセージ関数を挿入するには、次の手順を実行します。

- 1 **[挿入]** > **[新規ステップ]** を選択します。[ステップの追加] ダイアログ・ボックスが開きます。
- 2 **[エラー メッセージ]** または **[出力メッセージ]** ステップを選択して **[OK]** をクリックします。[エラーメッセージ] または [出力メッセージ] ダイアログ・ボックスが開きます。



- 3 **[メッセージ テキスト]** ボックスにメッセージを入力します。
- 4 **[OK]** をクリックするとメッセージが挿入され、ダイアログ・ボックスが閉じます。

`lr_error_message` 関数または `lr_output_message` 関数はスクリプトの現在の位置に挿入され、メッセージはテスト・サマリ・レポートに送信されます。

メッセージ関数の詳細については、『**Online Function Reference**』（英語版）（**[ヘルプ]** > **[関数リファレンス]**）を参照してください。

第 11 章

Quality Center を使ったスクリプト管理

VuGen と Quality Center の統合により、Quality Center を使って Vuser スクリプトを管理できます。

本章の内容

- ▶ Quality Center を使ったスクリプト管理について (189 ページ)
- ▶ Quality Center の接続と切断 (190 ページ)
- ▶ Quality Center プロジェクトからスクリプトを開く (194 ページ)
- ▶ Quality Center プロジェクトへのスクリプトの保存 (196 ページ)
- ▶ スクリプトのバージョン管理 (197 ページ)

Quality Center を使ったスクリプト管理について

VuGen は、HP の Web ベースのテスト管理ツール Quality Center と組み合わせて使えます。Quality Center は、Vuser スクリプト、シナリオ、および結果の保存と取得を効率よく行う手段を提供します。スクリプトを Quality Center プロジェクトに格納し、固有のグループに編成します。

VuGen で Quality Center プロジェクトにアクセスするには、VuGen を Quality Center がインストールされている Web サーバに接続する必要があります。ローカルとリモートのどちらの Web サーバにも接続できます。

Quality Center を使った作業の詳細については、『**Quality Center ユーザーズ・ガイド**』を参照してください。

Quality Center の接続と切断

Quality Center との間でスクリプトの保存と取得を行うには、Quality Center プロジェクトに接続する必要があります。テスト・プロセスにおいて、Quality Center プロジェクトはいつでも接続または切断できます。

Quality Center への接続

接続プロセスには次の 2 つの段階があります。最初に、ローカル Quality Center Web サーバまたはリモート Quality Center Web サーバに接続します。このサーバによって、VuGen と Quality Center プロジェクトの間の接続が処理されます。

次に、アクセスするプロジェクトを選択します。プロジェクトには、VuGen で作成したスクリプトが格納されます。

注：Quality Center プロジェクトはパスワードで保護されているため、ユーザ名とパスワードを指定する必要があります。

Quality Center に接続するには、次の手順を実行します。

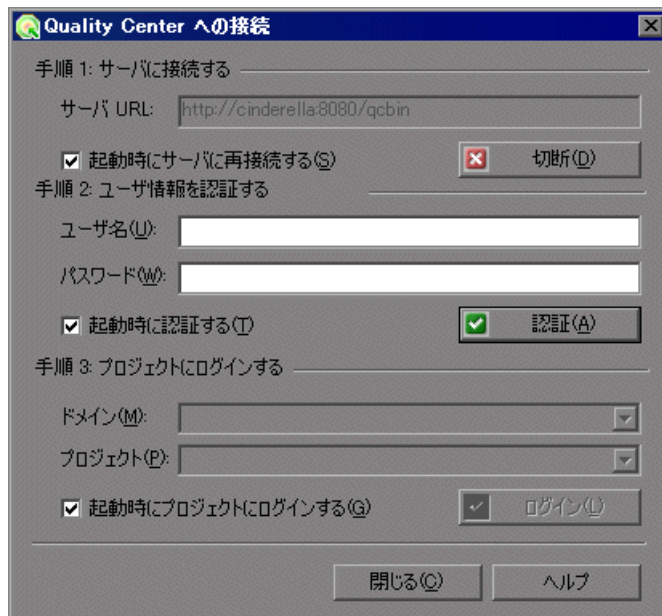
- 1 [ツール] > [Quality Center への接続] を選択します。
- 2 [Quality Center への接続 - サーバへの接続] ダイアログ・ボックスが表示されます。



- 3 [サーバの URL] ボックスに、Quality Center がインストールされている Web サーバの URL アドレスを入力します。

注：ローカル・エリア・ネットワーク（LAN）またはワイド・エリア・ネットワーク（WAN）を介してアクセス可能な Web サーバを利用できます。

- 4 次回の VuGen の起動時に、Quality Center サーバに自動的に再接続するには、**[起動時にサーバに再接続する]** チェック・ボックスを選択します。
- 5 **[接続]** をクリックします。**[Quality Center への接続]** ダイアログ・ボックスが表示されます。



サーバへの接続が確立されると、**[サーバ]** ボックスにサーバの名前が読み取り専用形式で表示されます。

- 6 ユーザ情報を認証します。
 - a **[ユーザ名]** ボックスに、Quality Center ユーザ名を入力します。
 - b **[パスワード]** ボックスに、Quality Center パスワードを入力します。
 - c **[認証]** をクリックして Quality Center サーバに対してユーザ情報を認証します。

ユーザ情報を認証したら、ユーザ情報の認証領域に、フィールドが読み取り専用形式で表示されます。[認証] ボタンは [ユーザを変更] ボタンに変わります。

別のユーザ名を使用して同じ Quality Center サーバにログインするには、[ユーザを変更] をクリックして新しいユーザ名とパスワードを入力し、再び [認証] をクリックします。

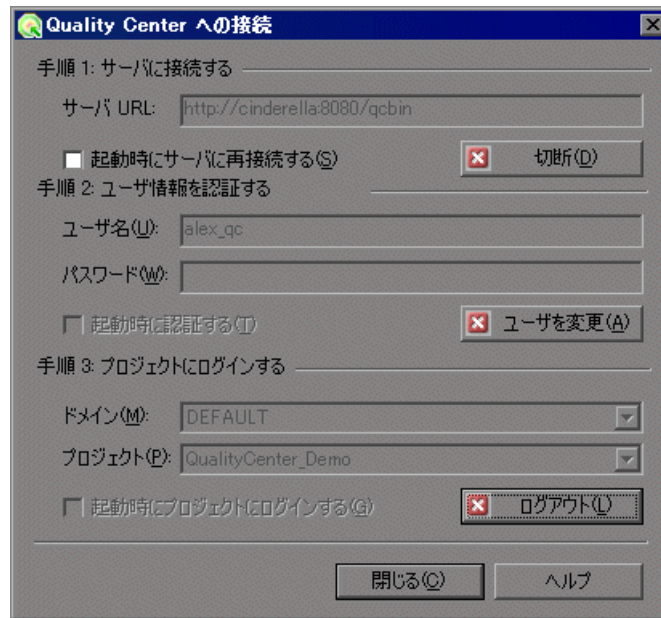
- 7 前述の [起動時にサーバに再接続する] を選択した場合、[起動時に認証する] オプションが有効になります。ユーザ情報を自動的に認証するには、次回 VuGen を起動するときに [起動時に認証する] を選択します。
- 8 プロジェクトへのログインに関する詳細を入力します。
 - a [ドメイン] ボックスで、Quality Center プロジェクトが保存されているドメインを選択します。接続する権限があるプロジェクトを含むドメインだけが表示されます（バージョン 7.5 より以前の TestDirector バージョンのプロジェクトを使って作業している場合は、[ドメイン] ボックスは関係ありません。次の手順に進んでください）。
 - b [プロジェクト] ボックスに Quality Center プロジェクト名を入力するか、またはリストからプロジェクトを選択します。接続する権限があるプロジェクトだけが表示されます。
 - c [ログイン] をクリックします。
- 9 起動時に、選択したプロジェクトにログインするには、[起動時にプロジェクトにログインする] を選択します。このオプションは、[起動時に認証する] チェック・ボックスを選択しているときのみ有効になります。
- 10 [閉じる] をクリックして、[Quality Center への接続] ダイアログ・ボックスを閉じます。

Quality Center からの切断

選択した Quality Center プロジェクトと Web サーバから VuGen を切断できます。

Quality Center から切断するには、次の手順を実行します。

- 1 [ツール] > [Quality Center への接続] を選択します。[Quality Center への接続] ダイアログ・ボックスが表示されます。



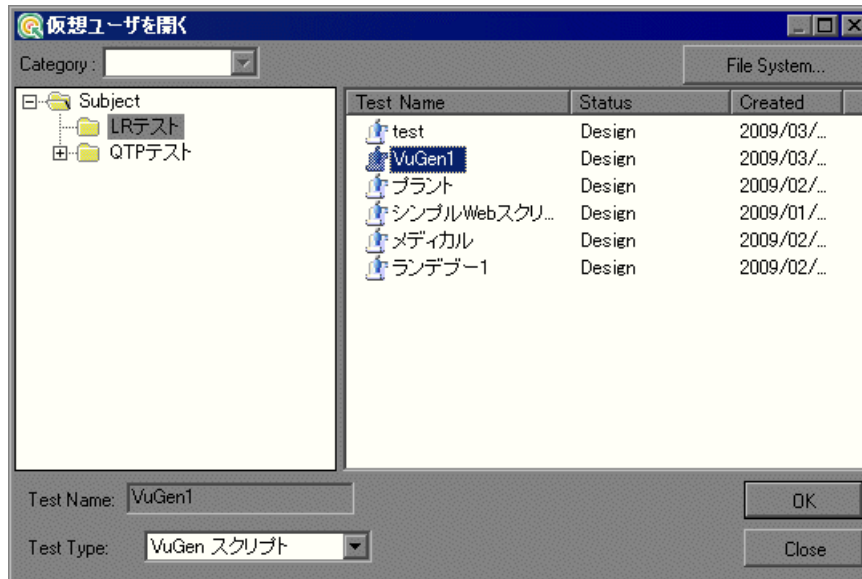
- 2 選択したプロジェクトから切断するには、[プロジェクトにログインする] で [ログアウト] をクリックします。同じサーバを使用しながら別のプロジェクトを開きたい場合は、[プロジェクト] ボックスに Quality Center プロジェクトの名前を入力するか、またはリストからプロジェクトを選択します。
- 3 Quality Center サーバから切断するには、[サーバへの接続] で [切断] をクリックします。
- 4 [閉じる] をクリックして、[Quality Center への接続] ダイアログ・ボックスを閉じます。

Quality Center プロジェクトからスクリプトを開く

Quality Center プロジェクトに接続すると、Quality Center からスクリプトを開くことができます。ファイル・システム内の格納場所ではなく、テスト計画ツリーの中でテストを探します。

Quality Center プロジェクトからスクリプトを開くには、次の手順を実行します。

- 1 Quality Center サーバに接続します（[ツール] > [Quality Center への接続]）。
- 2 [ファイル] > [開く] を選択します。Quality Center プロジェクトからテストを開くダイアログ・ボックスが表示され、テスト計画ツリーが表示されます。



注：ファイル・システムから直接スクリプトを開くには、[File System] ボタンをクリックします（Quality Center プロジェクトのスクリプトを開くダイアログ・ボックスに戻るには、[Quality Center] ボタンをクリックします）。

- 3 テスト計画ツリー内の適切なサブジェクトをクリックします。ツリーを展開して下位レベルを表示するには、閉じているフォルダをダブルクリックします。ツリーを閉じるには、開いているフォルダをダブルクリックします。

サブジェクトを選択すると、そのサブジェクトに属しているスクリプトが [テスト名] リストに表示されます。
- 4 [テスト名] リストからスクリプトを選択します。スクリプト名が読み取り専用の [テスト名] ボックスに表示されます。
- 5 **[OK]** をクリックしてスクリプトを開きます。VuGen によってスクリプトがロードされ、タイトル・バーに名前が表示されます。

[ファイル] メニューの最近使用したファイルのリストからスクリプトを開くこともできます。Quality Center プロジェクトから最近のスクリプトを選択したときに、そのプロジェクトに接続されていない場合は、[Quality Center への接続] ダイアログ・ボックスが表示されます。

最近使用したファイルのリストからテストを開く

[ファイル] メニューの最近使用したファイルのリストからスクリプトを開くことができます。Quality Center プロジェクト内にあるファイルを選択したけれど、Quality Center またはそのファイルの正しいプロジェクトに接続していない場合は、VuGen によって Quality Center に接続するように求められます。

そのスクリプトを使って作業を続けるにはプロジェクトにログインします。

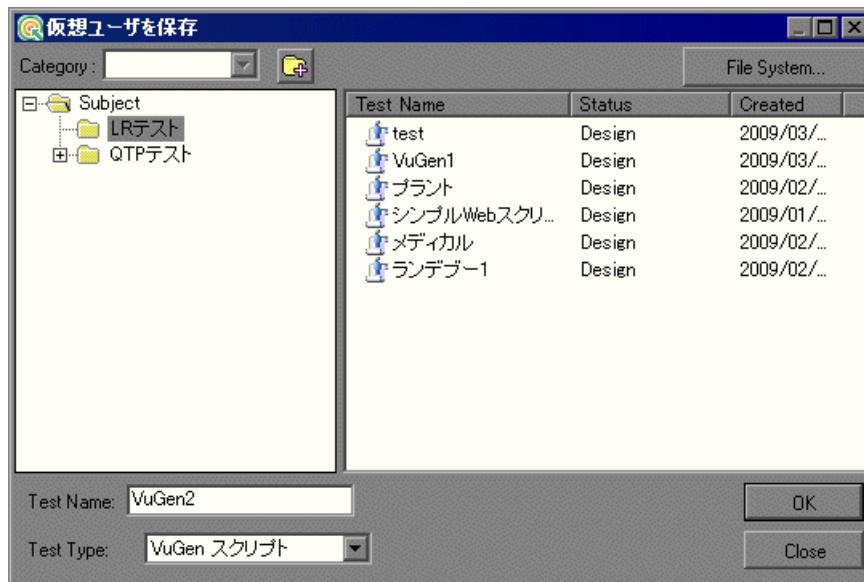
また、前回異なるユーザ名を使用して編集したスクリプトを開くことを選択した場合も、[Quality Center プロジェクトへの接続] ダイアログ・ボックスが開きます。表示されているユーザ名を使ってログインすることも、**[キャンセル]** をクリックして現在のユーザ名でログインすることもできます。

Quality Center プロジェクトへのスクリプトの保存

VuGen が Quality Center プロジェクトに接続されているときには、VuGen で新しいスクリプトを作成して、プロジェクトに直接保存することができます。スクリプトを保存するには、わかりやすい名前を付けて、テスト計画ツリーの中の適切なサブジェクトに関連付けます。これにより、各サブジェクトに対して作成されたスクリプトを把握し、テスト計画とテスト作成の進捗を表示できるようになります。

スクリプトを Quality Center プロジェクトに保存するには、次の手順を実行します。

- 1 Quality Center サーバに接続します（[ツール] > [Quality Center への接続]）。
- 2 [ファイル] > [名前を付けて保存] を選択します。Quality Center プロジェクトにテストを保存するダイアログ・ボックスが表示されます。



ファイル・システムに直接テストを保存するには、[File System] ボタンをクリックします。[テストを保存] ダイアログ・ボックスが開きます（[テストを保存] ダイアログ・ボックスからは、[Quality Center] ボタンをクリックすれば [Quality Center プロジェクトにテストを保存] ダイアログ・ボックスに戻ることができます）。

- 3 テスト計画ツリー内の適切なサブジェクトを選択します。ツリーを展開してサブレベルを表示するには、閉じているフォルダをダブルクリックします。ツリーを折りたたむには、開いているフォルダをダブルクリックします。
 - 4 [テスト名] ボックスに、スクリプトの名前を入力します。スクリプトを簡単に識別できるようなわかりやすい名前を使用します。
 - 5 [OK] をクリックしてスクリプトを保存し、ダイアログ・ボックスを閉じます。
- 次に Quality Center を起動したときに、Quality Center のテスト計画ツリーに新しいスクリプトが表示されます。

スクリプトのバージョン管理

バージョン・コントロール・サポート付きの Quality Center プロジェクトに接続すると、古いバージョンを維持しながら、自動化されたスクリプトの更新や変更ができます。これにより、各スクリプトに加えた変更の追跡、スクリプトのバージョン間の変更の確認、スクリプトの旧バージョンの復元が容易になります。

バージョン・コントロール・サポート付きのプロジェクトにスクリプトを保存すると、バージョン・コントロール・データベースにスクリプトが追加されます。バージョン・コントロール・データベースにスクリプトをチェックイン、チェックアウトすることで、スクリプトのバージョンが管理されます。

最新バージョンのスクリプトは、Quality Center のリポジトリにあり、Quality Center によってすべてのテストの実行に使用されます。

注： [ファイル] メニューの [Quality Center バージョン管理] オプションは、バージョン・コントロール・サポートの付いた Quality Center プロジェクトのデータベースに接続している場合にだけ使用できます。

本項の内容

- ▶ 「バージョン・コントロール・データベースへのスクリプトの追加」
- ▶ 「バージョン・コントロール・データベースからのスクリプトのチェックアウト」

- ▶ 「バージョン・コントロール・データベースへのスクリプトのチェックイン」
- ▶ 「[バージョンの履歴] ダイアログ・ボックスの使用方法」
- ▶ 「チェックアウト操作の取り消し」

バージョン・コントロール・データベースへのスクリプトの追加

[名前を付けて保存] を使ってバージョン・コントロール・サポート付きの Quality Center プロジェクトに新規のスクリプトを保存すると、VuGen によって自動的にスクリプトがプロジェクトに保存され、バージョン・コントロール・データベースにバージョン番号 1.1.1 でチェックインされ、その後作業を続けられるようにチェックアウトされます。

ステータス・バーにはそのつど各操作が表示されます。ただし、既存のスクリプトに変更を保存しても、そのスクリプトをチェックインすることにはなりません。スクリプトを保存して閉じて、チェックインを選択するまではスクリプトはチェックアウトしたままです。詳細については、198 ページ「バージョン・コントロール・データベースからのスクリプトのチェックアウト」を参照してください。

バージョン・コントロール・データベースからのスクリプトのチェックアウト

[ファイル] > [開く] を選択して、現在バージョン・コントロール・データベースにチェックインしているスクリプトを開くと、スクリプトは読み取り専用モードで開きます。

注：Quality Center プロジェクトからテストを開くダイアログ・ボックスには、プロジェクトの各スクリプトのバージョン・コントロール・ステータスを示すアイコンが表示されます。詳細については、194 ページ「Quality Center プロジェクトからスクリプトを開く」を参照してください。

チェックインしたスクリプトを確認できます。また、スクリプトを実行して結果を表示することもできます。

スクリプトを変更するには、スクリプトをチェックアウトする必要があります。スクリプトをチェックアウトすると、スクリプトは Quality Center によってユーザの一意のチェックアウト・ディレクトリ（初めてスクリプトをチェックアウトするときに自動的に作成される）にコピーされ、プロジェクト・データベースにロックされます。これにより、スクリプトに加えた変更を Quality Center プロジェクトのほかのユーザに上書きされないようにします。ただし、この場合でも、ほかのユーザはデータベースにチェックインされているスクリプトの最新バージョンを実行できます。

スクリプトを保存して閉じることはできますが、Quality Center データベースに戻されるまでスクリプトはロックされています。スクリプトは、スクリプトをチェックインするか、チェックアウトの操作を取り消すことによって解放されます。スクリプトのチェックインの詳細については、200 ページ「バージョン・コントロール・データベースへのスクリプトのチェックイン」を参照してください。チェックアウトの取り消しの詳細については、205 ページ「チェックアウト操作の取り消し」を参照してください。

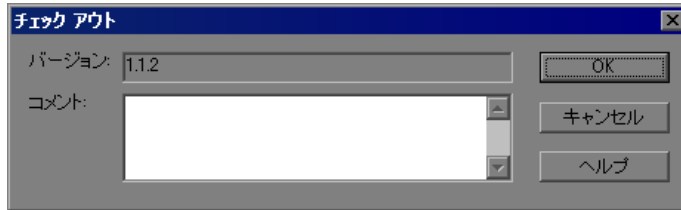
標準では、チェックアウト・オプションはスクリプトの最新バージョンにアクセスします。古いバージョンのスクリプトもチェックアウトできます。詳細については、202 ページ「[バージョンの履歴] ダイアログ・ボックスの使用方法」を参照してください。

最新バージョンのスクリプトをチェックアウトするには、次の手順を実行します。

- 1 チェックアウトするスクリプトを開きます。詳細については、194 ページ「Quality Center プロジェクトからスクリプトを開く」を参照してください。

注：開いたスクリプトが現在チェックインされていることを確認します。チェックアウトしているスクリプトを開くと、[**チェックアウト**] オプションは無効になります。ほかのユーザがチェックアウトしているスクリプトを開くと、[**バージョンの履歴**] オプションを除くすべての [**Quality Center バージョン管理**] オプションは無効になります。

- 2 [ファイル] > [Quality Center バージョン管理] > [チェックアウト] を選択します。[チェックアウト] ダイアログ・ボックスが開き、チェックアウトされるスクリプトのバージョンが表示されます。



- 3 [コメント] ボックスに変更の詳細を入力します。
- 4 [OK] をクリックします。読み取り専用のスクリプトが閉じて、書き込み可能スクリプトとして再び自動的に開きます。

バージョン・コントロール・データベースへのスクリプトのチェックイン

あるスクリプトがチェックアウトされていても、Quality Center ユーザは以前にチェックインされたバージョンを実行できます。たとえば、スクリプトのバージョン 1.2.3 をチェックアウトし、いくつかの変更を加えてスクリプトを保存するとします。バージョン 1.2.4 (または別の番号を割り当てる) としてスクリプトをバージョン・コントロール・データベースに再びチェックインするまで、Quality Center ユーザはバージョン 1.2.3 の実行を続行できます。

スクリプトの変更が終了し、Quality Center ユーザが新しいバージョンを使用できるようになったら、バージョン・コントロール・データベースにスクリプトをチェックインします。

注：Quality Center データベースにチェックインしない場合は、チェックアウト操作を取り消すことができます。詳細については、205 ページ「チェックアウト操作の取り消し」を参照してください。

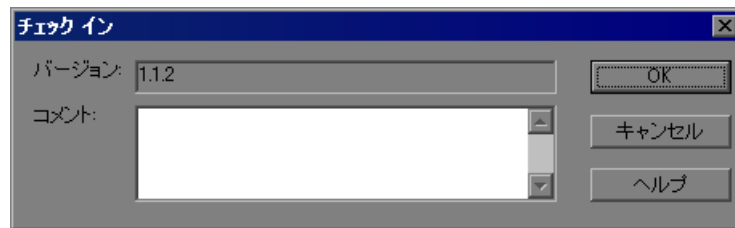
スクリプトをバージョン・コントロール・データベースに再びチェックインすると、Quality Center はチェックアウト・ディレクトリからスクリプトのコピーを削除し、Quality Center プロジェクトのほかのユーザがそのスクリプトのバージョンを利用できるようにデータベースのロックを解除します。

現在開いているスクリプトをチェックインするには、次の手順を実行します。

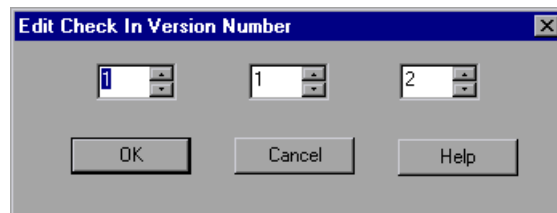
- 1 現在開いているスクリプトをチェックアウトしていることを確認します。詳細については、202 ページ「スクリプトのバージョン情報の表示」を参照してください。

注：開いているスクリプトが現在チェックインされていると、**[チェック イン]** オプションは無効になります。ほかのユーザがチェックアウトしているスクリプトを開くと、**[バージョンの履歴]** オプションを除くすべての **[Quality Center バージョン管理]** オプションは無効になります。

- 2 **[ファイル]** > **[Quality Center バージョン管理]** > **[チェック イン]** を選択します。**[チェック イン]** ダイアログ・ボックスが表示されます。



- 3 標準設定の新しいバージョン番号を受け入れて手順 7 に進むか、参照ボタンをクリックしてユーザ定義のバージョン番号を指定します。参照ボタンをクリックすると、**[Edit Check In Version Number]** ダイアログ・ボックスが開きます。



- 4 バージョン番号を、手作業またはバージョン番号の各要素の横にある上向き矢印と下向き矢印を使って修正します。最初の要素に入力できる数字は 1 ～ 900 です。第 2、第 3 の要素に入力できる数字は 1 ～ 999 です。バージョン・コントロール・データベースにある対象のスクリプトの最新バージョン番号よりも低いバージョン番号は入力できません。

- 5 [OK] をクリックしてバージョン番号を保存し、[Edit Check In Version Number] ダイアログ・ボックスを閉じます。
- 6 スクリプトのチェックアウト時に変更の詳細を入力した場合、その詳細は [コメント] ボックスに表示されます。コメントの入力や修正はこのボックスで行えます。
- 7 [OK] をクリックし、スクリプトをチェックインします。スクリプトが閉じて、読み取り専用のファイルとして再び自動的に開きます。

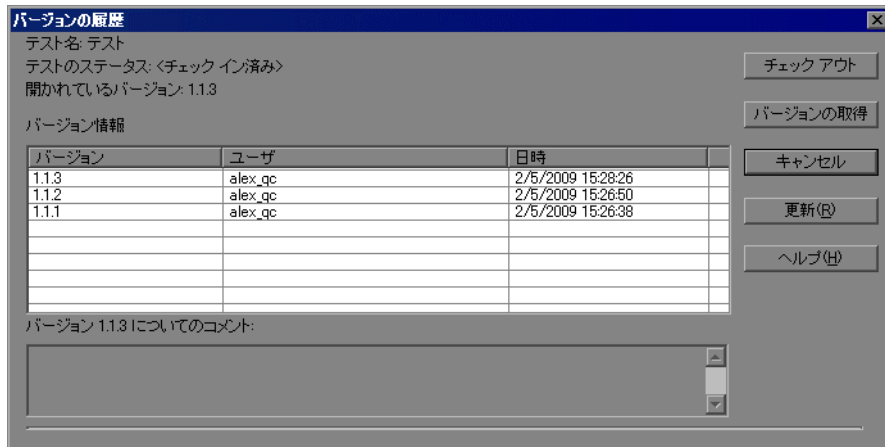
[バージョンの履歴] ダイアログ・ボックスの使用法

[バージョンの履歴] ダイアログ・ボックスを使って、現在開いているスクリプトに関するバージョン情報の表示や、古いバージョンのスクリプトの表示および取得が行えます。

スクリプトのバージョン情報の表示

Quality Center バージョン・コントロール・データベースに格納されている任意の開いているスクリプトのバージョン情報を、現在のステータスに関係なく表示できます。

スクリプトの [バージョンの履歴] ダイアログ・ボックスを開くには、スクリプトを開き [ファイル] > [Quality Center バージョン管理] > [バージョンの履歴] を選択します。



[バージョンの履歴] ダイアログ・ボックスには次の情報が表示されます。

- ▶ **[テスト名]** : 現在開いているスクリプトの名前。
- ▶ **[テストのステータス]** : スクリプトのステータス。次のいずれかになります。
 - ▶ **[チェックイン]** : スクリプトは、現在バージョン・コントロール・データベースにチェックインされています。現在読み取り専用形式で開いています。スクリプトをチェックアウトして編集できます。
 - ▶ **[チェックアウト]** : スクリプトをチェックアウトしています。現在読み取り / 書き込み可能な形式で開いています。
 - ▶ **[<ほかのユーザ>によるチェックアウト]** : スクリプトは現在別のユーザによってチェックアウトされています。現在読み取り専用形式で開いています。特定のユーザがスクリプトをチェックインするまでこのスクリプトをチェックアウトしたり編集したりすることはできません。
- ▶ **[開かれているバージョン]** : 現在、お使いの QuickTest コンピュータで開いているスクリプトのバージョン。
- ▶ **[バージョン詳細]** : スクリプトのバージョンの詳細。
 - ▶ **[バージョン]** : スクリプトのバージョンの一覧。
 - ▶ **[ユーザ]** : 各バージョンをチェックインしたユーザ。
 - ▶ **[日時]** : 各バージョンがチェックインされた日時。
- ▶ **[バージョンのコメント]** : 選択したスクリプトのバージョンがチェックインされたときに入力されたコメント。

以前のスクリプトのバージョンを使った作業

古いバージョンのスクリプトは、読み取り専用モードで表示するか、チェックアウトして最新バージョンとしてチェックインできます。

古いバージョンのスクリプトを表示するには、次の手順を実行します。

- 1 Quality Center スクリプトを開きます。最新バージョンのスクリプトが開きません。詳細については、194 ページ「Quality Center プロジェクトからスクリプトを開く」を参照してください。
- 2 **[ファイル]** > **[Quality Center バージョン管理]** > **[バージョンの履歴]** を選択します。[バージョンの履歴] ダイアログ・ボックスが表示されます。

- 3 **バージョン詳細リスト**から表示するバージョンを選択します。
- 4 **[バージョンの取得]** ボタンをクリックします。QuickTest から、スクリプトがチェックアウトされていないため読み取り専用モードで開くというメッセージが表示されます。
- 5 **[OK]** をクリックし、QuickTest メッセージを閉じます。選択したバージョンが読み取り専用モードで開きます。

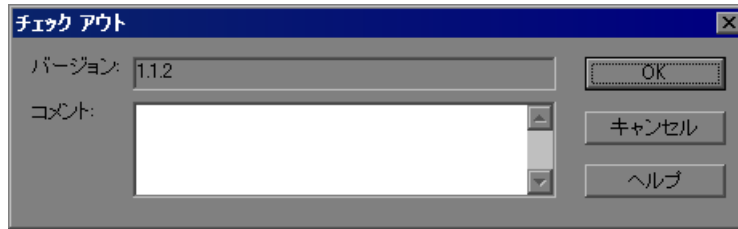
ヒント：QuickTest で現在開かれているバージョン番号を確認するには、[バージョンの履歴] ダイアログ・ボックスの **[現在開かれているバージョン]** の値を参照します。

[バージョンの取得] オプションを使用して読み取り専用モードで古いバージョンを開いたら、**[ファイル] > [Quality Center バージョン管理] > [チェックアウト]** を選択して、開いているスクリプトをチェックアウトできます。これは、[バージョンの履歴] ダイアログ・ボックスで **[チェックアウト]** ボタンを使用するのと同等の機能です。

古いバージョンのスクリプトをチェックアウトするには、次の手順を実行します。

- 1 Quality Center スクリプトを開きます。最新バージョンのスクリプトが開きます。詳細については、194 ページ「Quality Center プロジェクトからスクリプトを開く」を参照してください。
- 2 **[ファイル] > [Quality Center バージョン管理] > [バージョンの履歴]** を選択します。[バージョンの履歴] ダイアログ・ボックスが表示されます。
- 3 **バージョン詳細リスト**から表示するバージョンを選択します。
- 4 **[チェックアウト]** ボタンをクリックします。確認メッセージが開きます。

- 古いバージョンのスクリプトをチェックアウトすることを確認します。
[チェックアウト] ダイアログ・ボックスが開き、チェックアウトされるバージョンが表示されます。



- [コメント] ボックスに変更の詳細を入力します。
- [OK] をクリックします。開いていたスクリプトが閉じ、選択したバージョンが書き込み可能スクリプトとして開きます。
- 必要に応じてスクリプトを表示または編集します。
- 新規の最新バージョンとしてスクリプトを Quality Center データベースにチェックインするには、[ファイル] > [Quality Center バージョン管理] > [チェックイン] を選択します。変更したスクリプトを Quality Center にアップロードしない場合は、[ファイル] > [Quality Center バージョン管理] > [チェックアウト作業を元に戻す] を選択します。

スクリプトのチェックインの詳細については、200 ページ「バージョン・コントロール・データベースへのスクリプトのチェックイン」を参照してください。チェックアウトの取り消しの詳細については、205 ページ「チェックアウト操作の取り消し」を参照してください。

チェックアウト操作の取り消し

スクリプトをチェックアウトした後に、変更したスクリプトを Quality Center にアップロードしないと決定した場合は、ほかの Quality Center ユーザがチェックアウトできるように、チェックアウト操作を取り消す必要があります。

チェックアウト操作を取り消すには、次の手順を実行します。

- チェックアウトしたスクリプトをまだ開いていない場合は開きます。
- [ファイル] > [Quality Center バージョン管理] > [チェックアウト作業を元に戻す] を選択します。

- 3 **〔はい〕** をクリックし、チェックアウト操作の取り消しを確定します。チェックアウト操作が取り消されます。チェックアウトしたスクリプトが閉じ、以前チェックインしたバージョンが読み取り専用モードで再び開きます。

第 12 章

HP Performance Center によるスクリプトの管理

VuGen を HP Performance Center と統合することにより、Performance Center サーバへのスクリプトのアップロードおよび Performance Center サーバからのスクリプトのダウンロードが行えます。

本章の内容

- ▶ HP Performance Center によるスクリプトの管理について (208 ページ)
- ▶ Performance Center での Vuser スクリプトの表示 (208 ページ)
- ▶ VuGen から Performance Center への接続 (209 ページ)
- ▶ Vuser スクリプトのアップロード (210 ページ)
- ▶ Vuser スクリプトのダウンロード (213 ページ)
- ▶ ダウンロードしたスクリプトでの作業 (216 ページ)

HP Performance Center によるスクリプトの管理について

VuGen は、HP の Web 対応のグローバル負荷テスト・ツールである Performance Center と統合でき、異なる場所からシステムをテストできるようになっています。

VuGen のユーザ・インタフェースを使って、Performance Center サーバとの間でスクリプトのアップロードとダウンロードを実行できます。スクリプトを Performance Center にアップロードすることにより、スクリプトを Vuser クリプトのリストに追加して、テストで使用できるようになります。また、スクリプトをダウンロードして、編集したりローカルに保存したりすることもできます。

VuGen から Performance Center にアクセスしてアップロードやダウンロードを行うには、Performance Center がインストールされているサーバに接続する必要があります。

Performance Center を使った作業の詳細については、『**HP Performance Center ユーザーズ・ガイド**』を参照してください。

Performance Center での Vuser スクリプトの表示

Performance Center で Vuser スクリプトを実行するには、スクリプトが Performance Center サーバの Vuser スクリプト・リストに表示されている必要があります。Vuser スクリプト・リストには、プロジェクトで利用できるスクリプトがすべて表示されます。

Vuser スクリプト・リストを開くには、Performance Center のユーザ・サイト内の **[Project]** メニューから **[Vuser Scripts]** を選択します。

スクリプトをリストに追加するには、Performance Center インタフェースを介して、または直接 VuGen からサーバにアップロードします。

Performance Center を介してスクリプトをアップロードする方法の詳細については、『**HP Performance Center ユーザーズ・ガイド**』を参照してください。

VuGen からアップロードする方法の詳細については、210 ページ「Vuser スクリプトのアップロード」を参照してください。

VuGen から Performance Center への接続

VuGen と Performance Center が連携して動作することにより、Performance Center との間で Vuser スクリプトのアップロードとダウンロードを行う効率的な手段が提供されます。VuGen で Performance Center プロジェクトにアクセスするには、最初に VuGen を Performance Center がインストールされている Web サーバに接続する必要があります。接続後、Vuser スクリプトをアップロードまたはダウンロードできます。

VuGen を Performance Center に接続するには、次の手順を実行します。

- 1 VuGen で、[ツール] > [Performance Center への接続] を選択します。
[Performance Center 接続設定] ダイアログ・ボックスが開きます。



- 2 [URL] ボックスに、Performance Center がインストールされている Web サーバの URL アドレスを入力します。URL アドレスは、次の形式で指定します。
`http:// <サーバ名> /loadtest`
- 3 ユーザ名とパスワードを入力します。詳しくは Performance Center の管理者に問い合わせてください。
- 4 ログイン・プロセスを自動化するには、[ユーザ名とパスワードを記憶する] を選択します。指定したユーザ名とパスワードはレジストリに保存され、ダイアログ・ボックスを開くたびに表示されます。
- 5 VuGen の起動時に Performance Center サーバに自動的に接続するには、[起動時に自動的に接続する] を選択します。VuGen は、表示されたログイン情報を使って Performance Center への接続を試行します。

- 6 **[接続]** をクリックします。**[Performance Center への接続]** ダイアログ・ボックスに、接続ステータスが表示されます。

接続が確立されると、すべてのフィールドが読み取り専用で表示されます。

注：接続に失敗した場合は、ダイアログ・ボックスに接続に失敗した理由が表示されます。

Performance Center と Quality Center に同時に接続することはできません。

VuGen が Performance Center に接続されていない場合は、Vuser スクリプトをローカルのファイル・システムに保存できます。後で VuGen が Performance Center に接続されたら、VuGen でスクリプトを開き、それを以下に説明する方法で Performance Center にアップロードします。

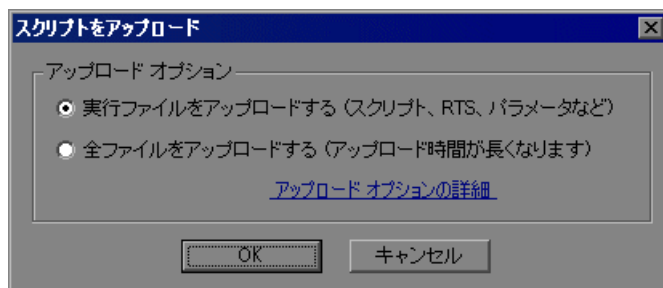
Vuser スクリプトのアップロード

ファイル・システムからサーバに Vuser スクリプトを転送することを、**アップロード**といいます。スクリプトをアップロードすると、Performance Center によって Vuser スクリプトのリストに表示されます。

VuGen から Performance Center サーバにスクリプトをアップロードする前に、資格情報を使ってサーバに接続する必要があります。接続の詳細については、209 ページ「VuGen から Performance Center への接続」を参照してください。

アップロード・オプション

VuGen では、アップロードの範囲を選択できます。要件に応じて、部分的なアップロードと完全なアップロードのいずれかを実行できます。次のアップロード・オプションを使用できます。



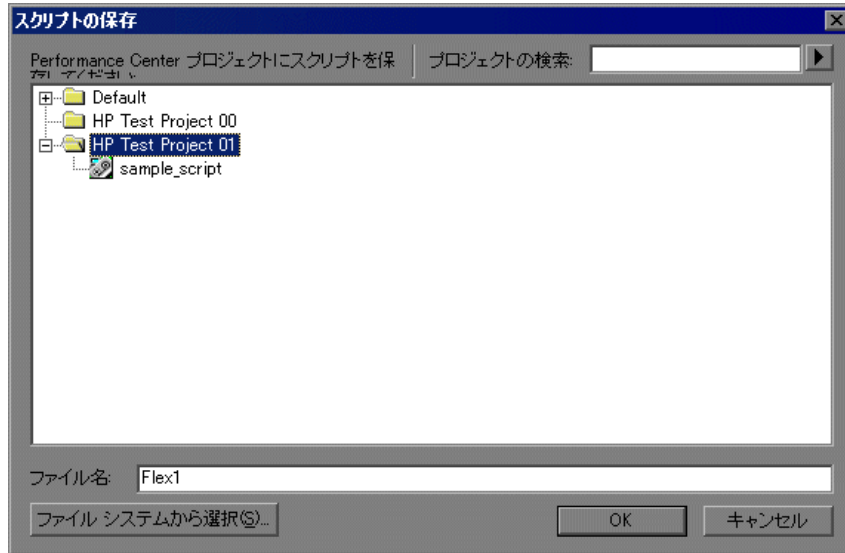
- ▶ **[実行ファイルをアップロードする]** : すべての主要なスクリプト・ファイル (usr, c, cfg, および xml ファイル) をサーバから削除します。データ・ファイルまたは古い記録データは削除されません。次に、VuGen はスクリプト・ファイル、実行環境設定、パラメータ・ファイルをアップロードします。
- ▶ **[全ファイルをアップロードする]** : VuGen は最初にすべてのスクリプト・ファイルおよびデータ・ファイルをサーバから削除します。現在のスクリプト・ファイルとデータ・ファイル (記録データや再生の結果ディレクトリを含む) がアップロードされます。

実行時ファイルのみをアップロードすると、VuGen はすべての記録データと再生結果をアップロードするのではなくスクリプト・ファイルだけをアップロードするため、時間が短縮されます。

注 : 実行時ファイルがすでにダウンロードされている場合は、標準設定ではダウンロードされたファイルだけがアップロードされます。再生後に生成されたスナップショットなど、新しく作成したファイルをアップロードする場合は、**[全ファイルをアップロードする]** を指定する必要があります。

VuGen から Performance Center にスクリプトをアップロードするには、次の手順を実行します。

- 1 Performance Center に接続されていることを確認します。[ツール] > [Performance Center への接続] を選択します。
- 2 VuGen で [ファイル] > [名前を付けて保存] を選択します。[スクリプトの保存] ダイアログ・ボックスが開きます。



- 3 [プロジェクトの検索] ボックスでプロジェクトの名前または名前の一部を入力して必要なプロジェクトを指定します。検索を開始するには、ボックスの右側にある矢印をクリックします。次の一致箇所に移動するには、矢印をもう一度クリックします。
- 4 スクリプトの保存先となるプロジェクトを選択します。[ファイル名] ボックスに、スクリプトの名前を入力します。

注：ファイル名は、英文字、数字、アンダーバーのみで構成し、250 文字以内で入力する必要があります。

- 5 [OK] をクリックします。[スクリプトをアップロード] ダイアログ・ボックスが表示されます。[実行ファイルをアップロードする] または [全ファイルをアップロードする] からアップロード・オプションを選択します。
- 6 [OK] をクリックすると、ファイルが Performance Center サーバにアップロードされます。

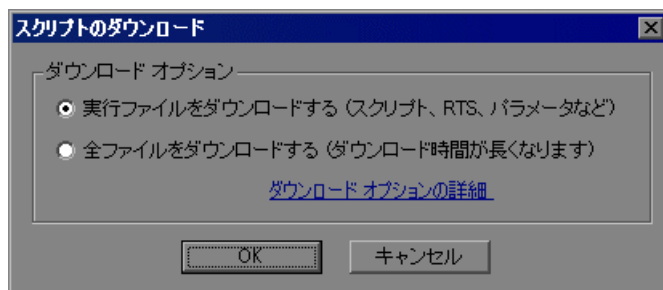
Vuser スクリプトのダウンロード

VuGen と Performance Center が連携して動作することにより、Performance Center から Vuser スクリプトをダウンロードし、VuGen で自動的に開いて編集するための効率的な手段が提供されます。スクリプト実行時のファイルのみをダウンロードするか、記録データや再生結果を含む完全なファイルをダウンロードするかを指定できます。

VuGen で Performance Center プロジェクトにアクセスするには、最初に VuGen を Performance Center がインストールされている Web サーバに接続する必要があります。接続後、ダウンロードするスクリプト・ファイルを選択できます。[Performance Center 接続設定] ダイアログ・ボックスから Performance Center に接続します。詳細については、209 ページ「VuGen から Performance Center への接続」を参照してください。

ダウンロード・オプション

要件に応じて、部分的なダウンロードと完全なダウンロードのいずれかを実行できます。次のダウンロード・オプションを使用できます。



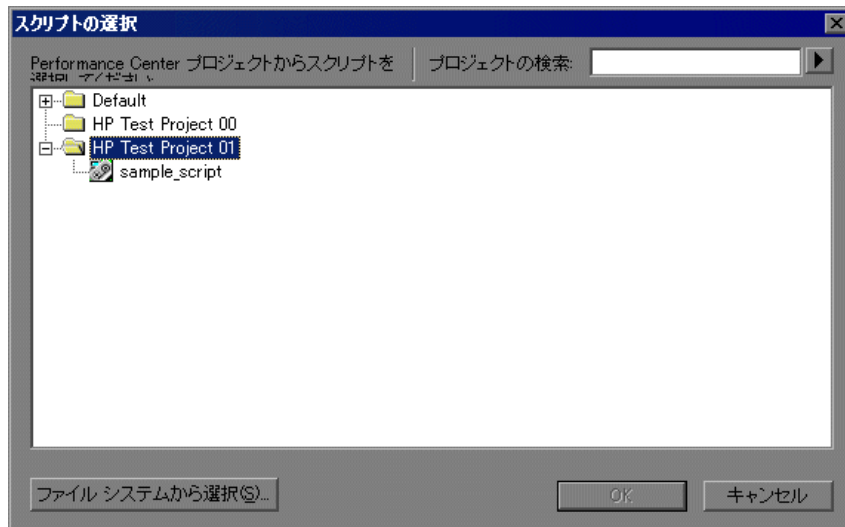
- ▶ **[実行ファイルをダウンロードする]** : VuGen は、ダウンロード時間を短縮するために、スクリプト・ファイルのみをダウンロードします。ダウンロードには、スクリプト・ファイル、実行環境の設定、およびパラメータ・ファイルが含まれます。
- ▶ **[全ファイルをダウンロードする]** : スクリプト・ファイルとデータ・ファイル（記録データや再生結果ディレクトリを含む）がダウンロードされます。

実行時のファイルのみを対象とする部分的なダウンロードは、スクリプト・ファイルがダウンロードされるだけなので高速です。スクリプト・ファイルとデータ・ファイルをすべてダウンロードすると、ダウンロードに多くの時間がかかります。

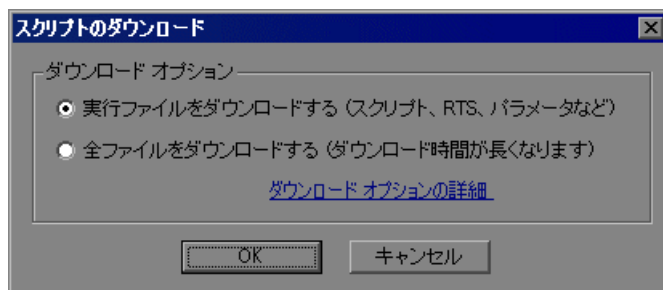
Performance Center サーバに接続すると、スクリプト・ファイルを VuGen にダウンロードできます。

Performance Center から Vuser スクリプトをダウンロードするには、次の手順を実行します。

- 1 Performance Center に接続されていることを確認します。VuGen のステータス・バーで **[PC に接続中]** ボタンを見つけます。
- 2 VuGen で、**[ファイル]** > **[開く]** を選択します。**[スクリプトの選択]** ダイアログ・ボックスが表示されます。



- 3 ダウンロードするスクリプトを含むプロジェクトを検索します。[**プロジェクトの検索**] ボックスでプロジェクトの名前または名前の一部を入力して必要なプロジェクトを指定します。検索を開始するには、ボックスの右側にある矢印をクリックします。次の一致箇所に移動するには、矢印をもう一度クリックします。
- 4 プロジェクトを展開し、ダウンロードするスクリプトを選択します。
- 5 [OK] をクリックします。[スクリプトのダウンロード] ダイアログ・ボックスが表示されます。



- 6 ダウンロード・オプションを [**実行ファイルをダウンロードする**] または [**全ファイルをダウンロードする**] のいずれかから選択します。
- 7 [OK] をクリックすると、Performance Center からファイルがダウンロードされます。ファイルのダウンロード状況を示す進行状況バーが表示されます。ダウンロードが完了すると、進行状況バーが閉じてスクリプトが表示されます。

ダウンロードしたスクリプトでの作業

スクリプトをダウンロードしたら、VuGen で編集できます。

標準設定では、ダウンロードしたファイルは **Temp** ディレクトリに保存されます。変更内容をローカルに保存することも、または編集したスクリプトを Performance Center サーバにアップロードすることもできます。

Performance Center に接続されている場合は、保存操作を行うと、スクリプトが Performance Center サーバに自動的にアップロードされます。

ソースが Performance Center サーバにあるダウンロードしたスクリプトは、[ファイル] メニューの [最近使用したスクリプト] リストに、[PC] というレフィックス付きで表示されます。

ダウンロードしたスクリプトに加えた変更内容を保存するには、次の手順を実行します。

- 1 Performance Center サーバにスクリプトを保存するには、[保存] をクリックします。VuGen にアップロード・オプションが表示されます（接続されている場合）。

再生後に生成されたスナップショットなど、新しく作成したファイルをアップロードする場合は、[全ファイルをアップロードする] を指定する必要があります。

- 2 スクリプトをローカルに保存するには、[名前を付けて保存] をクリックします。[テストを保存] ダイアログ・ボックスが開きます。[ファイル システムから選択] をクリックし、必要な場所を参照します。[OK] をクリックします。

第 2 部

パラメータ

第 13 章

VuGen パラメータを使った作業

ビジネス・プロセスを記録すると、記録中に実際に使用された値を含むスクリプトが VuGen によって生成されます。この記録された値とは異なる値を使って、スクリプトのアクション（クエリや送信など）を実行する必要がある場合を考えてみます。異なる値を使用するためには、記録された値をパラメータで置き換えます。このことを、スクリプトの「パラメータ化」と呼びます。

本章の内容

- ▶ VuGen パラメータの定義について（220 ページ）
- ▶ パラメータに関する制限（221 ページ）
- ▶ パラメータの作成（223 ページ）
- ▶ パラメータ・タイプについて（226 ページ）
- ▶ パラメータのプロパティの定義（228 ページ）
- ▶ 既存のパラメータの使用（230 ページ）
- ▶ パラメータ・リストの使用（233 ページ）
- ▶ パラメータ化オプションの設定（235 ページ）

VuGen パラメータの定義について

ビジネス・プロセスを記録すると、VuGen によって関数で構成された Vuser スクリプトが生成されます。関数内の引数の値は、記録セッション中に実際に使用された値です。

たとえば、Web アプリケーションの操作中に Vuser スクリプトを記録したとします。VuGen によって、図書館のデータベースから「UNIX」という文字列を探す次のステートメントが生成されたとします。

```
web_submit_form("db2net.exe",
  ITEMDATA,
  "name=library.TITLE",
  "value=UNIX",
  ENDITEM,
  "name=library.AUTHOR",
  "value=",
  ENDITEM,
  "name=library.SUBJECT",
  "value=",
  ENDITEM,
  LAST);
;
```

複数の Vuser と繰り返しを使用してスクリプトを再生するときは、UNIX という同じ値は使用したくありません。その代わりに、定数の値をパラメータに置き換えます。

```
web_submit_form("db2net.exe",
  ITEMDATA,
  "name=library.TITLE",
  "value=¥{Book_Title¥}",
  ENDITEM,
  "name=library.AUTHOR",
  "value=",
  ENDITEM,
  "name=library.SUBJECT",
  "value=",
  ENDITEM,
  LAST);
```

こうしておけば、Vuser を実行したときに、指定したデータ・ソースにある値でパラメータが置き換えられます。データ・ソースとして、ファイルまたは内部で生成された変数を使用できます。データ・ソースの詳細については、226 ページ「パラメータ・タイプについて」を参照してください。

Vuser スクリプトのパラメータ化には次の 2 つの利点があります。

- ▶ スクリプトのサイズが小さくなります。
- ▶ さまざまな値を使ってスクリプトをテストすることができます。たとえば、図書館のデータベースからいくつかの本を探す場合でも、submit 関数を一度書くだけで済みます。Vuser 特定の項目の探索を指示するのではなく、パラメータを使用します。再生時、Vuser はパラメータをさまざまな値に置き換えます。

パラメータ化には、次の作業が伴います。

- ▶ Vuser スクリプト内の定数値をパラメータで置き換える。
- ▶ パラメータのプロパティとデータ・ソースを設定する。

パラメータに関する制限

パラメータの対象にできるのは、関数内の引数だけです。関数の引数以外の文字列はパラメータ化できません。また、すべての関数の引数をパラメータ化できるわけでもありません。パラメータ化できる引数については、『**Online Function Reference**』（英語版）（[ヘルプ] > [関数リファレンス]）で各関数を参照してください。

lrd_stmt 関数を例にとってみます。この関数の構文は次のとおりです。

```
lrd_stmt (LRD_CURSOR FAR *mptCursor, char FAR *mpcText, long mliTextLen,
LRDOS_INT4 mjOpt1, LRDOS_INT4 mjOpt2, int miDBErrorSeverity);
```

『**Online Function Reference**』（英語版）によれば、パラメータ化できるのは **mpcText** 引数だけです。

記録された **lrd_stmt** 関数が次のようになっていたとします。

```
lrd_stmt(Csr4, "select name from sysobjects where name =¥"Kim¥" ", -1, 148, -99999, 0);
```

これは次のようにパラメータ化できます。

```
lrd_stmt(Csr4, "select name from sysobjects where name =¥" < name > ¥" ", -1, 148, -99999, 0);
```

注： `lr_eval_string` 関数を使えば、標準ではパラメータ化できない関数の引数でも「パラメータ化」することができます。さらに、`lr_eval_string` 関数を使えば、Vuser スクリプトの任意の文字列も「パラメータ化」できます。

VB, COM, および Microsoft .NET プロトコルには、パラメータを定義するのに `lr.eval string` 関数を使用する必要があります。

例：`lr.eval_string("{Custom_param}")`

`lr_eval_string` 関数の詳細については、『**Online Function Reference**』（英語版）を参照してください。

パラメータの作成

パラメータを作成するには、パラメータに名前を付け、そのタイプとプロパティを指定します。Vuser スクリプトに作成できるパラメータの数に制限はありません。パラメータを作成するには、次の手順を実行します。

手順 1：パラメータ化する引数を選択します。

スクリプト・ビューで作業している場合：


パラメータ化する引数を右クリックし、ポップアップ・メニューから **[パラメータで置換]** を選択します。

注：

- ▶ スクリプト・ビューで XML パラメータを作成する場合、境界タグを除いた内側の XML のみを選択する必要があります。たとえば、`<A>Belement<C>Celement</C>` という複合的なデータ構造をパラメータ化するには、`Belement<C>Celement</C>` の文字列全体を選択し、これをパラメータで置き換えます。
- ▶ Java Record Replay または Java Vuser スクリプトをパラメータ化する場合、文字列を部分的にではなく、全体をパラメータ化するようにします。

ツリー・ビューで作業している場合：

- 1 パラメータ化するステップを右クリックし、ポップアップ・メニューから **[プロパティ]** を選択します。該当する **[ステップのプロパティ]** ダイアログ・ボックスが開きます。

- 2  パラメータ化する引数の横にある **[ABC]** アイコンをクリックします。
[パラメータの選択または作成] ダイアログ・ボックスが開きます。



手順 2 : パラメータ名を指定します。

[**パラメータ名**] ボックスにパラメータの名前を入力します。このパラメータ名はスクリプト内で元の引数の代わりに表示されます。

パラメータ名は、スクリプトの実行中に置き換えられる情報のタイプに適した名前を付けてください。

たとえば、一般にユーザ名を入力する場合は、パラメータの名前を **Username** などとします。

注 : **unique** というパラメータ名は使用しないでください。VuGen によってすでに使用されています。

手順 3 : パラメータ・タイプを選択します。

パラメータを作成するときは、パラメータのデータ・ソースを指定します。これによって、**パラメータ・タイプ**が決まります。

データは、日付や時刻のように内部的に生成したり、ユーザ定義関数の結果として返すようにしたりできます。

パラメータの別のごく一般的な使用方法として、ユーザが定義した値が格納されているデータ・テーブルや外部ファイルから値を取得するように、**Vuser**を設定する方法があります。このようなパラメータを、**ファイル・タイプ・パラメータ**、および**テーブル・タイプ・パラメータ**と呼びます。

[**パラメータのタイプ**] リストから、[**ファイル**] を選択します。

各種のパラメータ・タイプの詳細については、226 ページ「パラメータ・タイプについて」を参照してください。

手順 4 : パラメータ・タイプのプロパティを定義します。

- 1 [**プロパティ**] をクリックします。[パラメータのプロパティ] ダイアログ・ボックスが開きます。
- 2 [**テーブルの作成**] をクリックします。メッセージ・ボックスが表示されます。[**OK**] をクリックします。

VuGen によって、引数の元の値を格納したセルを 1 つ持つテーブルが作成されます。

- 3 テーブルに別の値を追加するには、[**行の追加**] をクリックし、値を入力します。
値をさらにテーブルに追加するには、この手順を繰り返します。
- 4 [**閉じる**] をクリックして、[パラメータのプロパティ] ダイアログ・ボックスを閉じます。

詳細については、228 ページ「パラメータのプロパティの定義」を参照してください。

手順 5：そして引数をパラメータで置き換えます。

[**OK**] をクリックして、[パラメータの選択または作成] ダイアログ・ボックスを閉じます。

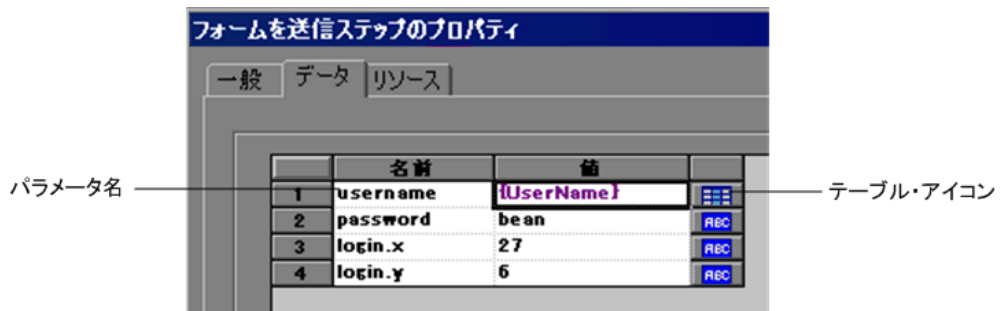
VuGen によって、スクリプト内の選択された文字列が、中括弧または丸括弧で囲まれたパラメータ名で置き換えられます。

注：標準のパラメータの括弧は、プロトコルの種類に応じて中括弧か大括弧のどちらかです。[パラメータ化] タブでパラメータに適した括弧を確認できます ([**ツール**] > [**一般オプション**])。詳細については、235 ページ「パラメータ化オプションの設定」を参照してください。



ツリー・ビューでは、[**ABC**] アイコンがテーブル・アイコンに置き換えられます。

次の例では、元の **username** の値は **jojo** でしたが、パラメータ **{UserName}** に置き換えられています。



パラメータ・タイプについて

パラメータを作成するときは、パラメータのデータ・ソースを指定します。次のタイプのデータ・ソースを指定できます。

- ▶ ファイルまたはテーブル・パラメータ・タイプ
- ▶ XML パラメータ・タイプ
- ▶ 内部データ・パラメータ・タイプ
- ▶ ユーザ定義関数のパラメータ

ファイルまたはテーブル・パラメータ・タイプ

既存のファイルか、VuGen または MS Query で作成したファイルに含まれるデータ。パラメータのごく一般的な使用法の 1 つに、データ・テーブルや外部ファイルから値を取得するように Vuser を設定するために使用するという方法があります。

データ・ファイル

データ・ファイルには、Vuser がスクリプトの実行中にアクセスするデータが格納されています。データ・ファイルはローカルにもグローバルにもできます。既存の ASCII ファイルを指定したり、VuGen を使って新しいファイルを作成したり、データベース・ファイルをインポートしたりできます。パラメータで使用する既知の値がたくさんあることがわかっている場合は、データ・ファイルが役立ちます。

データ・ファイルのデータは表形式で格納されます。1 つのファイルに多数のパラメータに対する値を格納できます。1 つのカラムに 1 つのパラメータに対するデータが保存されます。カラムの区切りはカンマなどの区切り文字で示されます。

次の例では、データ・ファイルに ID 番号と名前を格納しています。

```
id,first_name
120,John
121,Bill
122,Tom
```

注：英語以外の言語で作業を行うときは、パラメータ・ファイルを UTF-8 ファイルとして保存してください。[パラメータのプロパティ] ウィンドウで、[メモ帳で編集] をクリックします。メモ帳で、ファイルを UTF-8 文字コードのテキスト・ファイルとして保存します。

データ・テーブル

テーブル・パラメータ・タイプは、テーブルのセル値を設定することによってアプリケーションをテストする目的で使用します。ファイル・タイプでは出現するパラメータごとに 1 つのセル値を使用しますが、テーブル・タイプでは、値の配列と同じように、複数の行およびカラムをパラメータ値として使用します。テーブル・タイプを使用すると、テーブル全体を 1 回のコマンドで設定できます。これは、`sapgui_fill_data` 関数によってテーブル・セルを設定する SAPGUI Vuser では一般的です。

データ・ファイルまたはデータ・テーブルのパラメータのプロパティについては、第 14 章「ファイル、テーブル、および XML パラメータ・タイプ」を参照してください。

XML パラメータ・タイプ

XML 構造に含まれる複数の値データのプレースホルダとして使用されます。XML タイプのパラメータを使用すれば、構造全体を 1 つのパラメータで置き換えることができます。たとえば、**Address** という XML パラメータを使用して、連絡先の名前、住所、都市、郵便番号を置き換えることができます。このようなタイプのデータに XML パラメータを使用すれば、データを正確に入力でき、Vuser スクリプトをすっきりとパラメータ化することが可能になります。XML パラメータは、Web サービス・スクリプトあるいは、SOA サービスを対象に使用することをお勧めします。

内部データ・パラメータ・タイプ

内部データは Vuser の実行中に自動的に生成されるデータです。Date/Time (日時)、Group Name (グループ名)、Iteration Number (反復回数)、Load Generator Name (Load Generator 名)、Random Number (乱数)、Unique Number (一意の数)、Vuser ID などがあります。

内部データのパラメータのプロパティの定義方法については、267 ページ「内部データ・パラメータ・タイプのプロパティの設定」を参照してください。

ユーザ定義関数のパラメータ

外部の DLL の関数を使用して生成されたデータ。ユーザ定義関数は、パラメータを外部 DLL の関数から返される値で置き換えます。

ユーザ定義関数をパラメータとして割り当てる前に、その関数を含む外部ライブラリ (DLL) を作成します。関数の形式は次のとおりです。

```
__declspec(dllexport) char * <関数名> (char *, char *)
```

この関数に送られる引数は両方とも NULL です。

ライブラリを作成するときには、標準のダイナミック・ライブラリ・パスを使うことをお勧めします。そうすれば、ライブラリの完全パス名を入力する必要がなく、ライブラリ名を入力するだけで済みます。VuGen の bin ディレクトリが、標準のダイナミック・ライブラリ・パスです。このディレクトリにライブラリを追加できます。

ユーザ定義関数の例を以下に示します。

```
__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( &x)); t[24] = '\0'; return t;}
```

ユーザ定義関数のプロパティの定義方法については、276 ページ「ユーザ定義関数のプロパティの設定」を参照してください。

パラメータのプロパティの定義

パラメータのプロパティは、[パラメータのプロパティ] ダイアログ・ボックスか [パラメータ リスト] ダイアログ・ボックスで定義できます。

[パラメータのプロパティ] ダイアログ・ボックスにおけるパラメータのプロパティを定義するには、次の手順を実行します。

- 1 [パラメータのプロパティ] ダイアログ・ボックスを開きます。

[パラメータのプロパティ] ダイアログ・ボックスは、次のいずれかの方法で開くことができます。

- ▶ 新しいパラメータを作成するとき (223 ページ「パラメータの作成」を参照), [パラメータの選択または作成] ダイアログ・ボックスの [**プロパティ**] をクリックして, [パラメータのプロパティ] ダイアログ・ボックスを開きます。
- ▶ スクリプト・ビュー内でパラメータを選択し, 右クリック・メニューから [**パラメータのプロパティ**] を選択します。
- ▶ ツリー・ビューで, プロパティを定義する対象となるパラメータを含んでいるステップを右クリックし, [**プロパティ**] を選択します。選択したステップに対応する [ステップのプロパティ] ダイアログ・ボックスが開きます。
プロパティを定義する対象となるパラメータの横にあるテーブル・アイコンをクリックし, ポップアップ・メニューから [**パラメータのプロパティ**] を選択します。



次の例では, 「File」 タイプのパラメータのプロパティが表示されています。

パラメータのプロパティ - [UserName]

パラメータのタイプ(T): File

ファイルパス(F): UserName.dat 参照(R)...

列の追加(A)... 行の追加(R)... 列の削除(D)... 行の削除(B)...

| | Param1 |
|---|---------|
| 1 | joio |
| 2 | Mercury |
| 3 | Test |

メモ帳で編集(E)... データウィザード(W)...

パラメータをシミュレートする(S)...

列の選択

数順(M): 1

名前順(N):

ファイル形式

列の区切り文字(O): Comma

先頭データ行(I): 1

次の行を選択(S): Sequential

更新する対象(U): Each iteration

使用可能な値の終了後(F): Continue with last value

Controller で User の値を割り当てる

ブロックサイズを自動的に割り当てる(O)

各仮想ユーザに Q 件の値を割り当てる

開じる(O)

2 パラメータのプロパティを定義します。

- ▶ ファイルおよびテーブル・タイプ・パラメータのプロパティを定義する方法については、第 14 章「ファイル、テーブル、および XML パラメータ・タイプ」を参照してください。
- ▶ 内部データ・パラメータ・タイプのプロパティを定義する方法については、267 ページ「内部データ・パラメータ・タイプのプロパティの設定」を参照してください。
- ▶ ユーザ定義関数のプロパティを定義する方法については、228 ページ「ユーザ定義関数のパラメータ」を参照してください。

3 [パラメータのプロパティ] ダイアログ・ボックスを閉じます。

[閉じる] をクリックして、[パラメータのプロパティ] ダイアログ・ボックスを閉じます。

[パラメータ リスト] ダイアログ・ボックスにおけるパラメータのプロパティを定義するには、次の手順を実行します。



[パラメータ リスト] ボタンをクリックするか、[仮想ユーザ] > [パラメータ リスト] を選択します。プロパティを表示するパラメータを選択します。

詳細については、233 ページ「パラメータ・リストの使用」を参照してください。

既存のパラメータの使用

作成したパラメータは、VuGen によってパラメータ・リストに格納されます。既存のパラメータを使用して、1 つの引数を置き換えたり、複数出現する引数を置き換えたりできます。

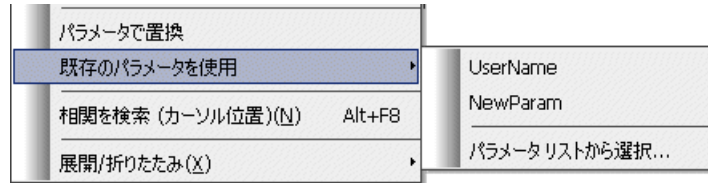
[パラメータ リスト] を使用すると、以前に定義したパラメータで引数を置き換えられるのと同時に、そのプロパティの表示や変更もできるので便利です。パラメータ・リストの使い方の詳細については、233 ページ「パラメータ・リストの使用」を参照してください。

定義済みパラメータを使用した文字列の置換

定義済みパラメータを引数に対して割り当てることができます。

定義済みパラメータで文字列を置換するには、次の手順を実行します。

- 1 スクリプト・ビューに入ります。
- 2 パラメータ化する引数を右クリックして、**[既存のパラメータを使用]** を選択します。サブメニューが開きます。



- 3 次のいずれかの方法でパラメータを選択します。
 - ▶ サブメニューのリストからパラメータを選択します。
 - ▶ **[パラメータ リストから選択]** を選択して **[パラメータ リスト]** ダイアログ・ボックスを開き、左側の表示枠からパラメータを選択します。

複数の出現の置き換え

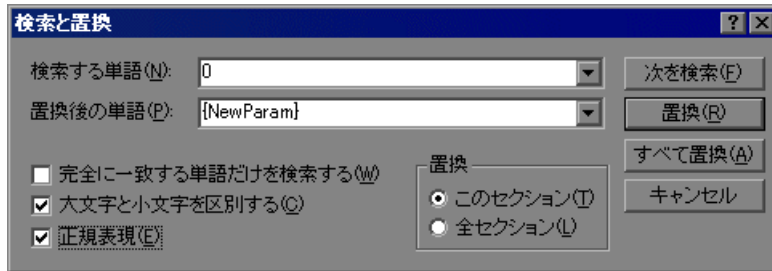
パラメータを作成すると、システムによって引数の元の値が記憶されます。**検索と置換**機能を使用すると、同じ引数が複数出現する場合に、そのうちの選択した引数、またはそれらすべての引数を、同じパラメータや既存の別のパラメータに置換できます。

複数出現する引数を特定のパラメータで置換するには、次の手順を実行します。

- 1 パラメータを右クリックし、メニューから **[一致する次の文字列を置換]** を選択します。

[検索と置換] ダイアログ・ボックスが開きます。**[検索する単語]** ボックスに、置換対象の値または引数が表示されます。**[置換後の単語]** ボックスに、括弧にはさまれたパラメータ名が表示されます。

- 必要に応じて [完全に一致する単語だけを検索する] や [大文字と小文字を区別する] のチェック・ボックスを選択します。正規表現 (., !, ?, + など) を使って検索するには, [正規表現] チェック・ボックスを選択します。



- [置換] または [すべて置換] をクリックします。

上の例では、値が **15** の引数がすべて、パラメータ **{NewParam}** に置換されます。

注: [すべて置換] の使用は、特に数字文字列の置換の場合は注意が必要です。指定した文字列がすべての出現箇所ですべて置換されます。

元の文字列の復元

VuGen では、パラメータ化を取り消し、記録された元の引数を復元することができます。

パラメータの元の値を復元するには、次の手順を実行します。

- ▶ [スクリプト ビュー]: パラメータを右クリックし、[既定値を復元] を選択します。
- ▶ [ツリー ビュー]:
 - ▶ パラメータが含まれているステップを右クリックして、[プロパティ] を選択します。
 - ▶ 元の値に戻すパラメータの横にあるテーブル・アイコンをクリックし、[パラメータを元に戻す] を選択します。



元の引数が復元されます。

パラメータ・リストの使用

パラメータ・リストを使用すると、パラメータの表示、作成、削除、選択、および変更が行えます。

パラメータ・リストを表示し、パラメータのプロパティを表示するには、次の手順を実行します。

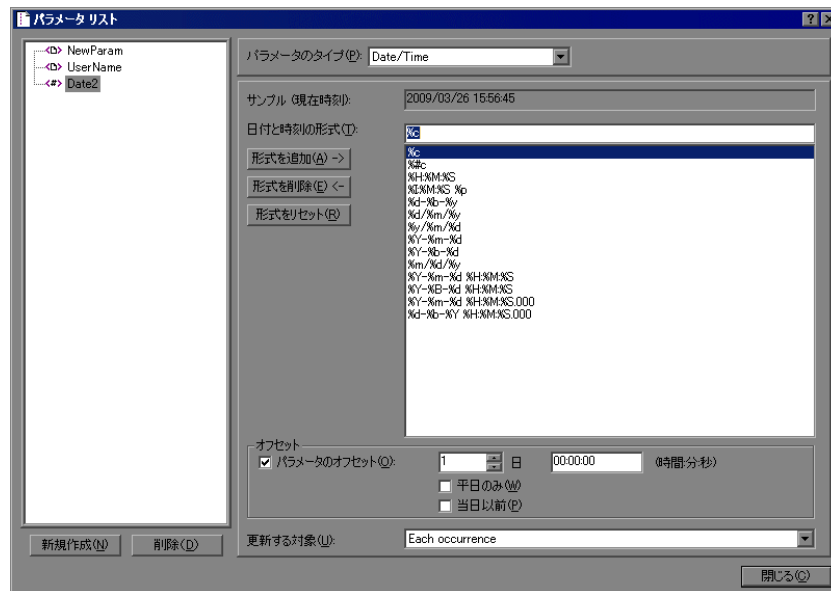


[**パラメータ リスト**] ボタンをクリックするか、[**仮想ユーザ**] > [**パラメータ リスト**] を選択します。プロパティを表示するパラメータを選択します。

パラメータ・リストには、作成したパラメータがすべて表示されます。これには、入力パラメータと出力パラメータの両方が含まれます。入力パラメータとは、スクリプトを実行する前に設計段階に値を定義するパラメータのことで、出力パラメータは設計段階に定義しますが、テストの実行中に値を取得します。出力パラメータは、多くの場合 Web サービス呼び出しとともに使用されます。

設計段階にスクリプトのパラメータを選択するときは注意して、空の出力パラメータではないことを確認します。

次の例では、「**Date/Time**」タイプのパラメータのプロパティを表示しています。



パラメータのプロパティを変更するには、次の手順を実行します。

左側のパラメータ・ツリーからパラメータを選択し、パラメータのタイプとプロパティを右側の表示枠で編集します。

パラメータ・プロパティの設定方法の詳細については、第 15 章「パラメータのプロパティの設定」および第 14 章「ファイル、テーブル、および XML パラメータ・タイプ」を参照してください。

新しいパラメータを作成するには、次の手順を実行します。

- 1 [パラメータ リスト] ダイアログ・ボックスで、**[新規作成]** をクリックします。新規パラメータは、仮の名前でパラメータ・ツリーに表示されます。
- 2 新規パラメータの名前を入力して、Enter キーを押します。

注： **unique** というパラメータ名は使用しないでください。VuGen によってすでに使用されています。

- 3 パラメータのタイプとプロパティを設定します。
- 4 **[閉じる]** をクリックして、[パラメータ リスト] ダイアログ・ボックスを閉じます。

注： VuGen によって新規パラメータが作成されますが、スクリプト内で選択されている文字列をそのパラメータで自動的に置き換えることはありません。

既存のパラメータを削除するには、次の手順を実行します。

- 1 パラメータ・ツリーからパラメータを選択し、**[削除]** をクリックします。[パラメータの削除] ダイアログ・ボックスが開きます。
- 2 パラメータ・ファイルをディスクから削除したければ、**[ディスクからパラメータ データ ファイルを削除する]** を選択します。
- 3 **[はい]** をクリックします。
- 4 **[ディスクからパラメータ データ ファイルを削除する]** を選択した場合は、警告メッセージが表示されます。**[はい]** をクリックして操作を確定します。

パラメータ化オプションの設定

パラメータ化オプションは、VuGen の [一般オプション] ウィンドウの [パラメータ化] タブで設定します。これらのオプションは次の項目を対象とします。

- ▶ パラメータの括弧
- ▶ グローバル・ディレクトリ

パラメータの括弧

パラメータを Vuser スクリプトに挿入すると、VuGen によってパラメータ名の前後にパラメータ用の括弧が追加されます。[パラメータ化] タブ ([ツール] > [一般オプション]) に、プロトコルの標準の括弧が表示されます。次の例では、Web プロトコルは中括弧を使用します。

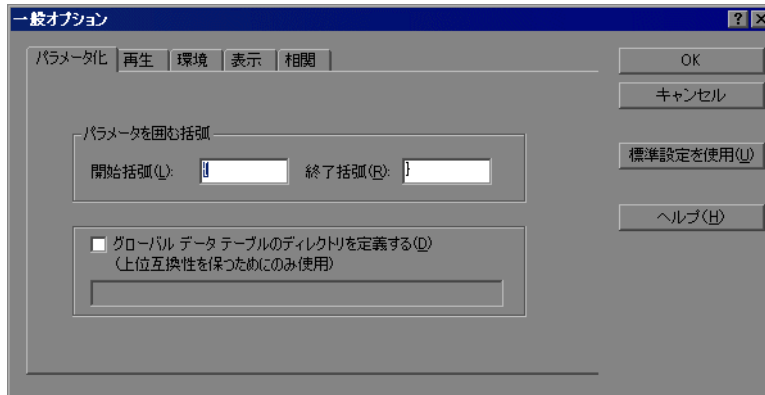
```
web_submit_form("db2net.exe",
    ITEMDATA,
    "name=library.TITLE",
    "value={Book_Title}",
    ENDITEM,
    "name=library.AUTHOR",
    "value=",
    ENDITEM,
    "name=library.SUBJECT",
    "value=",
    ENDITEM,
    LAST);
```

1 文字または複数の文字で構成される文字列を指定して、パラメータの括弧のスタイルを変更できます。スペース以外のすべての文字を使用できます。

注：標準のパラメータの括弧は、プロトコルの種類に応じて大括弧か中括弧のどちらかです。標準の括弧の種類を確認するには、新しいスクリプトを作成し、[パラメータ化] タブ ([ツール] > [一般オプション]) を確認します。

パラメータの括弧のスタイルを変更するには、次の手順を実行します。

- 1 VuGen で [ツール] > [一般オプション] を選択します。[一般オプション] ダイアログ・ボックスが開きます。
- 2 [パラメータ化] タブを選択し、使用する括弧を入力します。



- 3 [OK] をクリックして設定を適用し、ダイアログ・ボックスを閉じます。

グローバル・ディレクトリ

このオプションは、VuGen の前のバージョンとの互換性を保つためにだけ提供されています。以前のバージョン (4.51 またはそれ以前のバージョン) では、新しいデータ・テーブルを作成するときに、ローカルかグローバルかを指定していました。ローカルのテーブルは現在の Vuser スクリプトのディレクトリに保存され、スクリプトを実行している Vuser だけが使用できます。グローバルなテーブルはすべての Vuser スクリプトで使用できます。グローバル・ディレクトリはローカル・ドライブ上にもネットワーク・ドライブ上にも置けます。スクリプトを実行するすべてのマシンから、グローバル・ディレクトリを利用できることを確認してください。グローバル・テーブルの場所は、[一般オプション] ダイアログ・ボックスを使っていつでも変更できます。

VuGen の新しいバージョンでは、[パラメータのプロパティ] ダイアログ・ボックスまたは [パラメータ リスト] ダイアログ・ボックスの中でデータ・テーブルの場所を指定します。VuGen は、標準設定のスクリプト・ディレクトリや、ネットワーク上の別のディレクトリなど、指定した任意の場所のデータを取得できます。詳細については、226 ページ「データ・ファイル」を参照してください。

標準設定では、**[グローバル データ テーブルのディレクトリを定義する]** オプションは無効になっています。

グローバル・ディレクトリを設定するには、次の手順を実行します。

- 1 **[ツール]** > **[一般オプション]** を選択します。**[一般オプション]** ダイアログ・ボックスが開きます。
- 2 **[パラメータ化]** タブを選択します。
- 3 **[グローバル データ テーブルのディレクトリを定義する]** チェック・ボックスを選択し、グローバル・データ・テーブルがあるディレクトリを指定します。
- 4 **[OK]** をクリックして設定を適用し、ダイアログ・ボックスを閉じます。

第 14 章

ファイル，テーブル，および XML パラメータ・タイプ

パラメータのごく一般的な使用法の 1 つに，データ・テーブルや外部ファイルから値を取得するように Vuser を設定するために使用するという方法があります。データは，既存のファイルか，VuGen または MS Query で作成したファイルに含まれています。

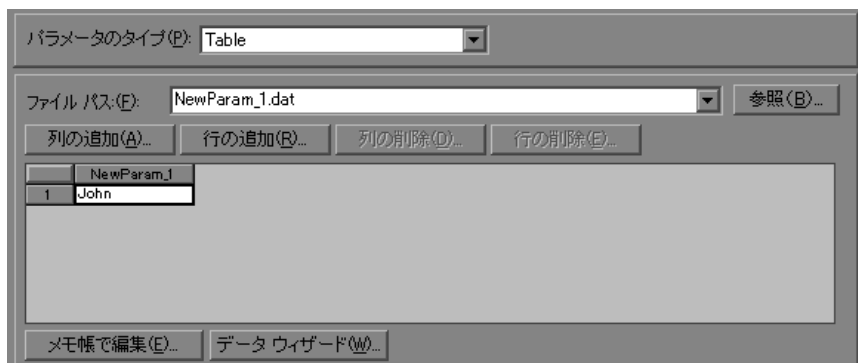
本章の内容

- ▶ データ・ファイルまたはデータ・テーブルの選択または作成 (240 ページ)
- ▶ ファイル・タイプのパラメータのプロパティ設定 (246 ページ)
- ▶ テーブル・タイプのパラメータのプロパティ設定 (248 ページ)
- ▶ ファイル・タイプまたはテーブル・タイプのパラメータにデータを割り当てる方法の選択 (250 ページ)
- ▶ XML パラメータのプロパティの設定 (255 ページ)

データ・ファイルまたはデータ・テーブルの選択または作成

ファイル・タイプまたはテーブル・タイプのパラメータを作成するときは，データを格納するための .dat ファイルを作成するか，既存の .dat ファイルを開く必要があります。そして，パラメータのほかのプロパティを定義します。たとえば，Vuser におけるパラメータへの値の割り当て方法を定義します。

新しいデータ・テーブルを作成するか，既存のデータ・ソースを [ファイルパス] リストから選択できます。



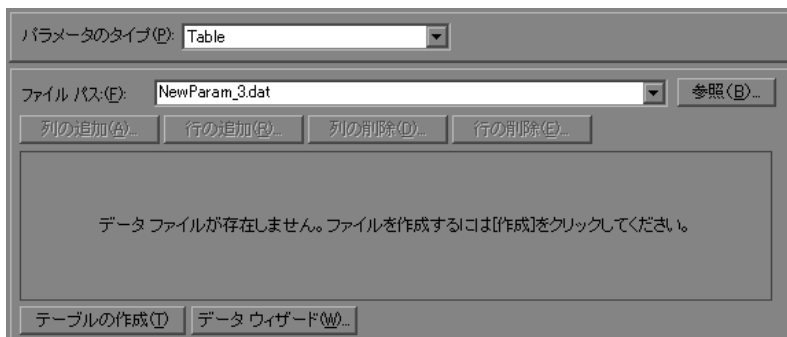
データのソース・ファイルまたはテーブルの選択は，次の手順で行います。

- 1 [パラメータのプロパティ] ダイアログ・ボックス，または [パラメータ リスト] を開きます。

手順の詳細については，228 ページ「パラメータのプロパティの定義」を参照してください。

- 2 テーブルを選択するか，または新しいテーブルを作成します。
 - ▶ ファイル・パス・リストにテーブル (.dat ファイル) が表示されない場合，または新しいテーブルを作成する場合は，[テーブルの作成] をクリックします。1 つのセルを持つ新しいテーブルが作成され，引数の元の値がテーブルの 1 番目のカラムに表示されます。
 - ▶ 既存のデータ・ファイルを開くには，.dat ファイルの名前を [ファイルパス] ボックスに入力するか，ドロップダウン・リストから名前を選択します。

または、**[参照]** をクリックして、既存のデータ・ファイルの場所を指定します。標準設定では、新しいデータ・ファイルはすべて **<パラメータ名>.dat** という名前で、スクリプトのディレクトリに格納されます。



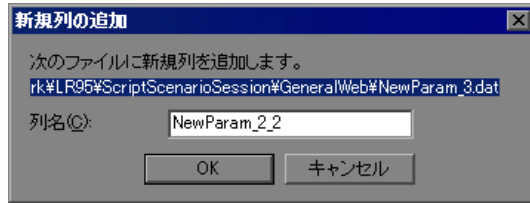
データ・ファイルが開き、最初の 100 行が表示されます。すべてのデータを表示するには、**[メモ帳で編集]** をクリックし、**[メモ帳]** でデータを表示します。

注： グローバル・ディレクトリを指定することもできます。グローバル・ディレクトリは、VuGen の前のバージョンとの互換性を保つためにだけ提供されています。詳細については、236 ページ「グローバル・ディレクトリ」を参照してください。

- ▶ 既存のデータベースからデータをインポートするには、**[データ ウィザード]** をクリックし、ウィザードの指示に従います。詳細については、243 ページ「既存のデータベースからのデータのインポート」を参照してください。

3 カラムと行をテーブルに追加します。

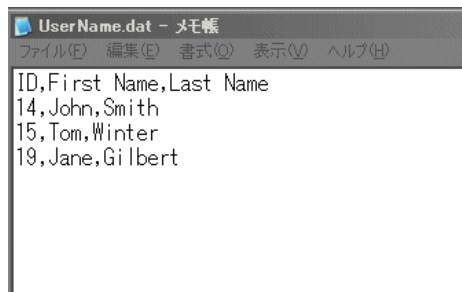
- ▶ テーブルにカラムを追加するには、[カラムの追加] を選択します。[新規列の追加] ダイアログ・ボックスを開きます。列名を入力して [OK] をクリックします。



- ▶ テーブルに行を追加するには、[行の追加] を選択します。

4 データ・ファイルを編集します。

- ▶ 任意のセルの中で値を入力します。
- ▶ データ・ファイルをメモ帳の中から編集するには、[メモ帳で編集] をクリックします。[メモ帳] が開いて、1行目にパラメータ名、2行目にパラメータの最初の値が表示されます。追加のカラム名と値をファイルに入力します。カンマやタブなどの区切り文字を使用してカラムの区切りを示します。テーブルの行ごと（新しいデータ行ごと）に改行します。



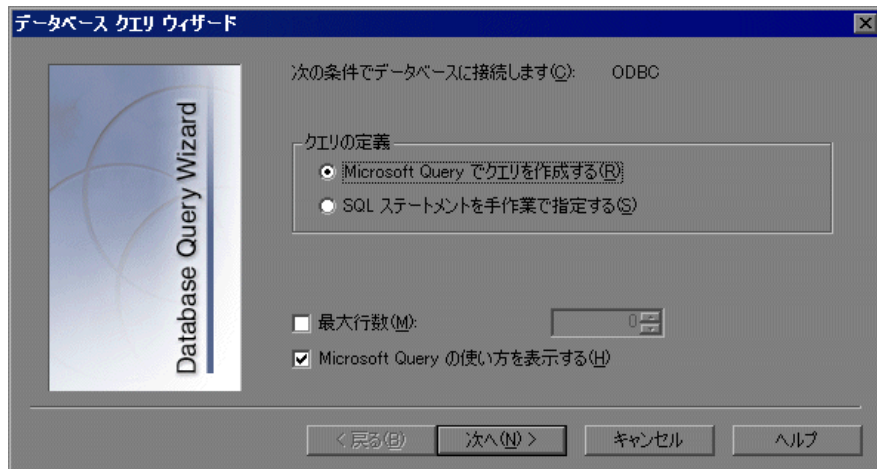
既存のデータベースからのデータのインポート

VuGen では、データベースからデータをインポートし、パラメータの値として使用できます。データをインポートする方法は 2 つあります。

- ▶ 新規クエリの作成
- ▶ SQL ステートメントの指定

VuGen では、ウィザードを実行し、その指示に従ってデータベースからデータをインポートすることができます。ウィザードで、データのインポート方法を指定します。MS Query を使って新しいクエリを作成するか、SQL ステートメントを指定します。データをインポートすると、**.dat** という拡張子を持つファイルに保存され、通常のパラメータ・ファイルとして利用できるようになります。

データベースのインポートを開始するには、まず [パラメータ リスト] ダイアログ・ボックス ([**仮想ユーザ**] > [**パラメータ リスト**]) の [**データ ウィザード**] をクリックします。[データベース クエリ ウィザード] が開きます。

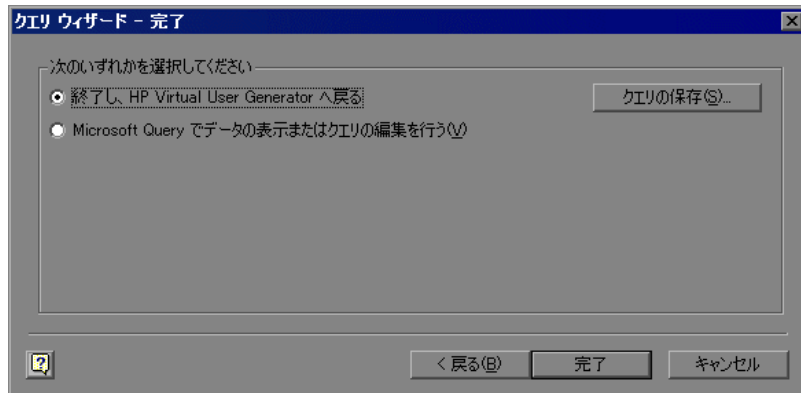


新規クエリの作成

新規クエリの作成には、Microsoft のデータベース・クエリ・ウィザードを使用します。システムに MS Query をインストールしておく必要があります。

新規クエリを作成するには、次の手順を実行します。

- 1 **[Microsoft Query でクエリを作成する]** を選択します。MS Query についての説明が必要な場合は、**[Microsoft Query の使い方を表示する]** を選択してください。
- 2 **[完了]** をクリックします。MS Query がマシンにインストールされていない場合は、利用できないことを示すメッセージが表示されます。処理を進める前に、Microsoft Office から MS Query をインストールします。
- 3 ウィザードに表示される指示に従って、必要なテーブルとカラムをインポートします。
- 4 データのインポートが終了したら、**[終了し、仮想ユーザ ジェネレータへ戻る]** を選択し、**[完了]** をクリックします。データベース・レコードが **[パラメータ リスト]** ボックスにデータ・ファイルとして表示されます。



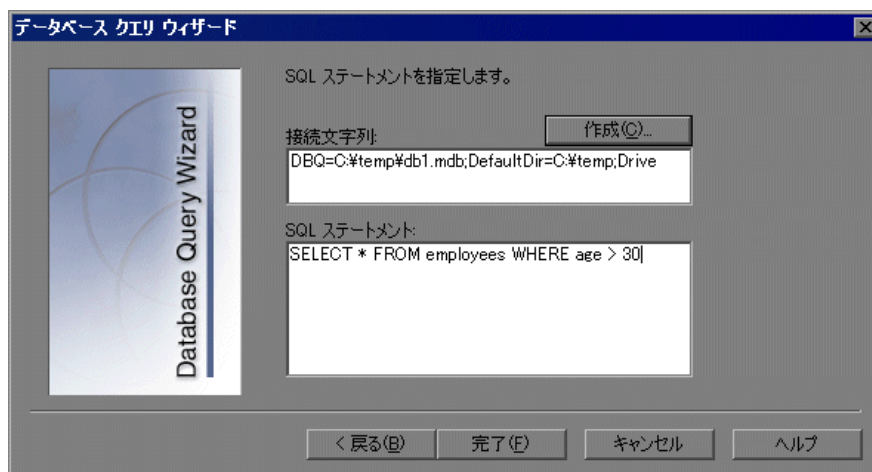
終了せずに MS Query でデータを表示または編集するには、**[Microsoft Query でデータの表示またはクエリの編集を行う]** を選択します。

- 5 データの割り当てプロパティを設定します。詳細については、246 ページ「ファイル・タイプのパラメータのプロパティ設定」を参照してください。

SQL ステートメントの指定

データベース接続と SQL ステートメントの指定を行うには、次の手順を実行します。

- 1 **[SQL ステートメントを手作業で指定する]** を選択します。**[次へ]** をクリックします。
- 2 新規接続文字列を指定するために、**[作成]** をクリックします。**[データ ソースの選択]** ウィンドウが開きます。
- 3 既存のデータ・ソースを選択するか、**[新規作成]** を選択してデータ・ソースを新規作成します。ウィザードの指示に従って、ODBC データ・ソースを作成します。作業が終わると、接続文字列が **[接続文字列]** ボックスに表示されます。
- 4 **[SQL ステートメント]** ボックスに、SQL ステートメントを入力します。



- 5 **[完了]** をクリックすると、SQL ステートメントが処理され、データがインポートされます。データベース・レコードが **[パラメータ リスト]** ボックスにデータ・ファイルとして表示されます。
- 6 データの割り当てプロパティを設定します。詳細については、246 ページ「ファイル・タイプのパラメータのプロパティ設定」を参照してください。

テーブルまたはファイルのデータを作成したら、割り当てプロパティを設定します。プロパティでは、使用するカラムと行を指定し、データをランダムに使用するか順次使用するかを指定します。プロパティは、ファイル・タイプ・パラメータとテーブル・タイプ・パラメータで別々に設定します。

注：パラメータのプロパティは、[パラメータ リスト] ダイアログ・ボックスでも設定できます。左側の表示枠でパラメータを選択し、右側の表示枠でそのプロパティを指定します。詳細については、233 ページ「パラメータ・リストの使用」を参照してください。

ファイル・タイプのパラメータのプロパティ設定

データ・ソースを選択したら、ファイルの割り当てプロパティを設定します。これらのプロパティは、VuGen にデータの使用方法を指示するものです。たとえば、これらのプロパティは、使用するカラム、新規の値を使用する頻度、一意の値がなくなったときにどうするかなどを指定します。

The screenshot shows a dialog box with the following settings:

- 列の選択 (Column Selection):**
 - 数順 (M): 1
 - 名前順 (N):
- ファイル形式 (File Format):**
 - 列の区切り文字 (C): Comma
 - 先頭データ行 (L): 1
- 次の行を選択 (S):** Unique
- 更新する対象 (U):** Each iteration
- 使用可能な値の終了後 (H):** Continue with last value
- Controller で Vuser の値を割り当てる (Assign Vuser values in Controller):**
 - ブロック サイズを自動的に割り当てる (Assign automatically by block size)
 - 各仮想ユーザに (N) 件の値を割り当てる (Assign (N) values to each virtual user)

ファイル・タイプのパラメータのプロパティを設定するには、次の手順を実行します。

- 1 パラメータ用の値を含むテーブルの列番号を指定します。[列の選択] セクションで、列番号か列名を指定します。

列番号で指定するには、[数順] を選択して列番号を選びます。列番号とはデータを含んでいる列のインデックスです。たとえば、パラメータのデータがテーブルの最初の列にある場合は、1 に設定します。

列名で指定するには、[名前順] を選択して、リストから列名を選びます。列ヘッダは、各列の最初の行 (0 行) にあります。列番号が変わる可能性がある場合、あるいはヘッダがない場合は、列の指定に列名を使用します。

- 2 **[ファイル形式]** セクションの **[列の区切り文字]** ボックスに、列の区切り文字を入力します。区切り文字とは、テーブルの列を区切るために使用する文字です。カンマ (Comma)、タブ (Tab)、またはスペース (Space) を指定できます。
- 3 **[ファイル形式]** セクションの **[先頭データ行]** ボックスで、Vuser スクリプト実行時に使用するデータの開始行を選択します。ヘッダは 0 行目です。ヘッダの後の最初の行から開始するには、「1」を指定します。ヘッダがない場合には、「0」を指定します。
- 4 **[次の行を選択]** リストからデータの割り当て方法を選択して、Vuser が Vuser スクリプト実行時にファイル・データをどのように選択するかを指定します。オプションには、**[Sequential]** (順次)、**[Random]** (ランダム)、または **[Unique]** (一意) があります。詳細については、250 ページ「ファイル・タイプまたはテーブル・タイプのパラメータにデータを割り当てる方法の選択」を参照してください。
- 5 **[更新する対象]** リストから更新オプションを選択します。**[Each iteration]** (反復ごと)、**[Each occurrence]** (出現ごと)、**[Once]** (一度) から選択します。詳細については、252 ページ「ファイル/テーブル/XML タイプのパラメータに対するデータの割り当て方法と更新方法」を参照してください。
- 6 データの割り当て方法として **[Unique]** (一意) を選択した場合には (手順 4)、以下を指定します。
 - ▶ **[使用可能な値の終了後]** : 一意の値がなくなった場合の処理として、**[Abort Vuser]** (Vuser の中止)、**[Continue in a cyclic manner]** (循環して継続)、**[Continue with last value]** (最後の値で継続) のいずれかを指定します。
 - ▶ **[Controller で Vuser の値を割り当てる]** (LoadRunner のユーザのみ) : Vuser に手作業でデータ・ブロックを割り当てるかどうか指定します。Controller にブロック・サイズを自動的に割り当てさせるか、必要な値の数を指定します。**[ブロックサイズを自動的に割り当てる]** または **[仮想ユーザに X 件の値を割り当てる]** を選択します。後者の場合には、割り当てる値の数を指定します。これを追跡するには、**[実行環境設定]** の **[ログ]** で、**[拡張ログ]** > **[パラメータ置換]** オプションを有効にします。十分なデータがない場合は、VuGen は Vuser ログに「No more unique values for this parameter in table <テーブル名>」 (<テーブル名>テーブルには、このパラメータのための一意の値がこれ以上ありません) という警告メッセージを表示します。

テーブル・タイプのパラメータのプロパティ設定

データ・テーブルを選択したら，その割り当てプロパティを設定します。これらのプロパティは，VuGen にテーブル・データの使用方法を指示するものです。たとえば，これらのプロパティは，使用するカラムと行，それらを使用する頻度，一意の値がなくなったときにどうするかなどを指定します。

テーブル・タイプのパラメータのプロパティを設定するには，次の手順を実行します。

- 1 パラメータ用の値を含むテーブルのカラム番号を指定します。[列] セクションで，どのカラムを使用するのかを指定します。または，[すべてを選択する] を選択します。

1 つまたは複数のカラムを番号で指定するには，[番号順に表示する] を選択し，カラム番号をカンマまたはダッシュで区切って入力します。カラム番号とはデータを含んでいるカラムのインデックスです。たとえば，パラメータのデータがテーブルの最初のカラムにある場合は，1 を選択します。

- 2 [列の区切り文字] ボックスに，カラムの区切り文字を入力します。区切り文字とは，テーブルのカラムを区切るために使用する文字です。カンマ (Comma)，タブ (Tab)，またはスペース (Space) を使用できます。
- 3 [行] セクションで，1 回の反復でいくつの行を使用するかを [反復ごとの行数] ボックスに指定します。

注：これは [更新する対象] フィールドを [Each iteration] (反復ごと) に設定したときのみ有効です。[更新する対象] が [Once] (一度) に設定されている場合は、すべての反復で同じ行が使用されます。

- 4 [第 1 行目のデータ] ボックスで、スクリプト実行時に使用するデータの開始行を選択します。ヘッダの後の最初の行から開始するには、「1」を入力します。何行のデータが使用可能かなどの、テーブルに関する情報を表示するには、[テーブル詳細] をクリックします。
- 5 データ表現の行区切り文字を [ログ表示に使用する行区切り文字] ボックスで指定します。この区切り文字は出力ログで行を区別するのに使用されます。パラメータ置換ログを有効にしている場合は、置換された値が再生ログに送信されます。再生ログ内の行区切り文字は、新しい行を示します。
- 6 [行数が不足する場合の対処法] ボックスで、反復のための十分な行数がテーブルにないときの処理方法を指定します。たとえば、設定するテーブルの行数が 3 で、データには 2 行しかないとします。2 つの行のみ設定する場合は、[パラメータは必要未満の行数を受け取ります] を選択します。[次の行を選択] ボックスの [Random] (ランダム) または [Sequential] (順次) で指定した方法に従って次の行を反復処理して取得する場合は、「[次の行を選択] の動作を使用」を選択します。
- 7 [次の行を選択] リストからデータの割り当て方法を選択して、Vuser が Vuser スクリプト実行時にテーブル・データをどのように選択するかを指定します。オプションには、[Sequential] (順次)、[Random] (ランダム)、または [Unique] (一意) があります。詳細については、250 ページ「ファイル・タイプまたはテーブル・タイプのパラメータにデータを割り当てる方法の選択」を参照してください。
- 8 [更新する対象] リストから更新方法を選択します。[Each iteration] (反復ごと) または [Once] (一度) から選択します。詳細については、252 ページ「ファイル/テーブル/XML タイプのパラメータに対するデータの割り当て方法と更新方法」を参照してください。

9 [Unique] (一意) を使用してデータを割り当てるよう選択した場合には、以下を指定します。

- ▶ [使用可能な値の終了後] : 一意の値がなくなった場合の処理として、
[Abort Vuser] (Vuser の中止), [Continue in a cyclic manner] (循環して継続), [Continue with last value] (最後の値で継続) のいずれかを指定します。
- ▶ [Controller で Vuser の値を割り当てる] (LoadRunner のユーザのみ) :
Vuser に手作業でデータ・ブロックを割り当てるかどうか指定します。
Controller にブロック・サイズを自動的に割り当てさせるか、必要な値の数を指定します。[ブロックサイズを自動的に割り当てる] または [仮想ユーザに X 件の値を割り当てる] を選択します。後者の場合には、割り当てる値の数を指定します。

これを追跡するには、[実行環境設定] の [ログ] で、[拡張ログ] > [パラメータ置換] オプションを有効にします。十分なデータがない場合は、VuGen は Vuser ログに「No more unique values for this parameter in table <テーブル名>」 (<テーブル名>テーブルには、このパラメータのための一意の値がこれ以上ありません) という警告メッセージを表示します。

ファイル・タイプまたはテーブル・タイプのパラメータにデータを割り当てる方法の選択

ファイルから値を取得して使う場合、VuGen ではソースからのデータをパラメータに割り当てる方法を指定できます。次の方法が使用できます。

- ▶ [Sequential] (順次)
- ▶ [Random] (ランダム)
- ▶ [Unique] (一意)

[Sequential] (順次)

[Sequential] (順次) 方式は、データを順次 (シーケンシャルに) Vuser に割り当てます。Vuser は実行時にデータ・テーブルにアクセスして、利用できる次の行を取得します。

データ・テーブルに十分な数の値がない場合、VuGen はテーブルの最初の値に戻り、テストの終了までその循環を繰り返します。

[Random] (ランダム)

[Random] (ランダム) 方式は、テスト実行の開始時に、データ・テーブルから値を無作為に選んで各 Vuser に割り当てます。

シナリオまたは Business Process Monitor プロファイルを実行するときには、乱数の生成に使われるシード値を指定できます。各シード値は、テスト実行に使用されるランダム値のシーケンスを表します。同じシード値を使用している場合はいつも同じ値のシーケンスがシナリオ内の Vuser に割り当てられます。テスト実行時に問題が生じ、同じ乱数シーケンスを使ってテストを再実行したいときに、このオプションを有効にします。

詳細については、**HP LoadRunner Controller**、**HP Performance Center**、または **HP Business Availability Center** を参照してください。

[Unique] (一意)

[Unique] (一意) 方式は、一意のシーケンシャルな値を Vuser のパラメータに割り当てます。

[一意] を選択する場合は、すべての Vuser とその反復回数に対して十分なデータがテーブルになければなりません。20 個の Vuser がいて 5 回の反復を実行したい場合には、テーブルに少なくとも一意の値が 100 個必要です。

データ・テーブルに十分な値がない場合は、処理方法を VuGen に指示できます。詳細については、246 ページ「ファイル・タイプのパラメータのプロパティ設定」または 248 ページ「テーブル・タイプのパラメータのプロパティ設定」を参照してください。

注： LoadRunner ユーザの場合、スクリプトで **[Unique]** (一意) のファイル・パラメーター化が使用されると、同じシナリオ内でそのスクリプトを使って複数の Vuser グループを実行すると、予期しないシナリオの結果が生じる場合があります。詳細については、『**HP LoadRunner Controller ユーザーズ・ガイド**』を参照してください。

ファイル/テーブル/XML タイプのパラメータに対するデータの割り当て方法と更新方法

ファイル・タイプ、テーブル・タイプ、XML タイプのパラメータの場合、データの割り当て方法と更新方法の選択が、シナリオの実行中に Vuser がパラメータを置換するために使用する値を左右します。

次の表は、選択したデータの割り当てプロパティと更新プロパティに応じて Vuser が使用する値をまとめたものです。

| 更新方法 | データの割り当て方法 | | |
|--|---|--|---|
| | 順次 | ランダム | 一意 |
| Each Iteration (反復ごと) | Vuser は反復ごとにデータ・テーブルから次の値を取得する | Vuser は反復ごとにデータ・テーブルから新しい乱数値を取得する | Vuser は反復ごとにデータ・テーブルの次の一意の位置から値を取得する |
| Each Occurrence (出現ごと) (データ・ファイルのみ) | Vuser は、同じ反復の中であっても、パラメータの出現ごとにデータ・テーブルから次の値を取得する | Vuser は、同じ反復の中であっても、パラメータの出現ごとにデータ・テーブルから新しい乱数値を取得する | Vuser は、同じ反復の中であっても、パラメータの出現ごとにデータ・テーブルから新しい一意の値を取得する |
| Once | 1 回目の反復で割り当てられた値が、Vuser ごとに、以降のすべての反復で使用される | 1 回目の反復で割り当てられた乱数値が、その Vuser のすべての反復で使用される | 1 回目の反復で割り当てられた一意の値が、Vuser の以降のすべての反復で使用される |

例

テーブルまたはファイルに次の値が格納されているとします。

Kim, David, Michael, Jane, Ron, Alice, Ken, Julie, Fred

順次方式

- ▶ 更新方法で [**Each iteration**] (反復ごと) を選択すると、すべての Vuser が 1 回目の反復では Kim を、2 回目の反復では David を、3 回目の反復では Michael を取得する、という具合になります。
- ▶ 更新方法で [**Each occurrence**] (出現ごと) を選択すると、すべての Vuser が 1 回目の出現では Kim を、2 回目の出現では David を、3 回目の出現では Michael を取得する、という具合になります。
- ▶ 更新方法で [**Once**] (一度) を選択すると、すべての Vuser がすべての反復で Kim を取得します。

注 : [**Sequential**] (順次) 方式を選択し、データ・テーブルに十分な数の値がない場合、VuGen はテーブルの最初の値に戻りテストの終了までその循環を繰り返します。

ランダム方式

- ▶ 更新方法で [**Each iteration**] (反復ごと) を選択すると、Vuser は反復ごとにテーブルから取得した乱数を使用します。
- ▶ 更新方法で [**Each occurrence**] (出現ごと) を選択すると、Vuser はパラメータの出現ごとに乱数を使用します。
- ▶ 更新方法で [**Once**] (一度) を選択すると、すべての Vuser がすべての反復で、最初にランダムに割り当てられた値を取得します。

一意方式

- ▶ 更新方法で [**Each iteration**] (反復ごと) を選択し、1 回のテスト実行で反復を 3 回実行するように指定した場合には、最初の Vuser は、1 回目の反復で Kim を、2 回目の反復で David を、3 回目の反復で Michael を取得します。2 つ目の Vuser は、Jane, Ron, Alice を取得します。3 つ目の Vuser は、Ken, Julie, Fred を取得します。
- ▶ 更新方法で [**Each occurrence**] (出現ごと) を選択すると、Vuser はパラメータの出現ごとに、リストから取得した一意の値を使用します。
- ▶ 更新方法で [**Unique**] (一意) を選択すると、最初の Vuser はすべての反復で Kim を取得し、2 番目の Vuser はすべての反復で David を取得する、という具合になります。

Controller での Vuser の振る舞い (LoadRunner のみ)

シナリオを設定してパラメータ化されたスクリプトを実行するときに十分な値がない場合、Vuser の振る舞いを指定できます。次の表に、パラメータの設定とシナリオの結果をまとめます。

- ▶ [次の行を選択] : [Unique] (一意)
- ▶ [更新する対象] : [Each iteration] (反復ごと)
- ▶ [使用可能な値の終了後] : [Continue with last value] (最後の値で継続)

| 状況 | 継続期間 | 結果の動作 |
|------------------------------|-----------------------------------|---|
| 値よりも反復が多い | [完了するまで実行する] | 一意の値が終了すると、各 Vuser は最後の値を使って継続しますが、値がもはや一意ではないことを示す警告メッセージが送信されます。 |
| 値よりも Vuser が多い | [Run indefinitely or Run for ...] | Vuser は一意の値を使い尽くします。その後にテストから「 Error: Insufficient records for param <パラメータ名> in table to provide the Vuser with unique data 」というエラー・メッセージが発行されます。これを回避するには、[パラメータのプロパティ] の [使用可能な値の終了後] オプションまたは [パラメータのプロパティ] の [次の行を選択] メソッドを変更します。 |
| 2 つのパラメータのうちの 1 つが値を使い果たしたとき | [Run indefinitely or Run for ...] | 値を使い果たしたパラメータは、2 番目のパラメータの値が一意でなくなるまで循環して継続します。 |

XML パラメータのプロパティの設定

特定の操作をエミュレートする Web サービス呼び出しを作成すると、その操作の引数に、多くの値を持つ複雑な構造が含まれる場合があります。XML タイプのパラメータを使用すれば、構造全体を 1 つのパラメータで置き換えることができます。

XML 要素の値セットをいくつか作成して、反復ごとに異なる値セットを割り当てることもできます。

XML パラメータ・タイプは、配列、Choice、および <any> 要素などの複雑なスキーマ・タイプをサポートします。

本項では、次の項目について説明します。

- ▶ 新規 XML パラメータの作成
- ▶ 値セットの定義
- ▶ 割り当て方法の設定
- ▶ XML パラメータのプロパティの変更

新規 XML パラメータの作成

Web サービスの [入力引数] を使って作業する場合、配列とそのサブ要素に遭遇することがあります。すべての配列要素の値を含む単一の XML パラメータを定義できます。

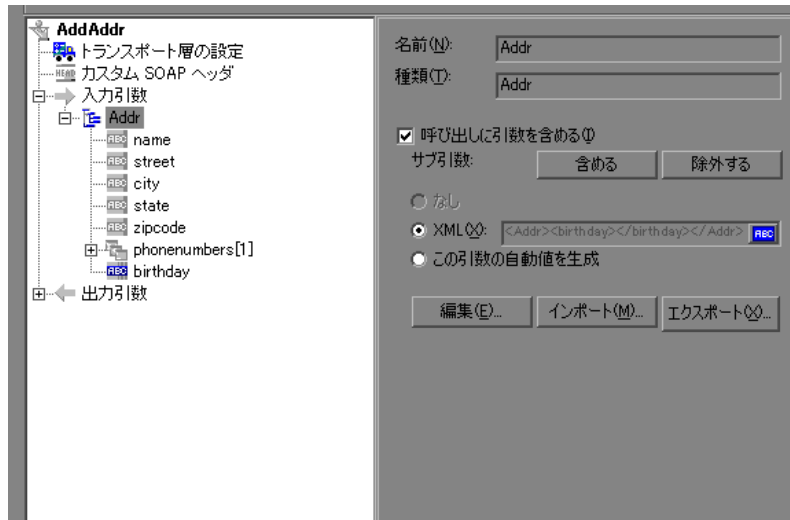
新しい XML タイプのパラメータは、ほかのパラメータ・タイプと同様に、[挿入] メニューから直接作成できます。Web サービス・タイプのスクリプトについては、Web サービス呼び出しプロパティから XML パラメータを直接作成します。

Web サービス呼び出しからの XML パラメータの作成

本項では、Web サービス・プロパティから XML パラメータを作成する方法について説明します。

Web サービス呼び出しプロパティから XML パラメータを作成するには，次の手順を実行します。

- 1 複合データ構造のルート要素を選択します。右側の表示枠には，引数の詳細が表示されます。



- 2 右側の表示枠で [XML] を選択し，[ABC] アイコンをクリックします。[パラメータの選択または作成] ダイアログ・ボックスが開きます。
- 3 [パラメータ名] ボックスにパラメータの名前を入力します。
- 4 [パラメータのタイプ] ボックスで，[XML] を選択します（選択されていない場合）。
- 5 [プロパティ] をクリックしてすぐに値セットを割り当てるか，[OK] をクリックしてダイアログ・ボックスを閉じ，後で値を割り当てます。

XML パラメータの作成 — 標準の方法

本項では，Web サービス呼び出しのプロパティを表示することなく，XML タイプのパラメータを作成する方法について説明します。これは，ほとんどのプロトコルおよびパラメータの型に対して値をパラメータ化する最も一般的な方法です。

Web サービス・スクリプトについては，前述の説明のとおり，Web サービス呼び出し内からパラメータを作成することをお勧めします。

新規 XML パラメータを作成するには、次の手順を実行します。

- 1 **[挿入]** > **[新規パラメータ]** を選択するか、スクリプト・ビューで定数値を選択し、右クリック・メニューで **[パラメータで置換]** を選択します。[パラメータの選択または作成] ダイアログ・ボックスが開きます。
- 2 **[パラメータ名]** ボックスにパラメータの名前を入力します。
- 3 **[パラメータのタイプ]** ボックスで、**[XML]** を選択します（選択されていない場合）。
- 4 **[プロパティ]** をクリックしてすぐに値セットを割り当てるか、**[OK]** をクリックしてダイアログ・ボックスを閉じ、後で値を割り当てます。

プロパティの設定方法の詳細については、255 ページ「XML パラメータのプロパティの設定」を参照してください。

値セットの定義

本項では、XML パラメータの値セットを作成する方法について説明します。

値セットとは、一連の値を含む配列のことです。**[カラムの追加]** および **[カラムの複製]** ボタンを使用すると、パラメータに対して複数の値セットを作成し、異なる反復に使用できます。

| スキーマ | 設定 1 | 設定 2 | 設定 3 |
|------------------|--------------|--------------|----------------|
| Addr | | | |
| name | John Doe | Tom Smith | Kim Jones |
| street | 2 Maple Ln. | 33 Acorn Dr. | 45 Jasper Ave. |
| city | Delray Beach | NIL | NIL |
| state | FL | AZ | MA |
| zip | 33452 | NIL | 02134 |
| zip4 | | | |
| phonenumber | | | |
| PhoneNumber [..] | | | |
| PhoneNumber[1] | NIL | NIL | NIL |

値セットを使用する場合、パラメータごとの配列要素の数は一定である必要がありません。

ある値セットには表示されるが、ほかの値セットには表示されないオプションの要素を使用できます。これにより、反復ごとに送信する値を変えることが可能になるため、一部の反復には特定の配列要素を含めて、それ以外の反復ではその配列要素を除外できます。

オプションの要素を除外するには、セルの左上角の小さな三角形をクリックして、塗りつぶされていない状態にします。

次の例では、**設定 1** と **設定 2** はオプションの要素「**name**」,「**street**」,「**state**」を使用します。**設定 3** は、「**street**」を使用しません。

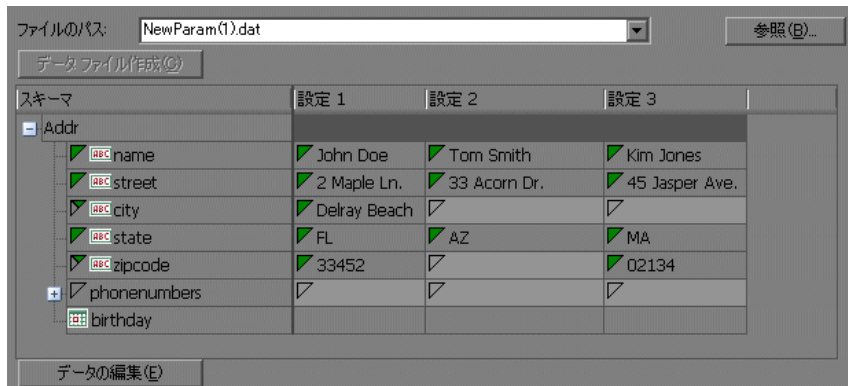
| スキーマ | 設定 1 | 設定 2 | 設定 3 |
|------------------|--------------|--------------|----------------|
| Addr | | | |
| name | John Doe | Tom Smith | Kim Jones |
| street | 2 Maple Ln. | 33 Acorn Dr. | 45 Jasper Ave. |
| city | Delray Beach | | |
| state | FL | AZ | MA |
| zip | 33452 | | 02134 |
| zip4 | | | |
| phonenumbers | | | |
| PhoneNumber [..] | | | |
| PhoneNumber[1] | | | |

値の変更の詳細については、『第 1 巻 – VuGen の使用』の を参照してください。

パラメータ要素の値を設定するには、次の手順を実行します。

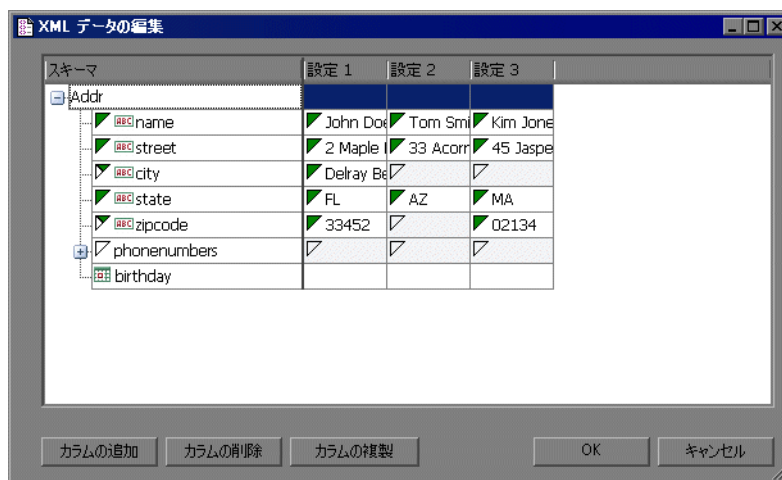
1 [パラメータのプロパティ] を表示します。

[パラメータのプロパティ] ダイアログ・ボックスが開いていない場合は、[**仮想ユーザ**] > [**パラメータ リスト**] を選択し、必要なパラメータを選択します。ダイアログ・ボックスに、パラメータ値が読み取り専用で表示されます。



2 [XML データの編集] ボックスを開きます。

[データの編集] ボタンをクリックして [XML データの編集] ダイアログ・ボックスを開きます。



3 XML パラメータに対する値のセットを定義します。

[設定] カラムに、スキーマに対応する値を挿入します。

行に [なし] と表示されている場合、その要素は nillable であることを意味しています。nillable 要素の値を含めるには、通常どおりに値を入力します。

値を nil としてマークするには、マークする [なし] アイコンをクリックします。これで、要素に割り当てた値が消去されます。次の例では、「city」要素は nillable ですが、設定 2 および設定 3 のみで nil としてマークされており、設定 1 ではマークされていません。

| スキーマ | 設定 1 | 設定 2 | 設定 3 |
|-------------------|--------------|--------------|----------------|
| Addr | | | |
| name | John Doe | Tom Smith | Kim Jones |
| street | 2 Maple Ln. | 33 Acorn Dr. | 45 Jasper Ave. |
| city | Delray Beach | NIL | NIL |
| state | FL | AZ | MA |
| zip | 33452 | NIL | 02134 |
| zip4 | | | |
| phonenumbers | | | |
| PhoneNumber [...] | | | |
| PhoneNumber[1] | NIL | NIL | NIL |

1 追加の値セットを作成します。

追加の値セットを挿入するには、[**カラム追加**] をクリックし、新しいカラムに追加する値のセットを挿入します。既存の値セットをコピーするには、コピーする値セットで行を選択し、[**カラムの複製**] をクリックします。

2 配列をコピーします。

配列要素とその子を複製するには、親ノードを選択して右クリック・メニューで [**配列要素の複製**] を選択します。

| スキーマ | 設定 1 | 設定 2 | 設定 3 |
|-------------------|----------|----------|----------|
| phone-numbers | | | |
| PhoneNumber [...] | | | |
| PhoneNumber[1] | [複製] | [複製] | [複製] |
| description | Home | Home | Home |
| phone-number | 888-8888 | 111-1111 | 444-4444 |
| PhoneNumber[2] | [複製] | [] | [複製] |
| description | Office | Office | Office |
| phone-number | 666-6666 | 222-2222 | 999-9999 |
| PhoneNumber[3] | [複製] | [] | [] |
| description | Mobile | Mobile | Mobile |
| phone-number | 333-3333 | 123-4567 | |

3 <any> 要素を処理します。

「任意」タイプの要素については、[スキーマ] カラムで [**<any>**] を右クリックし、利用可能なオプションの 1 つを選択します。これらのオプションはカーソルの位置によって変わります。

- ▶ [**配列要素の追加**]：ルート要素の下にサブ要素を追加します。
- ▶ [**子の追加**]：選択した要素にサブ要素を挿入します。
- ▶ [**兄弟の追加**]：選択した要素と同じレベルにサブ要素を追加します。
- ▶ [**XML の読み込み**]：XML ファイルから要素の値をロードします。
- ▶ [**XML の保存**]：XML ファイルとして配列を保存します。
- ▶ [**XML のコピー**]：選択した要素の完全な XML をクリップボードにコピーします。

各配列要素に意味のある名前を付けるには、**[名前の変更]** テキストをクリックします。



4 unnecessary columns are removed.

値セットを削除するには、該当する値セットを選択して **[カラムの削除]** をクリックします。

5 変更を保存します。

[適用] をクリックして変更内容を保存し、**[パラメータのプロパティ]** ダイアログ・ボックスでビューを更新します。

割り当て方法の設定

割り当て方法とは、使用する値セットおよびその使用方法を意味します。たとえば、反復ごとに新しい値セットを使用して、値セットを順次またはランダムに使用するように Vuser に指示できます。詳細については、252 ページ「ファイル/テーブル/XML タイプのパラメータに対するデータの割り当て方法と更新方法」を参照してください。

割り当て方法を定義するには、次の手順を実行します。

- 1 **[パラメータのプロパティ]** を開き、パラメータを選択します。
- 2 データ割り当て方法を定義します。

[次の値を選択] リストで、データの割り当て方法を選択して、Vuser が Vuser スクリプト実行時にファイル・データをどのように選択するかを指定します。オプションには、**[Sequential]** (順次)、**[Random]** (ランダム)、または **[Unique]** (一意) があります。詳細については、250 ページ「ファイル・タイプまたはテーブル・タイプのパラメータにデータを割り当てる方法の選択」を参照してください。

3 パラメータの更新オプションを選択します。

[**更新する対象**] リストから更新オプションを選択します。[**Each iteration**] (反復ごと), [**Each occurrence**] (出現ごと), [**Once**] (一度) から選択します。詳細については、252 ページ「ファイル/テーブル/XML タイプのパラメータに対するデータの割り当て方法と更新方法」を参照してください。

4 データの割り当て方法として [**Unique**] (一意) を選択した場合, [**使用可能な値の終了後**] オプションおよび [**Controller で Vuser 値を割り当てる**] オプションが有効になります。

▶ [**使用可能な値の終了後**] : 一意の値がなくなった場合の処理として, [**Abort Vuser**] (Vuser の中止), [**Continue in a cyclic manner**] (循環して継続), [**Continue with last value**] (最後の値で継続) のいずれかを指定します。

▶ [**Controller で Vuser の値を割り当てる**] (LoadRunner のユーザのみ) : Vuser に手作業でデータ・ブロックを割り当てるかどうか指定します。Controller にブロック・サイズを自動的に割り当てさせるか, 必要な値の数を指定します。[**ブロックサイズを自動的に割り当てる**] または [**仮想ユーザに X 件の値を割り当てる**] を選択します。後者の場合には, 割り当てる値の数を指定します。

これを追跡するには, [実行環境設定] の [ログ] で, [**拡張ログ**] > [**パラメータ置換**] オプションを有効にします。十分なデータがない場合は, Vuser ログに「**No more unique values for this parameter in table <テーブル名>**」 (<テーブル名>テーブルには, このパラメータのための一意の値がこれ以上ありません) という警告メッセージが表示されます。

5 [パラメータのプロパティ] ダイアログ・ボックスで, [**終了**] をクリックします。




入力引数のリストがパラメータ名で置き換えられ, ABC ボタンはテーブル・アイコンに置き換えられます。このボタンをクリックして, パラメータ・プロパティを編集したりパラメータを非パラメータ化したりできます。

XML パラメータのプロパティの変更

パラメータの値セットを変更する必要がある場合、Web サービスの [ステップのプロパティ] タブで変更できます。

XML パラメータのプロパティを変更するには、次の手順を実行します。

- 1 Web サービス・スクリプトのツリー・ビューで、[**プロパティの設定**] タブをクリックします。
- 2 [**入力引数**] の下で、XML パラメータを選択します。右側の表示枠には、パラメータの詳細が表示されます。
- 3 XML パラメータのプロパティを変更するには、[**XML**] ボックスの隣にある  テーブル・アイコン・ボタンをクリックし、[**パラメータ プロパティ**] を選択します。
- 4 257 ページ「値セットの定義」で説明した手順に従ってパラメータのプロパティを修正します。

第 15 章

パラメータのプロパティの設定

パラメータは、置き換えの対象となる情報の種類に基づいて定義されます。

本章の内容

- ▶ パラメータのプロパティの設定について (266 ページ)
- ▶ 内部データ・パラメータ・タイプのプロパティの設定 (267 ページ)
- ▶ ユーザ定義関数のプロパティの設定 (276 ページ)
- ▶ パラメータ形式のカスタマイズ (277 ページ)
- ▶ 更新方法の選択 (278 ページ)
- ▶ ファイル・タイプのパラメータのシミュレーション (279 ページ)
- ▶ ファイル・パラメータのシミュレータの使用方法 (281 ページ)

パラメータのプロパティの設定について

パラメータのプロパティを定義するときは、パラメータのデータ・ソースを指定します。

次のタイプのデータ・ソースに対してプロパティを定義します。

| データ・タイプ | データ・ソース |
|-----------------|---|
| 内部データ・パラメータ・タイプ | Vuser によって内部で生成されるデータ。Date/Time (日時), Group Name (グループ名), Iteration Number (反復回数), Load Generator Name (Load Generator 名), Random Number (乱数), Unique Number (一意の数), および Vuser ID |
| ユーザ定義関数 | 外部の DLL の関数を使用して生成されたデータ。 |
| ファイルとテーブル | 既存のファイルか、VuGen または MS Query で作成したファイルに含まれるデータ。 |

本章では、プロパティを「内部データ」および「ユーザ定義関数」のパラメータに割り当てる方法について説明します。

テーブルまたはファイル・タイプ・パラメータのプロパティの定義方法の詳細については、第 14 章「ファイル、テーブル、および XML パラメータ・タイプ」を参照してください。

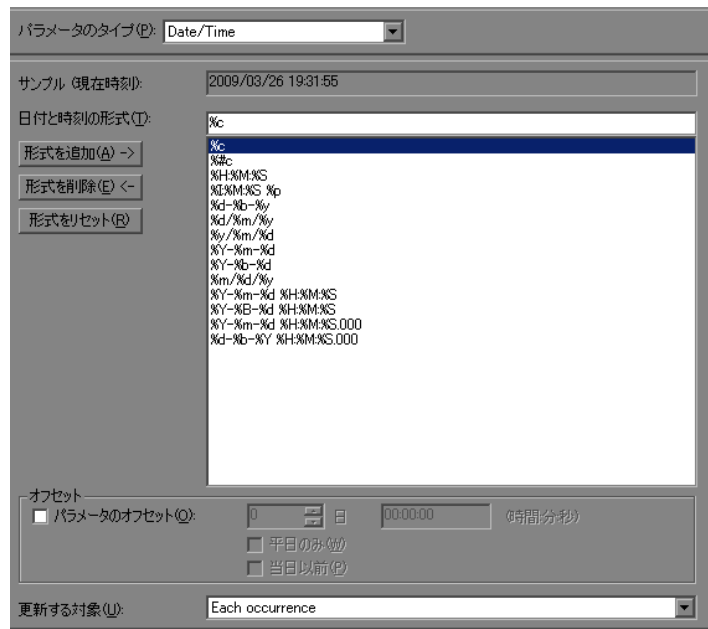
内部データ・パラメータ・タイプのプロパティの設定

本項では、Vuser によって内部で生成されるデータのプロパティの設定方法について説明します。内部データには次のようなデータが含まれます。

- ▶ Date/Time (日時)
- ▶ Group Name (グループ名)
- ▶ Iteration Number (反復回数)
- ▶ Load Generator Name (Load Generator 名)
- ▶ Random Number (乱数)
- ▶ Unique Number (一意の数)
- ▶ Vuser ID

Date/Time (日時)

[Date/Time] (日時) を選ぶと、パラメータは現在の日付 / 時刻で置き換えられます。日時の形式は、形式リストから選択するか、独自の形式を指定します。指定した形式は、スクリプトに記録されている日時の形式と一致しなければなりません。



VuGen では日時パラメータのオフセットを設定できます。たとえば、翌月の日付をテストする場合は、日付オフセットを 30 日に設定します。未来の時刻でのアプリケーションのテストを行う場合は、時刻オフセットを指定します。未来への前進オフセット（標準設定）、または過去への後退オフセットを指定できます。また、平日のみの日付値を使用し、土曜日と日曜日を除外するように VuGen に指示することもできます。

次の表に日付と時刻の記号の意味を示します。

| 記号 | 説明 |
|----|---------------------------|
| c | 年月日（数字）と時刻 |
| #c | 年月日（文字列）と時刻 |
| H | 時間（24 時間表示） |
| I | 時間（12 時間表示） |
| M | 分 |
| S | 秒 |
| p | 午前（AM）または午後（PM） |
| d | 曜日 |
| m | 月（01 ~ 12 までの数字） |
| b | 月（省略した文字列。たとえば Dec） |
| B | 月（省略しない文字列。たとえば December） |
| y | 年（短い形式。たとえば 03） |
| Y | 年（長い形式。たとえば 2003） |

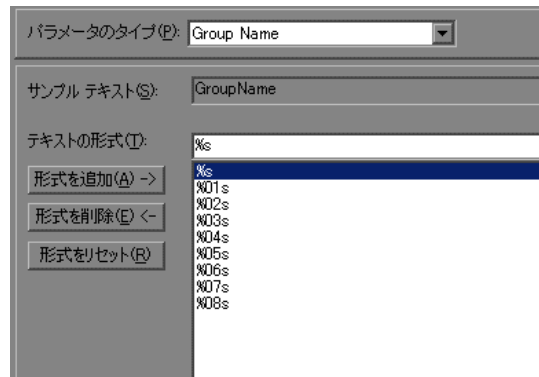
日付と時刻のパラメータのプロパティを設定するには、次の手順を実行します。

- 1 既存の日付と時刻の形式から選択するか、新しい形式を作成します。VuGen でどのように表示されるかは [サンプル（現在時刻）] ボックスで確認できます。パラメータ形式のカスタマイズの詳細については、277 ページ「パラメータ形式のカスタマイズ」を参照してください。

- 2 日付と時刻のオフセットを設定するには、[パラメータのオフセット] を選択して、日付および時刻の値に対する必要なオフセットを指定します。
週末を除く平日のみを使用するよう VuGen に指示するには、[平日のみ] を選択します。過去の日付をテストするために負のオフセットを指定するには、[当日以前] を選択します。
- 3 [Each occurrence] (出現ごと)、[Each iteration] (反復ごと)、[Once] (一度) のどれかを選択し、Vuser にパラメータ値の更新方法を指定します。詳細については、278 ページ「更新方法の選択」を参照してください。
- 4 [閉じる] をクリックして設定を適用し、[パラメータのプロパティ] ダイアログ・ボックスを閉じます。

Group Name (グループ名)

[Group Name] (グループ名) を選ぶと、パラメータは Vuser グループの名前で置き換えられます。シナリオ作成時に、Vuser グループの名前を指定します。VuGen でスクリプトを実行するとき、Vuser グループ名は常に「None」です。

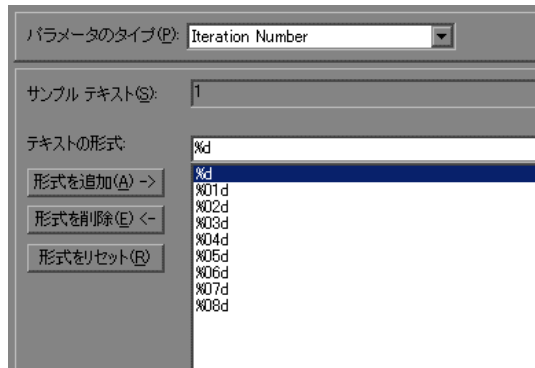


グループ名のパラメータのプロパティを設定するには、次の手順を実行します。

- 1 利用可能な既存の形式の中から 1 つを選択するか、新しい形式を作成します。形式を選択することによって、パラメータの文字列の長さを指定します。詳細については、277 ページ「パラメータ形式のカスタマイズ」を参照してください。
- 2 [閉じる] をクリックして設定を適用し、[パラメータのプロパティ] ダイアログ・ボックスを閉じます。

Iteration Number（反復回数）

[繰り返しの数字] を選ぶと、パラメータは現在の反復回数で置き換えられます。



反復回数のパラメータのプロパティを設定するには、次の手順を実行します。

- 1 利用可能な既存の形式の中から 1 つを選択するか、新しい形式を作成します。形式を選択することによって、パラメータの文字列の長さを指定します。詳細については、277 ページ「パラメータ形式のカスタマイズ」を参照してください。
- 2 [閉じる] をクリックして設定を適用し、[パラメータのプロパティ] ダイアログ・ボックスを閉じます。

Load Generator Name（Load Generator 名）

[Load Generator Name]（Load Generator 名）を選ぶと、パラメータが Vuser スクリプトの Load Generator の名前で置き換えられます。Load Generator とは、Vuser を実行しているコンピュータのことです。



Load Generator 名タイプのパラメータのプロパティを設定するには、次の手順を実行します。

- 1 利用可能な既存の形式の中から 1 つを選択するか、新しい形式を作成します。形式を選択することによって、パラメータの文字列の長さを指定します。詳細については、277 ページ「パラメータ形式のカスタマイズ」を参照してください。
- 2 **[閉じる]** をクリックして設定を適用し、[パラメータのプロパティ] ダイアログ・ボックスを閉じます。

Random Number (乱数)

[Random Number] (乱数) を選ぶと、パラメータは乱数で置き換えられます。最小値と最大値を指定することによって、乱数の範囲を設定します。

乱数を使用すれば、特定の範囲の値を使ったときのシステムの動作をテストできます。たとえば、従業員 ID 番号の範囲が 1 から 1,000 の場合に、50 人の従業員についてクエリを実行するには、50 個の Vuser を作成し、最小値を 1 にし、最大値を 1,000 にします。各 Vuser は 1 から 1,000 までの範囲の乱数を 1 つ受け取ります。

The screenshot shows a dialog box for configuring a parameter of type "Random Number". The "Parameter Type (P):" dropdown is set to "Random Number". The "Random Number Range" section has "Minimum (Q):" set to 1 and "Maximum (Q):" set to 100. The "Sample Value" is set to 1. The "Number Format (Q):" list contains several entries, with "%lu" selected. The "Update Target (U):" dropdown is set to "Each occurrence".

| Parameter Type (P): | Random Number |
|---------------------|--|
| 乱数の範囲: | 最小値 (Q): 1 |
| | 最大値 (Q): 100 |
| サンプル値: | 1 |
| 数字の形式 (Q): | %lu %01 lu %02 lu %03 lu %04 lu %05 lu %06 lu %07 lu %08 lu %lu |
| 更新する対象 (U): | Each occurrence |

乱数のパラメータのプロパティを設定するには、次の手順を実行します。

- 1 使用可能なパラメータ値のセットを規定する範囲を指定します。乱数の範囲の最小値と最大値を指定します。
- 2 **[数字の形式]** を選択することによって、乱数の長さを指定します。1 桁の数字には **%01lu** (または **%lu**)、2 桁の数字には **%02lu** というように指定します。VuGen でどのように表示されるかは **[サンプル値]** ボックスで確認できます。
- 3 **[Each occurrence]** (出現ごと)、**[Each iteration]** (反復ごと)、**[Once]** (一度) のどれかを選択し、Vuser にパラメータ値の更新方法を指定します。詳細については、278 ページ「更新方法の選択」を参照してください。
- 4 **[閉じる]** をクリックして設定を適用し、**[パラメータのプロパティ]** ダイアログ・ボックスを閉じます。

Unique Number (一意の数)

[Unique Number] (一意の数) を選ぶと、パラメータは一意の数字で置き換えられます。

一意の数のパラメータを作成するときには、開始値とブロック・サイズを指定します。ブロック・サイズは、各 Vuser に割り当てられる数値の範囲を示します。各 Vuser はその範囲の開始値で始まり、反復ごとにパラメータの値が大きくなります。たとえば開始値として **1** を、ブロック・サイズとして **500** を設定した場合、最初の Vuser は値 **1** を、次の Vuser は値 **501** を、それぞれの最初の反復で使います。

一意の数を構成する桁数とブロック・サイズによって反復の回数と Vuser の数が決まります。たとえば、数の桁数が 5 桁を上限としていてブロック・サイズが 500 の場合、使用できる数値は 100,000 個 (0 ~ 99,999) だけとなります。したがって、1 Vuser あたり 500 回の反復を実行する場合に実行できる Vuser 数は 200 のみとなります。

The image shows a configuration window for a parameter of type 'Unique Number'. The 'Range of Numbers' section has 'Start (S):' set to 1 and 'Block Size (B):' set to 100. The 'Sample Value' is 1. The 'Number Format (N):' list includes %01d through %08d and %d, with %01d selected. The 'Update Target (U):' is 'Each iteration' and the 'End of Available Values (H):' is 'Continue with last value'.

| | |
|----------------|--|
| パラメータのタイプ(P): | Unique Number |
| 数字の範囲: | 開始(S): 1 |
| | 仮想ユーザーごとのブロックサイズ(B): 100 |
| サンプル値: | 1 |
| 数字の形式(N): | %01d %02d %03d %04d %05d %06d %07d %08d %d |
| 更新する対象(U): | Each iteration |
| 使用可能な値の終了後(H): | Continue with last value |

また、ブロック内に一意の数がなくなったときにどのようなアクションを起こすかを、[**Abort Vuser**] (Vuser の中止), [**Continue in a cyclic manner**] (循環して継続), [**Continue with last value**] (最後の値で継続) から指定することができます。

一意の数を使えば、パラメータに対して使用可能なすべての値についてシステムの動作を確認することができます。たとえば、ID 番号が 100 から 199 の範囲にある全従業員についてクエリを実行するには、100 個の Vuser を作成し、開始番号を 100 に設定しブロック・サイズを 100 に設定します。各 Vuser には、100 から 199 までの一意の数字が ID として割り当てられます。

注： [Unique Number] (一意の数) パラメータ・タイプではインスタンスが 1 つ作成されます。複数のパラメータを定義してそれぞれに [Unique Number] (一意の数) パラメータ・タイプを指定しても、値が互いに重なることはありません。たとえば、2 つのパラメータで、5 回の反復を指定し、ブロック・サイズを 100 とすると、最初のグループの Vuser は 1, 101, 201, 301, 401 を使用し、2 つ目のパラメータを使用する次のグループは 501, 601, 701, 801, 901 を使用します。

一意のパラメータの定義後は、同じパラメータ・ファイルを指定することによって、ほかのスクリプトでも一意のパラメータを使用できます。このパラメータは、元のパラメータに割り当てられたすべてのプロパティを保持します。

一意の数のパラメータのプロパティを設定するには、次の手順を実行します。

- 1 開始値とブロック・サイズを指定します。たとえば、1 から始め、500 までの数を使用したい場合、[開始] ボックスに 1 と入力し、[仮想ユーザごとのブロック サイズ] ボックスに 500 と入力します。
- 2 [数字の形式] を選択することによって、一意の数値の長さを指定します。1 桁の数字には %01d (または %d)、2 桁の数字には %02d というように指定します。VuGen でどのように表示されるかは [サンプル値] ボックスで確認できます。
- 3 [Each occurrence] (出現ごと)、[Each iteration] (反復ごと)、[Once] (一度) のどれかを選択し、Vuser にパラメータ値の更新方法を指定します。詳細については、278 ページ「更新方法の選択」を参照してください。
- 4 一意の値がなくなった場合には、[使用可能な値の終了後] ボックスで、[Abort Vuser] (Vuser の中止)、[Continue in a cyclic manner] (循環して継続)、[Continue with last value] (最後の値で継続) から指定することができます。

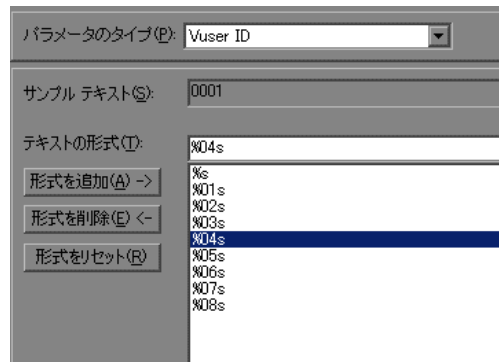
注： (LoadRunner ユーザの方へ) Controller でシナリオのスケジュールを設定する際、[使用可能な値の終了後] オプションは、スケジュール・ビルダの [継続時間] タブの [実行時間：(時間：分：秒)] オプションに対してのみ適用されます。[完了するまで実行する] オプションに対しては無視されるので注意してください。

- 5 **[閉じる]** をクリックして設定を適用し、[パラメータのプロパティ] ダイアログ・ボックスを閉じます。

Vuser ID

注： このパラメータ・タイプは主に LoadRunner のユーザが対象です。

[仮想ユーザ ID] を選ぶと、パラメータは、シナリオの実行中に Controller が Vuser に割り当てる ID 番号で置き換えられます。VuGen からスクリプトを実行する場合、Vuser ID は常に -1 です。



注： この ID は、[仮想ユーザ] ウィンドウに表示される ID 番号ではなく、実行時に生成される一意の ID 番号です。

Vuser ID タイプのパラメータのプロパティを設定するには、次の手順を実行します。

- 1 利用可能な既存の形式の中から 1 つを選択するか、新しい形式を作成します。形式を選択することによって、パラメータの文字列の長さや構造を指定します。詳細については、277 ページ「パラメータ形式のカスタマイズ」を参照してください。
- 2 **[閉じる]** をクリックして設定を適用し、[パラメータのプロパティ] ダイアログ・ボックスを閉じます。

ユーザ定義関数のプロパティの設定

[パラメータのプロパティ] ダイアログ・ボックスで, [パラメータのタイプ] リストから [User Defined Function] を選択します。

The screenshot shows a dialog box titled 'パラメータのプロパティ' (Parameter Properties). At the top, there is a dropdown menu labeled 'パラメータのタイプ (P):' (Parameter Type) with 'User Defined Function' selected. Below this is a text box labeled '関数名 (F):' (Function Name) containing 'MyFunctionName'. A section titled 'ライブラリ名' (Library Name) contains four text boxes: 'WinNT ライブラリ (W):' (WinNT Library) with 'MyDll.dll', 'Solaris ライブラリ (S):', 'HP/UX ライブラリ (H):', and 'AIX ライブラリ (A):'. To the right of these boxes is a button labeled '参照...' (Reference...). At the bottom of the dialog, there is a dropdown menu labeled '更新する対象 (U):' (Update Target) with 'Each occurrence' selected.

ユーザ定義関数のプロパティを設定するには, 次の手順を実行します。

- 1 [関数名] ボックスに関数名を指定します。DLL ファイルに作成した関数名をそのまま使用します。
- 2 [ライブラリ名] セクションの, 該当する [ライブラリ] ボックスにライブラリを指定します。必要に応じて, [参照] ボタンをクリックしてファイルを検索します。
- 3 値の更新方法を選択します。ユーザ定義関数の更新方法の詳細については, 278 ページ「更新方法の選択」を参照してください。

パラメータ形式のカスタマイズ

ほとんどのデータ・タイプでは、既存の形式を選択するか新しい形式を作成することによって、パラメータの形式をカスタマイズできます。

注：パラメータの形式は、記録された値と一致させるようにします。パラメータの形式が記録されている元の値と異なると、スクリプトを正しく実行できない場合があります。

パラメータの形式は、生成されるパラメータ文字列の長さや構造を規定します。生成されるパラメータ文字列は、実際のパラメータ値と、そのパラメータに付随するテキストから成ります。たとえば、%05s という形式（小数点の左に 5 桁）を指定した場合には、Vuser ID の 5 という数字は 1 桁の数字ですが、ほかの桁を 0 で埋めて 00005 と表示されます。数値を空白で埋めて長さを揃えるには、空白を使って揃える桁数をゼロ（0）を先頭に付けずに指定します。たとえば、%4s と指定した場合、Vuser ID の先頭に空白を追加することによってパラメータ文字列が 4 桁に揃えられます。

実際のパラメータ値の前後にテキスト文字列を指定できます。

たとえば、「Vuser No: %03s」という形式を指定した場合、Vuser ID の 1 番は「Vuser No: 001」と表示されます。

Date/Time（日時）、Group Name（グループ名）、Iteration Number（反復回数）、Load Generator Name（Load Generator 名）、Vuser ID の各パラメータ・タイプについて、形式を追加および削除することができます。

パラメータ・タイプに形式を追加するには、次の手順を実行します。

- 1 [パラメータのプロパティ] ダイアログ・ボックスで、形式を指定する対象となるパラメータ・タイプを選択します。
- 2 エディット・ボックス内に形式指定記号を入力し、**[形式を追加]** をクリックします。

注：形式をリストに追加すると、VuGen によって、形式が Vuser とともに保存され、以後使用できるようにします。

形式を削除するには、次の手順を実行します。

[パラメータのプロパティ] ダイアログ・ボックスで、既存の形式をリストから選択し、[形式を削除] をクリックします。

一連の元の形式を復元するには、次の手順を実行します。

[形式をリセット] をクリックします。

更新方法の選択

パラメータのタイプのいくつかを使用するときは、パラメータの値を更新する方法を指定できます。更新方法を設定するには、[更新する対象] ドロップダウン・リストから更新方法を選択します。次の更新方法があります。

- ▶ Each Occurrence (出現ごと)
- ▶ Each Iteration (反復ごと)
- ▶ Once (一度)

Each Occurrence (出現ごと)

「出現ごと」の方法を選択すると、Vuser はパラメータが出現するたびに新しい値を使います。パラメータを使っているステートメントどうしが関連していない場合に有用です。たとえば、ランダムなデータを使用する場合には、パラメータが現れるたびに異なる値を使うと有効です。

Each Iteration (反復ごと)

「反復ごと」の方法を選択すると、Vuser はスクリプトで反復ごとに新しい値を使います。そのパラメータが 1 つのスクリプトに複数回現れる場合、Vuser は 1 回の反復でそのパラメータが現れるたびに同じ値を使います。これは、パラメータを使うステートメントどうしが関連している場合に有用です。

注： 独自に反復カウントを使って反復処理を行うアクション・ブロックを作成した場合、そこにパラメータが含まれていて、VuGen に反復ごとに値を更新するよう指定した場合でも、VuGen が対象とする反復は当該ブロックの反復ではなく、スクリプト全体のグローバルな反復なので、ブロックの反復ではパラメータが更新されません。アクション・ブロックの詳細については、415 ページ「アクション・ブロックの作成」を参照してください。

Once (一度)

「一度」の方法を選択すると、Vuser はシナリオの実行時に、一度だけパラメータの値を更新します。Vuser はすべての反復でパラメータのすべての出現箇所と同じパラメータ値を使います。このタイプは日付と時刻を使う場合に有用です。

ファイル・タイプのパラメータのシミュレーション

ファイル・タイプのパラメータを作成した後に、ファイル・パラメータのシミュレータを使用して実際のシナリオでパラメータ置換をシミュレートできます。これにより、Controller でスクリプトを実行する前に、誤ったパラメータを修正できます。ファイル・パラメータのシミュレータは、LoadRunner ユーザにのみ関連しています。

注：すべての種類のパラメータ置換をシミュレートできるわけではありません。[次の行を選択] に [Same line as...] を選択したり、または [更新する対象] に [Each occurrence] (出現ごと) を選択したりすると、ファイル・パラメータのシミュレータは起動しません。

ファイル・パラメータのシミュレータを実行するには、次の手順を実行します。

- 1 [パラメータ リスト] ダイアログ・ボックスで、[パラメータをシミュレートする] をクリックします。[パラメータ シミュレーション] ダイアログ・ボックスが表示されます。



- 2 [仮想ユーザ数] ボックスからシミュレーションで実行する Vuser の数を選択します。
- 3 シナリオ選択モードを選択します。
 - ▶ [完了するまで実行する] を選択する場合、[実行する反復の回数] ボックスで実行する反復の回数を選択するか、または実行環境の設定から数値を取得します。
 - ▶ [無限に実行する] を選択する場合は、表示する反復数を定義する必要があります。実行環境の設定の反復回数は無視されます。選択する回数によって、各 Vuser の値の分布は変更されません。この回数は表示のみを対象としています。

注：

- ▶ **[無限に実行する]** は Controller のスケジューラの **[実スケジュール]** に準拠しています。
 - ▶ **[パラメータ リスト]** ダイアログの **[次の行を選択]** で **[Unique]** (一意) を選択した場合、各 Vuser には、シミュレータが (その Vuser のために) 置き換える値を取得する一意の範囲の行が割り当てられます。この設定を使う場合、**[Controller]** セクションの Vuser の割り当ての値には、標準設定で **[ブロック サイズを自動的に割り当てる]** が選択されています。この場合、シミュレーションを実行すると、範囲割り当てはシナリオ実行モードの選択に従って実施されます。標準設定の選択内容を **[仮想ユーザに X 件の値を割り当てる]** に変更すると、Vuser は指定した数値を割り当てられ、シナリオ実行モードの選択は無視されます。
-

4 **[シミュレート]** をクリックします。

注：ファイル・パラメータのシミュレータでは、256 回の反復および 256 の Vuser をシミュレートできます。

ファイル・パラメータのシミュレータの使用方法

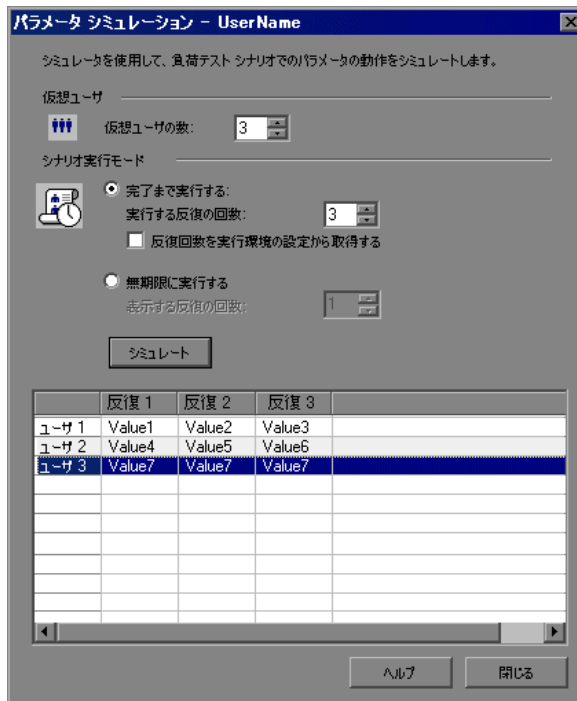
次の項ではファイル・パラメータのシミュレータの動作について説明し、シナリオ実行モードの設定の違いについて示します。

次の例では、**[パラメータ リスト]** ダイアログ・ボックスで次のように設定します。

- ▶ **新規パラメータの値** : Value1 から Value7
- ▶ **[次の行を選択]** : [Unique] (一意)
- ▶ **[使用可能な値の終了後]** : [Continue with last value] (最後の値で継続)
- ▶ **[Controller で Vuser の値を割り当てる]** : [ブロック サイズを自動的に割り当てる]

[シナリオ選択モード] : [完了まで実行する]

次の例では、ユーザは 3 つの Vuser を選択し、シナリオ実行モードを **[完了まで実行する]** に設定して 3 回の反復を選択しました。



シナリオ実行モードを **[完了まで実行する]** に設定すると、各 Vuser が受け取る行の数は反復の数と同じになります。テーブルに十分な行がなくなると、範囲割り当ては停止します。

シミュレーションが実行されると、最初の Vuser は最初の 3 つの値を取得します（これが反復数です）。2 番目の Vuser は次の 3 つの値を取得します。3 番目の Vuser は、最初の反復において残りの値を取得します。残りの反復については、[パラメータ リスト] ダイアログ・ボックスの **[使用可能な値の終了後]** オプションは **[Continue with last value]**（最後の値で継続）に設定されているため、3 番目の Vuser は同じ値で続行します。

4 番目の Vuser は失敗します。

[シナリオ選択モード] : [無限に実行する]

次の例では、ユーザは 3 つの Vuser を選択し、シナリオ実行モードを [無限に実行する] に設定して 3 回の反復が表示されるように選択しています。



シナリオ実行モードを [無限に実行する] に設定すると、各 Vuser の割り当て範囲は、.dat ファイル内のセルの数を Vuser 数で割ることによって計算されます。このシナリオでは、 $7/3=2$ です (シミュレータは、最も近い最小の整数を取得します)。

シミュレーションを実行すると、最初の Vuser は Value1 と Value2 を取得します。2 番目の Vuser は Value3 と Value4 を取得し、3 番目の Vuser は Value5 と Value6 を取得します。Vuser は 3 つしかないため、Value7 は振り分けられません。

注：テーブルの最初のカラムのセル上にマウスを移動すると、ツールチップに、その **Vuser** に割り当てられ値に関する情報が表示されます。

値が割り当てられていないセル上にマウスを移動すると、ツールチップに、値が割り当てられなかった理由が表示されます。

適切な値が割り当てられていない場合は、ツールチップは表示されません。

第 3 部

記録オプション

第 16 章

選択したプロトコルの記録オプション

本章では、選択したプロトコルの記録オプションについて説明します。

本章の内容

- ▶ EJB 記録オプション (288 ページ)
- ▶ RDP 記録オプション (289 ページ)
- ▶ Citrix 記録オプションについて (294 ページ)
- ▶ Citrix 記録オプションの設定 (300 ページ)
- ▶ データベース記録オプション (301 ページ)
- ▶ データベースの詳細記録オプション (302 ページ)
- ▶ AMF 記録モードの設定 (305 ページ)
- ▶ AMF コード生成オプション (309 ページ)
- ▶ Flex コード生成オプション (310 ページ)
- ▶ Microsoft .NET 記録オプション (312 ページ)
- ▶ WCF 記録オプション (318 ページ)
- ▶ RTE 設定オプション (320 ページ)
- ▶ RTE 記録オプション (321 ページ)
- ▶ SAPGUI 記録オプション (325 ページ)
- ▶ SAP-Web 記録オプション (328 ページ)
- ▶ 関連記録オプションの設定 (329 ページ)
- ▶ Web, ワイヤレス, Oracle NCA 記録オプション (331 ページ)

EJB 記録オプション

EJB Vuser の [Classpath] および [コード生成] の記録オプションを設定できません。本項では、[コード生成] 記録オプションについて説明します。[Classpath] 記録オプションの詳細については、第18章「Java 記録オプション」を参照してください。

[EJB オプション: コード生成] オプションを使用して、自動トランザクションと値チェックのセクションでプロパティを設定できます。また、初期化メソッドの保管場所も指定できます。

EJB コード生成記録オプションを設定するには、次の手順を実行します。

- 1 [記録開始] ダイアログ・ボックスの [オプション] をクリックします。コード生成オプションを編集するには [記録オプション] の [EJB オプション: コード生成オプション] ノードを選択します。



- 2 [自動トランザクション] オプションを有効にし、すべての EJB メソッドをトランザクションとして扱うように自動的にマークします。これで、すべてのメソッドを `lr.start_transaction` と `lr.end_transaction` 関数で囲むこととなります。標準設定では、このオプションは有効になっています（つまり、値が「true」に設定されています）。
- 3 [値チェックの挿入] オプションを有効にし、`lr.value_check` 関数を各 EJB メソッドの後に自動挿入します。この関数は、プリミティブな値および文字列で返される期待値をチェックします。
- 4 [EJB 初期化メソッド] を選択します。このメソッドには、EJB/JNDI 初期化プロパティが書き込まれます。利用可能なメソッドは、`init`（標準設定）と `action` です。

RDP 記録オプション

スクリプトを記録する前に、RDP の動作の記録方法とスクリプトの生成方法を設定できます。それには記録オプションを設定します。

次の項目について記録オプションの設定が可能です。

- ▶ 289 ページ「RDP ログイン・オプション」
- ▶ 290 ページ「RDP の基本コードの生成オプション」
- ▶ 292 ページ「RDP の高度なコード生成オプション」
- ▶ 293 ページ「RDP Agent Code Generation Options」

RDP ログイン・オプション

RDP ログイン・オプションでは、ターミナル・サービスのクライアント・セッションの実行方法を指定します。

RDP: ログイン

RDP セッション オプション:

| プロパティ | 値 |
|---|---|
| <input checked="" type="radio"/> RDP クライアント アプリケーションの実行 | |
| <input type="radio"/> カスタム接続ファイルを使用 | |
| ファイル名 | |
| <input checked="" type="radio"/> 標準設定接続ファイルを使用 | |
| | |
| | |
| | |
| | |
| | |
| | |

RDP クライアント アプリケーションの実行
ターミナル サービス クライアントを実行します。

- ▶ **[RDP クライアント アプリケーションの実行]**：ターミナル・サービスのクライアントを実行します。
- ▶ **[カスタム接続ファイルを使用]**：既存の接続ファイルを使用して、ターミナル・サービスのクライアントを実行します。このファイルには ***.rdp** 拡張子が必要です。ファイル・システムおよびネットワークのファイルを参照できます。

- ▶ **[標準設定接続ファイルを使用]**：ドキュメント・ディレクトリにある **Default.rdp** ファイルを使用して接続します。

RDP の基本コードの生成オプション

基本コードの生成オプションは、VuGen がスクリプトを作成する方法、すなわち詳細レベル、トリガ、タイムアウトを制御します。



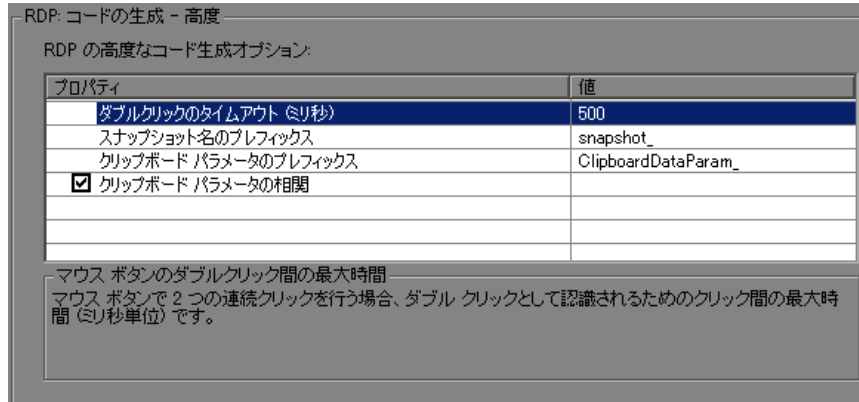
次のオプションを設定できます。

- ▶ **[スクリプト生成レベル]**：スクリプト生成時に使用されるスクリプト・レベルおよび API 関数タイプ。指定できるレベルは **[高]**、**[低]**、**[未処理]** です。
 - ▶ **[高]**：高レベルのスクリプトを生成します。キーボード・イベントは **rdp_type** 呼び出しに変換されます。同じ座標での連続する 2 回のマウス・クリックは、ダブルクリックと解釈されます。
 - ▶ **[低]**：低レベルのスクリプトを生成します。キーの解放/押下イベントは **rdp_key** イベントに変換されます。修飾キー (Alt, Ctrl, Shift) は、ほかの関数の **KeyModifier** パラメータとして使用されます。マウスの解放/押下/移動イベントは、マウスのクリック/ドラッグ・イベントに変換されます。
 - ▶ **[未処理]**：ネットワーク・バッファから入力イベントを抽出し、最も単純な形 (キーの解放/押下、マウスの解放/押下/移動) の呼び出しを生成することによって、未処理レベルのスクリプトを生成します。KeyModifier パラメータは使用されません。

- ▶ **[マウス移動呼び出しの生成]**：スクリプトに `rdp_mouse_move` 呼び出しを生成します。このオプションを有効にすると、スクリプト・サイズが非常に大きくなります（標準設定では無効）。
- ▶ **[未処理のマウス呼び出しの生成]**：有効になっているときは、スクリプト・レベルが未処理に設定されているかのように `rdp_mouse_button_up/down` 呼び出しが生成されます。キーボード呼び出しは、スクリプト・レベルのとおり生成されます。無効になっているときは、スクリプト・レベルのとおりマウス呼び出しが生成されます。スクリプト・レベルが未処理に設定されている場合、このオプションは無視されます（標準設定では無効）。
- ▶ **[未処理のキーボード呼び出しの生成]**：有効になっているときは、スクリプト・レベルが未処理に設定されているかのように `rdp_raw_key_up/down` 呼び出しが生成されます。マウス呼び出しは、スクリプト・レベルのとおり生成されます。無効になっているときは、スクリプト・レベルのとおりキーボード呼び出しが生成されます。スクリプト・レベルが未処理に設定されている場合、このオプションは無視されます（標準設定では無効）。
- ▶ **[常に接続名を生成する]**：有効になっているときは、関数呼び出しに `ConnectionName` パラメータが含まれます。無効になっているときは、スクリプトに複数の `rdp_connect_server` が現れている場合でも、関数にはこのパラメータしか含まれないようになります（標準設定では無効）。
- ▶ **[同期化ポイントの自動生成]**：同期化ポイントによって、ポップアップやそれ以外の制御の対象となるウィンドウまたはダイアログ・ボックスが特定の状態になるのを待っている間、スクリプトの再生を一時停止できます。このオプションにより、マウスのクリックおよびドラッグの前に自動的に `sync_on_image` 関数が生成されます（標準設定では有効）。**[同期化半径]** は、マウス操作から、同期化領域を定義する四角形の端までの距離です。標準設定値は 20 ピクセルです。
 - ▶ **[なし]**：同期化ポイントは自動的に追加されません。
 - ▶ **[四角形]**：クリックまたはドラッグした位置を中心とした四角形の同期化ポイントを作成します。**[同期化半径]** は、マウス操作から、同期化領域を定義する四角形の端までの距離です。標準設定値は 20 ピクセルです。
 - ▶ **[拡張]**：必要な場所（たとえばボタン）のみ選択されるように、また、UI の変更（たとえばボタンの移動）に反応するように設計された同期化ポイントを作成します。同期化領域が認識されないときは、四角形の同期化設定が使用されます。

RDP の高度なコード生成オプション

高度なコード生成オプションは、パラメータ名の標準のプレフィックスなど、VuGen がスクリプトを作成する方法を制御します。この設定は上級ユーザが変更することをお勧めします。

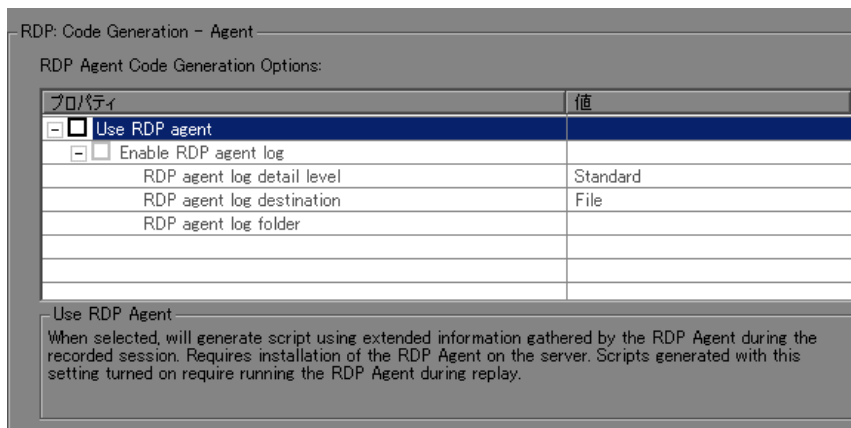


次のオプションを設定できます。

- ▶ **[ダブルクリックのタイムアウト (ミリ秒)]** : ダブルクリックとして認識されるための、マウス・ボタンの 2 つの連続クリック間の最大時間 (ミリ秒単位)。標準設定値は 500 ミリ秒です。
- ▶ **[スナップショット名のプレフィックス]** : 現在のスクリプトで生成されるスナップショット・ファイルの名前のプレフィックス。これは、スクリプトを結合する場合に便利です。スクリプトごとに異なるプレフィックスを指定できます。標準設定値は **snapshot_** です。
- ▶ **[クリップボード パラメータのプレフィックス]** : 現在のスクリプトで生成されるクリップボード・パラメータのプレフィックス。これは、スクリプトを結合する場合に便利です。スクリプトごとに異なるプレフィックスを指定できます。標準設定値は **ClipboardDataParam_** です。
- ▶ **[クリップボード パラメータの相関]** : ユーザが送信した記録済みクリップボード・テキストを、サーバから受信した同一のテキストを含む相関パラメータと置換します。

RDP Agent Code Generation Options

Agent Code Generation Options は、記録中に Microsoft Agent for Terminal Server エージェントと VuGen が連携する方法を制御します。



次のオプションを設定できます。

- ▶ **[Use RDP agent]** : 記録セッション中に RDP エージェントが収集した情報を使用してスクリプトを生成します。サーバに LoadRunner RDP エージェントがインストールされている必要があります。
- ▶ **[Enable RDP agent log]** : RDP エージェント・ログを有効にします。
 - ▶ **[RDP agent log detail level]** : RDP エージェント・ログで生成される詳細のレベルを設定します。**[Standard]** を指定すると最も低いレベルになり、**[Extended Debug]** を指定すると最も高いレベルになります。
 - ▶ **[RDP agent log destination]** : RDP エージェント・ログ・データの宛先を設定します。**[File]** を指定すると、リモート・サーバ側にのみログ・メッセージが保存されます。**[Stream]** を指定すると、VuGen マシンにログ・メッセージが送信されます。**[FileAndStream]** を指定すると、両方の宛先にログ・メッセージが送信されます。
 - ▶ **[RDP agent log folder]** : RDP エージェント・ログ・ファイルが生成されるリモート・サーバ上のフォルダのパス。

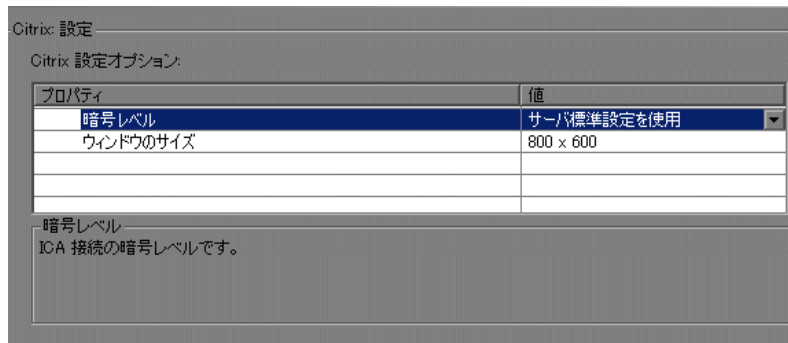
Citrix 記録オプションについて

次の項目について Citrix 記録オプションの設定が可能です。

- ▶ 設定記録オプション
- ▶ レコーダ記録オプション
- ▶ コード生成記録オプション
- ▶ ログイン記録オプション（シングル・プロトコルの Citrix ICA スクリプトの場合のみ）

設定記録オプション

[Citrix : 設定] 記録オプションでは、記録セッション中の Citrix クライアントに対するウィンドウ・プロパティと暗号化設定を設定します。



- ▶ [暗号レベル] : ICA 接続の暗号化レベル。[基本], [ログイン専用 128 ビット], [40 ビット], [56 ビット], [128 ビット], または [サーバ標準設定値の使用] から選択します。
- ▶ [ウィンドウのサイズ] : クライアント・ウィンドウのサイズ。[640 x 480], [800 x 600] (標準設定), [1024 x 768], [1280 x 1024], [1600 x 1200] から選択できます。

レコーダ記録オプション

[Citrix レコーダ オプション] で、記録中にタイトルの変更されたウィンドウ名を生成する方法を指定できます。画面のスナップショットをスクリプト・ファイルと一緒に保存するかどうか、また、テキスト同期化関数を生成するかどうかを指定することもできます。

| Citrix レコーダ オプション | |
|---|--|
| プロパティ | |
| <input type="checkbox"/> ウィンドウ名 | |
| <input type="radio"/> 新規のウィンドウ名をそのまま使用する | |
| <input type="radio"/> 新規名に共通のプレフィックスを使用する | |
| <input type="radio"/> 新規名に共通のサフィックスを使用する | |
| <input checked="" type="checkbox"/> 画面ショットを保存する | |
| ウィンドウ名 | |

ウィンドウ名

アプリケーションには、記録中にアクティブなウィンドウの名前が変わるものがあります。スクリプトをそのまま再生しようとするすると、Vuser は最初のウィンドウ名を使用するため再生が失敗します。記録オプションを使用して、ウィンドウの命名規則（VuGen がウィンドウを特定するのに使用する共通のプレフィックスまたはサフィックス）を指定できます。

たとえば、最初のウィンドウ名が「無題 - メモ帳」であり、アプリケーションの実行中にこのウィンドウの名前が「my_test - メモ帳」になる場合は、VuGen に共通のサフィックス（「メモ帳」）のみを使用するよう指示できます。

記録中のウィンドウ名の生成には次のオプションを使用できます。

- ▶ **[新規のウィンドウ名をそのまま使用する]**：ウィンドウのタイトルをそのままウィンドウ名として使用する場合に選択します（標準設定）。
- ▶ **[新規名に共通のプレフィックスを使用する]**：ウィンドウ・タイトルの先頭の共通文字列をウィンドウ名として使用する場合に選択します。
- ▶ **[新規名に共通のサフィックスを使用する]**：ウィンドウ・タイトルの末尾の共通文字列をウィンドウ名として使用する場合に選択します。

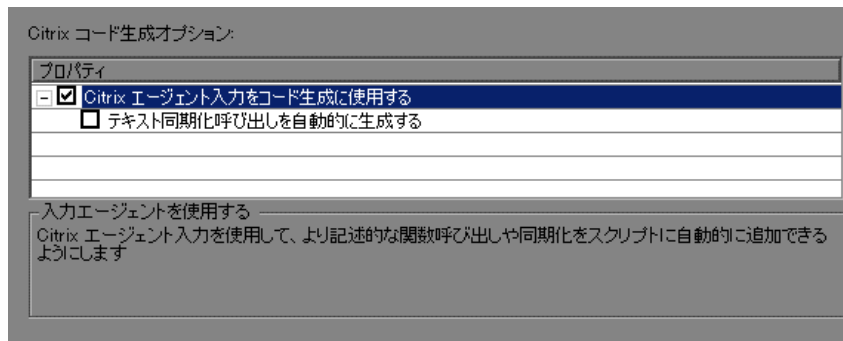
注：あるいは、記録後に実際のスクリプトでウィンドウ名を変更することもできます。スクリプト・ビューでウィンドウ名を見つけ、ウィンドウ名の先頭または末尾をワイルドカードの「*」で置き換えます。 `ctx_sync_on_window ("My Application*", ACTIVATE, ...CTX_LAST);`

スナップショット

[画面ショットを保存する]：該当する場合、スクリプトのステップごとに、Citrix クライアント・ウィンドウのスナップショットが VuGen によって保存されます。記録したアクションをより的確に理解するために、このオプションを有効にすることをお勧めします。ただし、スナップショットを保存すると使用するディスク領域が増え、記録セッションの速度が低下します。

コード生成記録オプション

[コード生成]：記録時に VuGen が情報をキャプチャする方法を設定できます。



- ▶ **[Citrix エージェント入力をコード生成に使用する]**：Citrix エージェント入力を使用して、同期化関数が追加されたより説明的なスクリプトを生成します（標準設定では有効）。
- ▶ **[テキスト同期化呼び出しを自動的に生成する]**：各マウス・クリックの前に、テキストの同期化を行う **テキストで同期**ステップを追加します（標準設定では無効）。

記録中に手動で追加したテキストの同期化ステップは、前述の設定の影響を受けません。前述のオプションを無効にしても、手動で追加したステップはスクリプトに現れます。記録時における**テキストで同期**ステップの追加の詳細については、『第 2 巻 - プロトコル』の「Citrix プロトコル」を参照してください。

前述のオプションは、スクリプトの再生成にも使用可能です。たとえば、当初**[テキスト同期化呼び出しを自動的に生成する]**を無効にしてスクリプトを記録した場合でも、テキストの同期化が含まれるように記録後にスクリプトを再生成できます。スクリプトの再生成の詳細については、103 ページ「Vuser スクリプトの再生成」を参照してください。

注： VuGen 8.1 以降を使用して、以前のバージョンの VuGen で作成したスクリプトを再生成した場合、そのスクリプトは以前のバージョンと互換性がなくなります。つまり、新しいスクリプトを作成したかのような動作になります。

ログイン記録オプション

[**Citrix：ログイン**] 記録オプションで、記録セッションに対する接続情報とログイン情報を設定します（NFUSE で作業を行っているときは、Web ページを通じてログインが実行されるため、ログイン・オプションは使用できません）。

ログイン情報を直接設定することも、VuGen に **ica** ファイルに格納されている既存の設定情報を使用させることもできます。

サーバの名前または VuGen が `ctx_connect_server` 関数を生成する接続を指定する必要があります。

```
ctx_connect_server("steel", "test", "test", "testlab", CTRX_LAST);
```

ログイン情報を省略すると、指定したサーバがクライアントによって検出されたときに、情報を入力するよう求められます。

Citrix セッションに関する次のユーザおよびサーバ情報を指定します。

[**ログイン情報**]：このセクションには次のログイン情報が含まれます。

- ▶ Citrix ユーザの**ユーザ名**
- ▶ Citrix ユーザの**パスワード**
- ▶ Citrix ユーザの**ドメイン**
- ▶ MetaFrame サーバがクライアントの識別に使用する**クライアント名**（任意）。

[**接続**]：このセクションには次のサーバ情報が含まれます。

- ▶ [**ネットワークのプロトコル**]：TCP/IP か TCP/IP+HTTP のどちらかを選択します。ほとんどの Citrix サーバは TCP/IP をサポートしています。しかし、一部のサーバは、ある決まった HTTP ヘッダを持つ TCP/IP のみ許可するように、管理者によって設定されています。通信上の問題が発生した場合は、TCP/IP + HTTP オプションを選択します。

- ▶ **[サーバ]** : Citrix サーバの名前。新しいサーバをリストに追加するには、**[追加]** をクリックし、サーバ名 (およびサーバの TCP/IP + HTTP 用ポート) を入力します。複数のサーバが適用されるのは **[公開アプリケーション]** を指定したときのみです。特定のアプリケーションのないデスクトップに接続しようとしている場合は、1 つのサーバのみが表示されます。
- ▶ **[発行されたアプリケーション]** : Citrix サーバで認識される、**発行されたアプリケーション** の名前。使用可能なアプリケーションのリストがドロップダウン・メニューに含まれています。発行されたアプリケーションが指定されていない場合、VuGen はサーバのデスクトップを使用します。発行されたアプリケーションの追加または名前変更を行った場合は、この記録オプションを閉じてから再度開き、新しいリストを表示します。

Citrix クライアントで、発行されたアプリケーションの名前を変更するには、Citrix サーバ・マシンで変更を行う必要があります。**[管理コンソール]** > **[アプリケーション]** を選択し、新しいアプリケーションを作成するか、既存のアプリケーションの名前を変更します。

発行されたアプリケーションを指定しなければ、Citrix のロード・バランシングは機能しません。サーバのデスクトップにアクセスする場合にロード・バランシングを使用するには、デスクトップをサーバ・マシンで発行されたアプリケーションとして登録し、**[発行されたアプリケーション]** ドロップダウン・リストからこの名前を選択します。

接続パラメータの ICA ファイルの使用

既存の .ica ファイルと関連するすべての設定情報がある場合は、**[接続パラメータに ICA ファイルを使用する]** を選択します。そしてその下のテキストボックスに、.ica ファイルのフル・パスを指定します。

ICA ファイルの形式の詳細については、『**第 2 巻 - プロトコル**』の 321 ページ「ICA ファイルについて」および Citrix の Web サイト (www.citrix.com) を参照してください。

Citrix 記録オプションの設定

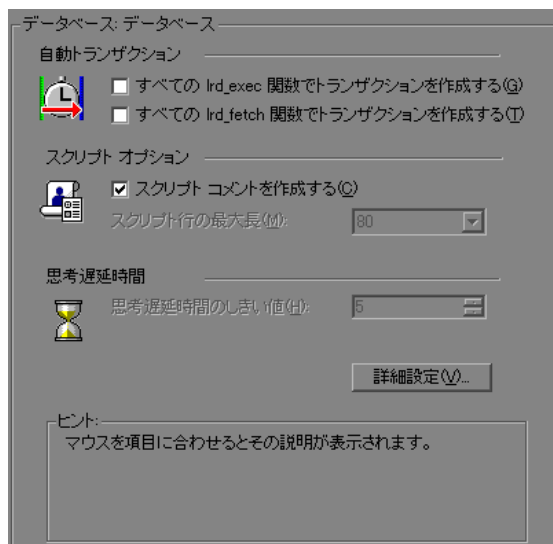
記録を行う前に、必要な記録オプションを設定します。

Citrix 記録オプションを設定するには、次の手順を実行します。

- 1 [記録オプション] ダイアログ・ボックスを開きます。[ツール] > [記録オプション] を選択するか、[記録開始] ダイアログ・ボックスで [オプション] ボタンをクリックします。キーボードのショートカット・キーは CTRL キー + F7 キーです。
- 2 [Citrix : ログイン] ノードを選択します (シングル・プロトコルの Citrix ICA スクリプトの場合のみ)。
 - ▶ 既存の ica ファイルと関連するすべての設定情報がある場合は、[接続パラメータに ICA ファイルを使用する] を選択します。ica ファイルのフル・パスを指定するか、[参照] ボタンをクリックして、ローカル・ディスクまたはネットワークのファイルの場所を見つけます。
 - ▶ ica ファイルがない場合は、[接続パラメータを定義する] を選択します。これは、標準設定です。[接続] 情報と [識別] 情報を入力します。
- 3 [Citrix: 設定] ノードを選択します。暗号化レベルとウィンドウ・サイズを選択します。
- 4 [Citrix: コード生成] ノードを選択します。Citrix エージェントからの情報を使用して、より説明的なスクリプトを生成するには、ノードのオプションを有効にします。
- 5 [Citrix: レコーダ] ノードを選択します。記録セッション中にタイトルが変わるウィンドウのウィンドウ名を生成する方法を指定します。
- 6 VuGen が各ステップのスナップショットを保存しないようにするには、[画面ショットを保存する] をクリアします。
- 7 Nfuse セッションの記録時には、Web の記録モードを「URL ベース」に設定します。[一般: 記録] 記録オプションを選択し、[URL ベースのスクリプト] ノードを選択します。
- 8 [OK] をクリックして設定を適用し、ダイアログ・ボックスを閉じます。

データベース記録オプション

データベース・セッションの記録を開始する前に、記録オプションを設定します。自動関数生成、スクリプト・オプション、思考遅延時間の基本的な記録オプションを設定できます。



- ▶ **[自動トランザクション]** : `lrd_exec` 関数と `lrd_fetch` 関数をすべてトランザクションとしてマークします。これらのオプションを有効にすると、VuGen によってすべての `lrd_exec` 関数または `lrd_fetch` 関数の前後に `lr_start_transaction` と `lr_end_transaction` が挿入されます。標準設定では、自動トランザクションは無効になっています。
- ▶ **[スクリプト オプション]** : 記録されたスクリプトに `lrd_stmt` のオプションの値を説明するコメントを生成します。また、スクリプト行の最大長を指定できます。標準設定の長さは 80 文字までです。
- ▶ **[思考遅延時間]** : VuGen ではオペレータの思考遅延時間が自動的に記録されません。思考遅延時間のしきい値を設定して、それよりも低い場合には思考遅延時間を無視するようにできます。記録された思考遅延時間がしきい値を越えていると、VuGen によって LRD 関数の前に `lr_think_time` ステートメントが挿入されます。記録された思考遅延時間がしきい値を越えている場合、`lr_think_time` ステートメントは生成されません。標準設定の値は 5 秒です。

データベース記録オプションを設定するには、次の手順を実行します。

- 1 [ツール] > [記録オプション] を選択します。[記録オプション] ダイアログ・ボックスが開きます。
- 2 `lrd_exec` ステートメントに対して自動トランザクションを有効にするには、[すべての `lrd_exec` 関数にトランザクションを作成する] を選択します。
`lrd_fetch` ステートメントに対して自動トランザクションを有効には、[すべての `lrd_fetch` 関数にトランザクションを作成する] を選択します。
- 3 スクリプトにわかりやすいコメントを挿入するように設定するには、[スクリプトコメントを作成する] を選択します。
- 4 VuGen エディタの行の最大長を変更するには、[スクリプト行の最大長] ボックスで最大長を指定します。
- 5 思考遅延時間のしきい値を標準設定の 5 秒から変更するには、[思考遅延時間のしきい値] ボックスに値を指定します。

ログのトレース・レベル、CtLib 関数の生成、コード生成バッファに関して詳細な記録オプションを設定することもできます。

データベースの詳細記録オプション

基本的な記録オプションのほかに、ログ・ファイルの詳細、CtLib 固有の関数、バッファ・サイズ、記録エンジンについて詳細な記録オプションも設定できます。

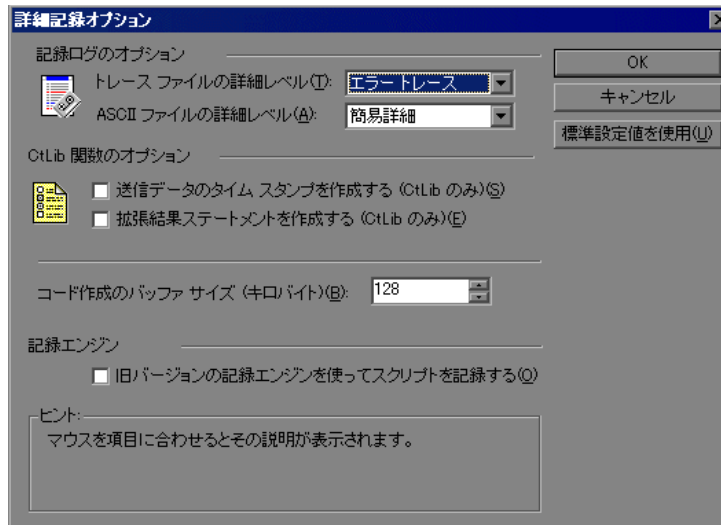
- ▶ [記録ログのオプション]: トレース・ファイルと ASCII ログ・ファイルの詳細レベルを設定できます。トレース・ファイルに対して指定できるレベルとしては、[オフ]、[エラー トレース]、[簡易トレース]、および [完全トレース] があります。[エラー トレース] に設定すると、エラー・メッセージだけがログに書き込まれます。[簡易トレース] に設定すると、エラーのほかに、記録中に生成された関数のリストがログに書き込まれます。[完全トレース] に設定すると、メッセージ、告示、警告などがすべてログに書き込まれます。

記録セッションに関して ASCII タイプのログが生成されるようにも設定できます。指定できるレベルとしては、[オフ]、[簡易詳細]、[完全詳細] があります。[簡易詳細] 設定では、すべての関数がログに書き込まれます。[完全詳細] 設定では、生成されたすべての関数とメッセージが ASCII コードでログに書き込まれます。

- ▶ **[CtLib 関数のオプション]** : 送信データのタイム・スタンプと、拡張結果ステートメントが作成されるように設定できます。
- ▶ **[送信データのタイムスタンプを作成する]** : 標準設定では、VuGen は **mpsReqSpec** パラメータに **TotalLen** キーワードと **Log** キーワードを設定して **lrd_send_data** ステートメントを生成します。[詳細記録オプション] ダイアログ・ボックスでは、**TimeStamp** キーワードも生成するように指定することができます。既存のスクリプトでこの設定を変更した場合は、**[ツール]** > **[スクリプトの再生成]** を選択して **Vuser** スクリプトを生成します。標準設定で **Timestamp** キーワードを生成することは推奨されません。記録中に生成されたタイム・スタンプは再生中に生成されるものと異なるため、スクリプトの実行が失敗するからです。このオプションを使用する必要があるのは、(**lrd_send_data** の後の **lrd_result_set** が失敗する) スクリプト実行中に試行が失敗した後のみです。オプションを指定すれば、生成されたタイム・スタンプと、**lrd_send_data** を実行して失敗したときのタイム・スタンプを相関できるようになります。
- ▶ **[拡張結果ステートメントを作成する]** : 標準設定では、結果セットを準備する段階で **lrd_result_set** 関数が生成されます。このオプションを選択すると、**lrd_result_set** 関数を拡張した関数である **lrd_result_set_ext** が生成されます。この関数は、結果セットを準備するほかに、**ct_results** からリターン・コードとタイプを発行します。
- ▶ **[コード作成のバッファサイズ]** : コード生成バッファの最大サイズをキロバイト単位で指定します。標準設定値は 128 キロバイトです。データベース・セッションが長い場合には、より大きなサイズを指定できます。
- ▶ **[記録エンジン]** : VuGen に対して、旧バージョンの LRD 記録エンジンを使ってスクリプトを記録するように指示して、VuGen の旧バージョンと互換性を保つことができます。このプロトコルはシングル・プロトコルのスクリプトにのみ適用されます。

詳細記録オプションを設定するには、次の手順を実行します。

- 1 [記録オプション] ダイアログ・ボックスの [データベース] ノードで [詳細設定] ボタンをクリックします。[詳細記録オプション] ダイアログ・ボックスが開きます。



- 2 [トレース ファイルの詳細レベル] を選択します。トレース・ファイルを無効にするには [オフ] を選択します。
- 3 ASCII ログ・ファイルを生成するには、[ASCII ファイルの詳細レベル] ボックスから詳細レベルを選択します。
- 4 CtLib の場合のみ : `lrd_send_data` 関数に対する TimeStamp キーワードを生成させるには、[送信データのタイムスタンプを作成する] オプションを選択します。
- 5 CtLib の場合のみ : `lrd_result_set` の代わりに `lrd_result_set_ext` を生成させるには、[拡張結果ステートメントを作成する] オプションを選択します。
- 6 コード生成バッファのサイズを、標準設定の 128 キロバイトから変更するには、[コード作成のバッファ サイズ] ボックスに使用する値を入力します。
- 7 下位互換性を維持するために旧バージョンの VuGen の記録エンジンを使うには、[旧バージョンの記録エンジンを使ってスクリプトを記録する] オプションを選択します。
- 8 [OK] をクリックして設定を保存して、[詳細記録オプション] ダイアログ・ボックスを閉じます。

AMF 記録モードの設定

AMF プロトコルおよび Web プロトコルを使用して Flash Remoting セッションからスクリプトを生成する方法を VuGen に指定できます。次のオプションがあります。

- ▶ AMF および Web
- ▶ AMF のみ
- ▶ Web のみ

標準設定では、VuGen はスクリプトに AMF 呼び出しのみを生成します。記録対象プロトコルを設定するには、[記録オプション] を開き ([ツール] > [記録オプション]), [一般: プロトコル] ノードを選択します。詳細については、86 ページ「プロトコルの追加と削除」を参照してください。

注：前述のいずれかのオプションで記録した場合でも、後でスクリプトを再生成してほかのプロトコルが含まれるように、または除外されるようにできます。スクリプトの再生成の詳細については、103 ページ「Vuser スクリプトの再生成」を参照してください。

AMF および Web

AMF プロトコルと Web プロトコルの両方を有効にした場合、VuGen はビジネス・プロセス全体の関数を生成します。AMF データに遭遇すると、適切な AMF 関数を生成します。

次のコードの抜粋では、VuGen は Web 関数 (**web_url**) と AMF 関数 (**amf_call**, **amf_define_envelope_header_set**) の両方を生成しています。

```
web_url("flash",
  "URL=http://testlab:8200/flash/", "Resource=0",
  ...
  "Snapshot=t1.inf",
  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);

amf_set_version("0");

amf_define_header_set("Id=amf_header_set",
  HEADER,
  "Name=amf_server_debug",
  "MustUnderstand=true",
  "Data=<object><boolean key=¥"coldfusion¥">true</boolean>
    <boolean key=¥""amfheaders¥">false</boolean>...
  LAST);

amf_call("flashgateway.samples.FlashJavaBean.testDocument",
  "Gateway=http://testlab:8200/flashservices/gateway",
  "AMFHeaderSetId=amf_header_set",
  "Snapshot=t3.inf",
  MESSAGE,
  "Method=flashgateway.samples.FlashJavaBean.testDocument",
  "TargetObjectId=/1",
  BEGIN_ARGUMENTS,
  "<xmlString><![CDATA[<TEST message=¥"test¥"><INSIDETEST/>
    </TEST>]]></"xmlString">",
  END_ARGUMENTS,
  LAST);
```

AMF のみ

Flash Remoting をエミュレートするのに AMF 呼び出しのみを必要とする場合は、Web 呼び出しを無効にして、AMF 呼び出しのみを生成できます。

次の例は、AMF プロトコルを有効にし、Web プロトコルを無効にして、前述のセッションを記録したものです。

```

Actions()
{
  amf_set_version("0");

  amf_define_header_set("Id=amf_header_set",
    HEADER,
    "Name=amf_server_debug",
    "MustUnderstand=true",
    "Data=<object><boolean key=¥"coldfusion¥">true</boolean>
      <boolean key=¥""amfheaders¥">false</boolean>...
    LAST);

  amf_call("flashgateway.samples.FlashJavaBean.testDocument",
    "Gateway=http://testlab:8200/flashservices/gateway",
    "AMFHeaderSetId=amf_header_set",
    "Snapshot=t3.inf",
    MESSAGE,
    "Method=flashgateway.samples.FlashJavaBean.testDocument",
    "TargetObjectId=/1",
    BEGIN_ARGUMENTS,
    "<xmlString><![CDATA[<TEST message=¥"test¥"><INSIDETEST
    /></TEST>]]></"
    "xmlString>",
    END_ARGUMENTS,
    LAST);
  ...

```

なお、この記録は、完全なビジネス・プロセスを示すものではありません。AMF を使用する Flash Remoting 呼び出しの例を示しているに過ぎません。

Web のみ

Web のみのオプションは、代替として Web HTTP 技術を使用します。AMF 呼び出しは生成されません。代わりに、Flash Remoting 情報を持つ **web_custom_request** 関数が生成されます。

AMF コード生成オプション

Flex アプリケーションを記録する際、状況によっては、VuGen で外部オブジェクトをデコードできないことがあります。これは、Flex 2 アプリケーションで使用されている独自のエンコード・スキームによるものです。

この問題に対処するには、コードが解読不能なときはユーザ定義要求関数を生成するように設定します。

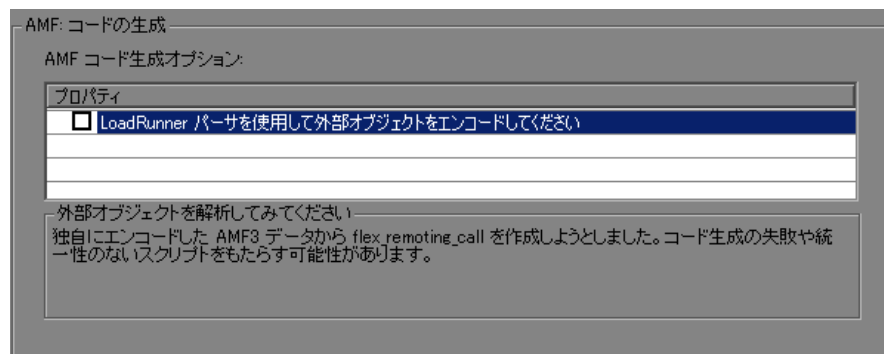
AMF 記録オプションを指定すると、VuGen は外部オブジェクトの AMF 呼び出し要求を生成しようとします。すべての外部オブジェクトを標準の AMF3 オブジェクトとしてデコードし、`amf_call` 関数を生成します。

このオプションを無効にすると（標準設定）、VuGen は、外部オブジェクトに遭遇したとき、未解析の AMF3 バイナリ・データが含まれるユーザ定義要求関数を生成します。

注：このオプションを有効にすると、コード生成の安定性が低下することがあります。

外部オブジェクトの AMF コード生成を設定するには、次の手順を実行します。

- 1 [記録オプション] ダイアログ・ボックスを開きます。[ツール] > [記録オプション] を選択するか、[記録開始] ダイアログ・ボックスで [オプション] ボタンをクリックします。キーボードのショートカット・キーは CTRL キー + F7 キーです。
- 2 [AMF: コードの生成] ノードを選択します。外部オブジェクトに対して標準の `amf_call` 関数を生成するには、この記録オプションを有効にします。



Flex コード生成オプション

Flex アプリケーションを記録する際、状況によっては、VuGen で外部オブジェクトをデコードできないことがあります。これは、Flex 2 アプリケーションで使用されている独自のエンコード・スキームによるものです。

この問題に対処するには、コードが解読不能なときは **flex_web_request** 関数を生成するように設定します。

この問題に対処するオプションは 2 つあります。

- 1) LoadRunner パーサを使用して、外部オブジェクトの Flex リモート要求を生成するよう試みます。すべての外部オブジェクトを標準の AMF3 オブジェクトとしてデコードし、Flex 関数を生成します。
- 2) サーバのパーサを使用して、外部オブジェクトの Flex リモート要求を生成するよう試みます。すべての外部オブジェクトを標準の AMF3 オブジェクトとしてデコードし、**flex_amf_call** 関数を生成します。

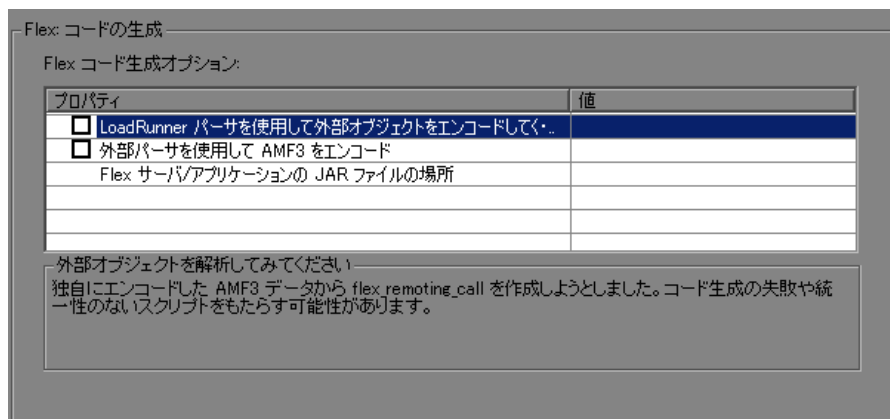
両方のオプションを無効にすると（標準設定）、VuGen は、外部オブジェクトに遭遇したとき、未解析の AMF3 バイナリ・データが含まれる **flex_web_request** 関数を生成します。

注： LoadRunner パーサによるエンコードを有効にすると、コード生成の安定性が低下することがあります。

外部オブジェクトの Flex コード生成を設定するには、次の手順を実行します。

- 1 [記録オプション] ダイアログ・ボックスを開きます。[ツール] > [記録オプション] を選択するか、[記録開始] ダイアログ・ボックスで [オプション] ボタンをクリックします。

- 2 [Flex: コードの生成] ノードを選択します。[LoadRunner パーサを使用して外部オブジェクトをエンコードしてください] チェック・ボックスを選択し、[OK] をクリックします。



- 3 スクリプトを再生成します。コード生成ログでエラーがないか確認します。スクリプトに **web_custom_request** ステップも **flex_custom_request** ステップもなければ、問題は解決されており、手順は完了します。
- 4 前述のステップが成功していなかった場合は、[ツール] > [記録オプション] に戻り、[Flex: コードの生成] ノードを選択します。
- 5 [外部パーサを使用して AMF3 をエンコード] チェック・ボックスを選択します。
- 6 サーバとアプリケーションの jar ファイルをマシン上の任意のディレクトリ、および、Load Generator の同じディレクトリにコピーします。サポートされているサーバおよび使用すべき jar ファイルの詳細については、HP ソフトウェア・サポートにお問い合わせください。
- 7 [ツール] > [記録オプション] を選択し、[Flex: コードの生成] ノードを選択します。[Flex サーバ/アプリケーションの JAR ファイルの場所] オプションの右側に、マシン上の各 jar ファイルの場所をセミコロンで区切って指定します。たとえば、C:\Jars\Myapp.jar;C:\Jars\MyServer.jar のように指定します。
- 8 [OK] をクリックします。
- 9 スクリプトを再生成します。コード生成ログでエラーがないか確認します。スクリプトに **web_custom_request** ステップも **flex_custom_request** ステップもなければ、問題は解決されており、手順は完了します。問題が解決されない場合は、HP ソフトウェア・サポートにお問い合わせください。

Microsoft .NET 記録オプション

スクリプトの記録オプションと Microsoft .NET 固有の記録オプションの両方を設定できます。本項では、Microsoft .NET 固有の記録オプションについて説明します。スクリプト記録オプションの詳細については、371 ページ「スクリプト生成オプションの設定」を参照してください。

Microsoft .NET 固有の記録オプションは、記録設定と .NET フィルタを対象とします。

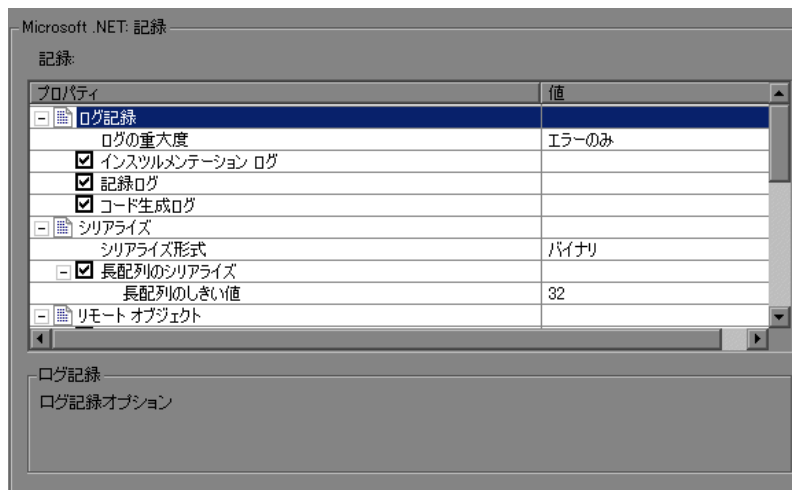
記録設定では、ログ、シリアライズ、デバッグ、およびログのトレース・レベルを制御できます。詳細については、313 ページ「.NET 記録オプション・ダイアログ・ボックスを開くには、[ツール] > [記録オプション] を選択し、[記録] ノードを選択します。」を参照してください。

フィルタ・オプションでは、記録用のフィルタを標準の .NET Remoting, ADO.NET, エンタープライズ・サービス、または WCF フィルタから選択できます。また、カスタム・フィルタを作成し、必要に応じて設定できます。詳細については、『第 2 巻 - プロトコル』の第 36 章「Microsoft .NET フィルタ」を参照してください。

WCF 全二重通信の記録の詳細については、318 ページ「WCF 記録オプション」を参照してください。

記録オプションの中には、コードの再生成に関係するものもあります。特定のオプションを指定してスクリプトを記録したら、コード生成オプションを変更して、別の設定または別の言語を使用してスクリプトを再生成できます。オプションを変更してスクリプトを再生成するには、[ツール] > **[スクリプトの再生成]** を選択します。[スクリプトの再生成] ダイアログ・ボックスで、**[オプション]** をクリックします。詳細については、103 ページ「Vuser スクリプトの再生成」を参照してください。

.NET 記録オプション・ダイアログ・ボックスを開くには、[ツール] > [記録オプション] を選択し、[記録] ノードを選択します。



次の領域に関して、スクリプトを設定できます。

- ▶ ログ
- ▶ シリアライズの設定
- ▶ リモート・オブジェクト
- ▶ WCF 全二重バインディング
- ▶ デバッグ・オプション
- ▶ フィルタ
- ▶ コード生成

ログ

- ▶ [ログ] オプションを使用して、記録ログ・ファイルに記録される情報の詳細度 ([ログの重大度]) を設定できます。ログのレベルは [エラーのみ] (標準設定) または [デバッグ] に設定します。重大度の設定は以降の説明に従って有効にするすべてのログに適用されます。ログを詳細に記録すると記録時間が大幅に増えることがあるので、HP のサポートによる特別な指示がある場合を除き、必ず [エラーのみ] ログを使用してください。

- ▶ **[インスツルメンテーション ログ]**：インスツルメンテーション・プロセスに関するメッセージをログ記録します（標準設定では有効）。
- ▶ **[記録ログ]**：記録中に発行されたメッセージをログ記録します（標準設定では有効）。
- ▶ **[コード生成ログ]**：コード生成段階で発行されたメッセージをログ記録します（標準設定では有効）。

シリアライズの設定

[シリアライズ] 設定ではシリアライズ形式を設定できます。

VuGen は記録中に未知のオブジェクトに遭遇した場合、オブジェクトがシリアライズをサポートしていればシリアライズを使用します。未知のオブジェクトは、たとえばフィルタに包含されていなかったために、その生成が記録されなかった入力引数などが考えられます。シリアライズによって、未知の引数をメソッドに渡すことによるコンパイルエラーがなくなります。オブジェクトがシリアライズされる場合、カスタム・フィルタを設定してこのオブジェクトを記録することをお勧めします。

- ▶ **[シリアライズ形式]**：シリアライズをサポートするクラスの記録時に VuGen によって作成されるシリアライズ・ファイルの形式です。[バイナリ]、[XML]、または [両方] を設定できます。バイナリ形式の利点は、圧縮率が高いため、より高速であるということです。バイナリ形式の不利な点は、XML と同じようにはデータを操作できない点です。
- ▶ **[長配列のシリアライズ]**：シリアル化可能なオブジェクトが含まれる長配列（たとえばプリミティブの配列）には、VuGen のシリアル化メカニズムを使用します。このオプションを有効にすると、配列サイズがしきい値以上の場合に、LrReplayUtils.GetSerializedObject 呼び出しが生成されます。
- ▶ **[長配列のしきい値]**：長配列とみなされる配列のしきい値サイズ。配列サイズがこのサイズ以上の場合、シリアル化可能なオブジェクトが検出されたときにシリアライズされます。

ヒント：XML のシリアル化の場合は、XML ファイルの内容を表示できます。ファイルの内容を表示するには、右クリック・メニューから **[XML の表示]** を選択します。

リモート・オブジェクト

本項では、プロキシ関連の記録オプションを定義します。

- ▶ **[インプロセス オブジェクトを記録する]**：サーバがクライアントと同じプロセスでホストされている場合に、クライアントとサーバの間のやり取りを記録します。このアクションは純粋なクライアント/サーバ・トラフィックではないため、通常は重要ではありません。インプロセス・メソッドが関係する場合、たとえば、特定のエンタープライズ・サービス・アプリケーションでは、このオプションを有効にしてインプロセス・メソッドをキャプチャできます（標準設定では無効）。

非同期呼び出し

次の項では、リモート・オブジェクトおよびコールバック・メソッドで使用する非同期呼び出しの処理方法を指定します。このオプションは、主に .NET Remoting と WCF の環境に関係します。

- ▶ **[標準設定で元のコールバックを呼び出す]**：スクリプトの生成時および再生時に、記録されたアプリケーションの元のコールバックを使用します。コールバック・メソッドがフィルタで明示的に除外されている場合は、このオプションが有効になっていてもコールバックは除外されます。GUIの更新など、ビジネス・プロセスに直接関係のないアクションをコールバックが実行している場合は、このオプションを必ず無効にしてください。
- ▶ **[非同期コールバックを生成する]**：このオプションは、元のコールバックが記録されなかった場合にコールバックを処理する方法を定義します。これは、前述のオプション（**[標準設定で元のコールバックを呼び出す]**）が無効になっているとき、あるいは、コールバックが明示的に除外されているときに関係してきます。

このオプションを有効にすると、再生中に元のコールバックの代わりに呼び出されるダミー・メソッドが作成されます。このダミー・コールバックは、スクリプトの **callbacks.cs** セクションに生成されます。

このオプションを無効にすると、コールバックの代わりに NULL 値が挿入され、イベントがその都度記録されます。

次のコードは、[非同期コールバックを生成する] が有効になっているときの Calculator クライアントのスクリプト生成を示しています。

```
lr.log("Event 2: CalculatorClient_1.Add(2, 3);");
Int32RetVal = CalculatorClient_1.Add(2, 3);
// Int32RetVal = 5;

callback_1 = new AsyncCallback(this.OnComplete1);
lr.log("Event 3: CalculatorClient_1.BeginAdd(2, 3, callback_1, null);");
IAsyncResult_1 = CalculatorClient_1.BeginAdd(2, 3, callback_1, null);
```

コールバック・メソッド OnComplete1 を表示するには、左側の表示枠で **callback.cs** ファイルをクリックします。

次のコードは、[非同期コールバックを生成する] が無効になっているときの スクリプト生成を示しています。コールバックの代わりに NULL が生成され、コールバック・イベントがその都度記録されています。

```
lr.log("Event 3: CalculatorClient_1.BeginAdd(2, 3, null, null);");
IAsyncResult_1 = CalculatorClient_1.BeginAdd(2, 3, null, null);

lr.log("Event 5: CalculatorClient_1.EndAdd(IAsyncResult_1);");
Int32RetVal = CalculatorClient_1.EndAdd(IAsyncResult_1);
// Int32RetVal = 5;

lr.log("Event 6: ((ManualResetEvent)(IAsyncResult_1.AsyncWaitHandle));");
ManualResetEvent_1 = ((ManualResetEvent)(IAsyncResult_1.AsyncWaitHandle));

lr.log("Event 7: ManualResetEvent_1.Close();");
ManualResetEvent_1.Close();
```

注： 特定の記録オプションを指定してスクリプトを記録した後で、記録オプションを変更する場合、スクリプトを再度記録する必要はありません。代わりに、新しい設定でスクリプトを再生成します。詳細については、103 ページ「Vuser スクリプトの再生成」を参照してください。

WCF 全二重バインディング

WCF (Windows Communication Foundation) を使用しているアプリケーションについてオプションを設定できます。詳細については、『第 2 巻 – プロトコル』の第 35 章「Microsoft .NET プロトコル」を参照してください。

デバッグ・オプション

デバッグ・オプションではスタックを追跡してサイズを指定できます。

- ▶ **[スタック トレース]** : スクリプト内の各呼び出しのスタックの内容を追跡します。これにより、アプリケーションで使用されたクラスおよびメソッドを確認できます。これは、フィルタに追加する参照、名前空間、クラス、またはメソッドを調べるのに便利です。追跡を有効にすると、記録時のアプリケーションのパフォーマンスに影響します。標準設定では、追跡は無効になっています。
- ▶ **[スタック トレースの制限指定]** : スタックに格納される呼び出しの最大数。標準設定値は 20 件までです。呼び出しの数が上限を超えた場合、VuGen はそれらを切り捨てます。

フィルタ

[すべてのアセンブリを無視することを標準設定にする] : 選択したフィルタによって明示的に含まれていないすべてのアセンブリを無視します。このオプションが無効の場合、VuGen は、記録中にロードされる各アセンブリで一致するフィルタ・ルールを検索します。

コード生成

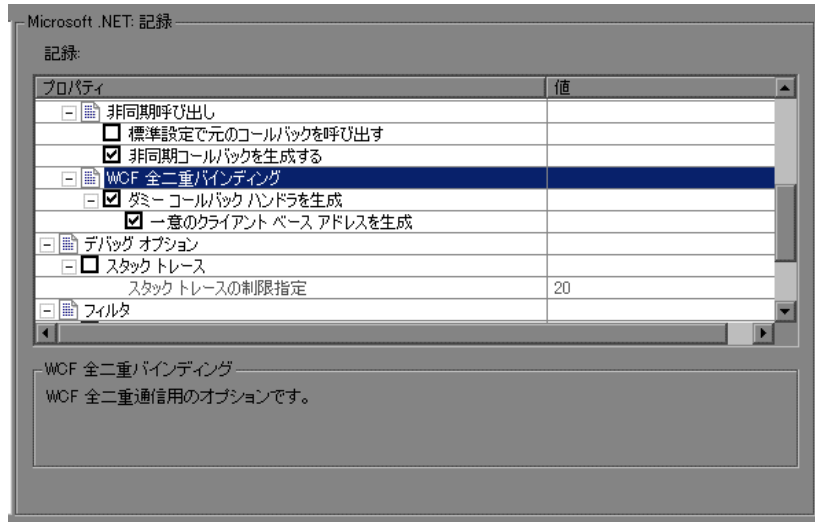
[コードの生成] オプションでは、コード生成中に警告およびスタック・トレースを表示するかどうかを指定できます。

- ▶ **[警告を表示する]** : コード生成中に発行された警告メッセージを表示します。
- ▶ **[スタック トレースを表示する]** : 記録されたスタック・トレースが存在する場合、表示します。
- ▶ **[すべてのイベント サブスクリプションを表示する]** : 記録されたすべてのイベント・サブスクリプションのコードを生成します (標準設定では無効)。このオプションが無効な場合、VuGen は、発行者 (イベントを発行するオブジェクト) と登録者 (イベントの通知を受け取るオブジェクト) の両方がフィルタに追加されているイベントのコードのみを生成します。

WCF 記録オプション

WCF 全二重通信の記録オプションにより、負荷テストに効果的なスクリプトを生成できます。次の項目について記録オプションの設定が可能です。

- ▶ ダミー・コールバック実装の生成
- ▶ 『第2巻 - プロトコル』のデュアル HTTP バインディングの記録



ダミー・コールバック実装の生成

[**ダミー コールバック ハンドラを生成**] 記録オプションを指定すると、全二重通信の元のコールバックをダミー・コールバックで置き換えるように設定されます。

ダミー・コールバック実装では、次のアクションが行われます。

- ▶ **引数の格納**。サーバは、再生中にハンドラを呼び出した場合、そのメソッドの引数をメモリ・マップのキー値に保存します。

- ▶ **再生の同期化**。次の応答が到着するまで、スクリプトの実行を停止します。VuGen は、記録中にコールバックが発生したポイントに同期化を配置します。スクリプトでは警告で示されます。

```
#warning: Code Generation Warning
// ここで次の応答を待機。
// 記録時の元のコールバックは次のとおり : ...
```

同期化の一部として、スクリプトは `GetNextResponse` を呼び出し、格納されている値を取得します。

```
Vuser<Callback_Name>.GetNextResponse();
```

ダミー・コールバック記録オプションの有効化

標準設定では、このオプションは有効になっています。

この記録オプションを有効にするには、次の手順を実行します。

- 1 [ツール] > [記録オプション] を選択します。
- 2 [Microsoft .NET: 記録] ノードを選択します。
[ダミー コールバック ハンドラを生成] を選択します。

RTE 設定オプション

ターミナル・エミュレーション中に使用される文字セットに合わせて、記録オプションを設定できます。標準設定の文字セットは ANSI です。漢字その他のマルチバイト・プラットフォームの場合は、DBCS（ダブルバイト文字セット）を指定できます。



[設定] 記録オプションを表示するには、ツールバーの [記録オプション] ボタンをクリックするか、[ツール] > [記録オプション] を選択します。[RTE] の [設定] ノードを選択します。

RTE: 設定

RTE 設定オプション:

| プロパティ | 値 |
|-------|------|
| 文字セット | ANSI |
| | |
| | |

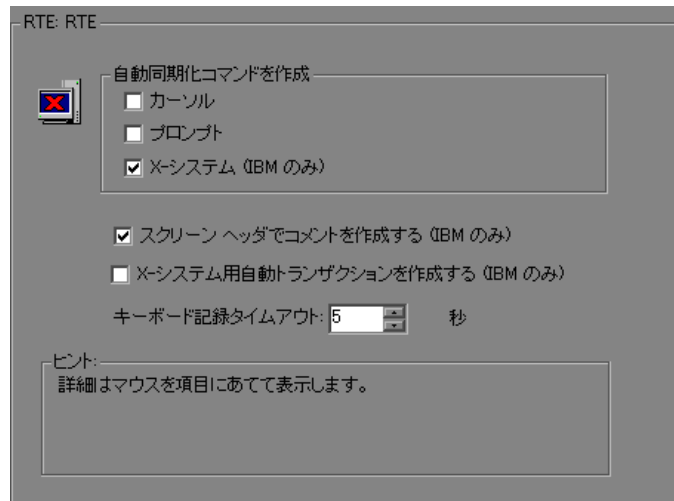
文字セット
ターミナル エミュレーションに使用される文字セット

RTE 記録オプション

記録オプションを設定することで、VuGen によって RTE 関数用に生成されるコードをカスタマイズできます。記録オプションの設定には [記録オプション] ダイアログ・ボックスを使います。



[記録オプション] ダイアログ・ボックスを表示するには、ツールバーの [記録オプション] ボタンをクリックするか、[ツール] > [記録オプション] を選択します。[RTE : RTE] ノードを選択します。



次の記録オプションを設定できます。

- ▶ 自動同期化コマンド
- ▶ 自動画面ヘッダ・コメント (IBM ターミナルのみ)
- ▶ 自動 X-System トランザクション (IBM ターミナルのみ)
- ▶ キーボード記録のタイムアウト

自動同期化コマンド

VuGen では、記録中にいくつかの TE 同期化関数を自動的に生成し、それらをスクリプトに挿入することができます。

- 1 記録中に新しい画面が表示されるたびに **TE_wait_sync** 関数を生成するよう、VuGen を設定できます。これには、[記録オプション] ダイアログ・ボックスの [**X-System**] チェック・ボックスを選択します。

標準設定では、記録中に新しい画面が表示されるたびに、VuGen によって **TE_wait_sync** 関数が生成されます。

注： IBM ブロック・モード・ターミナルのみを記録しているときは、VuGen によって **TE_wait_sync** 関数が生成されます。

- 2 それぞれの **TE_type** 関数の前に **TE_wait_cursor** 関数を生成するよう、VuGen を設定できます。これには、[記録オプション] ダイアログ・ボックスの [**カーソル**] チェック・ボックスを選択します。

標準設定では、**TE_wait_cursor** 関数は自動的に生成されません。

- 3 それぞれの **TE_type** 関数の前に **TE_wait_text** 関数を生成するよう、VuGen を設定できます (適切な場合)。これには、[記録オプション] ダイアログ・ボックスの [**プロンプト**] チェック・ボックスを選択します。

標準設定では、それぞれの **TE_type** 関数の前に **TE_wait_cursor** 関数が自動的に生成されません。

注： VT タイプのターミナルのみを記録しているときは、VuGen によって、意味のある **TE_wait_text** 関数が生成されます。ブロック・モード (IBM) ターミナルを記録しているときは、**TE_wait_text** 関数の自動生成は使用しないでください。

自動画面ヘッダ・コメント (IBM ターミナルのみ)

Vuser スクリプトの記録中に画面ヘッダ・コメントを自動的に生成し、それらのコメントをスクリプトに挿入するよう、VuGen を設定できます。

生成されたコメントをもとに、それぞれの新しい画面をターミナル・エミュレータ内で表示されているとおりに識別できるため、記録されたスクリプトが読みやすくなります。生成されるコメントには、ターミナル・エミュレータ・ウィンドウの 1 行目に表示されているテキストが格納されます。次のコメントは、Office Tasks という画面がターミナル・エミュレータに表示されていたことを示しています。

```
/* OFCTSK           Office Tasks           */
```

スクリプトの記録中に VuGen によってコメントが自動的に生成されるようにするには、[記録オプション] ダイアログ・ボックスの [スクリーンヘッダでコメントを作成する] チェック・ボックスを選択します。

標準設定では、画面コメントは自動的に生成されません。

注：コメントの自動生成は、IBM 5250 などのブロック・モードのターミナル・エミュレータを使用しているときのみ可能です。

自動 X-System トランザクション (IBM ターミナルのみ)

シナリオの実行中に、システムが X SYSTEM モードになっていた時間を記録するよう、VuGen を設定できます。この場合、VuGen によって、それぞれの **TE_wait_sync** 関数の後に **TE_wait_sync_transaction** 関数が挿入されます。**TE_wait_sync_transaction** 関数はそれぞれ、"default" という名前を持つトランザクションを作成します。**TE_wait_sync_transaction** 関数はそれぞれ、システムが以前の X SYSTEM 状態を保っていた時間を記録します。

記録中に **TE_wait_sync_transaction** ステートメントを挿入するよう VuGen を設定するには、[記録オプション] ダイアログ・ボックスの [**X-システム用自動トランザクションを作成する**] チェック・ボックスを選択します。

標準設定では、トランザクションは自動的に生成されません。

キーボード記録のタイムアウト

記録中にターミナル・エミュレータにテキストを入力すると、テキスト入力
VuGen によって監視されます。各キーストロークの後、VuGen は次のキース
トロークが発生するまで、指定の時間だけ待ちます。指定の時間内に次のキース
トロークが発生しなかった場合は、コマンドが完了したものとみなされます。
すると、VuGen によって、対応する **TE_type** 関数が生成され、スクリプトに挿
入されます。

連続するキーストロークの合間に VuGen が待つ時間の上限を設定するには、
[**キーボード記録タイムアウト**] ボックスにその時間量を入力します。

標準設定では、VuGen は連続するキーストロークの合間に、対応する **TE_type**
関数が生成されるまで最大 5 秒待ちます。

SAPGUI 記録オプション

記録オプションを使って、記録セッションのために SAP 関連の設定を行います。[記録オプション] ダイアログ・ボックスを開くには、[ツール] > [記録オプション] を選択するか、[記録開始] ダイアログ・ボックスで [オプション] ボタンをクリックします。キーボードのショートカット・キーは CTRL キー + F7 キーです。

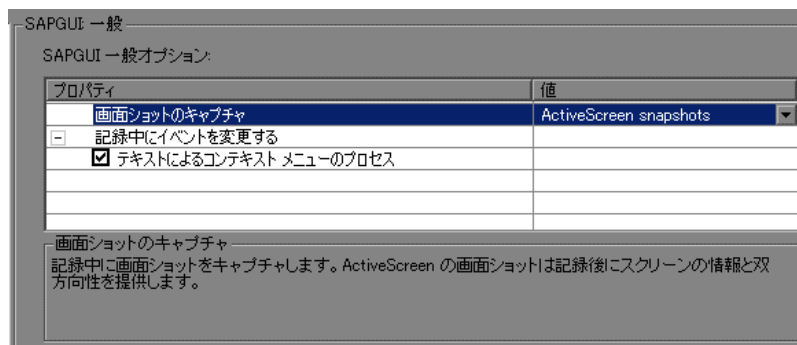
次の項目について記録オプションの設定が可能です。

- ▶ SAPGUI 一般記録オプション
- ▶ SAPGUI コード生成記録オプション
- ▶ SAPGUI 自動ログオン記録オプション

SAP-Web User タイプを使用してマルチ・プロトコルの User スクリプトを記録しようとしている場合は、その他の記録オプションについて 331 ページ「Web, ワイヤレス, Oracle NCA 記録オプション」を参照してください。

SAPGUI 一般記録オプション

これらの記録オプションを使って、次の項目について記録セッション中の一般的な設定を行います。



- ▶ **[画面ショットのキャプチャ]** : 記録中に表示される SAPGUI 画面のスナップショットの保存方法を指定します ([**ActiveScreen のスナップショット**], [**通常のスナップショット**], [**なし**])。ActiveScreen のスナップショットは記録後に双方向性と画面情報を提供しますが、より多くのリソースを必要とします。

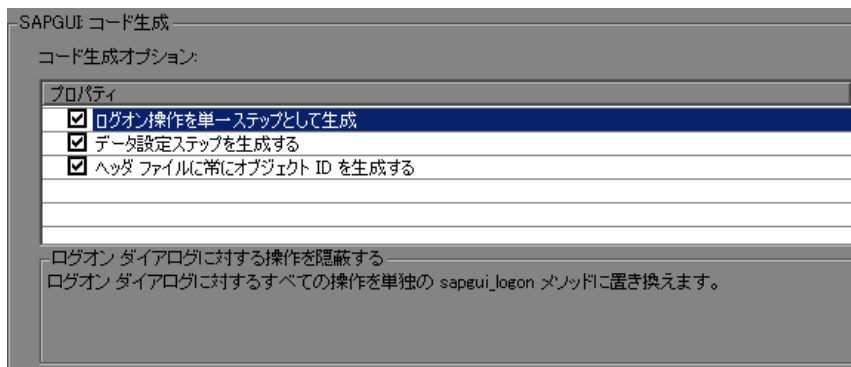
- ▶ **[テキストによるコンテキストメニューのプロセス]** : テキストによってコンテキスト・メニューを処理し、`sapgui_toolbar_select_context_menu_item_by_text` 関数を生成するように設定します。このオプションが無効のときは、IDによってコンテキスト・メニューが処理されます。これは、日本語文字を使用している場合に役立ちます。後者の場合は、コンテキスト・メニューに対して `sapgui_toolbar_select_context_menu_item` が生成されます。

[一般] 記録オプションを設定するには、次の手順を実行します。

- 1 [記録オプション] ダイアログ・ボックスを開き、**[SAPGUI : 一般]** ノードを選択します。
- 2 **[画面ショットのキャプチャ]** オプションのために、SAPGUI 画面のスナップショットの保存方法を指定します。
- 3 コンテキスト・メニューを処理する方法を選択します。**[テキストによるコンテキストメニューのプロセス]** を有効にします。IDによってコンテキスト・メニューを処理する場合は、このオプションを無効にします。
- 4 **[OK]** をクリックして設定を適用し、ダイアログ・ボックスを閉じます。

SAPGUI コード生成記録オプション

次の記録オプションは、コード生成の設定を行います。



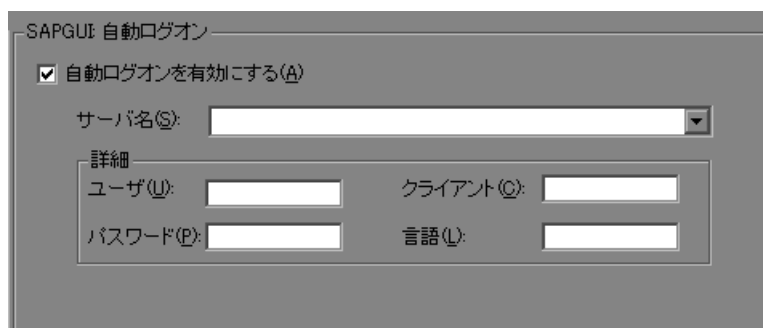
[コード生成] 記録オプションを設定するには、次の手順を実行します。

- 1 [記録オプション] ダイアログ・ボックスを開き、**[SAPGUI : コード生成]** ノードを選択します。

- 2 [ログオン操作を単一ステップとして生成] を選択し、すべてのログイン操作に対して、単一の `sapgui_logon` メソッドを生成するようにします。これによって、コードが簡略化されます。ログインで問題が生じた場合は、このオプションを無効にします。
- 3 各セルに対して個別のデータ挿入ステップを生成するのではなく、テーブルおよびグリッド・コントロールに対してデータ挿入ステップを生成するには、[データ設定ステップを生成する] を選択します。
- 4 よりコンパクトで読みやすいスクリプトを作成するには、[ヘッダファイルに常にオブジェクト ID を生成する] を選択します。これにより、オブジェクト ID がスクリプトではなく個別のヘッダ・ファイルに置かれるようになります。このオプションを無効にすると、一般スクリプト設定で指定された文字列の長さに従って ID が生成されます。このオプションを無効にしても読みやすくなるだけで、オーバーヘッドに違いはありません。
- 5 [OK] をクリックして設定を適用し、ダイアログ・ボックスを閉じます。

SAPGUI 自動ログオン記録オプション

これらの記録オプションを設定して、記録開始時に自動的にログオンするようにします。ログオン関数はスクリプトの `vuser_init` セクションに置かれます。サーバ名リストには SAP ログオン記述リスト上のすべてのサーバが含まれます。



[自動ログオン] 記録オプションを有効にして設定するには、次の手順を実行します。

- 1 [記録オプション] ダイアログ・ボックスを開き、[SAPGUI : コード生成] ノードを選択します。
- 2 [自動ログオンを有効にする] を選択します。

- 3 ログイン情報を入力します。
 - ▶ SAP サーバの名前。
 - ▶ SAP サーバの**ユーザ**の名前。
 - ▶ SAP サーバへログインするための**パスワード**。
 - ▶ MetaFrame サーバがクライアントの識別に使用する**クライアント名** (任意)。
 - ▶ インタフェースで使用する**言語**。
- 4 [OK] をクリックして設定を適用し、ダイアログ・ボックスを閉じます。

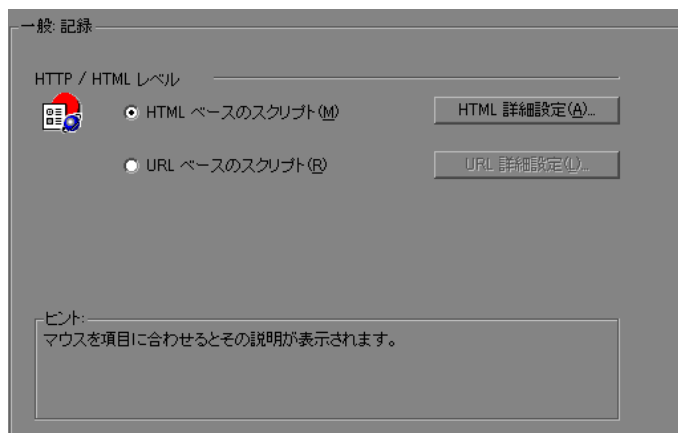
SAP-Web 記録オプション

記録オプションを使用して、VuGen による Vuser スクリプトの生成方法を設定します。

[一般：記録] ノードの推奨設定は次のとおりです。

SAP Workplace の記録の場合：[URL ベースのスクリプト]

SAP Portal の記録の場合：[HTML ベースのスクリプト] (標準設定)



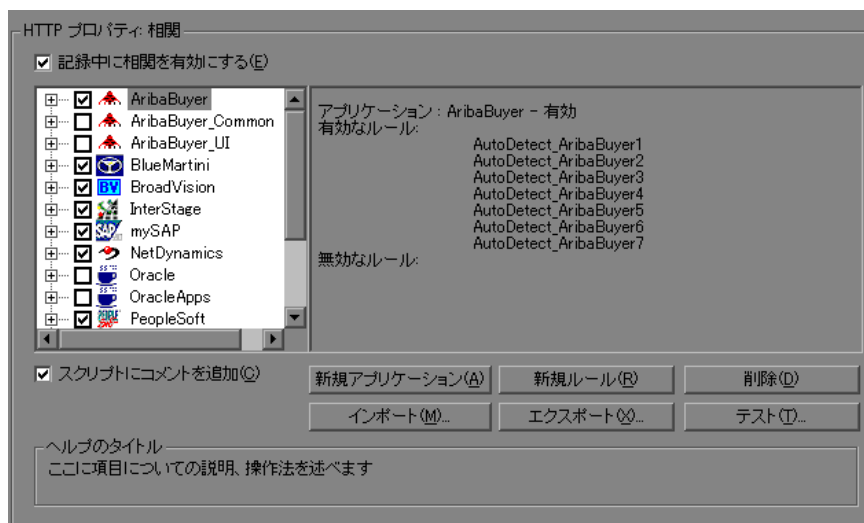
Web に関連する記録オプションの詳細については、331 ページ「Web, ワイヤレス, Oracle NCA 記録オプション」を参照してください。

関連記録オプションの設定

VuGen を使用して、記録中にステートメントを関連させるには、[**関連**] 記録オプションを設定します。これらのオプションは、Web Vuser スクリプトを開いた後、セッションの記録を開始する前に設定します。

[**関連**] 記録オプションを設定するには、次の手順を実行します。

- 1 スクリプトの作成後、記録を開始する前に [**ツール**] > [**記録オプション**] で [**HTTP プロパティ：関連**] ノードを選択します。



- 2 [**記録中に相関を有効にする**] オプションを選択します。
- 3 関連ルールを適用する対象となるサーバを指定します。サーバ名の隣のチェック・ボックスを選択し、指定サーバでルールを有効にします。サーバ・グループの特定のルールを有効にするには、「+」印をクリックしてツリーを展開し、必要なルールを選択します。
- 4 既存サーバに新しいルールを追加するには、既存エントリの中から 1 つを選択し、[**新規ルール**] をクリックします。ルールのプロパティは右側の表示枠で設定します。詳細については、『**第 2 巻 - プロトコル**』の 647 ページ「**関連のルールの設定**」を参照してください。
- 5 新しいアプリケーションにルールのセットを追加するには、[**新規アプリケーション**] をクリックします。次に [**新規ルール**] をクリックして、アプリケーションのルールを作成します。

- 6 既存ルールのプロパティを変更するには、ルールを左側の表示枠で選択し、右側の表示枠で変更をします。
- 7 アプリケーションやルールを削除するには、それを選択して **[削除]** ボタンをクリックします。選択した項目を削除する前に、VuGen から確認メッセージが表示されます。
- 8 関連ルールのセットをエクスポートするには、**[エクスポート]** をクリックして **.cor** ファイルを目的の場所に保存します。以前のセッションで作成した関連ルールのセットをインポートするには、**[インポート]** をクリックしてその場所からファイルを開きます。
- 9 **[OK]** をクリックします。

Web, ワイヤレス, Oracle NCA 記録オプション

[HTTP プロパティ: 詳細] では、次の設定が可能です。

- ▶ [お気に入り] インターネット記録オプション
- ▶ 記録エンジンの選択
- ▶ 記録スキーマの設定

[お気に入り] インターネット記録オプション

[お気に入り] オプションを使用して、思考遅延時間、コンテキストのリセット、スナップショットの保存、および **web_reg_find** 関数にかかわるコード生成の方法をカスタマイズできます。オプションのいくつかは、マルチ・プロトコル・モードでは利用できません。

- ▶ [思考遅延時間を記録する]: (ワイヤレスの場合のみ) この設定は標準で有効になっており、VuGen に対して、思考遅延時間を記録し、**lr_think_time** 関数を生成するよう指示します。また、[思考遅延時間しきい値] を設定することもできます。実際の思考遅延時間がしきい値より短かった場合、**lr_think_time** 関数は生成されません。
- ▶ [各アクションごとにコンテキストをリセットする]: (Web, Oracle NCA の場合のみ) この設定は、標準で有効になっており、VuGen に対してアクションごとに HTTP コンテキストをすべてリセットするよう指示します。この設定により、Vuser は新規ユーザがブラウザ・セッションを開始する様子をより正確にエミュレートできます。このオプションは、HTML コンテキストをリセットし、コンテキストを持たない関数が常にアクションの先頭に記録されるようにします。また、このオプションは、キャッシュをクリアし、ユーザ名とパスワードをリセットします。
- ▶ [スナップショットのリソースをローカルに保存する]: VuGen によって記録時と再生時にスナップショット・リソースのローカル・コピーが作成されます。この機能によって、より正確なスナップショットが作成され、よりすばやく表示することができます。
- ▶ [ページタイトルで **web_reg_find** 関数を生成する]: (Web, Oracle NCA の場合のみ) すべての HTML ページのタイトルに対して、**web_reg_find** 関数の生成を有効にします。VuGen によって、ページのタイトル・タグから文字列が取得され、それが **web_reg_find** の引数として使用されます。

- ▶ **[サブフレームで web_reg_find 関数を生成]** : 記録されたページのすべてのサブフレームにあるページのタイトルに対して、**web_reg_find** 関数の生成を有効にします。
- ▶ **[記録中、HTTP エラー時にスクリプトにコメントを追加する]** : HTTP 要求のエラーが発生するたびにスクリプトにコメントを追加します。エラー要求は、記録中に 400 番以降の値を持つサーバ応答を生成した要求として定義されます。
- ▶ **[サポート対象文字セット]** :
 - ▶ **[UTF-8]** : UTF-8 エンコーディングのサポートを有効にします。この設定を有効にすると、非 ASCII の UTF-8 文字がマシンのロケールで使用されているロケールに変換され、VuGen に正しく表示されます。このオプションは英語以外の UTF-8 エンコードのページでのみ有効にしてください。記録するサイトの言語がオペレーティング・システムの言語と一致している必要があります。英語以外の Web ページは、同じスクリプト内で異なるエンコーディング (ISO-8859-1 または shift_jis を組み合わせて) では記録できません。
 - ▶ **[EUC-JP]** : 日本語版 Windows のユーザは、このオプションを選択することで、EUC-JP 文字エンコーディングを使用している Web サイトに対するサポートを有効にできます。この設定を有効にすると、EUC-JP 文字がマシンのロケールで使用されているロケールに変換され、VuGen に正しく表示されます。VuGen によって、すべての EUC-JP (日本語 UNIX) 文字列をマシンのロケールに合わせて Shift-JIS (日本語版 Windows) エンコーディングに変換し、スクリプトに **web_sjis_to_euc_param** 関数を追加します (漢字のみ)。

記録エンジンの選択

VuGen の標準設定では、Web (HTTP/HTML) Vuser に対して、シングル・プロトコルの記録を含むすべての記録でマルチプロトコルの記録エンジンが使用されます。

下位互換性を維持するためにシングル・プロトコルの記録エンジンを使用するには、[詳細記録オプション] の **[以前の記録エンジンを使用してスクリプトを記録する]** オプションを選択します。このオプションを有効にすると、次の Web (HTTP/HTML) セッション記録でシングル・プロトコルが使用されます。

記録スキーマの設定

次の領域で記録スキーマを指定することによって、記録方法をさらにカスタマイズできます。

- ▶ ユーザ定義ヘッダの記録
- ▶ コンテント・タイプのフィルタリング
- ▶ リソース以外のコンテント・タイプの指定

ユーザ定義ヘッダの記録

Web Vuser は、サーバに送信されるすべての HTTP 要求とともに、いくつかの標準的な HTTP ヘッダを自動的に送信します。その他の HTTP ヘッダを記録するには、**[ヘッダ]** をクリックします。次の 3 つのモードがあります。**[ヘッダは記録しない]**、**[リスト内のヘッダを記録]**、**[リストに定義されていないヘッダを記録]** の 3 つです。最初のモードでは、ヘッダが記録されません。2 つ目のモードでは、チェック・マークを入れたユーザ定義ヘッダだけが記録されます。**[リストに定義されていないヘッダを記録]** を指定すると、チェック・マークを入れたヘッダと、ほかのリスクー・ヘッダを除くすべてのユーザ定義ヘッダが記録されます。

「リスクー・ヘッダ」 と呼ばれる標準ヘッダには、「Authorization」、
「Connection」、
「Content-Length」、
「Cookie」、
「Host」、
「If-Modified-Since」、
「Proxy-Authenticate」、
「Proxy-Authorization」、
「Proxy-Connection」、
「Referer」、
「WWW-Authenticate」があります。**[ヘッダリスト]** で選択しないかぎり、これらのヘッダは記録されません。標準設定は、**[ヘッダを記録しない]** です。

[リスト内のヘッダを記録] モードでは、チェック・マークを入れたヘッダが検出され、それらのヘッダに対してスクリプトに `web_add_auto_header` 関数が挿入されます。これは、明示的な指定のないかぎり記録されないリスクー・ヘッダを記録する場合に理想的なモードです。

[リストに定義されていないヘッダを記録] モードでは、チェック・マークを入れていないヘッダが記録中に検出され、それらのヘッダに対してスクリプトに `web_add_auto_header` 関数が挿入されます。

記録するユーザ定義ヘッダを決めるときに、すべてのヘッダを記録するように VuGen に指定して記録セッションを実行できます（下記の手順を参照）。その後、記録するヘッダと除外するヘッダを決定します。

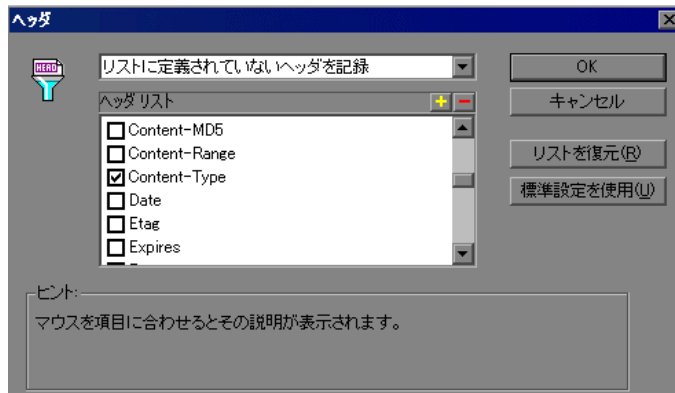
前出の画面では、[リスト内のヘッダを記録] モードで「Content-type」のヘッダを指定しています。VuGen によってヘッダが検出され、次のステートメントがスクリプトに追加されます。

```
web_add_auto_header("Content-Type"  
                    "application/x-www-form-urlencoded");
```

これはサーバに対して、アプリケーションの「Content-Type」が x-www-form-urlencoded であることを示しています。

ユーザ定義ヘッダの記録を制御するには、次の手順を実行します。

- 1 [記録オプション] ツリーで、[HTTP プロパティ: 詳細] ノードを選択します。
- 2 [ヘッダ] をクリックします。[ヘッダ] ダイアログ・ボックスが開きます。



- 3 次のいずれかの方法を使用します。
 - ▶ ヘッダを記録しないように設定するには、[ヘッダを記録しない] を選択します。
 - ▶ 特定のヘッダだけを記録するには、[リスト内のヘッダを記録] を選択し、ヘッダ・リストから必要なユーザ定義のヘッダを選びます。標準設定では、標準ヘッダ (**Accept** など) が選択されています。
 - ▶ すべてのヘッダを記録するには、[リストに定義されていないヘッダを記録] を選択し、リストからは項目を選択しません。

- ▶ 特定のヘッダだけを除外するには、**[リストに定義されていないヘッダを記録]** を選択し、除外するヘッダを選択します。
- 4 リストを対応する標準設定のリストに戻すには、**[リストを復元]** をクリックします。**[リスト内のヘッダを記録]** と **[リストに定義されていないヘッダを記録]** のそれぞれに、対応する標準設定のリストがあります。
- 5 **[OK]** をクリックして設定を受け入れ、**[ヘッダ]** ダイアログ・ボックスを閉じます。

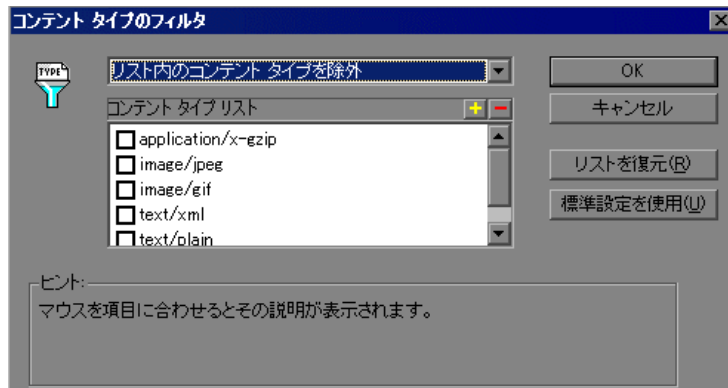
コンテンツ・タイプのフィルタリング

VuGen を使って、記録したスクリプトのコンテンツ・タイプをフィルタリングできます。記録するコンテンツ・タイプ、またはスクリプトから除外するコンテンツ・タイプを指定します。次の 3 つのモードがあります。**[コンテンツタイプをフィルタにかけない]**、**[リスト内のコンテンツタイプを除外]**、**[リストに定義されていないコンテンツタイプを除外]** の 3 つです。最初のモードで作業すると、コンテンツ・タイプがフィルタリングされません。2 つ目のモードでは、選択されたコンテンツ・タイプだけが除外されます。**[リストに定義されていないコンテンツタイプを除外]** を指定すると、チェック・マークを入れたコンテンツ・タイプ以外のすべてのコンテンツ・タイプが除外されます。標準では、フィルタは設定されていません。

たとえば、Web サイトのテキストと画像だけを対象とするには、**[リストに定義されていないコンテンツタイプを除外]** を選択し、タイプとして「**text/html**」、**image/gif**、**image/jpeg** を指定します。VuGen によってすべての HTML ページと画像が記録され、サイトに表示される「**text/css**」、**application/x-java スクリプト**」などのリソースが除外されます。

記録中に内容をフィルタリングするには、次の手順を実行します。

- 1 [記録オプション] ツリーで、[HTTP プロパティ：詳細] ノードを選択します。
- 2 [コンテンツタイプ] をクリックします。[コンテンツタイプのフィルタ] ダイアログ・ボックスが開きます。



- 3 次のいずれかの方法を使用します。
 - ▶ 内容をフィルタリングしないように設定するには、[コンテンツタイプをフィルタにかけない] を選択します。
 - ▶ 特定のコンテンツ・タイプだけを記録対象から除外するには、[リスト内のコンテンツタイプを除外] を選び、記録対象から除外するコンテンツ・タイプをリストから選択します。
 - ▶ 特定のコンテンツ・タイプだけを記録するには、[リストに定義されていないコンテンツタイプを除外] を選び、記録するコンテンツ・タイプを選択します。
- 4 リストを対応する標準設定のリストに戻すには、[リストを復元] をクリックします。[リスト内のコンテンツタイプを除外] と [リストに定義されていないコンテンツタイプを除外] のそれぞれに、対応する標準設定のリストがあります。
- 5 [OK] をクリックして設定を受け入れ、[コンテンツタイプのフィルタ] ダイアログ・ボックスを閉じます。

リソース以外のコンテンツ・タイプの指定

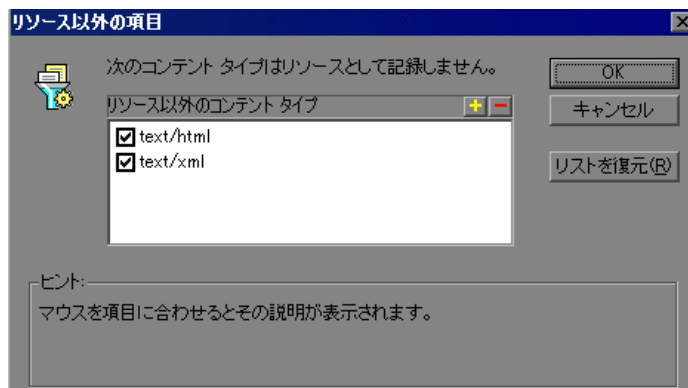
スクリプトを記録するときに、VuGen によって `web_url` 関数の `Resource` 属性を使って、再生中に対象リソースを取得するかどうかが表示されます。`Resource` 属性が 0 に設定されていると、対象リソースはスクリプトの実行中に取得されます。`Resource` 属性が 1 に設定されていると、そのリソース・タイプは `Vuser` によってスキップされます。

```
web_url("WebTours",
        "URL=http://localhost/WebTours/",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=",
        "Snapshot=t1.inf",
        "Mode=HTML",
        LAST);
```

特定のコンテンツ・タイプをリソースとして処理しないように除外することができます。たとえば、`gif` タイプのリソースがリソースとして処理されないように設定し、無条件にダウンロードされるように設定できます。VuGen によって `gif` タイプのリソースが検出されると、`Resource` 属性が 0 に設定され、再生時に `gif` を無条件にダウンロードされるようになります。

リソースとして記録しない内容を指定するには、次の手順を実行します。

- 1 [記録オプション] ツリーで、[HTTP プロパティ：詳細] ノードを選択します。
- 2 [リソース以外の項目] をクリックしてダイアログ・ボックスを開き、リソースとして記録しないコンテンツ・タイプのリストを表示します。



- 3 リストにコンテンツ・タイプを追加するには、「+」記号をクリックします。既存のエントリを削除するには、「-」をクリックします。
- 4 項目を有効にするには、その横のチェック・ボックスを選択します。
- 5 リストを標準設定のリストに戻すには、**[リストを復元]** をクリックします。
- 6 **[OK]** をクリックして設定を受け入れ、**[リソース以外の項目]** ダイアログ・ボックスを閉じます。

第 17 章

Click and Script 記録

Click and Script ソリューションでは、ユーザ・アクションに基づいて Web セッションを記録できます。本章では、Click and Script 記録と記録オプションについて説明します。

本章の内容

- ▶ Click and Script による記録について (339 ページ)
- ▶ Web (HTTP/HTML) Vuser スクリプトの表示 (340 ページ)
- ▶ Click and Script 記録オプションの設定 (341 ページ)
- ▶ GUI プロパティの詳細設定 (342 ページ)
- ▶ Web イベント記録の定義 (345 ページ)

以降の情報は、AJAX (Click and Script)、Web (Click and Script)、SAP (Click and Script)、Oracle Web Applications 11i、および PeopleSoft Enterprise の各プロトコルを対象とします。

Click and Script による記録について

Click and Script メカニズムでは、ユーザ・アクション GUI レベルで Web セッションを記録できます。VuGen は、Web インタフェースを対象としたアクションを直感的に表現する GUI レベルのスクリプトを作成します。たとえば、ボタンをクリックして情報を送信すると **web_button** 関数が生成され、エディット・ボックスにテキストを入力すると **web_edit_field** 関数が生成されます。SAP アプリケーションの場合は、**sap_button** 関数が記録されます。

純粋な Web セッションに対しては、Web (HTTP/HTML) Vuser タイプによって低レベルのスクリプトが作成されます。Web セッションに対するスクリプト・タイプの選択の詳細については、『第 2 巻 – プロトコル』の第 37 章「Web (HTTP/HTML, Click and Script) プロトコル」を参照してください。

Web (HTTP/HTML) Vuser スクリプトの表示

通常、Click and Script Vuser スクリプトには、ビジネス・プロセスを構成する複数のアクションが含まれています。GUI レベルで生成された記録済みの関数を見れば、記録されたセッションの間のユーザの正確なアクションを知ることができます。

たとえば、標準的な記録では、第 1 段階にサインインのプロセスが含まれています。ブラウザでサインイン・ページが開いたら、ユーザはユーザ名とパスワードを入力し、[Sign In] をクリックしてサインインします。

Web (Click and Script) Vuser では、エディット・フィールドに入力されたデータを表す `web_edit_field` 関数が生成されます。次の例では、ユーザは [userid] フィールドにテキストを入力し、暗号化される [pwd] フィールドにパスワードを入力しています。

```
vuser_init() {
    web_browser("WebTours",
        DESCRIPTION,
        ACTION,
        "Navigate=http://localhost:1080/WebTours/",
        LAST);

    web_edit_field("username",
        "Snapshot=t2.inf",
        DESCRIPTION,
        "Type=text",
        "Name=username",
        "FrameName=navbar",
        ACTION,
        "SetValue=jojo",
        LAST);

    web_edit_field("password",
        "Snapshot=t3.inf",
        DESCRIPTION,
        "Type=password",
        "Name=password",
        "FrameName=navbar",
        ACTION,
        "SetEncryptedValue=440315c7c093c20e",
        LAST);...
```

Click and Script 記録オプションの設定

スクリプトの記録を開始する前に、記録の対象と、記録後にスクリプトを生成する方法を指定するオプションを設定できます。

一般、HTTP プロパティ、ネットワークの項目について共通の記録オプションの設定が可能です。

次の項では、Click and Script メカニズムを使用するすべての Vuser タイプ (AJAX (Click and Script), Web (Click and Script), SAP (Click and Script), Oracle Applications, および PeopleSoft Enterprise の Vuser) に固有の GUI プロパティ記録オプションについて説明します。これらの記録オプションでは、記録対象イベント、および、各オブジェクトに含めるプロパティを指定できます。

- ▶ GUI プロパティの詳細設定
- ▶ Web イベント記録の定義

ほかの記録オプションの詳細については、該当する次の各項を参照してください。

- ▶ **[一般：スクリプト]**：第 19 章「スクリプト生成オプションの設定」を参照してください。
- ▶ **[一般：記録]**：第 20 章「Web Vuser の記録オプションの設定」を参照してください。
- ▶ **[ネットワーク：ポートの割り当て]**：395 ページ「ポートの割り当て設定」を参照してください。
- ▶ **[HTTP プロパティ：詳細]**：331 ページ「Web, ワイヤレス, Oracle NCA 記録オプション」を参照してください。**[スナップショットのリソースをローカルに保存する]** および **[ページタイトルで web_reg_find 関数を生成する]** オプションは GUI ベースのスクリプトには適用されません (次の GUI ベースのスクリプトの説明を参照)。
- ▶ **[HTTP プロパティ：相関]**：『第 2 巻 – プロトコル』のを参照してください。Oracle および PeopleSoft のサーバに対しては組み込みのルールが存在します。それらのルールを有効にするには、サーバ名の横にあるチェック・ボックスを選択します。

ほかにいくつかある HTTP プロパティは、Web (HTTP/HTML) スクリプトの場合にのみ設定できます。

GUI プロパティの詳細設定

VuGen では、次の項目について Click and Script 詳細オプションを設定できます。

- ▶ 記録設定
- ▶ コード生成設定

記録設定

記録設定により、何を記録するかを指定します。次の機能を有効または無効にできます。

[描画関連のプロパティ値の記録]

DOM オブジェクトのレンダリング関連プロパティ (**offsetTop** など) の値を記録して、再生時に利用できるようにします。これにより、再生速度が大幅に低下することがあります (標準設定では無効)。

[マウス イベントによる「クリック」を記録する]

Click () メソッドではなく、マウス・イベントをキャプチャすることでマウスのクリックを記録します。記録対象アプリケーションで DOM の click() メソッドが使用されている場合に有効にします。同じユーザ・アクションに対して複数の関数が生成されるのを防ぎます (標準設定では有効)。

[ソケット レベル データを記録する]

ソケット・レベル・データの記録を有効にします。このオプションを無効にした場合は、記録の前に、開始 URL を手動で追加する必要があります。また、スクリプトを HTML レベルで再生できなくなります (標準設定では有効)。

[AJAX ステップ用にスナップショットを生成]

AJAX ステップのスナップショットの生成を有効にします。このオプションは、標準設定では無効になっています。このオプションを有効にすると、記録中にエラーが発生することがあります。

コード生成設定

コード生成設定により，記録後のスクリプトの生成方法を指定できます。次の機能を有効または無効にできます。

[コンテキスト外ステップの生成を有効にする]

ActiveX コントロールや Java アプレットについては，URL ベースのスクリプトを作成するよう VuGen に指示することで，再生できます。これらの関数はネイティブの記録の一部ではないため，コンテキスト外の記録と呼ばれます（標準設定で無効）。

次の例は，コンテキスト外の記録オプションを有効にしてスクリプトを再生成したものです。

```
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);

...
```

このオプションを無効にすると、VuGen は、ActiveX コントロールおよび Java アプレットのコードを生成しません。次の例では、**web_image_link** 関数のみが生成され、クラス・ファイルが含まれる **web_url** 関数は生成されません。

```
web_image_link("Search Flights Button",
    "Snapshot=t5.inf",
    DESCRIPTION,
    "Alt=Search Flights Button",
    "FrameName=navbar",
    ACTION,
    "ClickCoordinates=58,9",
    LAST);
```

[ブラウザ タイトルの自動検証を有効にする]

ブラウザ・タイトルの自動検証を有効にします（標準設定では無効）。

タイトル検証のタイプをカスタマイズすることもできます。

▶ 次のタイピングでタイトル検証を行います。

[ナビゲーションのたびに]：ナビゲーションの後にのみ、タイトル検証を行います。フィールドが複数あるフォームの入力など、ユーザが同じページでいくつかの操作を実行した場合、タイトルは変わらないため、検証は不要です。

[ステップごとに]：ステップごとにタイトル検証を行い、ステップによってブラウザのタイトルが変更されていないことを確認します。ブラウザのタイトルが変更されると、スクリプトが失敗する原因となることがあります。

▶ [タイトルがない場合に、URL を使用してタイトル検証を行います。]：ブラウザ・ウィンドウにタイトルがない場合に、その URL を使用してステップごとにタイトル検証を行います。

Web イベント記録の定義

VuGen は、ドキュメントの表示中のマウスのクリックやキーの押下などのユーザ・アクションをきっかけとする HTML オブジェクト・イベントを記録することでスクリプトを作成します。

記録する必要のあるイベントが、VuGen の標準設定で自動的に記録されるものより多い場合や少ない場合があります。そのような場合、[Web イベント記録の定義] ダイアログ・ボックスで、3 つの定義済みの設定から、いずれかを選択することで標準のイベント記録設定を変更できます。また、特定の条件に合わせて、イベント記録の定義を個別にカスタマイズすることもできます。

本項では、Web イベントの VuGen による処理の設定方法について説明します。

- ▶ 標準で使用するイベント記録設定の選択
- ▶ イベント記録設定のカスタマイズ
- ▶ オブジェクトのリッスン・イベントの追加と削除
- ▶ イベントのリッスン設定と記録設定の変更
- ▶ イベント記録設定のリセット

たとえば VuGen では、通常はリンク・オブジェクト上の `mouseover` イベントは記録されません。しかし、マウスを対象の上に移動するマウスオーバ・ハンドラがリンクに関連付けられている場合は、`mouseover` イベントを記録することが重要になるかもしれません。この場合、リンク・オブジェクトがハンドラに関連付けられている場合には、リンク・オブジェクト上の `mouseover` イベントが必ず記録されるように、設定をカスタマイズできます。

注： イベント設定はグローバルな設定のため、設定を変更した後で記録されるすべてのテストに影響します。

イベント設定の変更は、すでに記録されたテストには影響しません。必要なイベントが VuGen で記録されなかった場合や、不要なイベントが記録された場合は、イベント記録設定を変更し、テストの中でその変更の影響を受ける部分を再度記録します。

[ユーザ定義 Web イベント記録設定] の設定に対する変更は、ブラウザを開いているときには効果がありません。[Web イベント記録の定義] ダイアログ・ボックスに必要な変更を行い、その変更を既存のテストに適用するには、開いているブラウザを更新し、新しい記録セッションを開始します。

標準で使用するイベント記録設定の選択

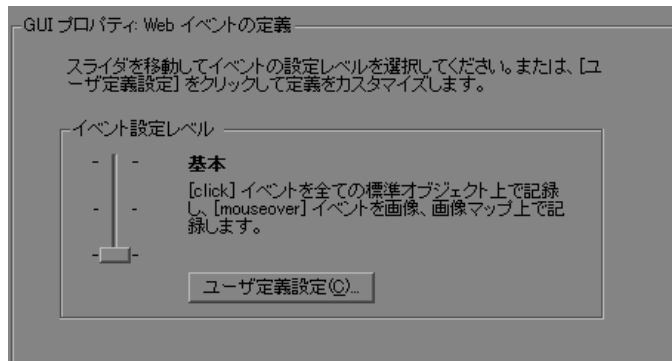
[Web イベント記録の定義] ダイアログ・ボックスでは、[基本]、[中]、[高] の 3 つの定義済みのイベント設定レベルから選択できます。標準設定の場合、VuGen では基本設定レベルから使用されます。必要なすべてのイベントが VuGen で記録されない場合は、より高いレベルのイベント設定が必要になります。

| レベル | 説明 |
|------|---|
| [基本] | <p>標準設定</p> <ul style="list-style-type: none"> ▶ 画像、ボタン、ラジオ・ボタンなどの標準的な Web オブジェクトに対するクリック・イベントを必ず記録します。 ▶ フォーム内での送信イベントを必ず記録します。 ▶ ハンドラまたは動作が関連付けられているその他のオブジェクトでのクリック・イベントを記録します。ハンドラおよび動作の詳細については、352 ページ「リッスン条件」を参照してください。 ▶ イメージおよびイメージ・マップ上の mouseover イベントは、そのイベントの後に同じオブジェクトに対するイベントが実行される場合にのみ記録します。 |
| [中] | <p>基本レベルで記録されるオブジェクトに加えて、HTML タグ・オブジェクトの <DIV>、、<TD> に対するクリック・イベントも記録します。</p> |

| レベル | 説明 |
|-----|--|
| [高] | <p>中位レベルで記録されるオブジェクトに加えて、ハンドラまたは動作が関連付けられているオブジェクトに対する <code>mouseover</code> イベント、<code>mousedown</code> イベント、および <code>double-click</code> イベントを記録します。</p> <p>ハンドラおよび振る舞いの詳細については、352 ページ「リッスン条件」を参照してください。</p> |

標準で使用するイベント記録の定義を設定するには、次の手順を実行します。

- 1 [記録オプション] ダイアログ・ボックスを開きます。[ツール] > [記録オプション] を選択します。
- 2 [GUI プロパティ : Web イベント設定] ノードを選択します。



- 3 スライダーを使って、標準で使用するイベント記録設定を選択します。[基本]、[中]、[高] を選択できます。

ヒント : [ユーザ定義設定] ボタンをクリックして、[ユーザ定義 Web イベント記録設定] ダイアログ・ボックスを開きます。ここでイベント記録の設定を定義できます。詳細については、「イベント記録設定のカスタマイズ」を参照してください。

- 4 [OK] をクリックします。

イベント記録設定のカスタマイズ

標準のイベント設定レベルに必要な記録が行われない場合は、[ユーザ定義 Web イベント記録設定] ダイアログ・ボックスを使って、イベント記録設定をカスタマイズできます。

画像、リンク、WebArea、WebButton など、標準的な Web 要素に対する Web イベントをカスタマイズできます。また、選択した HTML タグの記録動作を設定することもできます。必要な HTML タグ・オブジェクトを追加してから、その記録動作を設定します。たとえば、すべての **DIV** タグのすべての **mouseover** イベントを記録するように **VuGen** を設定できます。

[ユーザ定義 Web イベント記録設定] ダイアログ・ボックスでは、いくつかの方法でイベント記録をカスタマイズできます。次の設定ができます。

- ▶ **VuGen** で特別なリスン設定または記録設定を適用する対象となるオブジェクトの有効化または無効化。
- ▶ **VuGen** がリスンするべきイベントの追加または削除。
- ▶ イベントのリスンと記録の設定変更。

次のオプションを使って、イベント記録設定を変更できます。

| オプション | 説明 |
|---------|--|
| [イベント名] | <p>オブジェクトに関連付けられているイベントのリストが表示されます。</p> <ul style="list-style-type: none"> ▶ イベント表示枠にイベントを追加するには、[イベント] > [追加] を選択します。使用するイベントを選択します。 ▶ イベントを削除するには、[イベント名] カラムでイベントをクリックし、[イベント] > [削除] を選択します。 <p>詳細については、351 ページ「オブジェクトのリスン・イベントの追加と削除」を参照してください。</p> |

| オプション | 説明 |
|--------|---|
| [応答] | <p>VuGen がいつイベントをリッスンするかの基準</p> <ul style="list-style-type: none"> ▶ [常に] : 常にイベントをリッスンします。 ▶ [ハンドラの場合] : ハンドラが付加されているイベントをリッスンします。ハンドラは、Web ページに含まれているコードであり、通常はスクリプト言語で書かれている関数またはルーチンです。対応するイベントが発生したときに制御が渡されます。 ▶ [動作の場合] : DHTML 動作が付加されているイベントをリッスンします。DHTML 動作は、ページにおける特定の機能や振る舞いをカプセル化します。ページ上の標準的な HTML 要素に適用されている場合、その要素の標準設定の動作が拡張されます。 ▶ [ハンドラまたは動作の場合] : ハンドラまたは動作が付加されているイベントをリッスンします。 ▶ [しない] : イベントを一切リッスンしません。 <p>詳細については、352 ページ「イベントのリッスン設定と記録設定の変更」を参照してください。</p> |
| [記録] | <p>選択されたオブジェクトのイベントの記録を有効 / 無効にする、あるいは同じオブジェクトに対してその後イベントが発生した場合のみイベントの記録を有効 / 無効にします。</p> |
| [リセット] | <p>あらかじめ設定されているレベルにリセットします。[基本]、[中]、[高] を選択できます。</p> |

イベント記録の設定をカスタマイズするには、次の手順を実行します。

- 1 [記録オプション] ダイアログ・ボックスを開きます。[ツール] > [記録オプション] を選択します。
- 2 [GUI プロパティ : Web イベント設定] ノードを選択します。

- 3 **[ユーザ定義設定]** ボタンをクリックします。**[ユーザ定義 Web イベント記録設定]** ダイアログ・ボックスが開きます。



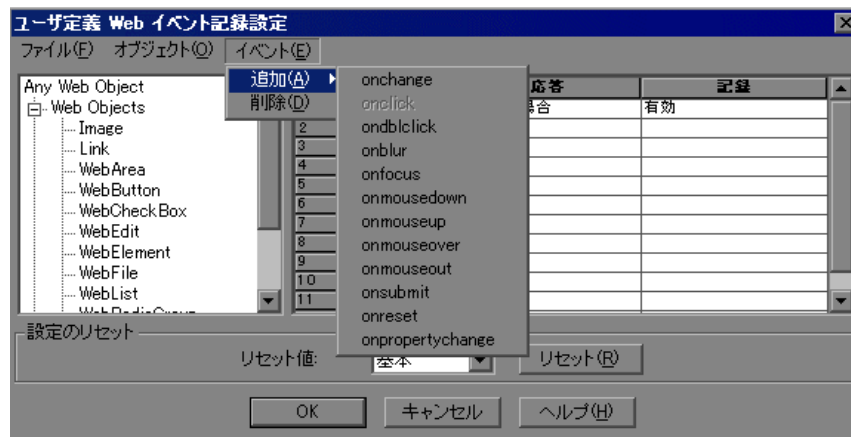
- 4 オブジェクトを指定します。
- ▶ 組み込み Web オブジェクトのいずれかを設定するには、左側の表示枠でそのオブジェクトを選択します。
 - ▶ 別の HTML タグ・オブジェクトを指定するには、**[オブジェクト]** > **[追加]** を選択し、イベントを記録する HTML タグの名前を入力します。
- 5 前述のように、右側の表示枠で、イベントごとに **[リッスン]** と **[記録]** の動作を指定します。記録したいイベントが右側の表示枠に表示されない場合は、後述のとおり、**[イベント]** > **[追加]** を選択してイベントを追加します。
- 6 **[OK]** をクリックしてカスタマイズを保存します。
- 7 あらかじめ設定されているレベルにリセットするには、**[リセット]** をクリックします。**[基本]**、**[中]**、**[高]** を選択できます。

オブジェクトのリッスン・イベントの追加と削除

VuGen がオブジェクトへリッスンするトリガとなるイベントのリストを変更できます。

オブジェクトにリッスン・イベントを追加するには、次の手順を実行します。

- 1 [ユーザ定義 Web イベント記録設定] ダイアログ・ボックスから、イベントを追加するオブジェクト（組み込み Web オブジェクトのいずれか、または HTML タグ・オブジェクト）を選択します。
- 2 [イベント] > [追加] を選択します。使用できるイベントのリストが開きます。



- 3 追加するイベントを選択します。イベントは [イベント名] 列にアルファベット順で表示されます。標準設定では、VuGen はハンドラが付加されているイベントをリッスンし、そのイベントを（それが何らかのレベルでリッスンされている限り）必ず記録します。

リッスン設定および記録設定の詳細については、「イベントのリッスン設定と記録設定の変更」下記を参照してください。

オブジェクトからリッスン・イベントを削除するには、次の手順を実行します。

- 1 [ユーザ定義 Web イベント記録設定] ダイアログ・ボックスの左側の表示枠から、イベントを削除するオブジェクトを選択します。
- 2 削除するイベントを [イベント名] カラムから選択します。
- 3 [イベント] > [削除] を選択します。[イベント名] 列からイベントが削除されます。

イベントのリッスン設定と記録設定の変更

オブジェクトごとにリスト表示される各イベントについて、リッスン条件を選択し、記録するかどうかを設定できます。

注：リッスンと記録の設定は相互に独立です。つまり、オブジェクトに対するイベントをリッスンしても、それを記録しないことや、オブジェクトに対するイベントをリッスンせずに、そのイベントを記録することができます。詳細については、354 ページ「イベントのリッスンと記録を行うためのヒント」を参照してください。

リッスン条件

イベントごとに、VuGen がいつイベントをリッスンするか指定できます。

- ▶ **[常に]**：オブジェクトでイベントが発生するたびにリッスンします。
- ▶ **[ハンドラの場合]**：イベントにイベント・ハンドラが関連付けられている場合にのみリッスンします。
- ▶ **[動作の場合]**：イベントに DHTML 動作が関連付けられている場合にのみリッスンします。
- ▶ **[ハンドラまたは動作の場合]**：イベントにハンドラまたは DHTML 動作が関連付けられている場合にのみリッスンします。
- ▶ **[しない]**：イベントを一切リッスンしません。

イベント・ハンドラは、Web ページに含まれているコードであり、通常はスクリプト言語で書かれている関数またはルーチンです。対応するイベントが発生したときに制御が渡されます。

DHTML 動作は、ページにおける特定の機能や振る舞いをカプセル化します。ページ上の標準的な HTML 要素に適用されている場合、その要素の標準設定の動作が拡張されます。

イベントに対するリッスン条件を指定するには、次の手順を実行します。

- 1 [ユーザ定義 Web イベント記録設定] ダイアログ・ボックスから、リッスン条件を変更する対象となるオブジェクトを選択します。
- 2 変更するイベントの行で、[応答] 列から必要なリッスン条件を選択します。

| イベント名 | 応答 | 記録 |
|---------------|--------------|----|
| onclick | ハンドラの場合 | 有効 |
| oncontextmenu | ハンドラの場合 | 有効 |
| onkeydown | 常に | 有効 |
| onmouseover | ハンドラの場合 | 無効 |
| onmouseup | 常に | 無効 |
| 6 | ハンドラの場合 | |
| 7 | 動作の場合 | |
| 8 | ハンドラまたは動作の場合 | |
| 9 | しない | |
| 10 | | |
| 11 | | |

リストからリッスン条件を選択します。[常に]、[ハンドラの場合]、[動作の場合]、[ハンドラまたは動作の場合]、または [しない] のいずれかを選択できます。

記録ステータス

イベントごとに、対象イベントを記録するようにも、記録しないようにも、あるいは次のイベントが選択されたイベントに依存している場合だけ記録するようにも設定できます。

- ▶ **[有効]** : VuGen が対象オブジェクトあるいはイベントの「バブリング先」である別のオブジェクトをリッスンしている場合、イベントが生じるたびにそれを記録します。
「バブリング」とは、子オブジェクトで発生したイベントが、イベントを処理するイベント・ハンドラに遭遇するまで、HTML コード内の階層をさかのぼる処理です。
- ▶ **[無効]** : 指定されたイベントを記録せず、イベント・バブリングがある場合はそれを無視します。
- ▶ **[次のイベントで有効]** : (Image オブジェクトおよび WebArea オブジェクトの場合にのみ該当) **有効** との唯一の違いは、以降のイベントが同じオブジェクトで発生した場合にのみイベントを記録することです。たとえば、マウスオーバー操作によって画像リンクが変わるとします。この画像の上をマウスが通過するたびに、mouseover イベントを記録する必要はないかもしれません。ただし、リンクが mouseover イベント後に表示される画像によってのみ有効になるので、mouseover イベントを、同じオブジェクトに対するクリック・イベントの前に記録することが重要となります。

イベントの記録ステータスを設定するには、次の手順を実行します。

- 1 [ユーザ定義 Web イベント記録設定] ダイアログ・ボックスから、記録ステータスを変更するオブジェクトを選択します。[任意の Web オブジェクト] を選択し、記録対象ページのすべての Web オブジェクトの記録ステータスを設定します。
- 2 変更対象のイベントの行で、記録ステータスを [記録] 列から選択します。

| イベント名 | 応答 | 記録 |
|-------------|---------|-----------|
| onclick | 常に | 有効 |
| onmouseover | ハンドラの場合 | 次のイベントで有効 |
| 3 | | 無効 |
| 4 | | 有効 |
| 5 | | 次のイベントで有効 |
| 6 | | |
| 7 | | |
| 8 | | |
| 9 | | |
| 10 | | |
| 11 | | |

イベントのリッスンと記録を行うためのヒント

理想的なリッスンと記録の設定を見つけるのが困難な場合があります。これらの設定を行うときには、次のガイドラインに留意します。

- ▶ オブジェクトのイベントを記録するには、VuGen がイベントをリッスンし、イベントが生じたときにそれを記録するように指定します。子プロジェクトのイベントは、親オブジェクトにそのイベントに対するハンドラまたは動作が含まれている場合にもリッスンできます。また、親オブジェクトのイベントは、子オブジェクトにそのイベントに対するハンドラまたは動作が含まれていても、リッスンできます。

ただし、ソース・オブジェクト（どの親オブジェクトがハンドラまたは動作を含んでいるかによらずイベントが実際に発生したオブジェクト）に対するイベントは記録しなければなりません。

たとえば、**onmouseover** イベント・ハンドラのあるテーブル・セルに 2 つのイメージが含まれているとします。ユーザがマウス・ポインタでどちらかのイメージに触れると、バブリングによってイベントがそのセルに送られます。このバブリングにはどちらのイメージが実際に触れられたかの情報が含まれています。この **mouseover** イベントは、次のようにして記録できます。

- ▶ WebTable の mouseover イベントに対する**応答**を [**ハンドラの場合**] に設定する一方で (イベントが生じたときに VuGen にイベントが「聞こえる」ように)、イベントは記録しないようにした上で、Image の mouseover イベントに対する**応答**は [**しない**] に設定する一方で、Image に対する記録は [**有効**] (WebTable レベルでリッスンした後の画像に対する mouseover イベントを記録します) に設定します。
- ▶ Image の mouseover イベントに対する**応答**を [**常に**] (イメージ・タグが動作またはハンドラを含んでいなくても mouseover イベントをリッスン) に設定し、Image オブジェクトに対する記録ステータスを [**有効**] (イメージに対する mouseover イベントを記録) に設定します。
- ▶ 多数のオブジェクト上の多数のイベントをリッスンするように設定すると、VuGen のパフォーマンスが低下することがあるので、リッスンの設定は、必要なオブジェクトに限定するようにします。
- ▶ まれに、イベントが発生したオブジェクト (ソース・オブジェクト) をリッスンしていると、イベントが妨害されることがあります。

イベント記録設定のリセット

ユーザ定義の設定を行った後で標準の設定に戻すには、標準の Web イベント設定を使用するように VuGen を設定します。

注 : 標準設定をリセットすると、ユーザ定義の設定は完全になくなります。

基本の設定レベルに戻すには、次の手順を実行します。

- 1 [記録オプション] で [**GUI プロパティ : Web イベント設定**] ノードを選択します。
- 2 [**標準設定値を使用**] をクリックします。標準設定スライダが再度表示され、すべてのイベント設定が**基本**イベント記録設定レベルに戻されます。

また、特定の (基礎の) ユーザ定義設定、[基礎]、[中]、または [高] を復元することもできます。

設定を特定のユーザ定義レベルにリセットするには、次の手順を実行します。

- 1 [記録オプション] で [GUI プロパティ : Web イベント設定] ノードを選択します。
- 2 [ユーザ定義設定] ボタンをクリックします。[ユーザ定義 Web イベント記録設定] ダイアログ・ボックスが開きます。
- 3 [戻した後の値] ボックスで、使用する定義済みイベント記録レベルを選択します。
- 4 [リセット] をクリックします。すべてのイベント設定が、選択したレベルの標準設定に戻されます。

第 18 章

Java 記録オプション

VuGen では、CORBA、RMI、JMS、または Jacada アプリケーションの記録方法を制御できます。標準の記録オプションを使用することも、必要に応じてそれらをカスタマイズすることもできます。

本章の内容

- ▶ Java 記録オプションの設定について (358 ページ)
- ▶ Java 仮想マシン (JVM) 記録オプション (359 ページ)
- ▶ クラスパス記録オプションの設定 (361 ページ)
- ▶ レコーダ・オプション (362 ページ)
- ▶ シリアル化オプション (364 ページ)
- ▶ 相関オプション (366 ページ)
- ▶ ログ・オプション (367 ページ)
- ▶ CORBA オプション (369 ページ)

Java 記録オプションの設定について

VuGen では、CORBA (Common Object Request Broker Architecture) または RMI (Remote Method Invocation) に対応した Java アプリケーションまたはアプレットを記録できます。EJB テストの記録の詳細については、『**第 2 巻 – プロトコル**』の第 17 章「Enterprise Java Beans (EJB) プロトコル」を参照してください。

記録をする前に、Java 仮想マシン (JVM)、およびコード生成段階で使われる記録オプションを設定します。記録オプションの設定は、必須ではありません。設定しない場合は、標準設定の値が使用されます。

本章で説明するオプションは、以前は **mercury.properties** ファイルに変更を加えることにより設定していました。

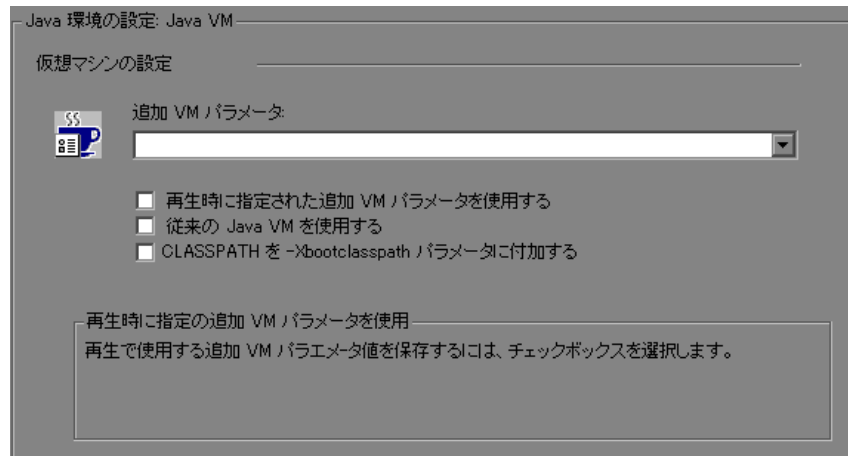
次の項目について記録オプションの設定が可能です。

- ▶ Java 仮想マシン (JVM) 記録オプション
- ▶ クラスパス記録オプションの設定
- ▶ レコーダ・オプション
- ▶ シリアル化オプション
- ▶ 関連オプション
- ▶ ログ・オプション

Java 仮想マシン (JVM) 記録オプション

Java VM オプションは、Java アプリケーションの記録時に使用する付加的なパラメータを指定します。

Vuser を記録する際、VuGen によって **Xbootclasspath** 変数に、標準のパラメータが設定されます。このダイアログ・ボックスを使って、**Xbootclasspath** に対して別のパラメータを設定すると、標準のパラメータに代えて、指定したコマンド・パラメータが使用されます。



また、単一のクラスパス文字列を作成するために、VuGen に対して、クラスパスを **Xbootclasspath** の前に追加する（文字列を挿入する）ように指定することもできます。

標準設定では、VuGen は記録中に従来型の VM を使用します。VuGen を別の仮想マシン（Sun の Java Hotspot VM）を使用するように設定することもできます。

Java 仮想マシンの記録オプションを設定するには、次の手順を実行します。

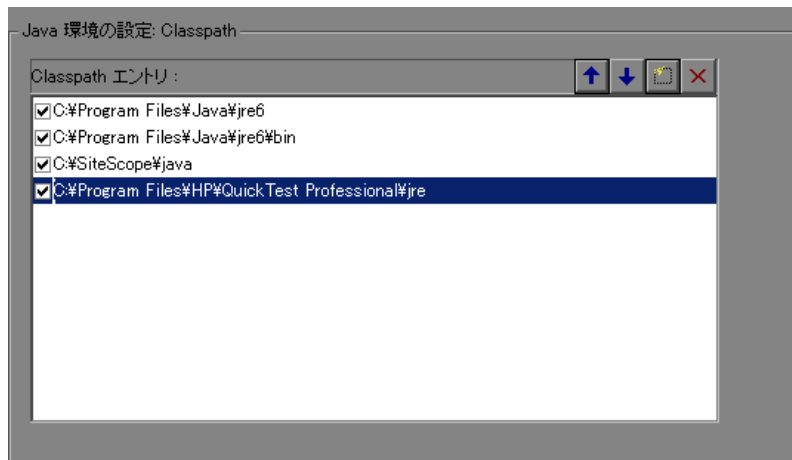
- 1 [記録開始] ダイアログ・ボックスの [オプション] をクリックします。記録オプションの [Java 環境の設定 : Java VM] ノードを選択します。
- 2 [追加 VM パラメータ] ボックスに、一連の Java コマンド・ライン・パラメータを指定します。任意の Java VM の引数をパラメータとして指定できます。よく使用する引数としては、デバッグ・フラグ (-verbose) や、メモリの設定 (-ms, -mx) があります。Java VM フラグの詳細については、JVM のドキュメントを参照してください。さらに、-D フラグの形式で Java アプリケーションに対してプロパティを渡すこともできます。

VuGen では、-Xbootclasspath 変数 (JDK 1.2 以上) に対して、標準のパラメータが自動的に設定されます。-Xbootclasspath に対して、追加のパラメータとしてパラメータの値を指定すると、VuGen によって標準設定の値の代わりにその設定が使用されます。
- 3 再生時に同じ追加 VM パラメータを使用するには、[再生時に指定された追加 VM パラメータを使用する] チェック・ボックスを選択します。
- 4 従来型の VM を使用するには、[従来の Java VM を使用する] チェック・ボックスを選択します (標準設定)。別の VM (Sun の Java HotSpot) を使用するには、このチェック・ボックスをクリアします。
- 5 クラスパスを Xbootclasspath の前に追加する (つまり、文字列を前置する) には、[CLASSPATH を -Xbootclasspath パラメータに付加する] チェック・ボックスを選択します。
- 6 [OK] をクリックしてダイアログ・ボックスを閉じ、記録を開始します。

クラスパス記録オプションの設定

[Java 環境の設 : Classpath] ノードでは、システムのクラスパス環境変数に含まれていない追加のクラスの場所を指定できます。これらのクラスは、Java アプリケーションの実行や、正しい記録を保障するのに必要な場合があります。

コンピュータまたはネットワークから必要なクラスを検索し、特定のテストの場合にそのクラスを無効にすることができます。また、クラスの順序を変更して、クラスパスのエントリを操作することもできます。



クラスパス記録オプションを設定するには、次の手順を実行します。

- 1 [記録開始] ダイアログ・ボックスの [オプション] をクリックします。記録オプションの [Java 環境の設定 : Classpath] ノードを選択します。
- 2 リストにクラスパスを追加するには、次の手順を実行します。



[クラスパスの追加] ボタンをクリックします。クラスパスのリストに新しい行が追加されます。

新しいクラスに対して、**jar**、**zip** またはほかのアーカイブ・ファイルのパスと名前を入力します。または、フィールドの右にある [参照] ボタンをクリックして、指定したいファイルを選択します。VuGen によって、クラスパスのリストに新しい場所が、ステータスが有効な状態で追加されます。

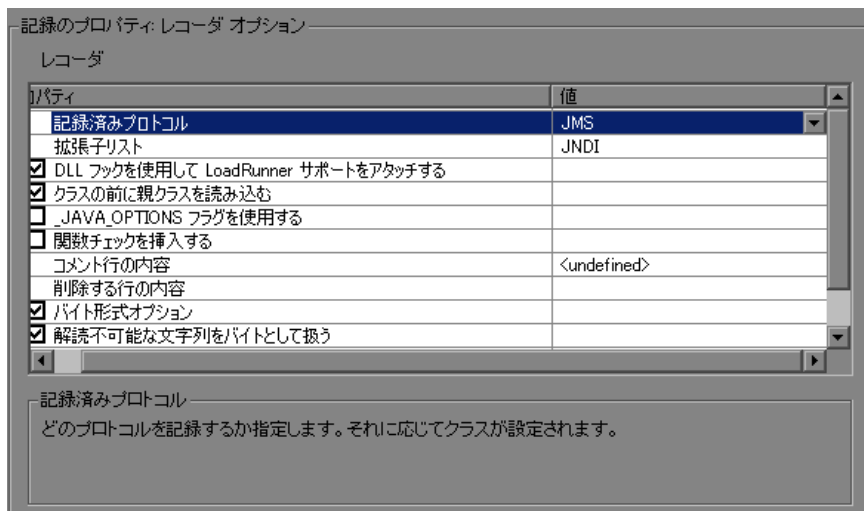


- 3 エントリを恒久的に削除するには、エントリを選択して [削除] ボタンをクリックします。

- 4 特定のテストの場合にだけクラスパスのエントリを無効にするには、エントリの左にあるチェック・ボックスをクリアします。
- ↓ 5 一覧の中でエントリを下方方向に移動するには、エントリを選択し、下向き矢印ボタンをクリックします。
- ↑ 6 クラスパス・エントリをリストの上方へ移動するには、対象エントリを選択して上向きの矢印ボタンをクリックします。
- 7 [OK] をクリックしてダイアログ・ボックスを閉じ、記録を開始します。

レコーダ・オプション

[レコーダオプション] では、記録するプロトコルおよびプロトコル固有の設定を指定します。



- ▶ [記録済みプロトコル]: 記録するプロトコル (RMI, CORBA, JMS, または Jacada) を指定します (標準設定は RMI)。
- ▶ [拡張子リスト]: サポートされているすべての拡張子のカンマ区切りリスト。各拡張子には、固有のフック・ファイルがあります (標準設定は JNDI)。
- ▶ [DLL フックを使用して LoadRunner サポートをアタッチする]: DLL フックを使用して、LoadRunner サポートを自動的に任意の JVM にアタッチします。

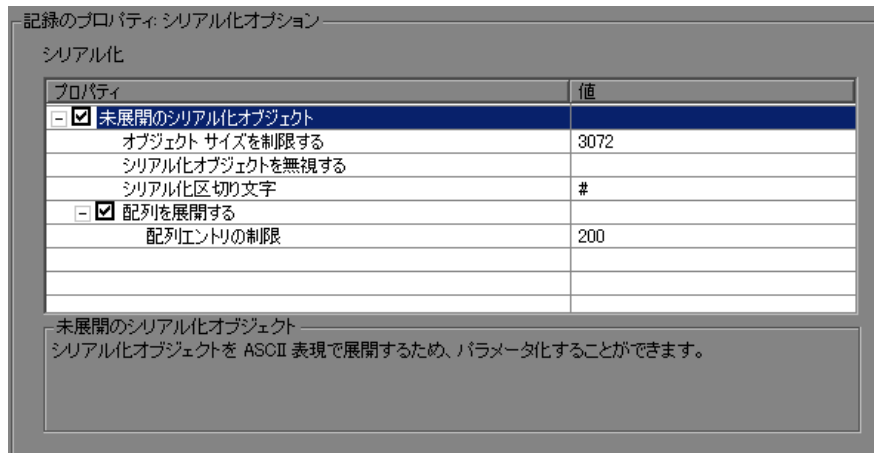
- ▶ **[クラスの前に親クラスを読み込む]**：子クラスの前に親クラスが読み込まれるように、読み込みの順序を変更します。これは、継承が深いツリーのフックを特定するのに役立ちます（標準設定では有効）。
- ▶ **[_JAVA_OPTIONS フラグを使用する]**：Version 1.2 以降の JVM に対して、使いたい JVM パラメータを指定する `_JAVA_OPTION` 環境変数を使用するようにします（標準設定では無効）。
- ▶ **[関数チェックを挿入する]**：再生中に受け取った戻り値を、記録中に生成された戻り値の期待値と比較する検証コードを挿入します。このオプションは、プリミティブ型の戻り値にのみ適用されます（標準設定では無効）。
- ▶ **[コメント行の内容]**：指定した文字列のいずれか 1 つを含むスクリプト行をすべてコメントアウトします。複数の文字列を指定するには、項目をカンマで区切ります。標準設定では、`<undefined>` を含む文字列がある行をコメントアウトします。
- ▶ **[削除する行の内容]**：指定した文字列のいずれか 1 つを含むスクリプト行をすべて削除します。複数の文字列を指定するには、項目をカンマで区切ります。この機能は、テストの目的に応じてスクリプトをカスタマイズするときに役立ちます。
- ▶ **[バイト形式オプション]**：可読文字を、必要なキャストを行うことによって、バイトまたは 16 進形式ではなく文字として表示します（標準設定では有効）。
- ▶ **[解読不可能な文字列をバイトとして扱う]**：不可読文字を含む文字列をバイトの配列として表します。このオプションは、パラメータとして呼び出しに渡される文字列に適用されます（標準設定では有効）。
- ▶ **[バイト配列形式]**：スクリプト内のバイト配列の形式で、**[標準]**、**[展開済みシリアル化オブジェクト]**、または **[未展開シリアル化オブジェクト]** から選択できます。非常に長いバイト配列を記録する場合は、シリアル化オブジェクト・オプションのいずれかを使用します。標準設定は **[標準]** です。
- ▶ **[LoadRunner コールバックを記録する]**：LoadRunner スタブ・オブジェクトをコールバックとして記録します。無効にした場合は、元のクラスがコールバックとして記録されます（標準設定では有効）。

Java 記録オプションを設定するには、次の手順を実行します。

- 1 [記録開始] ダイアログ・ボックスの [オプション] をクリックし、[記録プロパティ:レコーダオプション] ノードを選択します。
- 2 必要なオプションを設定します。チェック・ボックス付きのオプションには、オプションの横のチェック・ボックスを選択またはクリアします。文字列が必要なオプションには、必要な値を入力します。
- 3 すべてのオプションを標準の値に設定するには、[標準設定値を使用] をクリックします。
- 4 [OK] をクリックしてダイアログ・ボックスを閉じ、記録を開始します。

シリアル化オプション

[シリアル化オプション] では、オブジェクトのシリアル化の方法を制御できます。多くの場合、シリアル化は、オブジェクト値のパラメータ化のために ASCII 形式でオブジェクトを表示する際に適用されます。



次のオプションがあります。

- ▶ **[未展開のシリアル化オブジェクト]**：シリアル化されたオブジェクトを ASCII 形式に展開します。このオプションを指定すると、パラメータ化を行うために、オブジェクトの ASCII 値を表示できます（標準では有効）。
 - ▶ **[オブジェクト サイズを制限する]**：シリアル化可能なオブジェクトのサイズを指定した値に制限します。指定した値を超えるオブジェクトは、ASCII 形式でスクリプト内に表現されません。標準設定の値は 3072 です。
 - ▶ **[シリアル化オブジェクトを無視する]**：記録されたスクリプトの中で遭遇しても展開しないシリアル化オブジェクトを示します。オブジェクトはカンマで区切ります。
 - ▶ **[シリアル化区切り文字]**：ASCII 形式のオブジェクトで要素を区切る区切り文字を示します。VuGen では、これらの区切り文字に囲まれた文字列のみがパラメータ化されます。標準設定は「#」です。
- ▶ **[配列を展開する]**：シリアル化されたオブジェクトの配列の要素を ASCII 形式に展開します。このオプションを無効にし、オブジェクトに配列が含まれている場合、オブジェクトは展開されません。標準設定では、このオプションは有効になっています。つまり、シリアル化解除されたオブジェクトはすべて展開されます。
 - ▶ **[配列エントリの制限]**：レコーダによって、指定した数より多くの要素が含まれる配列が展開されないようにします。標準設定の値は 200 です。

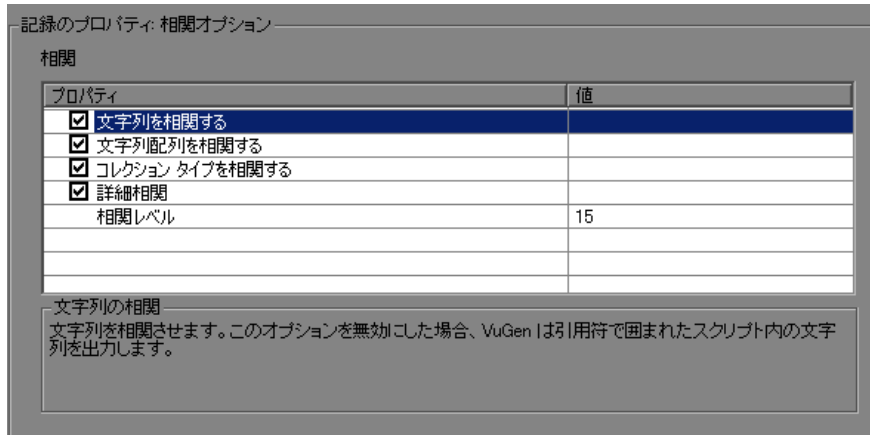
シリアル化のオプションを設定するには、次の手順で行います。

- 1 [記録開始] ダイアログ・ボックスの **[オプション]** をクリックし、**[記録プロパティ：シリアル化オプション]** ノードを選択します。
- 2 必要なオプションを設定します。すべてのオプションを標準の値に設定するには、**[標準設定値を使用]** をクリックします。
- 3 **[OK]** をクリックしてダイアログ・ボックスを閉じ、記録を開始します。

シリアル化の詳細については、『第 2 巻 – プロトコル』の第 16 章「Java – 相関」を参照してください。

相関オプション

[相関オプション] では、自動相関を有効にし、その深さを制御できます。



次のオプションがあります。

- ▶ [文字列を相関する]：相関が必要なすべての文字列を相関します。このオプションが無効の場合、相関が必要な文字列は引用符で囲んでスクリプトに出力されません（標準設定では無効）。
- ▶ [文字配列を相関する]：文字配列内のテキストを相関します（標準設定では有効）。
- ▶ [コレクションタイプを相関する]：JDK 1.2 以降の Collection クラスのオブジェクトを相関します（標準設定では無効）。
- ▶ [詳細相関]：CORBA コンテナの構造要素と配列で深い相関を有効にします（標準設定では有効）。
- ▶ [相関レベル]：相関のレベルの深さを、スキャン対象の内部コンテナの数として指定します（標準設定は 15）。

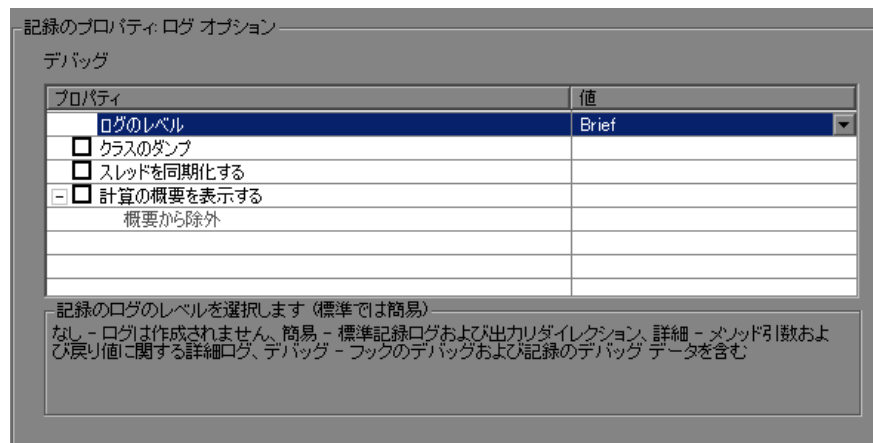
関連のオプションを設定するには、次の手順を実行します。

- 1 [記録開始] ダイアログ・ボックスの [オプション] をクリックし、[記録プロパティ: 関連オプション] ノードを選択します。
- 2 必要なオプションを有効にするか、値が必要なオプションに値を入力します。すべてのオプションを標準の値に設定するには、[標準設定値を使用] をクリックします。
- 3 [OK] をクリックしてダイアログ・ボックスを閉じ、記録を開始します。

関連の詳細については、『第 2 巻 – プロトコル』の第 16 章「Java – 関連」を参照してください。

ログ・オプション

[ログオプション] では、記録中に生成するデバッグ情報のレベルを指定します。



次のオプションがあります。

- ▶ [ログのレベル] : 生成する記録ログのレベル。
 - ▶ [なし] : ログ・ファイルは作成されません。
 - ▶ [簡易] : 標準記録ログおよび出力リダイレクションが生成されます。
 - ▶ [詳細] : メソッド、引数、および戻り値に関する詳細ログが生成されます。

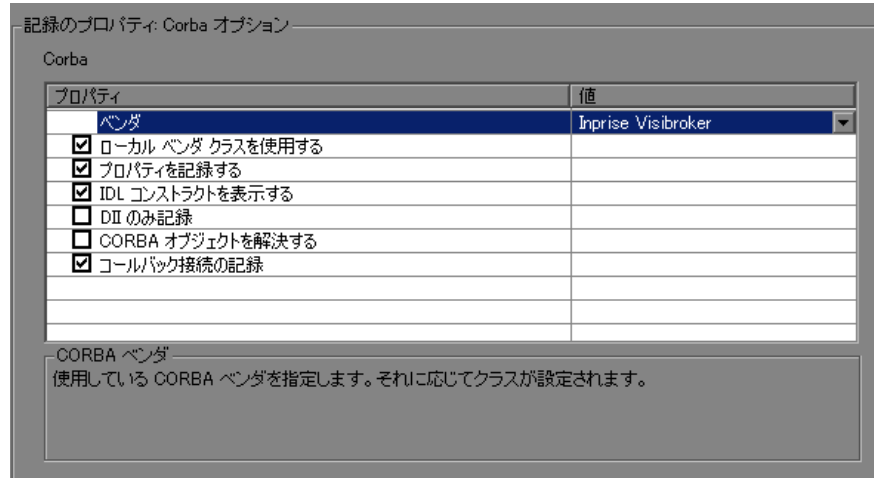
- ▶ **[デバッグ]**：前述のすべての情報に加えて、フックおよび記録のデバッグ情報が記録されます。
- ▶ **[クラスのダンプ]**：ロードしたすべてのクラスをスクリプト・ディレクトリにダンプします（標準設定では無効）。
- ▶ **[スレッドを同期化する]**：マルチ・スレッド・アプリケーションの場合に、各スレッド間の同期をとるようにします（標準設定では無効）。
- ▶ **[計算の概要を表示する]**：記録されたすべてのオブジェクトのダイジェストを生成します（標準設定では無効）。
 - ▶ **[概要から除外する]**：ダイジェスト計算に含めないオブジェクトの一覧です。

ログのオプションを設定するには、次の手順を実行します。

- 1 **[記録開始]** ダイアログ・ボックスの **[オプション]** をクリックし、**[記録プロパティ：ログオプション]** ノードを選択します。
- 2 ログのレベル（[なし]、[簡易]、[詳細]、または [デバッグ]）を選択します。
- 3 必要なオプションを有効にするか、値が必要なオプションに値を入力します。
- 4 すべてのオプションを標準の値に設定するには、**[標準設定値を使用]** をクリックします。
- 5 **[OK]** をクリックしてダイアログ・ボックスを閉じ、記録を開始します。

CORBA オプション

次のオプションは CORBA-Java を対象としています。このオプションでは、CORBA 固有の記録プロパティといくつかのコールバック・オプションを設定できます。



次のオプションがあります。

- ▶ **[ベンダ]** : CORBA ベンダ。 **Inprise Visibroker**, **Iona OrbixWeb**, または **Bea Weblogic**。
- ▶ **[ローカル ベンダ クラスを使用する]** : ローカル・ベンダ・クラスを使用して、**srv** フォルダを **BOOT** クラスパスに追加します。このオプションを無効にした場合は、**VuGen** がネットワーク・クラスを使用して、スクリプトのクラスをクラスパスに追加します（標準設定では有効）。
- ▶ **[プロパティを記録する]** : プロトコルに関連するシステム・プロパティとユーザ定義のプロパティを記録します（標準設定では有効）。
- ▶ **[IDL コンストラクトを表示する]** : パラメータを CORBA の呼び出しに対して渡すために使用される、インタフェース定義言語（IDL）の構成要素を表示します（標準設定では有効）。
- ▶ **[DLL のみ記録]** : DLL レベルのみで記録するよう **VuGen** に指示します（標準設定では無効）。

- ▶ **[CORBA オブジェクトを解決する]**：相関によって、CORBA オブジェクトが解決できなかった場合に、バイナリ・データを使って解決します（標準設定では無効）。
- ▶ **[コールバック接続の記録]**：各コールバック・オブジェクトについて、ORB への接続のための接続ステートメントを生成させます（標準設定では無効）。

CORBA 記録オプションを設定するには、次の手順を実行します。

- 1 [記録開始] ダイアログ・ボックスの **[オプション]** をクリックし、**[記録プロパティ：Corba オプション]** ノードを選択します。
- 2 必要なオプションを有効または無効にします。
- 3 すべてのオプションを標準の値に設定するには、**[標準設定値を使用]** をクリックします。
- 4 **[OK]** をクリックしてダイアログ・ボックスを閉じ、記録を開始します。

第 19 章

スクリプト生成オプションの設定

スクリプトの記録を開始する前に、必要なスクリプト言語とその言語に適用するオプションを指定します。

本章の内容

- ▶ スクリプト生成オプションの設定について (372 ページ)
- ▶ スクリプト言語の選択 (373 ページ)
- ▶ 基本オプションの適用 (374 ページ)
- ▶ 関連オプションについて (376 ページ)
- ▶ 記録オプションの設定 (377 ページ)

以降の情報は、マルチ・プロトコルの記録をサポートするすべての仮想ユーザ・スクリプトを対象とします。

スクリプト生成オプションの設定について

VuGen では、セッションを記録する前に、スクリプト生成言語を指定できます。使用可能なスクリプト生成言語は、プロトコルによって異なります。使用可能な言語としては、C、C#、Visual Basic、Visual Basic .NET、VB Script、および JavaScript です。標準設定では、対象プロトコルに応じた最も一般的な言語でスクリプトが生成されますが、この言語は [スクリプト] 記録オプションで変更できます。

ヒント：ある言語でスクリプトを記録した後に、別の言語でそのスクリプトを再生成することもできます。詳細については、103 ページ「Vuser スクリプトの再生成」を参照してください。

生成言語の選択後、スクリプトに何を含め、それをどのように生成するかをレコーダに指示する、言語固有の記録オプションを有効にできます。

記録するプロトコルの少なくとも 1 つにマルチ・プロトコルの機能がある場合は、スクリプト・オプションを使用できます。ただし、シングル・プロトコル・スクリプトで HTTP または WinSock を記録する場合は例外です。この場合、スクリプト・オプションは利用できません。

スクリプト言語の選択

セッションを記録するときに、VuGen は、ユーザの操作をエミュレートするスクリプトを作成します。標準のスクリプト生成言語は C 言語または C# 言語 (MS .NET の場合) です。FTP, COM/DCOM, およびメール・プロトコル (IMAP, POP3, SMTP) の場合は、Visual Basic, VBScript, および JavaScript でもスクリプトを生成できます。

- ▶ **[C 言語]** : 複雑な COM 構成要素と C++ オブジェクトを使用するアプリケーション用。
- ▶ **[C# 言語]** : 複雑なアプリケーションと環境を使用するアプリケーション用 (MS .NET プロトコルのみ)。
- ▶ **[Visual Basic .NET 言語]** : VB .NET アプリケーション用。VB の機能をすべて使用できます。
- ▶ **[Visual Basic for Applications]** : VB ベースのアプリケーション用。(VBScript とは異なり) VB の機能をすべて使用できます。
- ▶ **[Visual Basic Scripting]** : VBScript ベースのアプリケーション用 (ASP など)。
- ▶ **[Java Scripting]** : JavaScript ベースのアプリケーション用 (**js** ファイルやダイナミック HTML アプリケーションなど)。

記録セッションの後は、通常の C, C#, Visual Basic, VB Script, JavaScript コード、または制御フロー・ステートメントを使ってスクリプトに変更を加えられます。

次の項では、スクリプト編集オプションについて説明します。すべてのスクリプトについては、374 ページ「基本オプションの適用」を参照してください。C 以外のスクリプトに相関オプションを設定するには、376 ページ「相関オプションについて」を参照してください。

手順の詳細については、377 ページ「記録オプションの設定」を参照してください。

基本オプションの適用

基本スクリプト・オプションは生成されたスクリプトの詳細レベルを制御します。一部のオプションは特定の言語に対してのみ使用できます。

- ▶ **[記録停止時に AUT の全プロセスを閉じる]** : VuGen が記録を停止すると、すべての AUT (テスト対象アプリケーション) の処理が自動的に終了します (標準設定では無効)。
- ▶ **[プリミティブをロケールとして宣言する]** : プリミティブ値の変数をクラス変数ではなくローカル変数として宣言します (C, C#, .NET のみ。標準設定では有効)。
- ▶ **[バリエントを明示的に宣言する]** : ByRef バリエントを処理するため、バリエント・タイプを明示的に宣言します (Visual Basic for Applications のみ。標準設定では有効)。
- ▶ **[end_transaction 後に固定の思考時間を生成する]** : トランザクション終了後、固定思考遅延時間を秒単位で追加します。このオプションを有効にする場合は、思考遅延時間の値を指定できます。標準設定は 3 秒です (標準設定では無効)。
- ▶ **[記録されたイベントのログを生成する]** : 記録中に発生したすべてのイベントのログを生成します (標準設定では無効)。
- ▶ **[しきい値を超える場合に思考遅延時間を生成する]** : 思考遅延時間のしきい値を使用します。記録された思考遅延時間がしきい値に満たない場合、VuGen は思考遅延時間ステートメントを生成しません。しきい値も指定します。標準設定の値は 3 です。思考遅延時間が 3 秒以内の場合は、VuGen は思考遅延時間のステートメントを生成しません。このオプションを無効にすると、VuGen は思考遅延時間を生成しません (標準設定では有効)。
- ▶ **[呼び出し後情報を挿入する]** : 各メッセージ呼び出しの後に、その内容を表すログ・メッセージを挿入します (C 以外のみ。標準設定では有効)。
- ▶ **[出力パラメータ値を挿入する]** : 各呼び出しの後に出力パラメータ値を挿入します (C, C#, .NET のみ。標準設定では無効)。
- ▶ **[呼び出し前情報を挿入する]** : 各メッセージ呼び出しの前に、その内容を表すログ・メッセージを挿入します (C 以外のみ。標準設定では有効)。

- ▶ **[アクション ファイル内の最大行数]** : アクションの行数が指定されたしきい値を超えた場合に新しいファイルを作成します。標準のしきい値は 60000 行です (C, C#, .NET のみ。標準設定では無効)。
- ▶ **[プリミティブの戻り値に変数を再使用する]** : メソッド呼び出しから受け取るプリミティブに同じ変数を再利用します。これは **[プリミティブをロケールとして宣言する]** 設定に優先します (標準設定では有効)。
- ▶ **[長い文字列をパラメータで置換する]** : 最大文字数を超える文字列をパラメータに保存します。このオプションの初期の最大長は 100 文字です。パラメータと文字列全体は、次の形式で、スクリプトのフォルダ内の `lr_strings.h` ファイルに保存されます。

```
const char <paramName_uniqueID> ="string"
```

このオプションにより、スクリプトが読みやすくなります。スクリプトのパフォーマンスには影響しません (標準設定では有効)。
- ▶ **[フルネームを使用する]** : 新しい変数を宣言するときに完全な型名を使用します (C# および .NET のみ。標準設定では無効)。
- ▶ **[COM ローカル サーバとして作成されたプロセスを追跡する]** : 記録されたアプリケーションのサブプロセスの 1 つが COM ローカル・サーバとして作成されている場合は、そのアプリケーションの動作を追跡します (C および COM のみ。標準設定では有効)。
- ▶ **[配列に対してヘルパを使用する]** : ヘルパー関数を使って、バリエーションの配列からコンポーネントを抽出します (Java および VBScript のみ。標準設定では無効)。
- ▶ **[オブジェクトに対してヘルパを使用する]** : ヘルパー関数を使って、バリエーションのオブジェクトの参照が引数として関数に渡されたときに、その参照を抽出します (Java および VBScript のみ。標準設定では無効)。

手順の詳細については、377 ページ「記録オプションの設定」を参照してください。

相関オプションについて

相関を使って、テストの実行中に動的な値を保持できます。これらのオプションによって、記録時に VuGen によって自動的に行われた相関を拡張設定できます。すべての相関オプションは標準設定では無効になっています。相関オプションは、VB Applications, VBScript, JavaScript などの C 以外の言語にのみ適用されます。

- ▶ **[配列の相関]**：記録中に、文字列、構造体、数値など、すべてのデータ型の配列を追跡して相関させます（標準設定では有効）。
- ▶ **[大きい数の相関]**：記録中に、int, long int, 64 ビットの char, float, double などの長いデータ型を相関させます（標準設定では無効）。
- ▶ **[単純文字列の相関]**：単純で、配列ではない文字列や文章を相関させます（標準設定では無効）。
- ▶ **[小さい数の相関]**：記録中に、byte, char, および short int などの短いデータ型を相関させます（標準設定では無効）。
- ▶ **[構造の相関]**：複雑な構造要素を追跡して相関させます（標準設定では有効）。

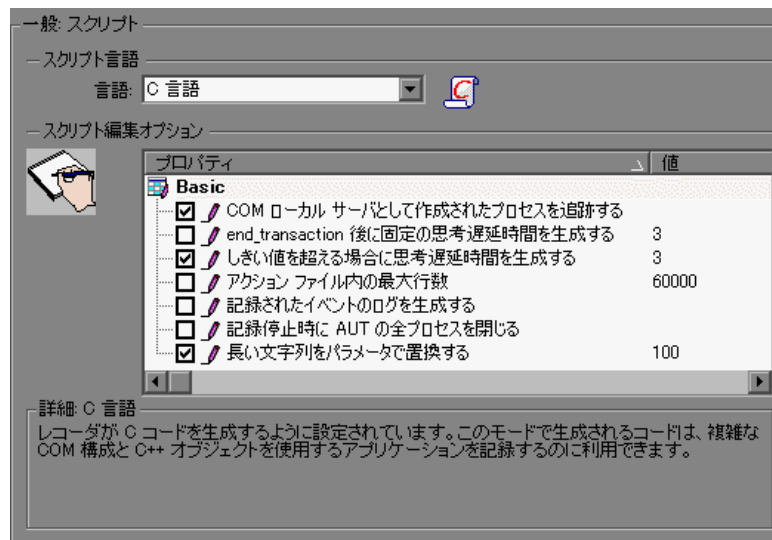
手順の詳細については、377 ページ「記録オプションの設定」を参照してください。

記録オプションの設定

[記録オプション] は、スクリプトに関連する記録を開始する前に設定します。使用できるオプションの数は、スクリプト生成言語によって異なります。

スクリプト記録オプションを設定するには、次の手順を実行します。

- 1 [記録オプション] ダイアログ・ボックスを開きます。メイン・メニューから [ツール] > [記録オプション] を選択するか、[記録開始] ダイアログ・ボックスの [オプション] をクリックします。[記録オプション] ダイアログ・ボックスが開きます。
- 2 [一般:スクリプト] ノードを選択します。



- 3 [スクリプト言語] ボックスで、コード生成モード ([C 言語] または [Visual Basic for Applications]) を選択します。複雑な構造要素と C++ コードを使用するアプリケーションを記録する場合は、C を使用します。スクリプト・ベースのアプリケーションを記録するには、Visual Basic モードを使用します。
- 4 [スクリプト編集オプション] セクションで必要なオプションを選択するには、オプションの横のチェック・ボックスを選択します。このオプションについては、前の項で説明しています。
- 5 [OK] をクリックし、設定を保存してダイアログ・ボックスを閉じます。

第 20 章

Web Vuser の記録オプションの設定

Web セッションを記録する前に、記録オプションをカスタマイズできます。

本章の内容

- ▶ 記録オプションの設定について (379 ページ)
- ▶ 記録レベルについて (380 ページ)
- ▶ 記録レベルの設定 (393 ページ)

以降の情報は、**Web (HTTP/HTML)**、**Web (Click and Script)**、**Web/WinSocket**、**Oracle Web Applications 11i**、**PeopleSoft Enterprise** の各 Vuser スクリプトを対象とします。

記録オプションの設定について

VuGen を使用して、ユーザが Web サイトで実行する標準的な操作を記録して、Web Vuser スクリプトを生成できます。

記録する前に、記録オプションを設定し、記録する情報、記録に使用するブラウザまたはクライアント、スクリプトの内容を指定します。

プロキシ設定やその他の詳細設定など、一般的な HTTP プロパティの記録オプションを設定できます。詳細については、331 ページ「**Web, ワイヤレス, Oracle NCA 記録オプション**」を参照してください。

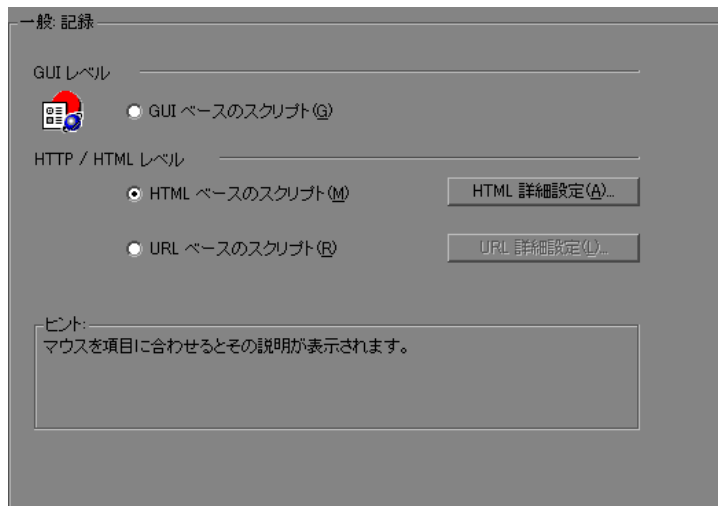
また、Web Vuser スクリプトに対して、関連記録オプションを設定することもできます。詳細については、『**第 2 巻 – プロトコル**』の第 42 章「**Web (HTTP/HTML) の関連ルール**」を参照してください。

記録レベルについて

記録レベルを選択することによって、Vuser スクリプトの生成時に、記録する情報と使用する関数が指定できます。記録レベルは、目的または環境によって選択します。指定できるレベルは、**GUI ベースのスクリプト**、**HTML ベースのスクリプト**、および **URL ベースのスクリプト**です。Web HTTP/HTML Vuser の場合は、後者の 2 つのオプションのみ使用可能です。

以下のガイドラインに従って、どの記録レベルを使用するかを決定します。

- ▶ JavaScript を使用するアプリケーションを含む大部分のアプリケーションでは、**[GUI ベースのスクリプト]** を使用します。このレベルは、PeopleSoft Enterprise および Oracle Web Applications 11i Vuser の場合にも推奨されます。
- ▶ アプレットおよび VB スクリプトを使用するブラウザ・アプリケーションの場合は、**[HTML ベースのスクリプト]** を作成します。
- ▶ 非ブラウザ・アプリケーションの場合は、**[URL ベースのスクリプト]** を使用します。



GUI ベースのスク립ト・オプションは、HTML アクションを **web_text_link** などのコンテキスト・センシティブ GUI 関数として記録するよう VuGen を設定します。

```
/* GUI ベース・モード : JavaScript をサポートしている CS タイプ関数 */  
vuser_init()  
{  
  web_browser("WebTours",  
    DESCRIPTION,  
    ACTION,  
    "Navigate=http://localhost:1080/WebTours/",  
    LAST);  
  
  web_edit_field("username",  
    "Snapshot=t2.inf",  
    DESCRIPTION,  
    "Type=text",  
    "Name=username",  
    "FrameName=navbar",  
    ACTION,  
    "SetValue=jojo",  
    LAST);  
  
  ...  
}
```

HTML ベースのスクリプト・レベルでは、HTML ユーザ・アクションごとにステップが生成されます。これらのステップもわかりやすいのですが、JavaScript コードの忠実なエミュレーションが反映されているわけではありません。

```
/* HTML ベース・モード：ユーザ・アクションを記述するスクリプト */  
...  
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 ベースのスクリプト・モード・オプションでは、ユーザのアクションによって送信されたサーバからのすべてのブラウザ要求とリソースが、VuGen によって記録されます。自動的にあらゆる HTTP リソースが URL ステップ (**web_url** ステートメント) として記録されます。ブラウザの通常の記録では、URL ベース・モードは相関関連の問題が起りやすいため推奨されません。ただし、アプレットや非ブラウザ・アプリケーションなどのページを記録している場合は、このモードが最適です。

URL ベースのスクリプトは、HTML ベースのスクリプトほどわかりやすくはありません。これは、すべてのアクションが **web_link** や **web_image** などではなく **web_url** のステップとして記録されるためです。

```
/* URL ベース・モード : web_url 関数のみ */
...
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/travel/calendar_functions.js",
        "Resource=1",
        "RecContentType=application/x-javascript",
        "Referer=",
        "Mode=HTTP",
        LAST);
...
```

マルチ・プロトコル・スクリプトを記録していなければ、記録中に記録レベルと詳細記録オプションを切り替えることができます。記録レベルを切り替える方法は、パフォーマンスのテストを行う上級ユーザ向けです。

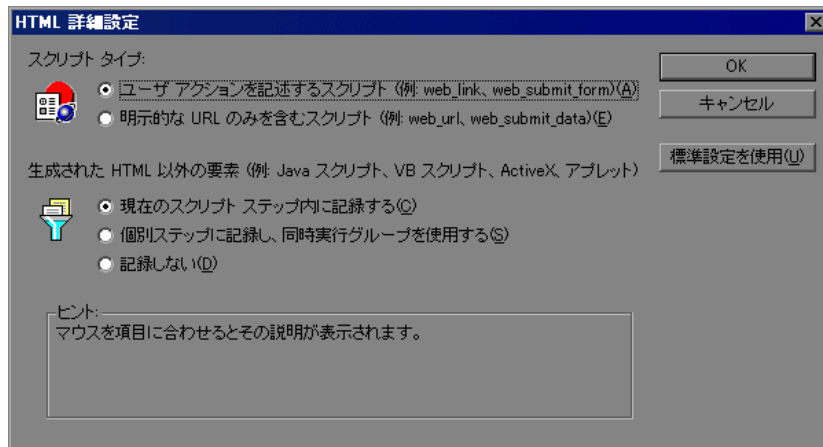
記録後、元の記録とは異なる方式でスクリプトを再生成することもできます。たとえば、スクリプトを HTML ベース・レベルで記録している場合、URL ベース・レベルで再生成できます。スクリプトを再生成するには、**[ツール]** > **[スクリプトの再生成]** を選択し、**[オプション]** をクリックして、再生成のための記録オプションを設定します。

HTML ベースのオプションの詳細設定

Web (HTTP/HTML) Vuser の標準設定の記録レベルである [HTML ベースのスク립ト] オプションを選択すると、VuGen によって現在の Web ページのコンテキストにおいて HTML アクションが記録されます。記録セッション中、リソースのすべては記録されませんが、再生時にはダウンロードされます。

次の範囲について、HTML ベースのスク립トの詳細オプションを設定できます。

- ▶ スクリプト・タイプの指定
- ▶ HTML 以外の要素の処理



スクリプト・タイプの指定

HTML ベースのスクリプトでは、次のタイプのスクリプトを指定できます。

- ▶ ユーザ・アクションを記述するスクリプト
- ▶ 明示的な URL のみを含むスクリプト

最初のオプション、[**ユーザアクションを記述するスクリプト**] は標準設定のオプションです。アクションに直接対応する関数が作成されます。URL (**web_url**)、リンク (**web_link**)、画像 (**web_image**)、フォーム送信 (**web_submit_form**) などの関数を作成します。コンテキスト・センシティブ記録と似た非常にわかりやすいスクリプトが作成されます。

```
/* HTML ベース・モード：ユーザ・アクションを記述するスクリプト */  
...  
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);  
...
```

2 つ目のオプションの [明示的な URL のみを含むスクリプト] では、すべてのリンク、画像および URL が `web_url` ステートメントとして記録されます。ただし、フォームの場合は、`web_submit_data` ステートメントとして記録されます。`web_link` 関数、`web_image` 関数、および `web_submit_form` 関数は生成されません。生成されるスクリプトは直観的ではなくなります。サイト内の多数のリンクが同じリンク・テキストを使用している場合に役立ちます。1 つ目のオプションを使用してサイトを記録すると、リンクの出現（インスタンス）が記録されますが、2 つ目のオプションを使用して記録すると、それぞれのリンクが URL のリストとして記録されます。このことにより、ステップの相関、およびパラメータ化を容易に行うことができます。

[明示的な URL のみを含むスクリプト] を選択した状態で記録したセッションの例を次に示します。

```
/* HTML ベース : 明示的な URL のみを含むスクリプト */
...
web_url("Click Here For Additional Restrictions",
        "URL=http://www.hplab.com/restrictions.html",
        "TargetFrame=",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://www.hplab.com/home?...
        "Snapshot=t4.inf",
        "Mode=HTML",
        LAST);

web_url("buttonhelp.gif",
        "URL=http://www.hplab.com/home?com/rstr?BV_EngineID...",
        "TargetFrame=main",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://www.hplab.com/home?...
        "Snapshot=t5.inf",
        "Mode=HTML",
        LAST);
...
```

HTML 以外の要素の処理

多くの Web ページには、アプレット、XML、ActiveX 要素、JavaScript など、HTML 以外の項目が含まれています。通常こうした HTML 以外の要素には、それ自身のリソースが含まれているか、取得されるかします。たとえば、記録された Web ページから呼び出された JavaScript の **js** ファイルによって、いくつかの画像がロードされることがあります。アプレットによって外部テキスト・ファイルがロードされることもあります。下記のオプションを使って、HTML 以外の要素をどのように記録するかを制御することができます。

次のオプションがあります。

- ▶ 現在のスクリプトステップ内に記録する（標準設定）
- ▶ 個別ステップに記録し、並行グループを使用する
- ▶ 記録しない

1 つ目のオプション [**現在のスクリプトステップ内に記録する**] では、HTML によらずに生成されたリソースについて新規関数が生成されません。すべてのリソースが、`web_url`、`web_link`、`web_submit_data` などの関連する関数の引数としてリストされます。Web 関数の引数となったリソースには **EXTRARES** フラグが付けられます。次の例では、ページにロードされた非 HTML 生成リソースがすべて `web_url` 関数の引数としてリストされています。

```
web_url("index.asp",
  "URL=http://www.daisy.com/index.asp",
  "TargetFrame=",
  "Resource=0",
  "RecContentType=text/html",
  "Referer=",
  "Snapshot=t2.inf",
  "Mode=HTML",
  EXTRARES,
  "Url=http://www.daisy.com/ScrollApplet.class", "Referer=", ENDITEM,
  "Url=http://www.daisy.com/board.txt", "Referer=", ENDITEM,
  "Url=http://www.daisy.com/nav_login1.gif", ENDITEM,
  ...
  LAST);
```

2 つ目のオプション [個別ステップに記録し、同時実行グループを使用する] では、非 HTML 生成リソースの 1 つ 1 つについて新しい関数が生成されます。そのページの関数 (`web_url` や `web_link` など) にはそれらが項目として含まれません。特定のリソースについて生成されたすべての `web_url` 関数は、同時実行グループ内に配置されます (`web_concurrent_start` と `web_concurrent_end` で囲まれます)。

次の例では、前述のセッションを、このオプションを設定した状態で記録したものです。アプレットとそのアプレットによってロードされたテキスト・ファイルに対して `web_url` 関数が生成されています。

```
web_url("index.asp",
    "URL=http://www.daisy.com/index.asp",
    "Resource=0",
    "RecContentType=text/html",
    "Referer=",
    "Snapshot=t2.inf",
    "Mode=HTML",
    LAST);

web_concurrent_start(NULL);
    web_url("ScrollApplet.class",
        "URL=http://www.daisy.com/ScrollApplet.class",
        "Resource=1",
        "RecContentType=application/octet-stream",
        "Referer=",
        LAST);

    web_url("board.txt",
        "URL=http://www.daisy.com/board.txt",
        "Resource=1",
        "RecContentType=text/plain",
        "Referer=",
        LAST);
web_concurrent_end(NULL);
```

3 つ目のオプション, [記録しない] では, 非 HTML 要素によって生成されたリソースが記録されないよう設定されます。

HTML ベースのモードで作業している場合は, VuGen によって **web_url** ステートメント内に **TargetFrame** 属性が挿入されます。この情報は, 実行時ブラウザとテスト結果レポートに Web ページを正しく表示するために使用されます。

```
web_url("buttonhelp.gif",
        "URL=http://www.hplab.com/home?com/rstr?BV_EngineID...",
        "TargetFrame=main",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://www.hplab.com/home?...",
        "Snapshot=t5.inf",
        "Mode=HTML",
        LAST);
```

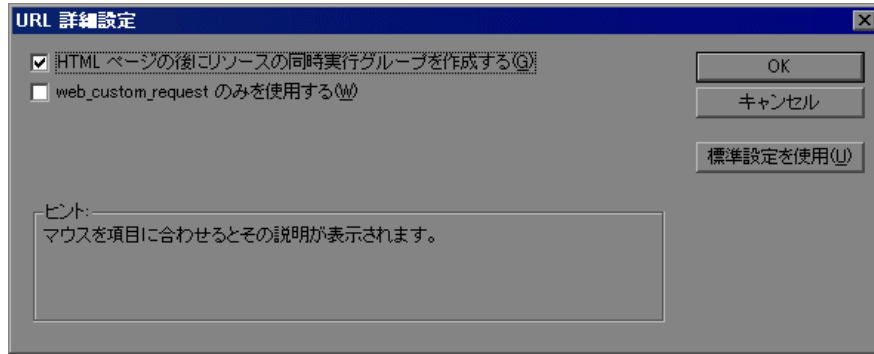
URL ベースのモードでの記録では, VuGen によってページ上のすべてのフレームの内容が記録され, **TargetFrame** 属性が省略されます。

URL ベースのオプションの詳細設定

URL ベース・モードのオプションを選択すると, VuGen によってサーバからのすべての要求とリソースが記録されます。自動的にあらゆる HTTP リソースが URL ステップ (**web_url** ステートメント) として, フォームの場合には, **web_submit_data** として記録されます。**web_link**, **web_image**, および **web_submit_form** 関数は生成されず, フレームも記録されません。

次の範囲について、URL 記録モードの詳細オプションを設定できます。

- ▶ リソースの処理
- ▶ ユーザ定義 HTTP 要求の生成



リソースの処理

URL ベースでの記録では、VuGen によって、ブラウザ要求の結果としてダウンロードされたすべてのリソースがキャプチャされます。標準設定ではこのオプションが有効になっており、URL の後、リソースが同時実行グループとして記録されます (**web_concurrent_start** と **web_concurrent_end** で囲まれます)。リソースには、画像などのファイルと **js** ファイルが含まれます。このオプションを無効にすると、リソースは別々の **web_url** ステップとしてリストされますが、同時実行グループとしてのマークは付きません。

次のコードは、[HTML ページの後にリソースの同時実行グループを作成する] オプションを有効にして記録したセッションを示すものです。

```
web_concurrent_start(NULL);
...
web_url("Click Here For Additional Restrictions",
        "URL=http://www.hplab.com/restrictions.html",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://www.hplab.com/home?...",
        "Snapshot=t4.inf",
        "Mode=HTTP",
        LAST);

web_url("buttonhelp.gif",
        "URL=http://www.hplab.com/home?com/rstr?BV_EngineID...",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://www.hplab.com/home?...",
        "Snapshot=t5.inf",
        "Mode=HTTP",
        LAST);
...
web_concurrent_end(NULL);
```

スクリプトに、**gif** ファイルと **js** ファイルが含まれていることに注目してください。このモードでは、**imp**, **txt**, カスケーディング・スタイル・シート (**css**) や、その他のグラフィック・ファイル、インポートされたファイルも記録の対象となります。

ユーザ定義 HTTP 要求の生成

ブラウザ以外のアプリケーションで記録している場合、すべての HTTP 要求をユーザ定義要求として記録するよう設定できます。VuGen によって内容に関係なくすべての要求に `web_custom_request` 関数が生成されます。

```
web_custom_request("www.hplab.com",
    "URL=http://www.hplab.com/",
    "Method=GET",
    "Resource=0",
    "RecContentType=text/html",
    "Referer=",
    "Snapshot=t1.inf",
    "Mode=HTTP",
    LAST);
```

EUC エンコードされた Web ページの有効化

(次の説明は日本語版 Windows のみに関するものです) Windows の標準文字セット以外を扱う場合には、コード変換が必要になることがあります。文字セットは、文字の集合と整数の集合との対応関係を表します。この対応関係によって、与えられた 1 つの文字について、整数との一意の組み合わせが成立します。EUC (Extended UNIX Code, 拡張 UNIX コード) と SJIS (Shift Japan Industry Standard, シフト JIS) は、Windows の標準文字セットではなく、Web サイトで日本語を表示するために使用されます。

Windows では SJIS コードを使いますが、UNIX では EUC コードを使います。Web サーバが UNIX マシン上にあり、クライアントが Windows の場合、コードが違うため、Web サイトの文字がクライアント・マシンで正しく表示されません。このことは、EUC コードの日本語文字を Vuser スクリプトで表示するときに影響します。

記録中、VuGen は HTTP ヘッダを通じて Web ページで使われているコードを検出します。文字セットに関する情報が HTTP ヘッダに存在しない場合は、HTML のメタ・タグを調べます。文字セットの情報がページから HTTP ヘッダまたはメタ・タグに送信されない場合、VuGen は EUC コードが使われていることを検知しません。

それでも Web ページで EUC コードが使われていることが事前にわかっていたら、VuGen に正しいコードで記録させることができます。ページを EUC コードで記録するには、[記録オプション] **記録** タブの [EUC] オプションを有効にします (日本語版 Windows でのみ表示されます)。

[EUC] オプションを有効にすると、EUC コードで書かれていない Web ページも強制的に EUC コードで記録されます。したがって、このオプションを有効にする必要があるのは、VuGen が HTTP ヘッダまたは HTML のメタ・タグからはコードを検知できず、ページが EUC コードで書かれていることが事前わかっている場合だけです。

記録中、VuGen は EUC コードの文字列を Web サーバから受信すると、それを SJIS に変換します。変換された SJIS 文字列はスクリプトの Action 関数に保存されます。しかし、再生を正常に実行するためには、文字列を再び EUC に変換してから Web サーバに送り返す必要があります。このため、VuGen は Action 関数の前に `web_sjis_to_euc_param` 関数を追加します。この関数で SJIS 文字列を EUC に再変換します。

次の例では、ユーザが EUC で書かれた Web ページに移動してリンクをクリックします。VuGen は Action 関数を記録し、`web_sjis_to_euc_param` 関数を Action 関数の前のスクリプトに追加します。

```
web_sjis_to_euc_param("param_link","Search");
web_link("LinkStep","Text={param_link}");
```

記録レベルの設定

本項では、記録レベルとそれらの詳細オプションの設定の手順について説明します。

記録オプションを設定するには、次の手順を実行します。

- 1 [ツール] > [記録オプション] を選択して、[記録オプション] ダイアログ・ボックスを開きます。
- 2 記録オプション・ツリー内の [一般：記録] ノードを選択します。
- 3 記録レベルとして [GUI ベースのスクリプト] (使用可能な場合)、[HTML ベースのスクリプト]、または [URL ベースのスクリプト] を選択します。
- 4 GUI ベースの記録の場合は、記録オプションを開いて (Ctrl+F7)、[GUI プロパティ：詳細] ノードを選択し、記録時にイベントをキャプチャするための追加オプションを設定します。

詳細については、342 ページ「GUI プロパティの詳細設定」を参照してください。

- 5 HTML ベースのモードの場合は、**[HTML 詳細設定]** をクリックし、スクリプトのタイプや非 HTML 要素の処理についての追加のオプションを設定します。スクリプトのタイプを選択します。

非 HTML リソースを処理する方法を選択します。詳細については、384 ページ「HTML ベースのオプションの詳細設定」を参照してください。

- 6 URL ベースのモードの場合、**[URL 詳細設定]** をクリックし、リソースの処理やキャッシュの有効化について追加のスクリプト・オプションを設定します。

リソースを記録し、同時実行グループとしてマークするには、**[HTML ページの後にリソースの同時実行グループを作成する]** を選択します（同時実行グループは、`web_concurrent_start` と `web_concurrent_end` で囲まれます）。

記録中にブラウザ・キャッシュを使用するには、**[キャッシュの使用を有効にする]** を選択します。このオプションを有効にした場合は、**[記録する前にキャッシュをクリアする]** チェック・ボックスをクリアしておくこと、記録前に、キャッシュがクリアされず、以前にアクセスしたページが使用されます。

すべての HTTP 要求を `web_custom_request` 関数として生成するには、**[web_custom_request のみを使用する]** を選択します。これらのオプションの詳細については、389 ページ「URL ベースのオプションの詳細設定」を参照してください。

- 7 日本語版 Windows のユーザの場合は、**[EUC]** オプションを選択して、VuGen で EUC エンコーディングを使用するよう設定します。

ページに EUC コード（日本語のコンテンツ）だけが使用されている Web サイトを記録する場合は、**[EUC]** オプションを選択します。VuGen は、EUC の文字列を SJIS に変換し、`web_sjis_to_euc_param` 関数を追加します。この情報がサーバからブラウザに（HTTP ヘッダまたは HTML メタ・タグで）送信される場合は、このオプションを有効にする必要はありません。

第 21 章

ポートの割り当て設定

ソケット・レベルでネットワーク・トラフィックを記録するプロトコルを使用する場合、トラフィックをどのように割り当てるかをポートに指定することができます。

本章の内容

- ▶ ポートの割り当て設定について (396 ページ)
- ▶ ポート割り当ての定義 (396 ページ)
- ▶ 新規サーバ・エントリの追加 (399 ページ)
- ▶ 高度なポート割り当てオプションの設定 (402 ページ)
- ▶ ポートの割り当て記録オプションの設定 (404 ページ)

以降の情報は、ソケット・レベルで記録するすべての **Vuser** スクリプト (HTTP, SMTP, POP3, IMAP, Oracle NCA, Windows Sockets) を対象とします。

ポートの割り当て設定について

ソケット・レベルでネットワーク・トラフィックを記録する Vuser スクリプト (HTTP, SMTP, POP3, FTP, IMAP, Oracle NCA, Windows Sockets) を記録する場合, [ポートの割り当て] オプションを設定できます。これらのオプションで, 特定のサーバとポートの組み合わせを通じて受け取るトラフィックを, 特定の通信プロトコルに割り当てることができます。

割り当ての対象にできる通信プロトコルは, FTP, HTTP, IMAP, NCA, POP3, SMTP および SOCKET です。割り当ては, サーバ名, ポート番号, または「サーバ: ポート」の組み合わせを指定することで作成できます。たとえば, **twilight** というサーバのポート 25 番からのトラフィックをすべて SMTP として扱うよう指定できます。また, **viper** というサーバからのすべてのトラフィックを, ポートに関係なく FTP プロトコルへ割り当てすることもできます。さらには, サーバ名に関係なく, ポート 23 のすべてのトラフィックを SMTP に割り当てすることも可能です。

マルチ・プロトコル・モードで記録する場合には, 少なくとも 1 つのプロトコルがソケット・レベルで記録していると, [ポートの割り当て] オプションが利用できます。ただし, シングル・プロトコル・スクリプトで HTTP または WinSock を記録する場合は例外です。この場合は, [ポートの割り当て] オプションは利用できません。

ポート割り当ての定義

VuGen はポート割り当ての設定を使用して, 特定のサーバとポートの組み合わせを通じて受け取るトラフィックを, 特定の通信プロトコルに割り当てます。次の項目についてポート割り当ての設定が可能です。

- ▶ **[キャプチャのレベル]**: キャプチャするデータのレベル (HTTP ベースのプロトコルの場合のみ)。
 - ▶ **[ソケット レベル データ]**: トラップを使用してソケット・レベルのデータだけをキャプチャします。この場合はポート割り当てが適用されます (標準設定)。
 - ▶ **[WinINet レベル データ]**: 特定の HTTP アプリケーションで使用されている WinINet.dll API のフックを使ってデータをキャプチャします。このようなフックを使用している最も一般的なアプリケーションは Internet Explorer です。このレベルにはポート割り当てを適用できません。

- ▶ **[ソケット レベルおよび WinINet レベルのデータ]** : 両方のメカニズムを使用してデータをキャプチャします。WinINet レベルでは、WinINet DLL を使用しているアプリケーションの情報が送信されます。ソケット・レベルでは、WinINet dll から発生したのではないと判断された場合にのみデータが送信されます。ポート割り当ては、WinINet から発生していないデータに適用されます。
- ▶ **[次のネットワーク レベルのサーバアドレス割り当て]** : プロトコルごとの割り当てを表示するよう指定します。たとえば、FTP の割り当てだけを表示したい場合には [FTP] を選択します。
- ▶ **[エントリの新規作成]** : [サーバエントリ] ダイアログ・ボックスが開き、新しい割り当てを入力できます。詳細については、399 ページ「新規サーバ・エントリの追加」を参照してください。
- ▶ **[エントリの編集]** : [サーバエントリ] ダイアログ・ボックスが開き、選択したエントリを編集できます。
- ▶ **[エントリの削除]** : 選択したエントリを削除します。
- ▶ **[オプション]** : [ポートの割り当ての詳細設定] ダイアログ・ボックスが開き、通信プロトコルと SSL レベルの自動検出を有効にできます。詳細については、402 ページ「高度なポート割り当てオプションの設定」を参照してください。

設定したポートとサーバの名前が全部ではない場合、VuGen は次の優先順位に従ってデータをサービスに割り当てます。

| 優先度 | ポート | サーバ |
|-----|-----------|-----------|
| 1 | 指定あり | 指定あり |
| 2 | 指定なし<すべて> | 指定あり |
| 3 | 指定あり | 指定なし<すべて> |
| 4 | 指定なし<すべて> | 指定なし<すべて> |

優先順位の高い割り当てがある場合に、それよりも優先順位の低い割り当てを指定しても、優先順位の低い割り当ては無効です。たとえば、**twilight** というサーバのポート番号 25 番のトラフィックを SMTP として扱うよう指定した後で、すべてのサーバのポート 25 番を HTTP として扱うよう指定しても、データは SMTP として扱われます。

さらに、次のガイドラインが適用されます。

- ▶ **ポート 0** : ポート番号 0 は任意のポートを表します。
- ▶ **強制割り当て** : ポート番号, サーバ名, または「サーバ:ポート」の組み合わせの割り当てを指定した場合, **VuGen** では, ネットワーク・トラフィックがそのサービスを使用するよう強制されます。たとえば, 「<任意のサーバ>」のポート 80 番が FTP を使用するよう指定した場合, **VuGen** では, 実際の通信が HTTP であったとしても, FTP プロトコルを使用してその通信が記録されます。この例では, **Vuser** スクリプトは空となる可能性があります。

ポート割り当てを定義すると, その内容は [ポートの割り当て] の一覧に表示されます。各項目のチェック・ボックスをクリアすることで, 一時的に割り当てを無効にできます。割り当てを無効にすると, **VuGen** は無効にした「サーバ:ポート」の組み合わせに割り当てられた, すべてのトラフィックを無視します。データが無関係な場合やプロトコルがサポートされていない場合は, ポートの割り当てを無効にしてください。

手順の詳細については, 404 ページ「ポートの割り当て記録オプションの設定」を参照してください。

新規サーバ・エントリの追加

[サーバエントリ] ダイアログ・ボックスを使用して、ポート割り当てのリストに新規エントリを作成します。

サーバエントリ

ソケット サービス

ターゲットサーバ: (任意のサーバ) ポート: (任意)

サービス ID: HTTP サービス タイプ: TCP

レコード タイプ: プロキシ 接続タイプ: 自動

SSL 設定

SSL バージョン: SSL 2/3

SSL 暗号: (標準設定 OpenSSL 暗号)

指定されたクライアント側証明書 (Base64/PEM) を使用する

クライアント証明書: [] ...

パスワード: []

指定されたプロキシサーバ証明書 (Base64/PEM) を使用する

プロキシ証明書: [] ...

パスワード: []

SSL のテスト

トラフィックの転送

次のローカルポートからターゲットサーバに転送する: []

詳細

接続のセキュリティレベルを設定します。このエントリはアナライザが保護された接続の暗号を解読するのに使用されます。

更新 キャンセル

ソケット サービス

- ▶ **[ターゲットサーバ]** : エントリ項目に登録する対象サーバの IP アドレスまたはホスト名。標準設定は「任意のサーバ」です。
- ▶ **[ポート]** : エントリ項目に登録する対象サーバのポート番号。ポート番号「0」は「すべてのポート」を意味します。
- ▶ **[サービス ID]** : 接続のタイプを識別するためにレコーダが使用するプロトコルまたはサービス名 (HTTP, FTP など)。新しい名前を指定することもできます。指定できる名前の長さは最長 8 文字です。
- ▶ **[サービス タイプ]** : サービスのタイプ。現在は TCP/IP に設定されています。

- ▶ **[レコードタイプ]**：記録のタイプ（直接か、それともプロキシ・サーバを経由するか）。
- ▶ **[接続タイプ]**：接続のセキュリティ・レベル。[非認証]，[SSL]，[自動] があります。[自動] を選択すると、レコーダによって SSL シグネチャについて最初の 4 バイトが検査されます。SSL 署名を検出すると、SSL が使用されていると推定されます。

注：SSL 接続は、次のメール・サービス・プロトコルと e ビジネス・プロトコルには適用されません：FTP, LDAP, SMTP, POP3, IMAP, DNS, MAPI。

SSL 設定

接続の種類に **[SSL]** もしくは **[自動]** を選択している場合には、**[SSL 設定]** セクションの設定を行います。この設定は新規エントリにのみ適用されます。この設定は、アプリケーションの SSL エンコーディングについて明らかな情報がある場合にのみ行ってください。それ以外の場合には、標準設定を使用します。

- ▶ **[SSL バージョン]**：クライアント・アプリケーションおよびサーバとの通信に使用する SSL のバージョン。標準設定では、SSL 2/3 が指定されています。ただし、サービスによっては SSL 3.0 または SSL 2.0 のみが必要になることがあります。新しいワイヤレス・アプリケーションでは新しいセキュリティ・アルゴリズム、TLS 1.0 を必要とします。
- ▶ **[SSL 暗号]**：リモートのセキュア・サーバに接続するのに使用する SSL 暗号を指定します。
- ▶ **[指定されたクライアント側証明書 (Base64/PEM) を使用する]**：リモート・サーバに接続する際に使用する標準のクライアント側の証明書を指定します。**txt**, **crt**, **pem** 形式のいずれかの証明書を指定するか一覧から探し、パスワードを入力します。
- ▶ **[指定されたプロキシサーバ証明書 (Base64/PEM) を使用する]**：サーバの証明書を要求するクライアント・アプリケーションに示す標準の証明書を指定します。**txt**, **crt**, **pem** 形式のいずれかの証明書を指定するか一覧から探し、パスワードを入力します。**[SSL テスト]** をクリックすると、サーバに対する認証情報をチェックできます。

トラフィックの転送

- ▶ **[次のローカルポートからターゲットサーバに転送する]**：特定のポートからのすべてのトラフィックを別のサーバに転送できます。このオプションは、特別な UNIX マシンの場合など、クライアント上で VuGen を正常に実行できない場合や、VuGen を介してアプリケーション・サーバを起動できない場合に特に役立ちます。クライアント・マシンに問題がある場合は、そのマシンからのトラフィックを捕獲して、サーバに渡すよう VuGen の設定を行います。こうすることで、VuGen はデータを処理し、アクションに対応するコードを生成することが可能となります。

たとえば、**host1** という UNIX のクライアントが、サーバ **server1** と、ポート番号 8080 経由で通信しており、[ポートの割り当て] には、**server1**、ポート 8080 のエントリが設定されていたとします。[サーバエントリ] ダイアログ・ボックスの **[トラフィックの転送]** セクションで、**[次のローカルポートからターゲットサーバに転送する]** チェック・ボックスを選択して、トラフィックの転送を有効にします。トラフィックの転送に使用するポート（この例では 8080）を指定します。

次にクライアントを **server1** ではなく、VuGen を実行している **host1** に接続します。VuGen はクライアント・マシンからの通信を受信し、その通信をローカルのポート 8080 を経由してサーバに転送します。トラフィックは VuGen を経由するため、トラフィックの分析と適切なコードの生成が可能となります。

手順の詳細については、404 ページ「ポートの割り当て記録オプションの設定」を参照してください。

高度なポート割り当てオプションの設定

VuGen の高度なポート割り当てオプションでは、**自動検出**オプションを設定できます。VuGen の自動検出では、サーバに送られるデータが解析されます。さらに、シグネチャ・データやデータのパターンが調べられ、プロトコルが特定されます。シグネチャを検出するため、最初の受信バッファまで、すべての送信バッファが蓄積されます。受信バッファが返されるまでに送信されたすべての送信バッファは単一のデータ**遷移**とみなされます。標準設定では、割り当ては定義されず、VuGen は自動検出を行います。一部のプロトコル (HTTP など) は、単一の遷移のなかでタイプが判別されます。ほかのネットワーク・プロトコルでは、タイプを判別されるまでにいくつかの遷移が必要です。このため、VuGen ではサーバとポートの組み合わせごとに一時バッファが作成されます。VuGen によって最初の遷移バッファが読み取られてもプロトコル・タイプが判別できない場合は、データが一時バッファに格納されます。そして、プロトコルを特定できるシグネチャが検出されるまで、着信バッファの読み取りが継続されます。

標準では、VuGen がプロトコルのシグネチャの検出に使用できるトランザクションは 4 つ、バッファは最大 2048 バイトまでです。VuGen が最大トランザクションに達するか、バッファ・サイズの上限に達してもプロトコルを特定できなかった場合、データは WinSock プロトコルに割り当てられます。VuGen に (マルチ・プロトコルを選択している場合) WinSock プロトコルを記録するように設定していない場合には、VuGen によってデータが破棄されます。

プロトコルのタイプを検出するために VuGen が読み取るバッファの最大サイズを変更することができます。また一時バッファのサイズを指定することも可能です。最初の送信バッファに格納されたデータの合計が、一時バッファのサイズより大きくなった場合、VuGen ではプロトコル・タイプの自動検出が行えなくなります。この場合には一時バッファのサイズを増やす必要があります。

- ▶ **[自動 SSL 検出を有効にする]** : SSL 通信を自動的に検出します。検出したい SSL のバージョンと標準の SSL 暗号の形式を指定します。これはポートの割り当てが、**[接続タイプ]** ボックスで **[自動]** に指定されているか、まったく指定のない場合にだけ適用されることに注意してください。サーバ、ポート、もしくは「サーバ : ポート」の組み合わせが、**[非認証]**、**[SSL]** のいずれかに指定されている場合には、自動 SSL 検出は適用されません。

- ▶ **[SOCKET ベース コミュニケーションの自動検出を有効にする]** : SSL 通信を自動的に検出します。必要な場合、VuGen によるプロトコルの検出が成功するまで 1 つずつ [移行しきい値の送受信] の最大数を増やします。また、VuGen によるプロトコルの検出が成功するまで、一度に 1024 バイト (1KB) ずつ [バッファサイズしきい値の送受信] の最大数を増やすこともできます。これらを行うと VuGen によるシグネチャのためにより多くのデータを調査することができるようになります。
- ▶ **[ログ レベル]** : 自動ソケット検出のログ・レベル (なし, 標準 (標準設定), デバッグ, 高度なデバッグ) を設定します。

前述のネットワーク・レベル・プロトコルを使用する場合は、VuGen がプロトコル・タイプを判別するよう、自動検出オプションをオンにした設定を推奨します。ほとんどの場合、VuGen のレコーダでは、これらのプロトコルのシグネチャが認識できます。そして、プロトコルの仕様に従ってプロトコルが自動的に処理されます。ただし、VuGen でプロトコルが認識されないこともあります。たとえば、次のような場合です。


- ▶ 既存のプロトコルに類似するプロトコル・シグネチャがあり、誤った処理が行われた場合。
- ▶ プロトコルに一意のシグネチャがない場合。
- ▶ プロトコルが SSL による暗号を使用しているため、WinSock レベルで認識されない場合。

前述のどの場合も、プロトコルをホストするサーバとポートを一意に識別する情報を提供することができます。

手順の詳細については、404 ページ「ポートの割り当て記録オプションの設定」を参照してください。

ポートの割り当て記録オプションの設定

[記録オプション] ダイアログ・ボックスは、次のいくつかの方法で開くことができます。

- ▶ ツールバー・ボタン：
- ▶ キーボードのショートカット：Ctrl キーを押しながら F7 キーを押します。
- ▶ [ツール] メニュー：[ツール] > [記録オプション] を選択します。

ポートの割り当て記録オプションを設定するには、次の手順を実行します。

- 1 [記録オプション] を開き、[ネットワーク：ポートマッピング] ノードを選択します。



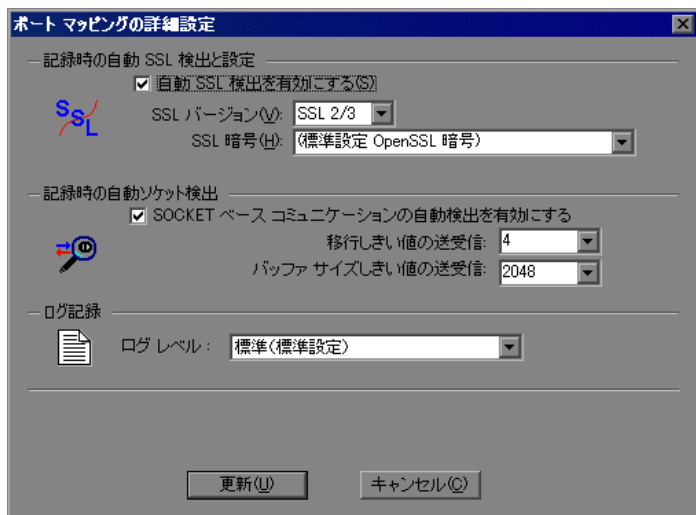
- 2 新しいサーバのポート割り当てを作成するには、[エントリの新規作成] をクリックします。[サーバエントリ] ダイアログ・ボックスが開きます。

- 3 [ソケット サービス] セクションで [サービス ID], [サービス タイプ], [ターゲットサーバ], [ポート], および [接続タイプ] を入力します。
- 4 接続の種類に [SSL] もしくは [自動] を選択している場合には、[SSL 設定] セクションの設定を行います。この設定は新規エントリにのみ適用されます。この設定は、アプリケーションの SSL エンコーディングについて明らかな情報がある場合にのみ行ってください。それ以外の場合には、標準設定を使用します。

[SSL バージョン], [SSL 暗号] を指定します。証明書を使用する場合には、[指定されたクライアント側証明書 (Base64/PEM) を使用する] または [指定されたプロキシサーバ証明書 (Base64/PEM) を使用する] を選択してユーザ情報を指定します。

[SSL のテスト] をクリックすると、サーバに対する認証情報をチェックできます。

- 5 トラフィックの転送ができるようにするには、[次のローカルポートからターゲットサーバに転送する] オプションを選択し、ポート番号を指定します。このオプションは、**ターゲット・サーバ**と**ターゲット・ポート**が一意 (<Any> が指定されていない) のときだけ、有効になります。
- 6 [更新] をクリックして、ダイアログ・ボックスを閉じます。
- 7 自動検出機能を設定するには、[オプション] をクリックします。[ポートの割り当ての詳細設定] ダイアログ・ボックスが開きます。



SSL 通信の自動検出を行うには、[自動 SSL 検出を有効にする] チェック・ボックスを選択し、バージョンと暗号の情報を指定します。

通信の種類を自動的に検出するには、[SOCKET ベース コミュニケーションの自動検出を有効にする] チェック・ボックスをチェックします。必要に応じて、移行の最大数を増やします。

[ログ レベル:] (なし、標準、デバッグ、高度なデバッグ) を選択します。

[更新] をクリックして設定を受け入れ、ダイアログ・ボックスを閉じます。

- 8 すべてのエントリを表示するには、[次のネットワークレベルのサーバアドレスの割り当て] ボックスから [(全 ID)] を選びます。

- 9 既存のエントリを修正する場合には、修正したいエントリを選択して、[**エントリの編集**] をクリックします。エントリのサーバ名、ポート番号は変更できないことに注意してください。変更できるのは、接続の種類とセキュリティ設定だけです。
- 10 割り当てを恒久的に削除するには、エントリをリストから選び、[**エントリの削除**] をクリックします。特定のエントリについて、一時的に割り当て設定を無効にするには、項目の横のチェック・ボックスをクリアします。割り当てを有効にするにはチェック・ボックスを選びます。
- 11 [OK] をクリックします。

第 4 部

実行環境設定

第 22 章

実行環境の設定

Vuser スクリプトを記録した後で、スクリプトの実行環境を設定します。これらの設定は、実行時のスクリプトの振る舞いを指定します。

本章の内容

- ▶ 実行環境の設定について (412 ページ)
- ▶ 実行論理の設定 (マルチ・アクション) (413 ページ)
- ▶ ペースの設定 (418 ページ)
- ▶ 実行環境のペースの設定 (マルチ・アクション) (419 ページ)
- ▶ ペースの設定と実行論理オプションの設定 (シングル・アクション) (420 ページ)
- ▶ 実行環境設定のログの設定 (422 ページ)
- ▶ 思考遅延時間の設定 (427 ページ)
- ▶ 実行環境の追加属性の設定 (429 ページ)
- ▶ その他の設定 (430 ページ)
- ▶ VB 実行環境の設定 (435 ページ)

実行環境の設定について

Vuser スクリプトを記録した後、そのスクリプトの実行環境を設定できます。実行環境の設定は、スクリプトの実行方法を規定します。これらの設定は、Vuser スクリプトのディレクトリにある **default.cfg** ファイルに格納されます。実行環境の設定は、VuGen、Controller、または Business Process Monitor を使ってスクリプトを実行するときに、Vuser に適用されます。

実行環境の設定を行うことによって、さまざまな種類のユーザの動作をエミュレートできます。たとえば、サーバの出力にすぐに応答するユーザをエミュレートすることも、作業を停止して考えてから応答するユーザをエミュレートすることもできます。また実行環境の設定では、Vuser がアクションを反復する回数も指定できます。

実行環境設定の表示と設定は、[実行環境設定] ダイアログ・ボックスを使って行います。これらの設定は、以下のいずれかの方法で開くことができます。



- ▶ VuGen ツールバーの **[実行環境の設定]** ボタンをクリックします。
- ▶ キーボードのショートカット・キー、**F4** キーを押します。
- ▶ **[仮想ユーザ]** > **[実行環境の設定]** を選択します。

LoadRunner Controller からでも実行環境の設定を変更できます。詳細については、各製品のドキュメントを参照してください。

注： LoadRunner の場合は、標準の実行環境の設定で VuGen のデバッグ環境と Controller の負荷テスト環境がサポートされます。この標準の設定は、次のとおりです。

- ▶ **[思考遅延時間]**：VuGen では無効になっていますが、Controller では記録されたとおりに再生されます。
 - ▶ **[ログ]**：VuGen では標準で有効になっていますが、Controller では無効になっています。
 - ▶ **[HTML 以外のリソースをダウンロードする]**：VuGen と Controller の両方で有効になっています。
-

本章で説明する一般的な実行環境設定は、すべてのタイプの Vuser スクリプトに適用されます。説明する項目は次のとおりです。

- ▶ 実行論理（反復）
 - ▶ ペースの設定
 - ▶ ログ
 - ▶ 思考遅延時間
 - ▶ その他
 - ▶ 追加属性

WinSock やデータベース（Oracle 2-tier, Sybase, MSSQL など）など、複数のアクションをサポートしないプロトコルでは、反復オプションとペースの設定オプションはどちらもペースの設定ノードで処理できます。多くのプロトコルには、追加の実行環境設定があります。これらのプロトコルの Vuser 固有の実行環境の設定については、該当する項を参照してください。

実行論理の設定（マルチ・アクション）

注：次の項の内容は、複数のアクションで動作するプロトコルを対象としています。[実行環境設定] の [一般] の下に [実行論理] ノードがあれば、複数アクション対応のプロトコルです。シングル・アクション・プロトコルについては、420 ページ「ペースの設定と実行論理オプションの設定（シングル・アクション）」を参照してください。

どの Vuser スクリプトにも、**vuser_init**、**実行 (Action)**、および **vuser_end** の 3 つのセクションがあります。Vuser がスクリプトの実行時に「**実行**」セクションを繰り返し実行するように指定できます。この繰り返しを「**反復**」といいます。

反復を複数回実行する場合、Vuser スクリプトの **vuser_init** セクションと **vuser_end** セクションは繰り返されません。

[実行環境設定] ダイアログ・ボックスを開き、[一般: 実行論理] ノードを選択します。



- ▶ [反復回数]: 反復の回数です。Vuser は、すべての Action セクションを指定した回数だけ繰り返し実行します。

注: LoadRunner Controller では、スケジュールの設定でシナリオの継続期間を指定すると、その設定が Vuser の反復の設定に優先します。つまり、この継続時間が 5 分（標準設定）に設定してあると、実行環境の設定で反復回数を 1 回に設定しても、Vuser は 5 分間反復し続けます。

複数のアクションが含まれるスクリプトを実行するときに、アクションの実行方法を指定し、Vuser がどのようにアクションを実行するかを設定できます。

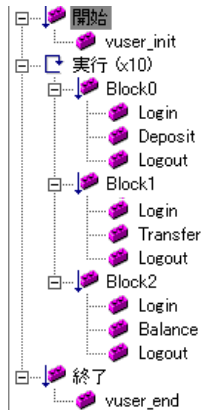
- ▶ [アクション ブロック]: アクション・ブロックは、スクリプト内のアクションのグループです。各ブロックのプロパティ（順序、反復、重み付け）を個別に設定できます。
- ▶ [順番]: スクリプト内のアクションの順序を設定できます。アクションを順番に実行するか、ランダムに実行するかを指定することもできます。

- ▶ **[反復]**：「実行」セクション全体の反復回数を設定する以外に、各アクションまたはアクション・ブロックの反復回数を設定することもできます。これはたとえば、製品を探すのに多数のクエリを実行するけれども、購入する場合は 1 回だけというような商用サイトでの操作のエミュレーションに役立ちます。
- ▶ **[重み付け]**：アクションをランダムに実行するアクション・ブロックには、**重み付け**（ブロック内の各アクションの割合）を設定できます。

アクション・ブロックの作成

アクション・ブロックは、Vuser スクリプト内のアクションのグループです。アクションをグループ化して個別のアクション・ブロックを作成し、同じアクションを複数のブロックに追加できます。アクション・ブロックまたは各アクションを順番（Sequential）に実行するか、ランダム（Random）に実行するかを設定できます。標準設定の Sequential モードでは、Vuser は、反復のツリー・ビューに表示されている順番でブロックまたはアクションが実行されます。

次の例では、**Block0** は預け入れ、**Block1** は振り替えを実行し、**Block2** は残高要求を送信します。**Login** と **Logout** のアクションは、3 つのブロックに共通です。



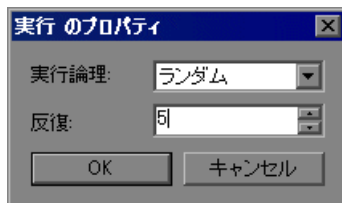
各ブロックの順序、反復を個別に設定します。

アクションとアクション・ブロックを設定するには、次の手順を実行します。

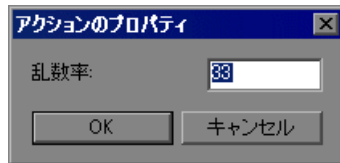
- 1 記録またはプログラミングによって、必要なアクションをすべて作成します。
- 2 [実行環境設定] ダイアログ・ボックスを開きます。[一般: 実行論理] ノードを選択します。
- 3 新規アクション・ブロックを追加します。[ブロックの挿入] をクリックします。VuGen によって、使用可能な次のインデックスが付いた新しいアクション・ブロック (**Block0**, **Block1**, **Block2**) が挿入ポイントに挿入されます。
- 4 ブロックにアクションを追加します。[アクションの挿入] をクリックします。[アクションを選択] リストが開きます。



- 5 ブロックに追加するアクションを選択し、[OK] をクリックします。VuGen によって、現在のブロックまたはセクションに新規アクションが挿入されます。
- 6 ブロックに追加するアクションごとに、手順 3 を繰り返します。
- 7 アクションまたはアクション・ブロックを削除するには、対象を選択して [削除] ボタンをクリックします。
- 8 項目の位置を変えるには、[上へ移動] または [下へ移動] をクリックします。
- 9 反復回数とアクションの実行論理を設定するには、[プロパティ] をクリックします。各アクションまたはアクション・ブロックのプロパティ・ダイアログ・ボックスが開きます。



- 10 **[実行論理]** リストから **[順次]** または **[ランダム]** を選択します。これにより VuGen にアクションを順番に実行するかランダムに実行するかを指定します。
- 11 **[反復]** ボックスで反復の回数を指定します。アクション・ブロック内でパラメータを定義し、反復ごとにパラメータ値を更新するよう VuGen に指示した場合の「反復ごと」というのは個々のブロックの反復ではなくグローバルな反復のことです。
- 12 **[OK]** をクリックします。
- 13 実行論理が「ランダム」のブロックには、各アクションの重み付けを設定します。アクションを右クリックして、**[プロパティ]** を選択します。**[アクションのプロパティ]** ダイアログ・ボックスが開きます。



選択したブロックまたはアクションに対して、必要な割合を指定します。**[乱数率]** ボックスで、対象アクションの割合を指定します。割合の合計は 100% にならなければなりません。

- 14 プロパティを設定する各要素に対して、前述の手順を繰り返します。

ペースの設定

注：次の項の内容は、複数のアクションで動作するプロトコルを対象としています。[実行環境設定] の [一般] の下に [**実行論理**] ノードがあれば、複数アクション対応のプロトコルです。シングル・アクション・プロトコルについては、420 ページ「ペースの設定と実行論理オプションの設定 (シングル・アクション)」を参照してください。

[実行環境の設定] の [ペースの設定] を設定することで、反復の間隔を変えることができます。ペースは、アクションの反復の間で待機する時間を **Vuser** に指示します。各反復の開始間隔として次のオプションを指定できます。

- ▶ [**前回の反復が終了次第すぐ**]：直前の反復の終了後、すぐに次の反復を開始します。
- ▶ [**前回の反復が終了後**], [**一定**] / [**値の範囲**] [遅延間隔] [X 秒]：直前の反復の終了後、指定した時間が経過したら、次の反復を開始します。正確な秒数または時間の範囲を指定します。たとえば、直前の反復が終了してから 60 ～ 90 秒後に次の反復を開始するように指定できます。

スクリプト実行時に、反復が終了してから次の反復を開始するまでに **Vuser** が待機した時間は、**VuGen** の実行ログに示されます。

- ▶ [**一定**] / [**値の範囲**] [間隔] [各 X 秒]：反復と反復の間の時間を指定します。秒単位で固定時間または時間の範囲を定義します。たとえば、新しい反復を 30 秒ごとに開始したり、30 ～ 45 秒ごとのランダムな間隔で開始したりするように指定できます。反復は、直前の反復が終了してからのみ開始します。

スケジュールが設定された各反復は、前回の反復が完了した後に開始されません。スクリプト実行時に、反復が終了してから次の反復を開始するまでに **Vuser** が待機した時間は、**VuGen** の実行ログに示されます。

たとえば、4 秒ごとに新しい反復を開始するように定義すると、次のようになります。

- ▶ 最初の反復が 3 秒かかると、**Vuser** は次の反復で 1 秒待機します。
- ▶ 最初の反復が 2 秒かかると、**Vuser** は 2 秒待機します。

- ▶ 最初の反復が 8 秒かかると、次の反復は、最初の反復が開始してから 8 秒後に開始します。VuGen の実行ログには、反復のペースを守れなかったことを示すメッセージが表示されます。

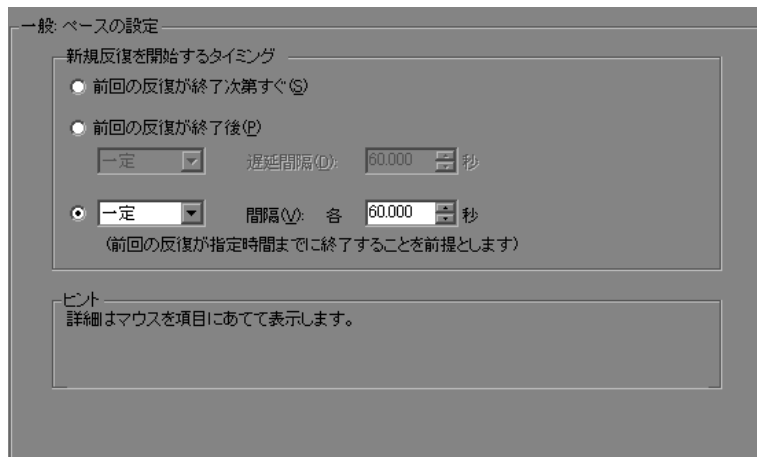
ペースの設定のオプションの詳細については、419 ページ「実行環境のペースの設定（マルチ・アクション）」を参照してください。

実行環境のペースの設定（マルチ・アクション）

ペースの設定のオプションを使用して、アクションの反復の間隔を設定することでアクションのペースを設定できます。

反復の間隔を設定するには、次の手順を実行します。

- 1 [実行環境設定] ダイアログ・ボックスを開き、[一般：ペースの設定] ノードを選択します。



- 2 [新規反復を開始するタイミング] セクションで、次のいずれかを選択します。
 - ▶ [前回の反復が終了次第すぐ]
 - ▶ [前回の反復が終了後]
 - ▶ [一定] または [値の範囲]

- 3 [前回の反復が終了後] オプションには次の設定があります。
 - ▶ 遅延の種類を [一定] か [値の範囲] から選択します。
 - ▶ [一定] で値を指定するか, [値の範囲] で値の範囲を指定します。
- 4 [各…秒] オプションには次の設定があります。
 - ▶ 間隔の種類を [一定] か [値の範囲] から選択します。
 - ▶ [一定] で値を指定するか, [値の範囲] で値の範囲を指定します。
- 5 [OK] をクリックします。

ペースの設定と実行論理オプションの設定 (シングル・アクション)

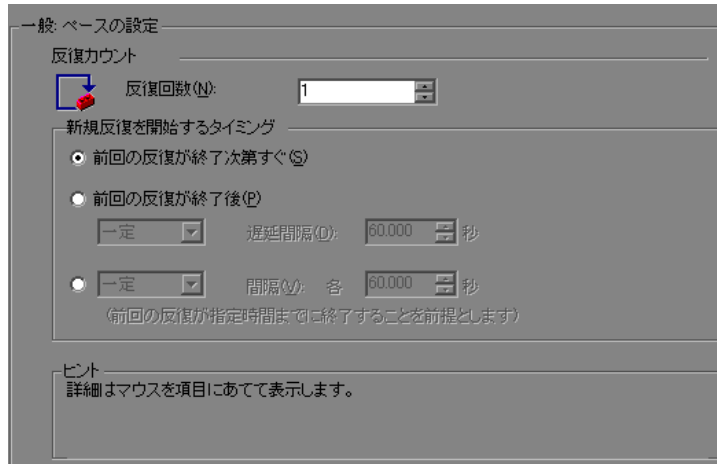
注: 次の項の内容は、複数のアクションではなく単一のアクションを使用するプロトコルを対象としています。[実行環境設定] の [一般] の下に [ペースの設定] ノードと [実行論理] ノードがなければ、単一アクションのプロトコルです。

Vuser がスクリプトの実行時に「**Action**」セクションを繰り返し実行するように指定できます。この繰り返しを「**反復**」といいます。反復を複数回実行する場合、Vuser スクリプトの **vuser_init** セクションと **vuser_end** セクションは繰り返されません。

反復とペースの設定を設定するには、次の手順を実行します。



- 1 VuGen ツールバーの [実行環境の設定] ボタンをクリックするか, [仮想ユーザ] > [実行環境の設定] を選択して, [実行環境設定] ダイアログ・ボックスを開きます。[ペースの設定] ノードをクリックして, 反復とペースのオプションを表示します。



- 2 **[反復回数]** ボックスで反復の回数を指定します。Vuser は、すべての Action セクションを指定した回数だけ繰り返し実行します。
- 3 **[新規反復を開始するタイミング]** セクションで、次のいずれかを選択します。
 - ▶ [前回の反復が終了次第すぐ]
 - ▶ [前回の反復が終了後]
 - ▶ [一定] または [値の範囲]
- 4 **[前回の反復が終了後]** オプションには次の設定があります。
 - ▶ 遅延の種類を [一定] か [値の範囲] から選択します。
 - ▶ [一定] で値を指定するか、[値の範囲] で値の範囲を指定します。
- 5 **[各 … 秒]** オプションには次の設定があります。
 - ▶ 間隔の種類を [一定] か [値の範囲] から選択します。
 - ▶ [一定] で値を指定するか、[値の範囲] で値の範囲を指定します。
- 6 **[OK]** をクリックします。

ペースの設定のオプションの概要については、418 ページ「ペースの設定」を参照してください。

実行環境設定のログの設定

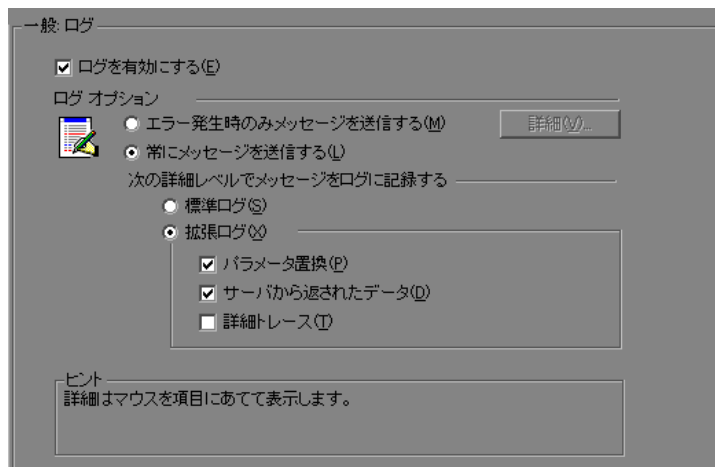
実行中、Vuser は自身に関する情報と、サーバとの通信に関する情報をログに書き込みます。Windows 環境では、この情報はスクリプトのディレクトリにある **output.txt** というファイルに格納されます。UNIX 環境では、この情報は標準出力に送られます。ログの情報は、デバッグの際に役立ちます。

ログの実行環境を設定して、出力へ送られるログ記録の情報量を指定できます。**標準**ログまたは**拡張**ログを選択できます。また、ログの記録を一切無効にすることもできます。ログを無効にすることは、多数の Vuser を実行する場合に有用です。何十、何百という Vuser が実行時の情報をディスクに記録すると、システムの速度が通常より遅くなる場合があります。開発時は、再生に関する情報を得られるようにログ記録を有効にしておきます。ログの記録を無効にするときは、その前に必ずスクリプトが正しく動作することを確認してください。

注： `lr_error_message` 関数と `lr_output_message` 関数を使って、出力ログにメッセージを送信するように Vuser スクリプトをプログラミングできます。



[**実行環境の設定**] ボタンをクリックするか、[**仮想ユーザ**] > [**実行環境の設定**] を選択して、[実行環境設定] ダイアログ・ボックスを表示します。[**一般：ログ**] タブをクリックして、ログのオプションを表示します。



ログを有効にする

このオプションは、再生中の自動ログ記録を有効にします。つまり、VuGen によって実行ログにログ・メッセージが書き込まれます。このオプションは、自動ログ記録と `lr_log_message` を通じて発行されるログ・メッセージだけを対象とします。`lr_message`、`lr_output_message`、`lr_error_message` を使って手作業で送信されるメッセージは、変わらずに発行されます。

ログのオプション

実行環境設定のログの設定では、開発の段階に応じてログ記録のレベルを調整できます。

ログ・メッセージをログに送信する条件を指定できます。[**エラー発生時のみメッセージを送信する**] または [**常にメッセージを送信する**] のどちらかを選択します。開発時には、ログ記録を有効にしておけます。スクリプトのデバッグが終了し、正しく動作することを確認したら、エラー・ログのみを有効にできます。

エラーの発生時にのみメッセージを送信する場合（JIT（ジャスト・イン・タイム）メッセージングとも呼ばれる）は、ログのキャッシュ・サイズなど、追加の詳細オプションを設定できます。詳細については、425 ページ「ログのキャッシュ・サイズの設定」を参照してください。

ログ詳細レベルの設定

ログに記録される情報の種類を指定したり、すべてのログ記録を無効にしたりできます。

注：[**実行環境設定**] ダイアログ・ボックスの [**一般**] タブで [**エラー処理**] に [**エラーでも処理を継続する**] を設定している場合でも、エラー・メッセージは出力ウィンドウに送信されます。スクリプトのログ詳細レベルを変更しても、`lr_message`、`lr_output_message`、および `lr_log_message` の各関数の動作は変更されず、メッセージは送信され続けます。

- ▶ **[標準ログ]**：スクリプトの実行中に実行された関数や送信されたメッセージの標準ログが生成されます。これらの標準ログはデバッグで使用します。大規模な負荷テスト・シナリオまたはプロファイルでは、このオプションは無効にしてください。

ログの記録レベルを「**標準ログ**」に設定したスクリプトをシナリオまたはプロファイルに追加した場合、ログ記録モードは自動的に **[JIT ログ記録]** に設定されます。ただし、ログ記録モードが無効になっているか、**[拡張ログ]** に設定されている場合は、スクリプトをシナリオまたはプロファイルに追加しても、ログの設定には何の影響もありません。

- ▶ **[拡張ログ]**：警告やメッセージを含めた詳細なログが作成されます。大規模な負荷テスト・シナリオまたはプロファイルでは、このオプションは無効にしてください。

[拡張ログ] オプションでは、拡張ログにどの情報を追加するかを指定できます。

- ▶ **[パラメータ置換]**：このオプションを選択すると、スクリプトに割り当てられたすべてのパラメータがそれらの値とともにログに記録されます。詳細については、第 13 章「VuGen パラメータを使った作業」を参照してください。
- ▶ **[サーバから返されたデータ]**：このオプションを選択すると、サーバによって返されたすべてのデータがログに記録されます。
- ▶ **[詳細トレース]**：このオプションを選択すると、セッション中に **Vuser** が送信したすべての関数とメッセージがログに記録されます。このオプションは、**Vuser** スクリプトをデバッグするときに役立ちます。

VuGen によるイベントのログの記録の程度（標準、パラメータ置換など）は **メッセージ・クラス** ともいいます。メッセージ・クラスには、標準（Brief）、詳細（Extended）、パラメータ（Parameters）、結果データ（Result Data）、完全トレース（Full Trace）の 5 つがあります。

手作業でスクリプト内にメッセージ・クラスを設定する場合は、

lr_set_debug_message 関数を使用します。これは、スクリプトのごく一部分だけについてデバッグ情報を受け取る場合などに便利です。

たとえば、ログの設定を [標準ログ] に設定し、スクリプトの特定のセクションについて拡張ログを取得するとします。この場合は、`lr_set_debug_message` 関数を使って、スクリプト内の対象の場所でメッセージ・クラスを [拡張] に設定します。拡張モードのタイプ (Parameter, Result Data, Full Trace) を指定するときは、この関数をもう一度呼び出す必要があります。標準ログ・モードに戻るには、`lr_set_debug_message` 関数を呼び出して標準モードを指定します。メッセージ・クラスの設定の詳細については、『**Online Function Reference**』(英語版) ([ヘルプ] > [関数リファレンス]) を参照してください。

ログのキャッシュ・サイズの設定

実行環境のログ設定の詳細オプションでは、ログのキャッシュ・サイズを指定できます。ログのキャッシュには、テスト実行に関する未処理のデータが格納され、エラーが生じた場合に参考にできます。キャッシュの内容が指定したサイズを超えると、一番古い項目が削除されます。標準設定のサイズは 1 KB です。

ログ記録は、次のように行われます。

- 1 エラーが生じたときにだけメッセージをログに記録するには、[**エラー発生時のみメッセージを送信する**] を選択します。
- 2 テスト実行に関する情報がファイルに書き出されずに、ログのキャッシュに格納されます。この情報が 1 KB を超えると、古いデータから上書きされます。[実行ログ] タブは、ログ・ファイルの内容を表示するので、この場合には空のままとなります。
- 3 エラーが発生すると (内部エラーか `lr_error_message` を使用したプログラムによるものかを問わず)、キャッシュの内容がログ・ファイルと [実行ログ] タブに移されます。これを参照することで、エラーが発生するまでの状況を確認できます。

エラーが発生して VuGen によってキャッシュの内容をログ・ファイルに書き出されると、ログ・ファイルのサイズはキャッシュのサイズよりも大きくなります。たとえば、キャッシュ・サイズが 1KB ならば、ログ・ファイルのサイズは 50 KB になることもあります。これは正常な動作であり、未処理のデータをわかりやすい文章に整えるために必要なオーバーヘッドを反映しているに過ぎません。

JIT モードの場合、`lr_message` 関数および `lr_log_message` 関数の出力は、エラー発生時に出力がログのキャッシュにあった場合にのみ、[出力] ウィンドウおよびログ・ファイルには送られません。個々のメッセージ文字列については、[実行ログ] を参照してください。

CtLib サーバ・メッセージのログの記録

CtLib Vuser スクリプト（クライアント / サーバ・タイプのプロトコルの下にある Sybase CtLib）を実行する場合、CtLib クライアントによって生成されるメッセージはすべて、標準ログおよび出力ファイルのログに記録されます。標準設定では、サーバ・メッセージはログに記録されません。サーバ・メッセージのログの記録を（デバッグ用に）有効にするには、次の行を Vuser スクリプトに挿入します。

```
LRD_CTLIB_DB_SERVER_MSG_LOG;
```

VuGen によって、すべてのサーバ・メッセージが標準ログに書き込まれます。

サーバ・メッセージを（標準ログ以外の）出力に送信するには、次のように入力します。

```
LRD_CTLIB_DB_SERVER_MSG_ERR;
```

サーバ・エラーを記録しない標準のモードに戻るには、次の行をスクリプトに入力します。

```
LRD_CTLIB_DB_SERVER_MSG_NONE;
```

注：生成されるサーバ・メッセージは長く、ログの書き込み処理によってシステムの処理速度が低下することがあるため、サーバ・メッセージのログの記録はスクリプト内の特定のコード・ブロックだけを対象に有効にします。

思考遅延時間の設定

Vuser の思考遅延時間は、実際のユーザがアクションとアクションの間で待つ時間をエミュレートしたものです。たとえば、ユーザはサーバからデータを受け取ったときに、データを数秒間かけて確認してから応答するかもしれません。この遅れを「思考遅延時間」といいます。VuGen は `lr_think_time` 関数を使って、Vuser スクリプトに思考遅延時間の値を記録します。次に示す記録された関数は、次のアクションを実行する前にユーザが 8 秒待機したことを示します。

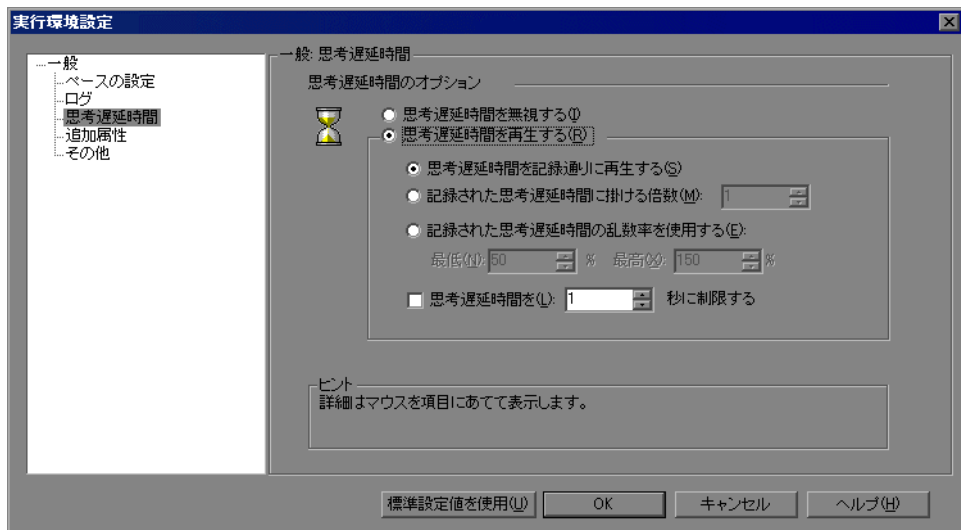
```
lr_think_time(8);
```

標準設定では、Vuser スクリプトを実行して前述の `lr_think_time` ステートメントに到達すると、Vuser は次のアクションを実行する前に 8 秒間待機します。思考遅延時間の設定を行うことにより、記録された思考遅延時間をスクリプト実行時に Vuser にどのように適用するかを設定できます。

`lr_think_time` 関数の詳細と手作業での変更方法については、『**Online Function Reference**』（英語版）（[ヘルプ] > [関数リファレンス]）を参照してください。



VuGen ツールバーの [実行環境の設定] ボタンをクリックするか、[仮想ユーザ] > [実行環境の設定] を選択して、[実行環境設定] ダイアログ・ボックスを開きます。[一般：思考遅延時間] ノードをクリックして、思考遅延時間のオプションを表示します。



思考遅延時間のオプション

標準設定では、Vuser スクリプトを実行すると、Vuser は記録セッション中にスクリプトに記録された思考遅延時間の値を使用します。VuGen では、記録された思考遅延時間の使用、思考遅延時間の無視、記録された思考遅延時間を基に指定した値の使用が可能です。

- ▶ **[思考遅延時間を無視する]**：記録された思考遅延時間を無視します。つまり、すべての `lr_think_time` 関数を無視してスクリプトを再生します。
- ▶ **[思考遅延時間を再生する]**：2 つ目の思考遅延時間オプションを選択すると、記録された思考遅延時間を使用できます。
 - ▶ **[思考遅延時間を記録通りに再生する]**：再生時に、`lr_think_time` 関数の引数の値が使用されます。たとえば、`lr_think_time(10)` は 10 秒間待機します。
 - ▶ **[記録された思考遅延時間に掛ける倍数]**：再生時に、記録された思考遅延時間の倍数を使用します。これにより再生中に適用される思考遅延時間を増減させることができます。たとえば、4 秒間の思考遅延時間が記録されている場合に、その値を 2 倍するように指定すれば、Vuser の思考遅延時間は 8 秒になります。思考遅延時間を 2 秒に減らすには、記録された時間に 0.5 を掛けます。
 - ▶ **[記録された思考遅延時間の乱数率を使用する]**：記録された思考遅延時間のランダムな割合 (%) を使用します。思考遅延時間の範囲を指定することによって、思考遅延時間の値の範囲を設定します。たとえば、思考遅延時間の引数が 4 の場合、下限を 50%、上限を 150% に設定すると、最短の思考遅延時間は 2 (50%)、最長の思考遅延時間は 6 (150%) となります。
 - ▶ **[思考遅延時間を X 秒に制限する]**：思考遅延時間の最大値を制限します。

実行環境の追加属性の設定

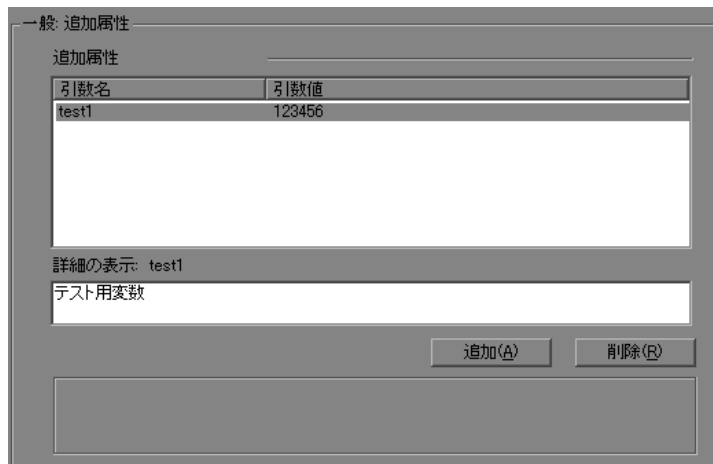
[追加属性] ノードを使用して Vuser スクリプトの追加属性を指定できます。
[追加属性] の設定は、すべてのタイプの Vuser スクリプトに適用されます。

`lr_get_attrib_string` を使用すると、テスト実行中のある時点で取得できるコマンド・ライン引数を指定できます。このノードを使用して、作成されたスクリプトに外部パラメータを渡すことができます。

追加属性を設定するには、次の手順を実行します。



- 1 [実行環境の設定] ボタンをクリックするか、[仮想ユーザ] > [実行環境の設定] を選択して、[実行環境設定] ダイアログ・ボックスを表示します。左の表示枠のツリーで [一般:追加属性] ノードを選択します。



- 2 [追加] をクリックして、新しいコマンド・ライン引数のエントリを追加します。属性の名前と値を入力します。
- 3 選択した引数を削除するには、[削除] をクリックします。

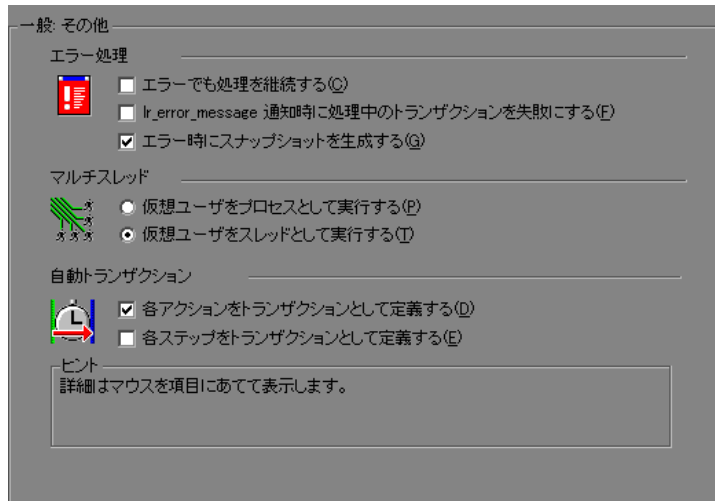
その他の設定

Vuser スクリプトでは、次に示すその他の実行環境オプションを設定できます。ただし、マルチ・スレッドと自動トランザクションのオプションは HP Business Availability Center には適用されません。

- ▶ エラー処理
- ▶ マルチ・スレッド
- ▶ 自動トランザクション



[**実行環境の設定**] ボタンをクリックするか、[**仮想ユーザ**] > [**実行環境の設定**] を選択して、[**実行環境設定**] ダイアログ・ボックスを表示します。左の表示枠のツリーで [**一般：その他**] ノードを選択します。



[**その他**] の設定は、すべてのタイプの Vuser スクリプトに適用されます。

エラー処理

- ▶ [**エラーでも処理を継続する**] : エラーが発生しても Vuser がスクリプト実行を継続するように指定します。標準ではこのオプションは無効になっているため、エラーが発生すると Vuser は終了します。

- ▶ **[lr_error_message 通知時に処理中のトランザクションを失敗にする]** :
lr_error_message 関数が発行されたすべてのトランザクションを「失敗」としてマークするように指定します。If ステートメントをプログラミングし、ある条件が満たされたときに lr_error_message 関数が発行されるようにします。
- ▶ **[エラー時にスナップショットを生成する]** : エラー発生時にスナップショットを生成します。スナップショットは、Vuser ログでエラーが発生した行をダブルクリックすると表示できます。

負荷テスト環境で **[エラーでも処理を継続する]** オプションと **[エラー時にスナップショットを生成する]** オプションを両方とも有効にすることはお勧めしません。このように設定すると、Vuser のパフォーマンスに悪影響を与える可能性があります。

データベース Vuser のエラー処理

データベース・プロトコル (LRD) を使っているときに、スクリプトの特定のセグメントに対するエラー処理を制御できます。対象のセグメントを指定するには、セグメントを LRD_ON_ERROR_CONTINUE ステートメントと LRD_ON_ERROR_EXIT ステートメントで囲みます。Vuser によって指定したセグメント全体に新しいエラー設定が適用されます。[エラーでも処理を継続する] を指定すると、VuGen がエラーに遭遇したとき、それを無視したことを示すメッセージが発行されます。

たとえば、[エラーでも処理を継続する] 機能を有効にしている、Vuser が次に示すスクリプトのセグメントの再生中にエラーに遭遇した場合には、スクリプトの実行は継続されます。

```
lrd_stmt(Csr1, "select..."...);
lrd_exec(...);
```

Vuser に、特定のセグメントでエラーが発生した場合を除き、エラーが発生してもスクリプト全体の実行を継続するように指示するには、[エラーでも処理を継続する] オプションを選択し、除外するセグメントを LRD_ON_ERROR_EXIT ステートメントと LRD_ON_ERROR_CONTINUE ステートメントで囲みます。

```
LRD_ON_ERROR_EXIT;
lrd_stmt(Csr1, "select..."...);
lrd_exec(...);
LRD_ON_ERROR_CONTINUE;
```

LRD_ON_ERROR ステートメントを使用する以外に、「重要度のレベル」に基づいてエラー処理を制御する方法もあります。LRD_ON_ERROR ステートメントは、データベース関連、不正なパラメータなど、すべてのタイプのエラーを検出します。データベース操作のエラー（エラー・コード 2009）が発生したときにだけ、Vuser を終了させたい場合には、関数の重要度のレベルを設定します。データベース操作を実行するすべての関数では、関数の最後のパラメータである **miDBErrorSeverity** で示される重大度レベルが使用されます。

VuGen では次の重要度のレベルがサポートされています。

| 定義 | 意味 | 値 |
|-------------------------------|---|---|
| LRD_DB_ERROR_SEVERITY_ERROR | データベース・アクセス・エラーが発生した時点で、スクリプトの実行を終了します（標準設定）。 | 0 |
| LRD_DB_ERROR_SEVERITY_WARNING | データベース・アクセス・エラーが生じてもスクリプトの実行を継続しますが、警告を出します。 | 1 |

たとえば、次のデータベース・ステートメントが失敗した場合（たとえば、テーブルが存在していないなど）、スクリプトの実行は終了します。

```
lrd_stmt(Csr1, "insert into EMP values ('Smith',301)¥n", -1, 1, 1, 0);
```

データベース操作エラーが発生しても、スクリプトの実行を続けるようにするには、ステートメントの重要度を 0 から 1 に変えます。

```
lrd_stmt(Csr1, "insert into EMP values ('Smith',301)¥n", -1, 1, 1, 1);
```

注： [エラーでも処理を継続する] を有効にすると、その設定が重要度「0」に優先し、データベース・エラーが発生したときでも、スクリプトの実行は継続されます。また [エラーでも処理を継続する] を無効にしても、重要度を「1」に指定すると、データベース・エラーが発生してもスクリプトの実行は継続されます。

RTE Vuser のエラー処理

RTE Vuser を使用する場合は、個々の関数のエラー処理を制御できます。動作を変える関数の手前に `lr_continue_on_error(0)`; ステートメントを挿入します。Vuser は、スクリプト実行の終了まで、または別の `lr_continue_on_error` ステートメントが発行されるまで、新しい設定を使用します。

たとえば、[エラーでも処理を継続する] 機能を有効にしている、Vuser が次に示すスクリプトのセグメントの再生中にエラーに遭遇した場合には、スクリプトの実行は継続されます。

```
TE_wait_sync();
TE_type(...);
```

特定のセグメントを除くスクリプト全体で、エラーが発生しても Vuser が実行を継続するように指定するには、[エラーでも処理を継続する] オプションを選択し、次のように除外するセグメントを `lr_continue_on_error` ステートメントで囲み、0 を渡して [エラーでも処理を継続する] オプションを無効にし、1 を渡して有効にします。

```
lr_continue_on_error(0);
TE_wait_sync();
lr_continue_on_error(1);
....
```

マルチ・スレッド

Vuser ではマルチ・スレッド環境がサポートされています。マルチ・スレッド環境の主な利点は、Load Generator ごとに多数の Vuser を実行できることです。スレッドとして実行できるのは、スレッドセーフのプロトコルだけです (HP Business Availability Center には適用されません)。

注 : Sybase-CtLib, Sybase-DbLib, Informix, Tuxedo, PeopleSoft-Tuxedo の各プロトコルはスレッドセーフではありません。

- ▶ マルチ・スレッドを実行するには、**[仮想ユーザをスレッドとして実行する]** をクリックします。
- ▶ マルチ・スレッドを無効にして、各 Vuser を個別のプロセスとして実行するには、**[仮想ユーザをプロセスとして実行する]** をクリックします。

Controller は、ドライバ・プログラム（たとえば **mdrv.exe**、**r3vuser.exe** など）を使って Vuser を実行します。各 Vuser を個別のプロセスとして実行した場合、同じドライバ・プログラムが Vuser のインスタンスごとにメモリ上でいくつも実行（およびロード）されます。同じドライバ・プログラムをメモリにロードすると、多くの RAM とその他のシステム・リソースが消費されます。そのため、Load Generator で実行できる Vuser の数が限定されることもあります。

別の方法として、各 Vuser をスレッドとして実行すると、Controller は 50 の Vuser に対してドライバ・プログラム（**mdrv.exe** など）のインスタンスを 1 つだけ起動します（標準設定）。このドライバ・プロセス/プログラムは、いくつもの Vuser を起動し、各 Vuser はスレッドとして実行されます。これらのスレッド化された Vuser は、親ドライバ・プロセスのメモリのセグメントを共有します。ドライバ・プロセス/プログラムを何度も再ロードする必要がなくなるので、メモリ空間を大幅に節約でき、1 台の Load Generator で実行できる Vuser の数を増やせます。

自動トランザクション

LoadRunner が Vuser スクリプト内の各ステップまたはアクションをトランザクションとして処理するように指定できます（HP Business Availability Center には適用されません）。これを「自動トランザクションの使用」といいます。LoadRunner は、ステップ名またはアクション名をトランザクションの名前として割り当てます。標準設定では、各アクションに対して自動トランザクションが有効になっています。

- ▶ 各アクションに対して自動トランザクションを無効にするには、**[各アクションをトランザクションとして定義する]** チェック・ボックスをクリアします（標準設定では有効）。
- ▶ 各ステップに対して自動トランザクションを有効にするには、**[各ステップをトランザクションとして定義する]** チェック・ボックスを選択します。（標準設定では無効）。

自動トランザクションを無効にしている場合でも、記録中および記録後に手作業でトランザクションを挿入できます。手作業によるトランザクションの挿入の詳細については、第 6 章「Vuser スクリプトの拡張」を参照してください。

注：シナリオの実行中に診断ブレークダウン・データ（J2EE）を生成する必要がある場合は、自動トランザクションを使用してはなりません。各トランザクションの開始と終了を手作業で定義してください。

VB 実行環境の設定

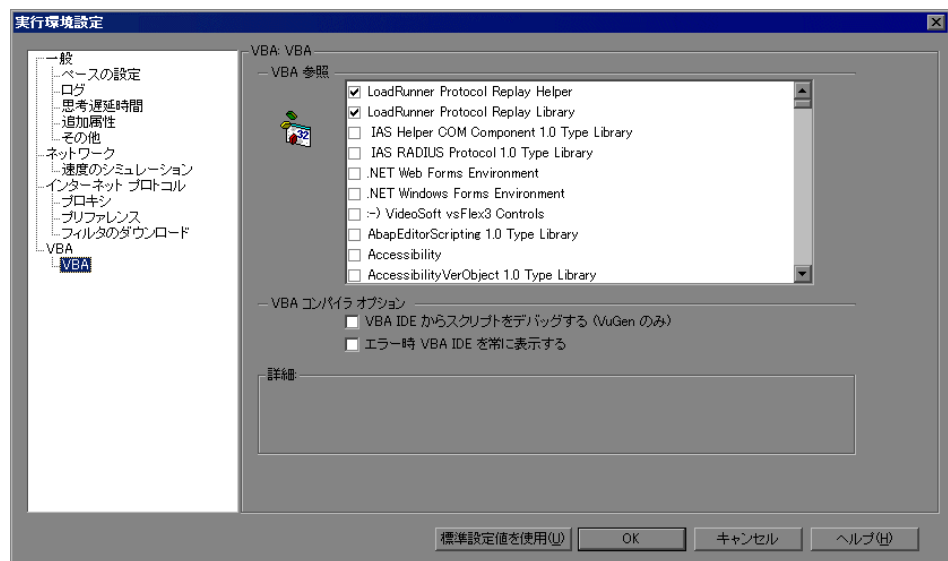
Visual Basic によるスクリプトを実行する前に、再生時に参照するライブラリを指定します。VuGen は、マシンに格納されているすべてのライブラリの一覧を表示します。



実行環境設定の表示と設定を行うには、[実行環境設定] ダイアログ・ボックスを使用します。[実行環境設定] ダイアログ・ボックスを表示するには、VuGen ツールバーの **[実行環境の設定]** ボタンをクリックします。

VBA の実行環境を設定するには、次の手順を実行します。

- 1 **[実行環境設定]** ダイアログ・ボックスを開き、**[VBA : VBA]** ノードを選択します。



2 **[VBA 参照]** セクションで、スクリプト実行中に使用する参照ライブラリを選択します。ライブラリを選択すると、そのライブラリの詳細とバージョンがダイアログ・ボックスの下部に表示されます。

3 コンパイラに関する適切なオプションを選択します。

Visual Basic IDE（統合開発環境）を使ってデバッグを行えるようにするには、**[VBA IDE からスクリプトをデバッグする (VuGen のみ)]** を選択します。

スクリプト実行中に Visual Basic IDE を常に表示しておくには、**[エラー時 VBA IDE を常に表示する]** を選択します。

4 **[OK]** をクリックして実行環境の設定を適用します。

第 23 章

ネットワーク実行環境の設定

ネットワークの速度をシミュレートするには、ネットワーク実行環境を設定します。

本章の内容

- ▶ ネットワーク実行環境の設定について (438 ページ)
- ▶ ネットワーク速度の設定 (438 ページ)
- ▶ プロキシ・オプションの設定 (439 ページ)
- ▶ ブラウザのエミュレーション・プロパティの設定 (444 ページ)
- ▶ インターネット・プリファレンスの設定 (449 ページ)
- ▶ Web サイトのフィルタリング (458 ページ)
- ▶ デバッグ情報の取得 (460 ページ)
- ▶ HTML 圧縮の実行 (461 ページ)
- ▶ Web ページのコンテンツ・チェック (462 ページ)

以降の情報は、すべてのインターネット関連プロトコル、Citrix ICA、Oracle NCA、WinSock を対象とします。

すべての Vuser に適用される一般的な実行環境の設定については、第 22 章「実行環境の設定」を参照してください。

ネットワーク実行環境の設定について

Vuser スクリプトを作成した後、実行環境の設定を行います。この設定によって、Vuser が正しく実際のユーザをエミュレートできるようインターネット環境を構成できます。インターネット関連の実行環境設定では、プロキシ、ブラウザ、速度のシミュレーション、その他の設定が行えます。

実行環境設定を行うには、[実行環境設定] ダイアログ・ボックスを開き、適切なノードを選択します。

[実行環境設定] ダイアログ・ボックスを表示するには、次のいずれかの手順を実行します。

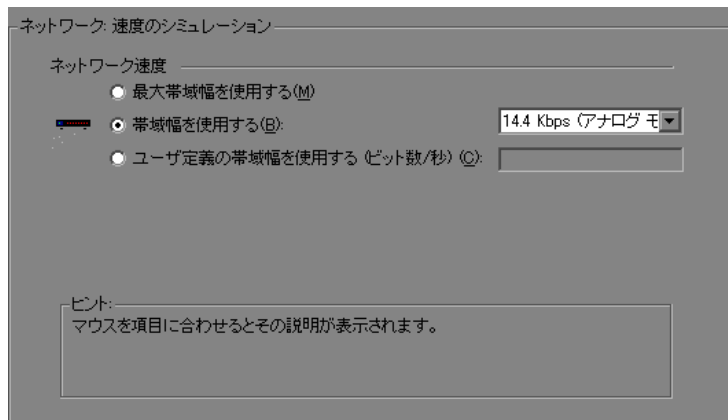


- ▶ VuGen ツールバーの [実行環境の設定] ボタンをクリックします。
- ▶ キーボードのショートカット・キー、**F4** キーを押します。
- ▶ [仮想ユーザ] > [実行環境の設定] を選択します。

LoadRunner Controller からでも実行環境の設定を変更できます。詳細については、各製品のドキュメントを参照してください。

ネットワーク速度の設定

[実行環境設定] ダイアログ・ボックスの [ネットワーク:速度のシミュレーション] ノードで、テスト環境におけるモデムのエミュレーションを設定します。



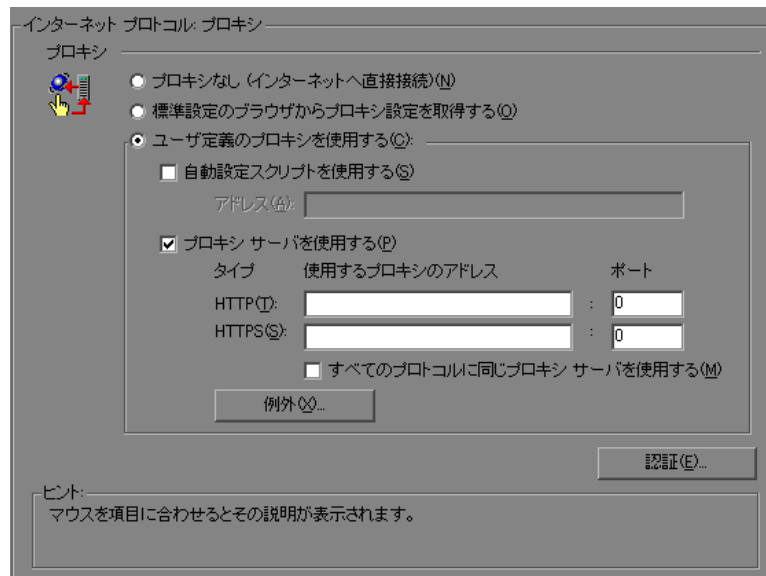
速度のシミュレーション

速度のシミュレーション設定を使って、テスト環境をエミュレートするのに最も近い帯域幅を選択できます。次のオプションがあります。

- ▶ **[最大帯域幅を使用する]**：標準では、帯域幅のエミュレーションは無効になっており、Vuser はネットワーク上で利用できる最大の帯域幅で実行されます。
- ▶ **[帯域幅を使用する]**：Vuser をエミュレートする帯域幅のレベルを指定します。アナログ・モデム、ISDN、DSL をエミュレートするため、14.4Kbps から 512 Kbps の範囲の速度を選択できます。
- ▶ **[ユーザ定義の帯域幅を使用する]**：Vuser をエミュレートする帯域幅の上限を指定します。帯域幅をビットで指定します（1 キロビット = 1024 ビット）。

プロキシ・オプションの設定

[実行環境設定] ツリーの **[インターネット プロトコル: プロキシ]** ノードで、プロキシ関連の設定を行います。



[実行環境設定] には、次のようなプロキシの設定オプションがあります。

- ▶ **[プロキシなし]** : すべての Vuser が、インターネットに常に直接接続します。つまり、プロキシ・サーバを使用せずに接続します。
- ▶ **[標準設定のブラウザからプロキシ設定を取得する]** : すべての Vuser が実行されているマシンから、通常使うブラウザとなっているプロキシの設定を使用する。
- ▶ **[ユーザ定義のプロキシを使用する]** : すべての Vuser が同じユーザ定義のプロキシ・サーバを使用する。実際のプロキシ・サーバの構成情報や、自動設定のための設定スクリプト (**.pac** ファイル) へのパスなどで設定を行います (441 ページ「自動プロキシ設定の設定方法」を参照)。

サーバの構成情報で指定する場合、IP アドレス、名前、ポート番号を使用します。HTTP サイト用にプロキシ・サーバを 1 つ指定し、HTTPS サイト (セキュア・サイト) 用に別のプロキシ・サーバを指定することもできます。

プロキシ情報を指定した後、プロキシ・サーバの認証情報と、プロキシ・ルールに対する例外を指定できます。

注 : 再生中にプロキシの応答を待つように Vuser に指示し、プロキシが基本認証をサポートしていると仮定しないようにするには、次のステートメントを追加します。`web_set_sockets_option("PROXY_INITIAL_BASIC_AUTH", "0");`

認証

プロキシ・サーバで Vuser ごとに認証を必要とする場合は、このダイアログ・ボックスを使用して、適切なパスワードとユーザ名を入力します。

- ▶ **[ユーザ名]** : Vuser がプロキシ・サーバへのアクセスに使用するユーザ名を入力します。
- ▶ **[パスワード]** : Vuser によるプロキシ・サーバへのアクセスに必要なパスワードを入力します。

注 : 認証を記録時に動的に追加する場合や、複数のプロキシ・サーバの認証を追加する場合は、`web_set_user` 関数を使用します。詳細については、『**Online Function Reference**』(英語版) ([ヘルプ] > [関数リファレンス]) を参照してください。

例外

すべての Vuser が指定されたプロキシ・サーバを使用するように指定できます。この場合、Vuser に直接アクセスさせる（つまり、プロキシ・サーバを使用せずにアクセスさせる）URL がある場合は、それらの URL のリストをテキスト・ボックスに入力します。

- ▶ **[次で始まるアドレスにはプロキシサーバを使用しない]**：プロキシ・サーバから除外するアドレスを入力します。各項目はセミコロンで区切ります。
- ▶ **[ローカルアドレス（イントラネット）にはプロキシサーバを使用しない]**：イントラネットからのアドレスなど、ローカル・アドレスをプロキシ・サーバから除外するには、このチェック・ボックスを選択します。

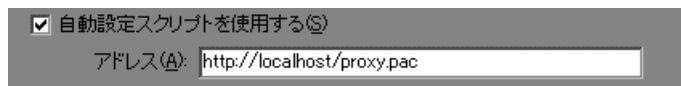
自動プロキシ設定の設定方法

自動プロキシ設定は、ほとんどのブラウザでサポートされています。この機能では、プロキシ割り当て情報を含む JavaScript ファイル（通常は .pac 拡張子付き）を指定できます。このスクリプトは、URL に基づいて、どの場合にはプロキシ・サーバを使用して、どの場合には直接サイトへ接続するのかをブラウザに指定します。また、このスクリプトでは、アドレスごとに異なるプロキシ・サーバを使うよう指定することができます。

設定スクリプトを使用するように VuGen または Internet Explorer ブラウザを設定できます。自動プロキシ設定のファイルを指定すると、Vuser はテストの実行時にそのプロキシ・ファイルから得られたルールを使用します。

VuGen で設定スクリプトを指定するには、次の手順を実行します。

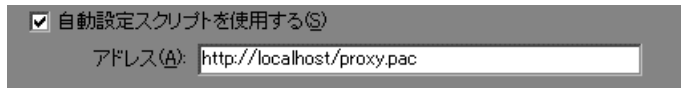
- 1 **[仮想ユーザ]** > **[実行環境設定]** を選択し、**[インターネット プロトコル：プロキシ]** ノードを選択します。
- 2 **[ユーザ定義のプロキシを使用する]** を選択し、**[自動設定スクリプトを使用する]** オプションを選択します。スクリプトの場所を指定します。



Internet Explorer (IE) で設定スクリプトを指定するには、次の手順を実行します。

- 1 **[ツール]** > **[インターネット オプション]** を選択し、**[接続]** タブを選択します。

- 2 **[LAN の設定]** ボタンをクリックします。[LAN 設定] ダイアログ・ボックスが開きます。
- 3 **[自動設定スクリプトを使用する]** オプションを選択し、スクリプトの場所を指定します。



Vuser の振る舞いを追跡するには、テキスト実行中にログを生成し、[実行ログ] タブまたは `mdrv.log` ファイルを調べます。ログには、各 URL に対して使用されたプロキシ・サーバが表示されます。次の例では、URL `australia.com` へ直接接続していますが、URL `http://www.google.com` についてはプロキシ・サーバ `aqua` が使われています。

```
Action1.c(6): t=1141ms: FindProxyForURL returned DIRECT
Action1.c(6): t=1141ms: Resolving australia.com
Action1.c(6):t=1141ms:Connecting to host 199.203.78.255:80
...
Action1.c(6):t=1281ms:Request done "http://australia.com/GetElementByName.htm"
...
Action1.c(6):web_url was successful, 357 body bytes, 226 header bytes
Action1.c(15):web_add_cookie was successful
Action1.c(17): t=1391ms: FindProxyForURL returned PROXY aqua:2080
Action1.c(17):t=1391ms:Auto-proxy configuration selected proxy aqua:2080
Action1.c(17):t=1391ms:Resolving aqua
Action1.c(17):t=1391ms:Connecting to host 199.203.139.139:2080
...
Action1.c(17):t=1578ms:"http://www.google.com/" (RelFrameld=1) へ送出した 168 バイトの要求ヘッダ
Action1.c(17): GET http://www.google.com/ HTTP/1.1¥r¥n
```

プロキシ実行環境の設定

本項では、プロキシ実行環境の設定に必要な手順について説明します。

プロキシを設定するには、次の手順を実行します。

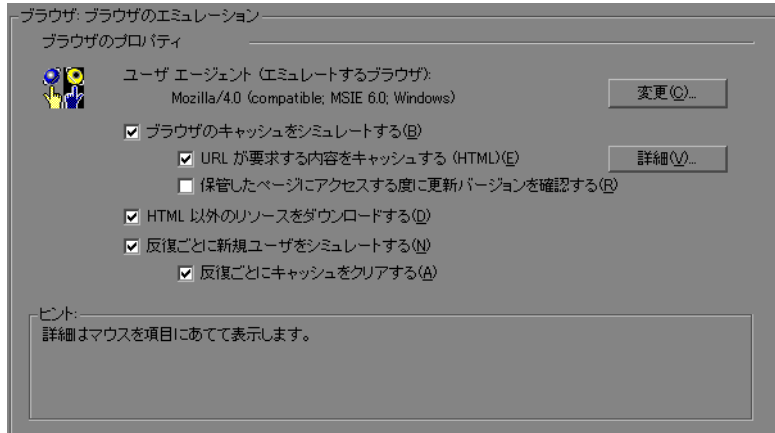


- 1 **[実行環境設定]** ダイアログ・ボックスを開きます。VuGen ツールバーにある **[実行環境の設定]** ボタンをクリックするか、**[仮想ユーザ]** > **[実行環境の設定]** を選択します。

- 2 **[インターネット プロトコル: プロキシ]** ノードをクリックします。
- 3 プロキシ・オプションを選択します。**[プロキシなし (インターネットへ直接接続)]**, **[標準設定のブラウザからプロキシ設定を取得する]**, **[ユーザ定義のプロキシを使用する]** があります。
- 4 ユーザ定義のプロキシを指定した場合は、以下を実行します。
 - ▶ HTTP および HTTPS プロキシ・サーバの IP アドレスを指定します。
 - ▶ **pac** ファイルまたは JavaScript ファイルを使用してプロキシを指定するには、**[自動設定スクリプトを使用する]** オプションを選択してスクリプトの場所を指定します。http:// で始まる Web の場所 (たとえば, **http://hostname/proxy.pac**) またはファイル・サーバ上の場所 (たとえば, **C:¥temp¥proxy.pac**) を指定できます。
- 5 Vuser がプロキシ・サーバを使わずに直接アクセスする URL を指定するには、**[例外]** をクリックし、それらの URL をリストに追加します。**[例外]** ダイアログ・ボックスでは、ローカル (イントラネット) アドレスへの直接アクセスも指定できます。
- 6 プロキシ・サーバが各 Vuser の認証を必要とする場合は、**[認証]** をクリックして、適切なパスワードとユーザ名を入力します。
- 7 セキュリティが保護されたサイト用に専用サーバを指定するのではなく、すべてのインターネット・プロトコル (HTTP, HTTPS) 用に同じプロキシ・サーバを使用するよう Vuser に指示するには、**[すべてのプロトコルに同じプロキシサーバを使用する]** チェック・ボックスを選択します。

ブラウザのエミュレーション・プロパティの設定

[実行環境設定] ダイアログ・ボックスの [ブラウザ: ブラウザのエミュレーション] ノードでは、テスト環境におけるブラウザのプロパティについて設定します。



ブラウザのプロパティ

次の項目についてブラウザのプロパティの設定が可能です。

- ▶ エミュレートするユーザ・エージェント・ブラウザ
- ▶ ブラウザのキャッシュをシミュレートする
- ▶ HTML 以外のリソースをダウンロードする
- ▶ 反復ごとに新規ユーザをシミュレートする

キャッシュと新しいリソースのチェックに関する詳細オプションを設定することもできます。

エミュレートするユーザ・エージェント・ブラウザ

Vuser が Web サーバに要求を送信するときは、要求に HTTP ヘッダが含まれています。テキストの 1 行目には、メソッド (通常、「GET」または「POST」)、リソース名 (たとえば「pclt/default.htm」)、プロトコルのバージョン (たとえば「HTTP/1.0」) が含まれています。2 行目以降には、「ヘッダー情報」が「属性名, コロン, 値」の形式で含まれています。要求は、空白行で終わります。

Vuser のヘッダには、エミュレートされているブラウザの種類を示す「**User-Agent**」ヘッダが含まれています。次に例を示します。

User-Agent:Mozilla/3.01Gold (WinNT; I)

前述の例は、ブラウザは Intel 社の CPU を搭載したマシン上の Windows NT で動作する Mozilla/3.01Gold がエミュレーション対象のブラウザであることを示しています。

ブラウザのエミュレーション・ノードから **[変更]** をクリックし、ヘッダに含めるブラウザ情報を指定します。Web Vuser に対して、いくつかの標準的なブラウザをエミュレートするように指定できます。または、ブラウザ以外の HTTP アプリケーションの場合は、そのアプリケーションと組み合わせる HTTP クライアントを指定できます。この場合は、以降の HTTP ヘッダに含める「**ユーザ定義のブラウザを使用する**」を示す文字列を指定しなければなりません。標準設定では、ユーザ・エージェントは Microsoft Internet Explorer 5.5 ブラウザ・エージェントをエミュレートします。

ブラウザのキャッシュをシミュレートする

このオプションを選択すると、Vuser はキャッシュを利用しながらブラウザをシミュレートします。キャッシュは頻繁にアクセスするドキュメントのローカル・コピーを保存するのに使われます。キャッシュを使うことにより、Vuser のネットワーク接続時間を削減します。標準設定では、キャッシュ・シミュレーションは有効になっています。キャッシュが無効になっている場合、Vuser はすべてのキャッシュ機能を見捨て、要求ごとにすべてのリソースをダウンロードします。

キャッシュ・シミュレーションが無効になっているとき、リソースがページ上に複数回表示されていても、各リソースはページごとに一度だけダウンロードされます。リソースは画像、フレーム、または別の種類のスクリプト・ファイルです。

LoadRunner や Performance Center など複数の Vuser を実行しているときは、各 Vuser はそれぞれのキャッシュを使用し、キャッシュから画像を取得します。このオプションが無効にすると、すべての Vuser で、キャッシュが利用されずにブラウザがエミュレートされます。

次のように実行環境の設定を変更して、使用しているブラウザの設定を Internet Explorer に合わせるすることができます。

| ブラウザの設定 | 実行環境の設定 |
|----------------------------------|---|
| [ページを表示するごとに確認する] | [ブラウザのキャッシュをシミュレートする] を選択して、[保管したページにアクセスする度に更新バージョンを確認する] を有効にします。 |
| [Internet Explorer を起動するごとに確認する] | [ブラウザのキャッシュをシミュレートする] だけを選択します。 |
| [自動的に確認する] | [ブラウザのキャッシュをシミュレートする] だけを選択します。 |
| [確認しない] | [ブラウザのキャッシュをシミュレートする] を選択して、[保管したページにアクセスする度に更新バージョンを確認する] を無効にします。 |

VuGen では、次の 2 つのブラウザ・キャッシュ・オプションを設定できます。

- ▶ **[URL が要求する内容をキャッシュする (HTML)]**：このオプションを有効にすると、VuGen は HTML 内容を要求する URL のみをキャッシュに格納します。内容は、解析、検証、または相関に必要となる場合があります。このオプションを選択すると、HTML 内容は自動的にキャッシュに保存されます。標準設定では、このオプションは有効になっています。

ヒント：仮想ユーザのメモリ使用量を減らすには、テストの明示的な要件でなければ、このオプションを無効にします。

キャッシュする内容の種類のリストに内容の種類をさらに追加するには、**[詳細設定]** をクリックします。詳細については、448 ページ「URL が要求する内容をキャッシュする際の詳細設定」を参照してください。親オプションの **[ブラウザのキャッシュをシミュレートする]** を有効にした場合は、このオプションを無効にしても、VuGen によって画像ファイルが格納されます。

- ▶ **[保管したページにアクセスする度に更新バージョンを確認する]**：指定した URL について、キャッシュに保存されているものより新しいバージョンがあるかどうかをブラウザがチェックするようにします。このオプションを有効にすると、VuGen によって「If-Modified-Since」属性が HTTP ヘッダに追加されます。このオプションを有効にすると、ページの最新版が表示されるようになりますが、シナリオまたはセッションの実行時に発生するトラフィックが増えます。標準設定では、このオプションは無効となっており、ブラウザは新しいリソースの有無をチェックしません。このオプションは、エミュレートするブラウザの設定と一致するよう設定します。

HTML 以外のリソースをダウンロードする

Vuser は、再生中に Web ページにアクセスするときに画像ファイルをロードします。これには、ページとともに記録された画像イメージと、明示的にはページとともに記録されていない画像イメージの両方が含まれます。実際のユーザは、Web ページにアクセスするとき、画像がロードされるのを待ちます。したがって、エンド・ユーザ時間を含め、システム全体をテストする場合には、このオプションを有効にします（標準設定では有効）。パフォーマンスを向上させるために、実際のユーザをエミュレートしない場合は、このオプションを無効にします。

ヒント：Web ページにアクセスするたびに画像が変わり（広告業者のバナーなど）、画像チェックで不一致が生じる場合には、このオプションを無効にします。

反復ごとに新規ユーザをシミュレートする

このオプションを選択すると、VuGen は、反復を行う前にすべての HTTP コンテキストを **init** セクションの終了時の状態にリセットします。この設定により、Vuser は新規ユーザがブラウザ・セッションを開始する様子をより正確にエミュレートできます。クッキーをすべて削除するには、すべての TCP 接続を閉じ（キープ・アライブを含む）、エミュレートされたブラウザのキャッシュをクリアします。次に HTML フレーム階層をリセットして（フレームは番号 1 から番号付けされる）ユーザ名およびパスワードをクリアします。標準設定では、このオプションは有効になっています。

- ▶ **[反復ごとにキャッシュをクリアする]** : Web ページに初めてアクセスしたユーザをシミュレートするために、反復ごとにブラウザのキャッシュをクリアします。Vuser がブラウザのキャッシュに格納された情報を使用して、最近ページにアクセスしたユーザをシミュレートできるようにする場合は、チェック・ボックスをクリアしてこのオプションを無効にします。

URL が要求する内容をキャッシュする際の詳細設定

[詳細設定] ダイアログ・ボックスでは、キャッシュに格納する URL 内容タイプを指定できます。このダイアログ・ボックスには [実行環境設定] ダイアログ・ボックスの **[ブラウザ: ブラウザのエミュレーション]** ノードからアクセスできます。

複数のグループに対して詳細設定を同時に変更することはできません。グループごとに個別に設定を編集してください。

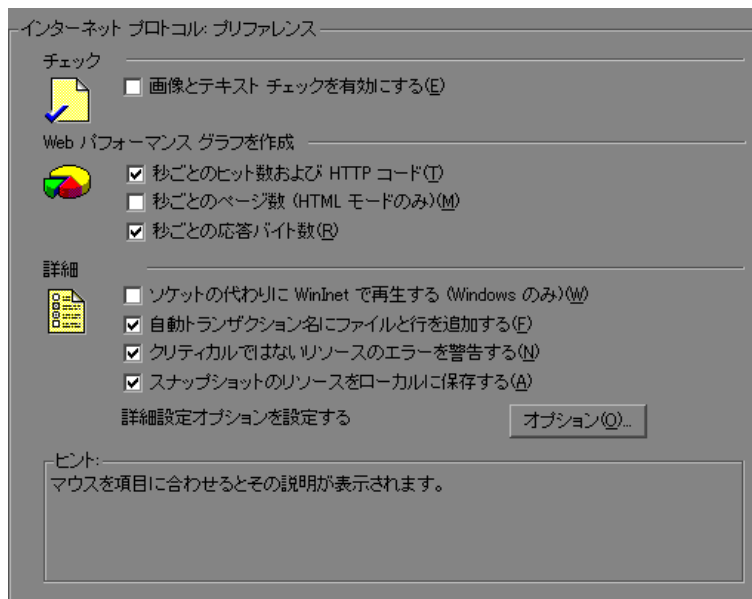
内容タイプを追加するには、次の手順を実行します。

- 1 **[HTML ページ以外にも内容が必要な URL を指定する]** オプションを有効にします。
- 2 プラス記号をクリックし、`text/plain`、`text/xml`、`image/jpeg`、`image/gif` などの追加内容タイプを選択します。内容名をテキスト・ボックスに入力します。
- 3 内容タイプをリストから削除するには、その内容タイプを選択してマイナス記号をクリックします。

インターネット・プリファレンスの設定

[実行環境設定] ダイアログ・ボックスの [インターネットプロトコル: プリファレンス] ノードを使用して、次の項目に関連する設定を行います。

- ▶ 画像とテキストのチェック
- ▶ Web パフォーマンス・グラフの作成
- ▶ Web 実行環境の詳細オプション



画像とテキストのチェック

[**画像とテキスト チェックを有効にする**] オプションを選択すると、Vuser は再生中に **web_find** または **web_image_check** 検証関数を実行して、画像とテキストの検証チェックを行うことができます。このオプションは、HTTP モードで記録されたステートメントにのみ適用されます。Vuser で検証チェックを実行すると、検証チェックを実行しない Vuser より多くのメモリが消費されます (標準設定では無効)。

Web パフォーマンス・グラフの作成

このオプションを選択すると、Vuser によって Web パフォーマンス・グラフの作成に使用するデータが収集されます。テストの実行中はオンライン・モニタ、テストの実行後は Analysis を使用して、[秒ごとのヒット数および HTTP コード]、[秒ごとのヒット数 (HTML モードのみ)]、[秒ごとの応答バイト数] のグラフを表示できます。Analysis を使用して、テスト実行後にコンポーネント・ブレイクダウン・グラフを表示できます。Vuser に収集させるグラフ・データのタイプを選択します。

注： Web パフォーマンス・グラフを使用しない場合は、メモリを節約するためにこれらのオプションを無効にしておきます。

Web 実行環境の詳細オプション

- ▶ **[ソケットの代わりに WinInet で再生する]**：このオプションを選択すると、VuGen は標準のソケット再生の代わりに WinInet 再生エンジンを使用します。VuGen には、Sockets ベース（標準設定）および WinInet ベースの 2 つの HTTP 再生エンジンがあります。WinInet は Internet Explorer で使用されるエンジンであり、IE ブラウザに組み込まれている機能をすべてサポートしています。WinInet 再生エンジンの制約として、スケーラブルでないこと、UNIX をサポートしていないことが挙げられます。さらに、スレッドでの作業時に WinInet エンジンがモデム速度および接続数を正確にエミュレートしないことがあります。

VuGen 独自のソケット・ベースの再生は、負荷テストにおけるスケーラビリティに優れた小型軽量のエンジンです。また、スレッドでの使用時にも正確に機能します。ソケット・ベースのエンジンの制限事項としては、SOCKS プロキシがサポートされていない点があります。このタイプの環境を記録する場合は、WinInet 再生エンジンを使用してください。

- ▶ **[自動トランザクション名にファイルと行を追加する]**：トランザクション名にファイル名と行番号を追加し、自動トランザクション時に一意となるトランザクション名を作成します（標準設定では有効）。

注： このオプションを選択すると、ログ・ファイルに記録される情報が増えるため、より多くのメモリが必要になります。

- ▶ **[クリティカルではないリソースのエラーを警告する]** : ダウンロードに失敗した画像や Java アプレットなど、負荷テストにとってさほど重要ではない項目で、失敗した関数についての警告ステータスを返します。標準設定では、このオプションは有効になっています。特定の警告をエラーとみなしてテストを失敗させる場合は、このオプションを無効にできます。特定の内容タイプを重要なものとして設定するには、非リソースのリストにその内容タイプを加えます。詳細については、337 ページ「リソース以外のコンテンツ・タイプの指定」を参照してください。
- ▶ **[スナップショットのリソースをローカルに保存する]** : VuGen はスナップショット・リソースをローカル・マシン上にファイルとして保存します。この機能を使用することで、実行時ビューアはスナップショットをより正確に作成し、よりすばやく表示できます。

インターネット・プリファレンスのその他のオプション

[プリファレンス] ノードの [詳細] セクションの [オプション] ボタンをクリックすると、各種の詳細オプションを設定できます。設定できるオプションは、[DNS キャッシング]、[HTTP バージョン]、[Keep-Alive の HTTP 接続]、[サーバサイド圧縮の受け付け]、[Accept-Language ヘッダー]、[HTTP 要求の接続タイムアウト]、[HTTP 要求の受信タイムアウト]、[ネットワークのバッファサイズ]、および [ステップ ダウンロード時間のタイムアウト] です。

HTTP

- ▶ **[HTTP のバージョン]** : HTML バージョンとしてバージョン 1.0 または 1.1 のどちらを使用するかを指定します。バージョン情報は、Vuser が Web サーバに要求を送信するときの HTTP 要求ヘッダに含まれています。標準設定は 1.1 です。HTTP 1.1 では次の機能がサポートされています。
 - ▶ 持続的な接続。下の「HTTP 接続の Keep-Alive」を参照してください。
 - ▶ HTML 圧縮。461 ページ「HTML 圧縮の実行」を参照してください。
 - ▶ 仮想ホスティング。複数のドメイン名が同一の IP アドレスを共有します。
- ▶ **[HTTP 接続の Keep-Alive]** : Keep-Alive (キープ・アライブ) は、持続的な接続を可能にする HTTP 拡張機能を表す用語です。こうした長時間の HTTP セッションでは、複数のリクエストを同一の TCP 接続を介して送信できます。これにより、Web サーバとクライアントのパフォーマンスが向上します。

この Keep-Alive オプションは、Keep-Alive 接続をサポートする Web サーバがあって初めて機能します。この設定により、Vuser スクリプトを実行するすべての Vuser に対して Keep-Alive HTTP 接続を有効にすることができます（標準では有効）。

- ▶ **[Accept-Language 要求ヘッダ]**：受け入れた言語のカンマ区切りリストを提供します。たとえば、**en-us**、**fr** などがあります。
- ▶ **[HTTP エラーを警告とする]**：HTTP エラーによりリソースのダウンロードが失敗した場合に、エラーの代わりに警告を発行します。
- ▶ **[HTTP 要求接続タイムアウト (秒)]**：Vuser が、ステップ内の特定の HTTP 要求への接続を中断するまで待機する時間（秒単位）です。タイムアウトは、要求に対してサーバが安定しユーザに応答するための猶予を与えるものです（標準設定の値は 120 秒です）。このタイムアウトは、Vuser が **wap_connect** 関数によって開始された WAP 接続を待機する時間にも適用されます。
- ▶ **[HTTP 要求受信タイムアウト (秒)]**：Vuser がステップ内の特定の HTTP 要求への応答を受信するまで待機する時間（秒単位）です。タイムアウトは、要求に対してサーバが安定しユーザに応答するための猶予を与えるものです（標準設定の値は 120 秒です）。
- ▶ **[zlib ヘッダを要求]**：リクエスト・データを **zlib** 圧縮ライブラリ・ヘッダとともにサーバに送信します。標準では、サーバに送信されるリクエストには **zlib** ヘッダが含まれます。このオプションを使用すると、**zlib** ヘッダがリクエストに含まれないブラウザ以外のアプリケーションをエミュレートできます。これらのヘッダを除外するには、このオプションを「**No**」に設定します（標準設定は「**Yes**」です）。
- ▶ **[サーバサイド圧縮を受け入れる]**：再生で圧縮データを受け入れることができることをサーバに示します。使用可能なオプションは、「**None**（圧縮なし）」、「**gzip**（gzip 圧縮を受け入れる）」、「**gzip, deflate**（gzip または deflate 圧縮を受け入れる）」、および「**deflate**（deflate 圧縮を受け入れる）」です。圧縮データを受け入れると、CPU の処理量が大幅に増えることがあります。標準設定は、「**gzip, deflate**（gzip または deflate 圧縮を受け入れる）」です。

一般

- ▶ **[DNS のキャッシュ]**：このオプションを選択すると、Vuser はドメイン・ネーム・サーバによって解決されたホストの IP アドレスをキャッシュに保存します。これにより、それ以降の同じサーバに対する呼び出しに費やす時間を節約できます。ロード・バランサーなどの技術によって自動的に IP アドレスが変更される場合には、このオプションを確実に無効にしておき、Vuser がキャッシュの値を使用しないようにします（標準では有効）。
- ▶ **[UTF-8 から、または UTF-8 への変換]**：受信した HTML ページおよび送信されたデータの UTF-8 への変換と UTF-8 からの変換をします。記録オプションで UTF-8 のサポートを有効にします。詳細については、331 ページ「Web, ワイヤレス, Oracle NCA 記録オプション」を参照してください（標準設定では無効）。
- ▶ **[リソースによって発生したステップ タイムアウトを警告とする]**：タイムアウトの時間内にロードされなかったリソースが原因でタイムアウトが発生した場合に、エラーではなく警告を発行します。リソース以外の場合は、エラーが発行されません（標準では無効）。
- ▶ **[HTML の Content-Type を解析する]**：要求した HTML について、次の特定の内容タイプが指定されていれば、その応答を解析します。**HTML**, **text/html**, **TEXT** (任意のテキスト), **ANY** (任意の内容タイプ)。text/xml は HTML としては解析されません。標準値は **[TEXT]** です。

タイムアウト設定は主に、該当システム環境下では、許容タイムアウト値を変えるのが望ましいと判断できる上級ユーザ向けです。ほとんどの場合、標準の値で問題ないはずですが、サーバが妥当な時間内に応答しない場合は、タイムアウト時間を長くするのではなく、接続に関するほかの問題がないか調べてください。タイムアウト時間を長くすると、スクリプトが不必要に待機することになります。

- ▶ **[ステップ ダウンロード タイムアウト (秒)]**：Vuser がスクリプトのステップの実行を中止するまでに待機する時間です。このオプションはページのダウンロードに特定秒数以上は待たないユーザの操作をエミュレートするのに使用します。

- ▶ **[ネットワーク バッファ サイズ]** : HTTP 応答の受信に使用するバッファ・サイズの最大値を設定します。データのサイズが指定サイズより大きいと、サーバはデータをチャンクに分けてで送信するため、システムのオーバーヘッドが増加します。Controller で複数の Vuser を実行すると、各 Vuser は自身のネットワーク・バッファを使用します。この設定は、主にネットワーク・バッファ・サイズがスクリプトのパフォーマンスに影響を与える可能性があるとして判断した上級ユーザを対象にしています。標準設定は 12 キロバイトです。最大サイズは 0x7FFF FFFF です。
- ▶ **[NTLM 情報を出力します]** : NTLM ハンドシェイクに関する情報を標準ログに出力します。
- ▶ **[SSL 情報を出力します]** : SSL ハンドシェイクに関する情報を標準ログに出力します。
- ▶ **[エラーとして発行された一致エラー数の上限]** : LB (左境界) または RB (右境界) を使用して、コンテンツ・チェックにエラーとして発行されたエラーの一致数を制限します。これは、文字列が見つかったときに失敗となる (Fail=Found) 一致に適用されます。以降の一致は、情報メッセージとして一覧表示されます。標準設定値は 10 です。
- ▶ **[同じページに対する 'META refresh' の上限]** : META 更新がページごとに実行される最大回数。標準設定値は 2 です。
- ▶ **[UTF-8 による ContentCheck の値]** : ContentCheck XML ファイルの値を UTF-8 で格納します。

認証

- ▶ **[認証再思考時の固定思考遅延時間 (ミリ秒)]** : ユーザによる認証情報 (ユーザ名とパスワード) の入力をエミュレートするため、思考遅延時間を Vuser スクリプトに自動的に追加します。この思考遅延時間はトランザクションに含まれません (標準設定値は 0 です)。
- ▶ **[NTLM2 セッションセキュリティを無効にする]** : より基本的な NTLM 2 セッション・セキュリティ応答ではなく、完全な NTLM 2 ハンドシェイク・セキュリティを使用します (標準設定は「No」です)。
- ▶ **[Windows のネイティブ NTLM 実装を使用する]** : 固有 API の代わりに NTLM 認証のための Microsoft セキュリティ API を使用します (標準設定は「No」です)。

- ▶ **[統合認証を有効にする]** : Kerberos ベースの認証を有効にします。サーバによって認証スキームが提示されている場合は、別のスキームに優先して **[Negotiate]** を使用しません (標準設定は「No」です)。
- ▶ **[大きい KDC 負荷をかける]** : 前の反復で取得した資格情報を再利用しません。この設定を有効にすると、KDC (キー配布サーバ) にかかる負荷が高くなります。サーバにかかる負荷を減らすには、このオプションを「**Yes**」に設定して、前の反復で取得した資格情報を再利用します。このオプションは Kerberos 認証が使用されている場合にのみ関係します (標準設定は「No」です)。

ログ

- ▶ **[出力バッファの行の長さ]** : 行の折り返しを無効にして、要求、応答ヘッダ、ボディ、または JavaScript ソース、あるいはこれらすべてを印刷するための行の長さです。
- ▶ **[出力バッファはバイナリ ゼロのみエスケープする]** :
 - ▶ **[Yes]** : 要求、応答ヘッダ、ボディ、または JavaScript ソース、あるいはこれらすべてを印刷する場合に、バイナリのゼロ値のみをエスケープします。
 - ▶ **[No]** : 印刷不能な文字や制御文字をすべてエスケープします。

Web (Click and Script) 固有

- ▶ **[一般]** :
 - ▶ **[ホームページの URL]** : ブラウザを開いたときに表示されるホーム・ページの URL (標準設定値は about:blank)。
 - ▶ **[DOM ベースのスナップショット]** : サーバ応答の代わりに DOM からスナップショットを生成するよう VuGen に指示します (標準設定値は「Yes」)。
 - ▶ **[HTTP による文字セットの変換]** : "Content-Type:.....; charset=..." HTTP 応答ヘッダに基づいて文字セットの変換を実行します。[UTF-8 から、または UTF-8 への変換] に優先します。
 - ▶ **[META による文字セットの変更があったら解析しなおす]** : META タグによる文字セットの変更があったら解析しなおします。[HTTP による文字セットの変換] が有効な場合にのみ設定できます。[自動] は、最初の反復で使用したときにのみ解析が有効になることを意味します。

- ▶ **[JavaScript エラー時に失敗する]** : JavaScript の評価エラーが発生した場合に、Vuser が失敗します。標準設定値は「No」で、JavaScript エラーの後に警告メッセージだけが表示され、スクリプトの実行が継続されます。
- ▶ **[新規ウィンドウオブジェクトごとに標準クラスを初期化する]** : 有効にすると、スクリプト (SRC コンパイル済みスクリプト) はキャッシュされません。
- ▶ **[無効になっている対象要素を無視する]** : 無効になっている Vuser スクリプト関数による対象要素を無視します。
- ▶ **[タイマ]** :
 - ▶ **[ステップ終了時にタイマを最適化する]** : 可能であれば、期限前に、ステップの終了時に期限切れになる `setTimeout/setInterval/ < META 更新 >` を実行します (標準設定は「Yes」です)。
 - ▶ **[単独 `setTimeout/setInterval` しきい値 (秒)]** : `window.setTimeout` および `window.setInterval` メソッドの上限タイムアウトを指定します。遅延時間がこのタイムアウトを超えた場合、これらのメソッドは渡された関数を呼び出さなくなります。これにより、ユーザが指定した時間、次の要素をクリックするまでに待機する操作がエミュレートされます (標準設定値は 5 秒です)。
 - ▶ **[累積 `setTimeout/setInterval` しきい値 (秒)]** : `window.setTimeout` および `window.setInterval` メソッドのタイムアウトを指定します。遅延時間がこのタイムアウトを超えた場合、それ以上の `window.setTimeout` および `window.setInterval` 呼び出しは無視されます。タイムアウトはステップごとに累積されます (標準設定値は 30 秒です)。
 - ▶ **[ステップ終了時に `setInterval` を設定しなす]** : **0** なら「No」、**1** なら「1 回」、**2** なら「Yes」です。
- ▶ **[履歴]** :
 - ▶ **[履歴のサポート]** : テスト実行での `window.history` オブジェクトのサポートを有効にします。[有効]、[無効]、[自動] のいずれかを選択できます。[自動] (標準設定) を選択すると、Vuser は最初の反復で使用されている場合にのみ `window.history` オブジェクトをサポートします。このオプションを無効にすると、パフォーマンスが向上します。
 - ▶ **[履歴の最大サイズ]** : 履歴リストに保存するステップの最大数 (標準設定値は 100 ステップ)。

▶ [ナビゲータのプロパティ] :

- ▶ [navigator.browserLanguage] : ナビゲータ DOM オブジェクトの **browserLanguage** プロパティに設定されているブラウザ言語です。標準設定は記録された値です。標準設定では、古い記録エンジンで作成されたスクリプトは、**en-us** を使用します。
- ▶ [navigator.systemLanguage] : ナビゲータ DOM オブジェクトの **systemLanguage** プロパティに設定されているシステム言語です。標準設定は記録された値です。標準設定では、古い記録エンジンで作成されたスクリプトは、**en-us** を使用します。
- ▶ [navigator.userLanguage] : ナビゲータ DOM オブジェクトの **userLanguage** プロパティに設定されているユーザ言語です。標準設定は記録された値です。標準設定では、古い記録エンジンで作成されたスクリプトは、**en-us** を使用します。

▶ [画面のプロパティ] :

- ▶ [screen.width] : screen DOM オブジェクトの **width** プロパティをピクセル単位で設定します (標準設定値は 1024 ピクセルです)。
- ▶ [screen.height] : screen DOM オブジェクトの **height** プロパティをピクセル単位で設定します (標準設定値は 768 ピクセルです)。
- ▶ [screen.availWidth] : screen DOM オブジェクトの **availWidth** プロパティをピクセル単位で設定します (標準設定値は 1024 ピクセルです)。
- ▶ [screen.availHeight] : screen DOM オブジェクトの **availHeight** プロパティをピクセル単位で設定します (標準設定値は 768 ピクセルです)。

▶ [メモリ管理] :

- ▶ [DOM メモリ割り当ての標準ブロック サイズ] : DOM のメモリ割り当てに対して標準のブロック・サイズを設定します。この値が小さすぎると、余分な `malloc` 呼び出しが増え、実行時間が遅くなる場合があります。ブロック・サイズが大きすぎると、不必要に大きな領域が占有される場合があります (標準設定値は 16384 バイトです)。
- ▶ [動的に作成される DOM オブジェクトためのメモリ マネージャ] : [Yes] — 動的に作成される DOM オブジェクトためのメモリ・マネージャを使用します。[No] — メモリ・マネージャは使用しません (例 : SAP のように同じドキュメントで複数の DOM オブジェクトが動的に作成される場合)。[自動] — プロトコルが推奨する標準設定値を使用します (標準設定では、SAP を除くすべてのプロトコルが「はい」)。

- ▶ **[JavaScript ランタイム メモリのサイズ (KB)]** : JavaScript 実行時メモリのサイズをキロバイト単位で指定します (標準設定値は 256 KB) です。
- ▶ **[JavaScript のスタック メモリ サイズ (KB)]** : JavaScript スタック・メモリのサイズをキロバイト単位で指定します (標準設定値は 32 KB) です。

Web サイトのフィルタリング

再生中における Vuser のダウンロード元となる Web サイトを指定できます。除外するサイトか、追加するサイトのどちらかを指定できます。許可または禁止するリソースを、URL、ホスト名、またはホストサフィックスを指定して制御します。

「**URL**」は、Web サイトの完全な URL アドレスで、`http://` または `https://` で始まります。「**ホスト**」は、ドメインを持つホスト・マシンの名前 (`www.hp.com` など) です。

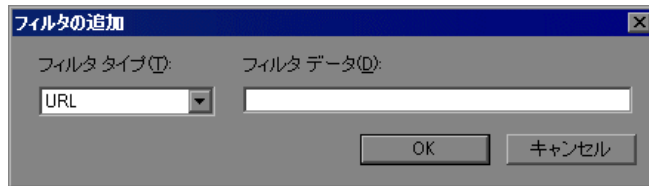
「**ホストのサフィックス**」は、複数のホスト名に共通のサフィックスです (例 : `hp.com`) これは、共通のドメインに複数の Web サイトがある場合に便利です。

除外するサイトを指定した場合は、リストで指定したサイトを除くすべての Web サイトからリソースをダウンロードします。追加するサイトを指定した場合は、[追加] リスト内のサイト以外のすべての Web サイトについて、それらからのリソースを除外します。



フィルタする Web サイトのリストを作成するには、次の手順を実行します。

- 1 **インターネット プロトコル: フィルタのダウンロード** ノードをクリックします。
- 2 オプションとして、**リストされたアドレスのみを含める** または **リストされたアドレスを除外する** を選択します。
- 3 エントリをリストに追加します。エントリを追加するには、**追加** をクリックします。**フィルタの追加** ダイアログ・ボックスが開きます。



フィルタのタイプとして、**URL**、**ホスト**、または **ホストのサフィックス** を選択し、URL などのフィルタ・データを入力します。URL を入力するときは、必ず **http://** または **https://** で始まる完全な URL を入力します。**OK** をクリックします。

- 4 エントリを編集するには、エントリを選択して **編集** をクリックします。
- 5 エントリを削除するには、エントリを選択して **削除** をクリックします。すべてのエントリを削除するには、**全て削除** をクリックします。

デバッグ情報の取得

Vuser スクリプトを実行すると、実行情報が [出力] ウィンドウまたはログ・ファイルに表示されます。[実行環境の設定] の [ログ] ノードを使って、[出力] ウィンドウとログ・ファイルに送られる情報量を制御できます。詳細については、422 ページ「実行環境設定のログの設定」を参照してください。

デバッグ情報には、次のものがあります。

- ▶ ログ情報
- ▶ トランザクションの失敗
- ▶ ゲートウェイとの接続ステータス（接続中、切断中、リダイレクト中のいずれか）（WAP のみ）

より詳細なデバッグ情報を得るには、**default.cfg** ファイルを編集します。
[Web] セクションで、**LogFileWrite** フラグを **1** に設定します。生成されるトレース・ファイルには、スクリプト実行時のすべてのイベントが記録されます。

負荷テストを実施する場合は、必ず **LogFileWrite** フラグをクリアして、Vuser が大きなトレース・ファイルを作成してリソースを浪費しないようにしてください。

HTML 圧縮の実行

HTTP 1.1 をサポートするブラウザは、圧縮されている HTML ファイルを展開できます。サーバは送信するファイルを圧縮して、データ転送に必要な帯域幅を節約します。圧縮は、自動的に、または手作業で有効にできます。

VuGen で自動的に圧縮を有効にするには、[実行環境設定] の [インターネット プロトコル: プリファレンス] ノードを使用します。[オプション] をクリックして [詳細オプション] を開き、[サーバサイド圧縮を受け入れる] オプションを有効にします。標準設定では、このオプションは有効になっています。詳細については、451 ページ「インターネット・プリファレンスのその他のオプション」を参照してください。

圧縮を手作業で追加するには、スクリプトの先頭に次の関数を入力します。

```
web_add_auto_header("Accept-Encoding", "gzip");
```

サーバによって圧縮データが送信されていることを検証するには、**Content-Encoding:gzip** という文字列が実行ログのサーバの応答セクションにあることを確認します。ログには展開前と後のデータ・サイズも表示されます。

圧縮は、大規模なデータ転送には大きな効果があります。つまり、データが大きければ大きいほど、圧縮が効果的になります。より大きなデータで作業するには、ネットワークのバッファ・サイズを増やして、データを 1 つのチャンクとして受け取るようにします（「ネットワーク・バッファ・サイズ」を参照）。

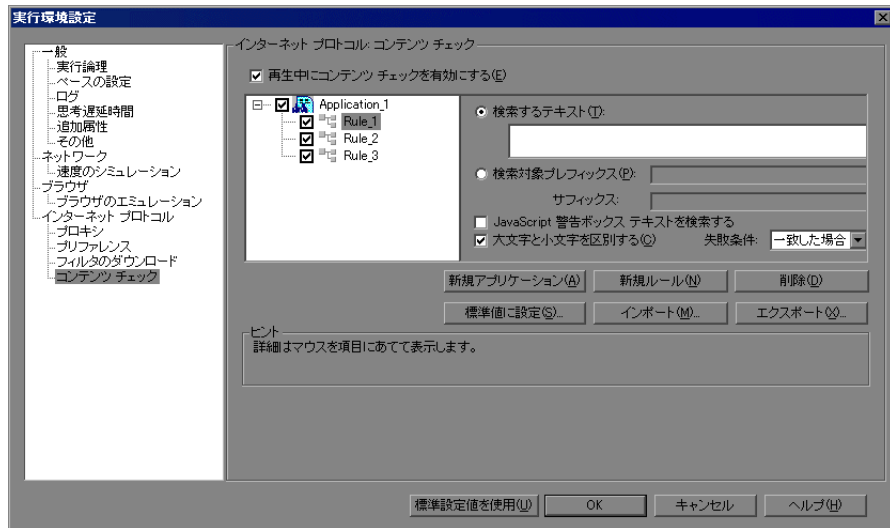
Web ページのコンテンツ・チェック

VuGen のコンテンツ・チェックのメカニズムを使って、ページ内容の特定の文字列を検査することができます。このオプションは、標準的ではないエラーを検出するのに役立ちます。通常の運用では、アプリケーション・サーバで障害が発生するとブラウザが汎用の HTTP エラー・ページを表示して、エラーの性質を通知します。この標準のエラー・ページは VuGen に認識され、エラーとして処理されるので、スクリプトは失敗として報告されます。しかし、アプリケーション・サーバの中には、VuGen がエラー・ページとして認識しない固有のエラー・ページを発行するものもあります。エラー・ページはサーバによって送信され、エラーが発生したことを示す特定の形式のテキスト文字列を含んでいます。

たとえば、エラーが発生したときに、「**ASP Error**」というテキストを含んでいるユーザ定義ページを発行するアプリケーションがあるとします。その場合には、VuGen に対して、サーバから返されたすべてのページでこのテキストを検索するように指定します。VuGen はこの文字列を検出すると、再生を失敗とします。なお、VuGen はページ本体は検索しますが、ヘッダは検索しません。

[実行環境設定] の [**インターネット プロトコル: コンテンツ チェック**] で、検索する文字列を指定します。複数のコンテンツ・チェック・ルールを使って、複数のアプリケーションの内容を定義できます。以降では、次の項目について説明します。

- ▶ コンテンツ・チェック・ルールについて
- ▶ コンテンツ・チェック・ルールの定義



コンテンツ・チェック・ルールについて

VuGen のコンテンツ・チェックのメカニズムを使って、ページ・コンテンツの特定の文字列を検査することができます。このオプションは、標準的ではないエラーを検出するのに役立ちます。通常の運用では、アプリケーション・サーバで障害が発生するとブラウザが汎用の HTTP エラー・ページを表示して、エラーの性質を通知します。この標準のエラー・ページは VuGen に認識され、エラーとして処理されるので、スクリプトは失敗として報告されます。しかし、アプリケーション・サーバの中には、VuGen がエラー・ページとして認識しない固有のエラー・ページを発行するものもあります。エラー・ページはサーバによって送信され、エラーが発生したことを示す特定の形式のテキスト文字列を含んでいます。

たとえば、エラーが発生したときに、「ASP Error」というテキストを含んでいるユーザ定義ページを発行するアプリケーションがあるとします。その場合には、VuGen に対して、サーバから返されたすべてのページでこのテキストを検索するように指定します。VuGen はこの文字列を検出すると、再生を失敗とします。なお、VuGen はページ本体は検索しますが、ヘッダは検索しません。

- ▶ **[再生中にコンテンツ チェックを有効にする]**：再生中のコンテンツ・チェックを有効にします（標準で有効です）。なお、アプリケーションに対するルールを定義した後でも、このオプションを無効にすることで、テスト実行ごとにルールを無効にできます。

ルール情報

右の表示枠には、検索するテキストの検索条件が表示されます。検索するテキストそのもの、または対象テキストの前後にあるテキストを表すプレフィックスとサフィックスを指定できます。

- ▶ **[検索するテキスト]**：検索するテキスト文字列を指定します。
- ▶ **[検索対象プレフィックス/サフィックス]**：検索するテキスト文字列のプレフィックスとサフィックスを指定します。
- ▶ **[大文字と小文字を区別する]**：検索時に大文字と小文字を区別します。
- ▶ **[JavaScript 警告ボックステキストを検索]**：JavaScript 警告ボックス内のテキストのみを検索します（Web（Click and Script）、PeopleSoft Enterprise、および Oracle Web Applications 11i Vuser のみ）。

アプリケーションとルールの追加と削除

- ▶ **[新規アプリケーション]**：左側の表示枠のアプリケーション・リストに新規アプリケーションを自動的に追加します。標準の名前は、Application_<**インデックス番号**>で、最初は **Application_1** から始まります。新しいアプリケーションを作成したら、**[新規ルール]** をクリックしてアプリケーションにルールを追加します。アプリケーションの名前を変更するには、名前をダブルクリックします。
- ▶ **[新規ルール]**：右側の表示枠にルール条件が表示され、現在選択されているアプリケーションの新しいルールを入力できるようになります。ルールはスクリプトとともに標準の XML ファイルに保存されます。このルール・ファイルをエクスポートすれば、ほかのユーザとの共有や、ほかのマシンでのインポートができます。
- ▶ **[削除]**：選択したルールまたはアプリケーションを削除します。

ルールのインポートとエクスポート

- ▶ **[インポート]** / **[エクスポート]** : ルール・ファイルをインポートまたはエクスポートします。アプリケーションとルールの指定は、**xml** という拡張子を持ったルール・ファイルに格納されます。ルールをファイルにエクスポートすることでほかのマシンでも利用できるようになります。ほかのルール・ファイルをインポートすることも可能です。選択してインポートしたルールが既存のルールと矛盾する場合、矛盾するルールであることを示す警告が **VuGen** によって表示されます。その場合、既存のスクリプトに対して作成したルールを、インポートしようとしているルールと統合するか、現在のルールを上書きするように指定できます。**[エクスポート]** をクリックすると、**[エクスポートするアプリケーションの選択]** ダイアログ・ボックスが表示されます。

標準のルールの設定

- ▶ **[標準値に設定]** : コンテンツ・チェックには、**インストール**、**標準設定**、**スクリプトごと** の 3 つのタイプがあります。「インストール」ルールは製品のインストール時に自動的に定められます。「標準設定」ルールはマシンで実行するすべてのスクリプトに適用されます。「スクリプトごと」のルールは、現在のスクリプトに対して定義されているものです。ルールを変更または追加したとき、その変更は現在のスクリプトにのみ適用されます。**VuGen** で、あるルールを標準設定ルールの一覧に加えて、そのマシンのすべてのスクリプトに適用するには、**[標準値に設定]** をクリックします。

複数のスクリプトを使う場合や、製品のアップグレードを行う場合には、標準設定のルールとスクリプトごとのルール間に矛盾が生じる場合があります。その場合、**VuGen** から、ルールを統合するかどうかの確認を求められます。ルールを統合すると（推奨）、アプリケーションのルール・リストにルールが追加されます。

この操作は、左側の表示枠のアプリケーション・リストで有効になっているアプリケーションにのみ適用されます。現在のスクリプトで有効になっているアプリケーションがなければ、**Defaults** ファイルでも有効なアプリケーションはありません。既存の **Defaults** ファイルを上書きするには、**[はい]** をクリックします。操作を取り消して、既存の **Defaults** ファイルを維持する場合は、**[いいえ]** をクリックします。

ルールは標準の **XML** ファイルに保存されます。このルール・ファイルをエクスポートすれば、ほかのユーザとの共有や、ほかのマシンでのインポートができます。

[標準値に設定] をクリックして上書きを承認すると、VuGen によって次の処理が行われます。

- 1 Defaults ファイルの全アプリケーションに**無効**の印を付けます。
- 2 現在のスクリプトで**有効**になっているアプリケーションについて、Defaults ファイル内でのアプリケーションの有無に応じて、統合またはコピーを行います。アプリケーションが存在する場合、現在のスクリプトのルールが Defaults ファイルのルールに統合されます。アプリケーションが Defaults ファイルになければ、ルールは VuGen によって Defaults ファイルにそのままコピーされます。
- 3 スクリプトにおいて有効となっていたアプリケーションについて、Defaults ファイルでも**有効**の印を付けます。現在のスクリプトで**有効**になっているアプリケーションがなければ、Defaults ファイルでも**有効**の印が付くアプリケーションはありません。

標準設定値を使用

Defaults ファイルからルールをインポートします。このボタンをクリックすると、アプリケーションとその標準設定のリストを示すダイアログ・ボックスが表示されます。これらのルールのインポートまたは変更を行うことができます。このルールが既存のルールと矛盾する場合、矛盾するルールであることを示す警告が VuGen によって表示されます。また、標準設定ファイルで定義されているルールを現在有効なルールに統合することもできます。

アプリケーションのすべての標準設定を使用するには [標準設定値を使用] をクリックします。これにより、標準設定ファイルから定義がインポートされます。このとき、アプリケーションとその標準設定のリストを示すダイアログ・ボックスが表示されます。これらの定義のインポートまたは変更を行うことができます。このルールがほかのルールと矛盾する場合、VuGen から矛盾するルールであることを示す警告が表示されます。また、標準設定ファイルで定義されているルールを現在有効なルールに統合することや、現在有効なルールで上書きすることもできます。

コンテンツ・チェック・ルールの定義

[実行環境設定] の [インターネット プロトコル: コンテンツ チェック] を使って、Web ページの内容を検査するルールを定義します。

コンテンツ・チェック・ルールを定義するには、次の手順を実行します。

- 1 [実行環境設定] ダイアログ・ボックスを開き、[インターネット プロトコル：コンテンツ チェック] ノードを選択します。
- 2 [再生中にコンテンツ チェックを有効にする] オプションを選択します。
- 3 [新規アプリケーション] ボタンをクリックして、内容を検査するアプリケーションのリストに新しいエントリを追加します。
- 4 既存のアプリケーションに対するルールを追加するには、[新規ルール] をクリックします。各アプリケーション・サーバに1つまたは複数のルールを適用できます。左側の表示枠でルールの横のチェック・ボックスのオン/オフを切り替えて、ルールの有効/無効を切り替えます。
- 5 実際のテキスト文字列を検索するには、[検索するテキスト] を選択して、検索したいテキストを入力します。最良の結果が得られるように、可能なかぎり具体的に入力してください。たとえば、「エラー」ではなく「ASP エラー」のように、アプリケーション固有のテキストを入力するようにします。
- 6 文字列の前後に付くテキストに基づいて検索するには、[検索対象プレフィックス/サフィックス] を選択し、前置記号と後置記号を指定します。
- 7 大文字と小文字を区別して指定するには、[大文字と小文字を区別する] を選択します。
- 8 ルールが、マシンに格納されているすべてのスクリプトに適用されるよう標準として設定するには、ルールまたはアプリケーションを選択して [標準値に設定] をクリックします。
- 9 ルールのファイルをエクスポートするには、[エクスポート] をクリックして、保存場所を指定します。
- 10 ルールのファイルをインポートするには、[インポート] をクリックし、ファイルの場所を探します。
- 11 アプリケーションやルールを削除するには、ルールを選択して [削除] ボタンをクリックします。
- 12 アプリケーション全体に標準の設定を使用するには、[標準設定値を使用] をクリックします。アプリケーションの一覧と標準の設定を含むダイアログ・ボックスが表示されます。矛盾がある場合は、ルールを上書きまたは統合できます。

第 24 章

選択したプロトコルの実行環境の設定

Vuser スクリプトを記録した後で、スクリプトの実行環境を設定します。これらの設定は、実行時のスクリプトの振る舞いを指定します。

本章の内容

- ▶ RDP の実行環境の設定 (470 ページ)
- ▶ Citrix の実行環境の設定 (475 ページ)
- ▶ .NET 環境の実行環境の設定 (479 ページ)
- ▶ Oracle NCA の実行環境の設定 (482 ページ)
- ▶ SAPGUI の実行環境の設定 (485 ページ)
- ▶ Java および EJB の実行環境の設定 (489 ページ)
- ▶ RTE の実行環境の設定 (491 ページ)
- ▶ WAP の実行環境の設定 (494 ページ)
- ▶ MMS の実行環境の設定 (500 ページ)
- ▶ Flex の実行環境の設定 (502 ページ)

RDP の実行環境の設定

RDP Vuser スクリプトを作成したら、実行環境を設定します。これらの設定によって、スクリプト実行時の Vuser の振る舞いを制御できます。

次の項目について RDP 固有の実行環境を設定できます。

- ▶ RDP 設定実行環境の設定
- ▶ RDP 同期化実行環境の設定
- ▶ RDP 詳細実行環境の設定
- ▶ RDP エージェント実行環境の設定

RDP 設定実行環境の設定

[RDP : 設定] を使用して、(仮想) RDP クライアントの振る舞いを設定します。



- ▶ **[RDP クライアント バージョン エミュレーション]** : 再生中に作成する RDP パケットのバージョン。[As Recorded] または特定のバージョン番号を指定できます。
- ▶ **[RDP キャッシュ使用を有効にする]** : RDP でのデータ・キャッシュの並べ替えをサポートします (標準設定では有効)。
- ▶ **[リモート デスクトップ解像度 (ピクセル)]** : アプリケーションを実行するウィンドウのサイズ。[As Recorded] または特定のサイズを指定できます。

- ▶ **[リモート デスクトップの色数]**：再生時の色深度の設定。[As Recorded] または特定の深度を指定できます。
- ▶ **[接続時に次のプログラムを開始]**：RDP 接続を開き、指定されたアプリケーションを起動します。[プログラムパスとファイル名] および [開始フォルダ] (任意) を指定します。

RDP 同期化実行環境の設定

[RDP : 同期化] の設定では、さまざまな関数の標準のタイミング値を指定します。

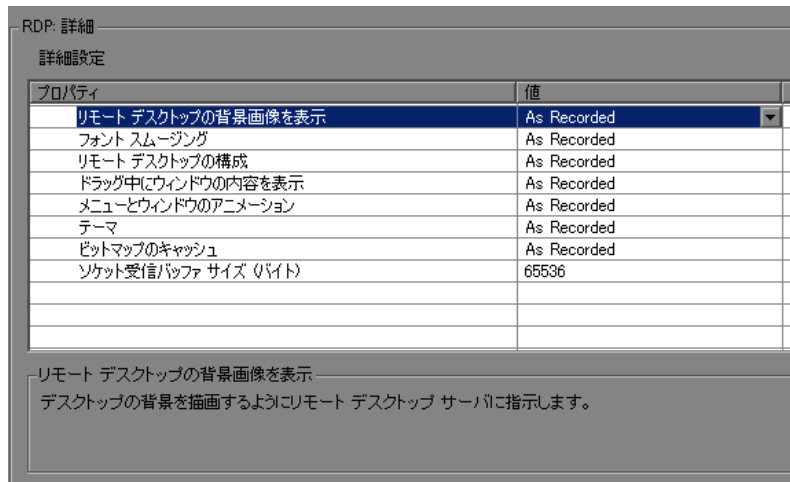


- ▶ **[標準設定の同期のタイムアウト (秒)]**：同期化操作を待機する時間 (秒単位)。0 ~ 1000 の値を入力します。標準設定値は 60 です。
- ▶ **[画像の同期の標準設定の許容レベル]**：画像の同期化を実行する際の許容レベル。[Exact], [Low], [Medium] (標準設定), [High] のいずれかを選択します。[High] は、変更や不一致に対して最も寛大です。[Low] はおよそ 95% の一致を要求, [Medium] はおよそ 85% の一致を要求, そして [High] はおよそ 70% の一致を要求します。また, [Exact] は 100% の一致を要求します。
- ▶ **[標準設定の入力原点]**：標準設定の入力操作の原点。
 - ▶ **[Recorded]**：入力原点が指定されていないすべての入力操作に座標を使用します。これが標準設定です。
 - ▶ **[Synched]**：前のいずれかの同期化関数で保存された最新のオフセットを、入力原点が指定されていない各入力操作の記録済み座標に追加します。

- ▶ **[標準設定のオフセット追加]**：後続のすべての関数のために、同期中に移動した画像のオフセットを保存します（標準設定では「いいえ」）。
- ▶ **[タイムアウトで画像の同期化ステップを失敗にする]**：同期化中に画像が見つからなかった場合の処理方法を指定します。「はい」（標準設定）を指定すると、失敗のステータスが設定され、Vuser は「エラー時も処理を継続する」の設定に従います。「いいえ」を指定すると、LR_NOT_FOUND フラグが返され、ステップが警告を報告し、スクリプトが続行されます。
- ▶ **[同期化の失敗を無効にする] ダイアログ**：選択した場合は、「同期化の失敗」ダイアログ・ボックスが開かなくなります（標準設定ではオフ）。
- ▶ **[タイピング速度 (ミリ秒 / 文字)]**：キーボード・コマンドの連続した文字を送信する間隔（ミリ秒単位）。0 ～ 1000 の値を入力します。標準設定値は 150 です。

RDP 詳細実行環境の設定

[詳細設定] ダイアログ・ボックスでは、RDP の詳細な実行環境の設定を編集できます。

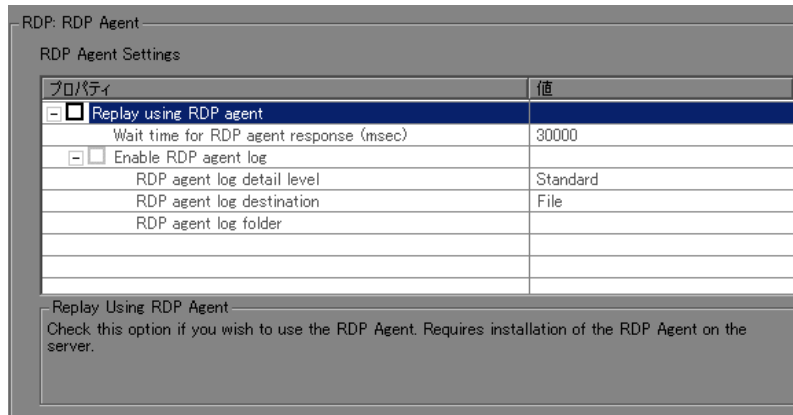


- ▶ **[リモート デスクトップの背景画像を表示]**：リモート・デスクトップにデスクトップの背景画像が表示されないようにして、リモート・デスクトップのアプリケーションを実行することができます。この設定を無効にすれば、リモート・デスクトップ・サーバのシステム・リソースを節約できます。

- ▶ **[フォント スムージング]**：リモート・デスクトップ・サーバがフォント・スムージングを使用できるようにします。この設定を無効にすれば、リモート・デスクトップ・サーバのシステム・リソースを節約できます。
- ▶ **[リモート デスクトップの構成]**：リモート・デスクトップの構成を有効にします。
- ▶ **[ドラッグ中にウィンドウの内容を表示]**：ウィンドウのドラッグ中にウィンドウの内容を表示します。この設定を無効にすれば、リモート・デスクトップ・サーバのシステム・リソースを節約できます。
- ▶ **[メニューとウィンドウのアニメーション]**：リモート・デスクトップ・サーバでメニューとウィンドウのアニメーションを可能にします。この設定を無効にすれば、リモート・デスクトップ・サーバのシステム・リソースを節約できます。
- ▶ **[テーマ]**：リモート・デスクトップ・サーバでの Windows のテーマの使用を可能にします。この設定を無効にすれば、リモート・デスクトップ・サーバのシステム・リソースを節約できます。
- ▶ **[ビットマップのキャッシュ]**：リモート・デスクトップ・サーバがビットマップ・キャッシュを使用できるようにします。この設定を有効にすれば、リモート・デスクトップ・サーバのシステム・リソースを節約できます。
- ▶ **[ソケット受信バッファ サイズ (バイト)]**：ソケットの受信バッファに割り当てるバイト数。バッファが小さすぎると、一杯になってサーバが切断される原因となります。バッファが大きすぎると、ローカルのシステム・リソース（メモリ）をより多く使用することになります。

RDP エージェント実行環境の設定

[RDP : エージェント] 実行環境の設定は、再生中に Microsoft Terminal Server の RDP エージェントと VuGen が連携する方法を制御します。



次のオプションを設定できます。

- ▶ **[Use RDP agent]** : 記録中に RDP エージェントを使用し、記録セッション中に RDP エージェントによって収集された情報を使用してスクリプトを生成するように指定します。サーバに LoadRunner RDP エージェントがインストールされている必要があります。
- ▶ **[Enable RDP agent log]** : RDP エージェント・ログを有効にします。この機能は、デバッグの目的でのみ使用します。
 - ▶ **[RDP agent log detail level]** : RDP エージェント・ログで生成される詳細のレベルを設定します。[Standard] を指定すると最も低いレベルになり、[Extended Debug] を指定すると最も高いレベルになります。
 - ▶ **[RDP agent log destination]** : RDP エージェント・ログ・データの宛先を設定します。[File] を指定すると、リモート・サーバ側にのみログ・メッセージが保存されます。[Stream] を指定すると、VuGen マシンにログ・メッセージが送信されます。[FileAndStream] を指定すると、両方の宛先にログ・メッセージが送信されます。
 - ▶ **[RDP agent log folder]** : RDP エージェント・ログ・ファイルが生成されるリモート・サーバ上のフォルダのパス。このオプションに何も指定せず、エージェント・ログの宛先を [File] に設定した場合、ログは、サーバ上のユーザの一時フォルダに保存されます。

Citrix の実行環境の設定

Citrix Vuser スクリプトを作成したら、実行環境を設定します。これらの設定によって、スクリプト実行時の Vuser の振る舞いを制御できます。[設定] ノードでの Citrix 実行環境の設定は、Citrix クライアントのプロパティと一致していません。これらの設定は、サーバにかかる負荷に影響します。接続のプロパティを表示するには、[Citrix Program Neighborhood] で ICA 接続を表すアイコンを選択し、右クリックして表示されるメニューから [アプリケーションセットの設定] を選択します。[デフォルトのオプション] タブを選択します。

注： Citrix Vuser では IP スプーフィングはサポートされていません。

一般的な実行環境の設定の詳細については、第 22 章「実行環境の設定」を参照してください。[速度のエミュレーション] プロパティの設定の詳細については、第 23 章「ネットワーク実行環境の設定」を参照してください。

次の項目について Citrix 固有の実行環境を設定できます。

- ▶ Citrix 設定実行環境の設定
- ▶ Citrix 同期化実行環境の設定

Citrix 設定実行環境の設定

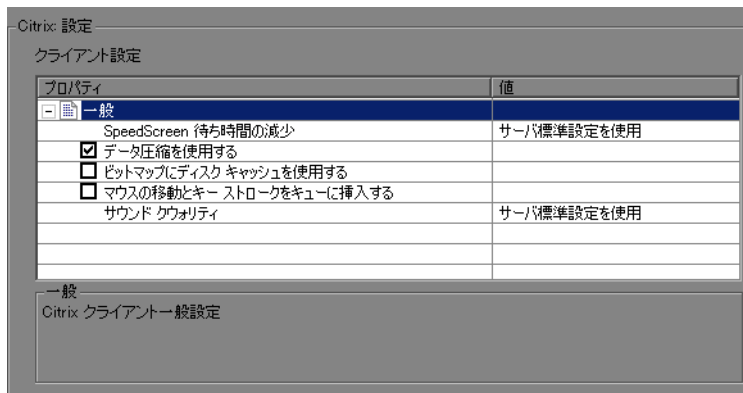
画面のレイテンシ、データ圧縮、ディスク・キャッシュ、マウスの動きのキューイングに関する設定です。

実行環境を設定するには、次の手順を実行します。



- 1 [実行環境設定] ダイアログ・ボックスを開きます。VuGen ツールバーにある [実行環境の設定] ボタンをクリックするか、[仮想ユーザ] > [実行環境の設定] を選択します。

2 [Citrix : 設定] ノードを選択します。[一般] プロパティを指定します。



必要なクライアント設定オプションを設定します。

- ▶ [SpeedScreen 待ち時間の減少] : ネットワークの速度が遅いときにユーザとのやり取りを向上させるために使用する機能。ネットワークの速度に応じてこの機能を「On」または「Off」にします。「Auto」オプションを選択すると、現在のネットワーク速度に基づいてオプションのオン/オフが切り替わります。ネットワーク速度がわからない場合は、このオプションを [サーバ標準設定値の使用] に設定し、マシンの標準設定を使用するようにします。
- ▶ [データ圧縮を使用する] : Vuser に、転送するデータを圧縮するよう指示します。このオプションを有効にするには、このオプションの左側にあるチェック・ボックスを選択します。無効にするには、チェック・ボックスをクリアします。帯域幅が限られている場合は、データ圧縮を有効にします（標準設定では有効）。
- ▶ [ビットマップにディスク キャッシュを使用する] : Vuser に、ビットマップや、よく使用するグラフィカル・オブジェクトをローカルのキャッシュに格納するよう指示します。このオプションを有効にするには、このオプションの左側にあるチェック・ボックスを選択します。無効にするには、チェック・ボックスをクリアします。帯域幅が限られている場合は、このオプションを有効にします（標準設定では無効）。

- ▶ **[マウスの移動とキーストロークをキューに挿入する]**：マウスの動きとキーストロークのキューが作成され、これらがパケットとして低い頻度でサーバに送られます。これにより、低速接続の場合のネットワーク・トラフィックが減少します。このオプションを有効にすると、セッションにおけるキーボードとマウスの動きに対する応答が鈍くなります。このオプションを有効にするには、このオプションの左側にあるチェック・ボックスを選択します。無効にするには、チェック・ボックスをクリアします（標準設定では無効）。
- ▶ **[サウンドクォリティ]**：サウンドのクォリティを指定します。**[サーバ標準設定値の使用]**、**[Sound Off]**、**[High sound quality]**、**[Medium sound quality]**、および **[Low sound quality]** があります。クライアント・マシンに 16 ビットの Sound Blaster 互換サウンド・カードが搭載されていない場合は、**[サウンドオフ]** を選択します。サウンドのサポートを有効にすると、クライアント・マシンの発行されたアプリケーションからサウンド・ファイルを再生できるようになります。

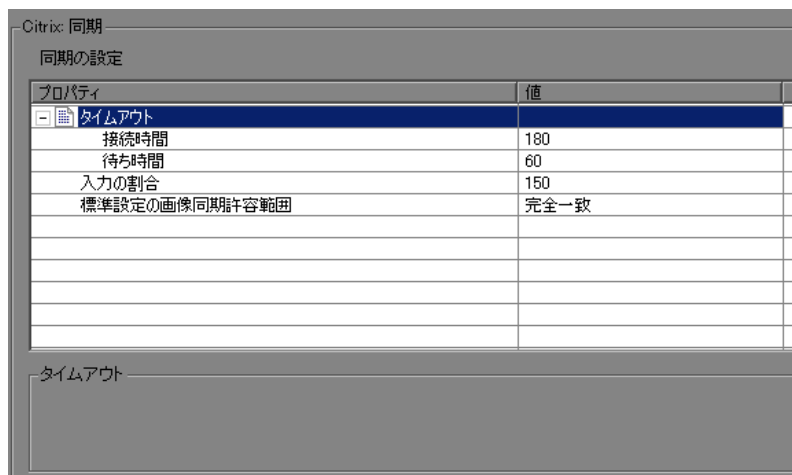
Citrix 同期化実行環境の設定

同期化設定は接続時間と待ち時間に関係します。

同期化の実行環境の設定を行うには、次の手順を実行します。



- 1 VuGen ツールバーにある **[実行環境の設定]** ボタンをクリックするか、**[仮想ユーザ]** > **[実行環境の設定]** を選択します。
- 2 **[Citrix : 同期]** ノードを選択します。



3 [接続時間] には、確立されている接続を終了する前に、アイドル状態で待機する秒数を示します。標準設定値は 180 秒です。

4 [待ち時間] には、接続を終了する前に、同期ポイントでアイドル状態のまま待機する秒数を示します。標準設定値は 60 秒です。

スクリプトの特定のセクションに対して待ち時間を設定するには、[挿入] > [新規ステップ] をクリックし、**待機時間設定**ステップを挿入します。新しい待ち時間は、挿入したポイントから、スクリプトの最後、または次の **待機時間設定**ステップまでを対象として適用されます。

5 [入力の割合] には、キーストローク間の遅延時間をミリ秒で指定します。

6 [標準設定の画像同期許容範囲] のレベルを指定します。この設定により、同期化されているとみなされるために 2 つの画像が共有しなければならない等しさのレベルが制御されます。

▶ [完全一致] : (標準設定) 100% の一致が必要。

▶ [Low] : 95% の一致が必要。

▶ [Medium] : 85% の一致が必要。

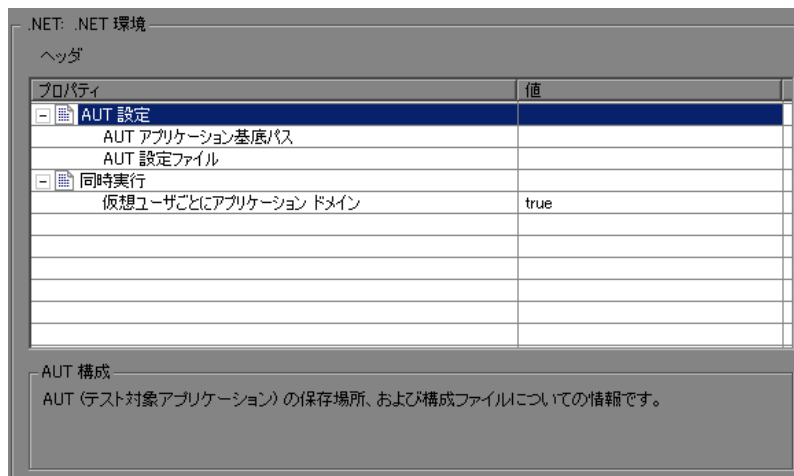
▶ [High] : 70% の一致が必要。

同期化の詳細については、『第 2 巻 - プロトコル』の第 18 章「Citrix プロトコル」を参照してください。

7 [OK] をクリックして設定を保存し、ダイアログ・ボックスを閉じます。

.NET 環境の実行環境の設定

Microsoft .NET Vuser スクリプトを実行する前に、.NET 環境設定を [実行環境設定] ダイアログ・ボックスから指定できます。



ペースや反復のオプションを設定するための一般的な Microsoft .NET スクリプトの実行時設定を設定することもできます。詳細については、第 22 章「実行環境の設定」を参照してください。

AUT 設定

[AUT アプリケーション基底パス]：再生時に DLL のロード元となる AUT (テスト対象アプリケーション) の基底ディレクトリです。標準設定では、記録時に必要なすべての DLL はスクリプトのディレクトリに格納されます。このオプションを使用して、見つからなかった AUT の DLL ファイルの場所を指定します。これは通常は、記録対象アプリケーションのインストール先パスです。AUT は、スクリプトを実行するマシンにインストールされている必要があります。このボックスを空白のままにしておくと、VuGen は再生時にアプリケーションの基底ディレクトリとしてローカルの `script\bin` ディレクトリを使用します。

[**AUT 設定ファイル**]：記録されるアプリケーションの設定ファイルの名前。
VuGen は、AUT 設定ファイルを `script\bin` ディレクトリにコピーし、ローカルに保存されたファイルをロードします。別の場所を指定するには、完全パスを使用します。ファイル名だけが指定され、ファイルが `script\bin` フォルダに存在しない場合、VuGen はアプリケーションの基底ディレクトリからファイルをロードします。

同時実行

- ▶ [**仮想ユーザごとにアプリケーション ドメイン**]：各 Vuser を個別のアプリケーション・ドメインで実行できるようにします（標準設定では有効）。Vuser を個別のアプリケーション・ドメインで実行すると、静的変数が Vuser の間で共有されないため、互いをロックすることなしに Vuser を個別に実行できます。

ADO.NET プロバイダは、負荷テストの精度に大きな影響を与える **接続プール** と呼ばれる機能をデプロイします。すべての Vuser で使用されるアプリケーション・ドメインが 1 つだけのときは常に、接続プールは有効です。.NET Framework はデータベース接続を開いたまま維持し、新しい接続が要求された場合にそれらの再利用を試みます。多くの Vuser が 1 つのアプリケーション・ドメインで実行されるため、Vuser が互いに干渉する可能性があります。この動作は直線的ではなく、そのことが Vuser の精度を下げる場合があります。標準設定では**有効**で、Vuser ごとに個別の接続プールが割り当てられます。つまり、Vuser ごとに接続プールはあるけれども、Vuser は互いに干渉しません。この設定により精度は向上しますが、スケーラビリティは低下します。

このオプションを無効にした場合、データベースの接続プールを手作業で無効にする必要があります。

次の表に、手作業で接続プールを無効にする方法を示します。

| プロバイダ | オプション |
|--|--|
| .NET Framework Data Provider for SQL Server | "Pooling=false" または "Pooling=no" |
| .NET Framework Data Provider for Oracle | "Pooling=false" または "Pooling=no" |
| .NET Framework Data Provider for ODBC | 接続プールは ODBC Driver Manager により管理されます。接続プールを有効または無効にするには、ODBC データ・ソース・アドミニストレータを使用します（[コントロールパネル] または [管理ツール] フォルダにあります）。[接続プール] タブでは、インストールされている ODBC ドライバごとに接続プール・パラメータを指定できます。 |
| .NET Framework Data Provider for OLE DB | "OLE DB Services=-2" |
| Oracle Data Provider for .NET | "pooling=false" |
| Adaptive Server Enterprise ADO.NET Data Provider | "Pooling=False" |

.NET リソースを指定するには、次の手順を実行します。

- 1 実行環境の設定を開きます。F4 キーを押すか、[仮想ユーザ] > [実行環境の設定] を選択します。
- 2 左側の表示枠で [.NET 環境] ノードをクリックします。
- 3 [AUT アプリケーション基底パス] ボックスに、DLL の基底フォルダを設定します。
- 4 [AUT 設定ファイル] ボックスに、記録対象アプリケーションのパスを設定します。
- 5 [仮想ユーザごとにアプリケーション ドメイン] を有効に設定することをお勧めします（標準設定）。

Oracle NCA の実行環境の設定

スクリプトを実行する前に、実行環境の設定を行って、スクリプトが実ユーザを正確にエミュレートするようにできます。すべてのプロトコルに共通の一般的な実行環境の設定（思考遅延時間、間隔、ログ記録など）については、第 22 章「実行環境の設定」を参照してください。ネットワークに関する設定については、第 23 章「ネットワーク実行環境の設定」を参照してください。

次の項では、Oracle NCA Vuser 専用の実行環境の設定について説明します。これらの実行環境の設定を行うことで、エミュレートする通信パラメータを指定できます。

Oracle NCA クライアント・エミュレーションの実行環境設定

Oracle NCA クライアントが正確にエミュレートされるように、ネットワークを設定できます。



次のオプションを設定できます。

ソケット・モード

クライアントとの通信がより高速な HTTP レベルではなくソケット・レベルで実行されます。

[**タイムアウト (秒)**] : サーバからの応答を Oracle NCA Vuser が待機する時間です。標準設定値は -1 です。-1 の場合、タイムアウトは無効になり、クライアントはいつまでも待機します。

プラグマ・モード

プラグマ・モードでは、Oracle によって定義されているプラグマ・モードで通信が行われます。プラグマ・モードの通信は、HTTP および Servlet よりも上位の通信レベルで、メッセージを定期的送信するという特徴があります。このモードでは、クライアントはサーバが直ちにデータを返さないことを認識します。サーバは、要求されたデータを送ることができるまで、所定の間隔でメッセージを送信します。

- ▶ [**再試行限度**] : クライアントがエラーを発行するまでにサーバから受け付ける **IfError** メッセージの最大数を指定します。**IfError** メッセージは、サーバからクライアントに定期的送られるメッセージで、できるだけ早くデータを返すことを通知するものです。
- ▶ [**再試行間隔**] : **IfError** メッセージが生じた場合の再試行の間隔を指定します。
- ▶ [**再試行間隔をトランザクションに含める**] : 再試行の間の間隔もトランザクション持続時間に含めます。

プラグマ・モードでの記録の詳細については、『第 2 巻 – プロトコル』の第 45 章「Oracle NCA プロトコル」を参照してください。

ハートビート

Oracle サーバに送信されるハートビートを有効または無効にします。ハートビートは、サーバとの通信が正常に行われていることを確認する処理です。Oracle NCA サーバに高い負荷がかかっている場合は、ハートビートを無効にします。ハートビートを有効にした場合は、ハートビート・メッセージをサーバに送信する間隔を設定できます。

- ▶ [**ハートビートを有効にする**] : 標準設定では、ハートビートはサーバへ送信される信号です。これを無効にするには、チェック・ボックスをクリアします。
- ▶ [**頻度**] : ハートビート信号の頻度です。標準設定値は 120 秒です。

フォーム

記録時に検出された Oracle Forms のバージョンを示します。

- ▶ **[バージョン]** : この設定は、記録を行った後にサーバがアップグレードされた場合にのみ変更します。

診断

Oracle Applications のデータベース層の診断モジュールに関する情報を提供します。

- ▶ **[アプリケーションのバージョン]** : Oracle Application のバージョン。このオプションは、Oracle Application を使用する場合に使用できます (ユーザ定義の Oracle NCA アプリケーションでは使用できません)。これは、Oracle データベース・ブレイクダウンを使用する場合にのみ必要です。

クライアントのエミュレーションを設定するには、次の手順を実行します。



- 1 **[実行環境設定]** ダイアログ・ボックスを開きます。**[仮想ユーザ]** > **[実行環境の設定]** を選択するか、VuGen ツールバーの **[実行環境の設定]** ボタンをクリックします。
- 2 実行環境設定ツリーから **[Oracle NCA : クライアントのエミュレーション]** ノードを選択します。
- 3 ネットワーク・タイムアウト値を秒単位で設定します。クライアントにサーバの応答をいつまでも待機させるには、標準設定値 -1 を使用します。
- 4 プラグマ・モードで作業をするときは、クライアントがエラーを発行するまでに受け付ける **IfError** メッセージの再試行回数 **[最大エントリ数]** を指定します。標準設定値は 5 です。
- 5 Oracle NCA サーバへのハートビートの送信を有効にするには、**[ハートビートを有効にする]** オプションを選択します。次の行に、ハートビートを送信する間隔を秒単位で指定します。標準設定値は 120 秒です。
- 6 **[OK]** をクリックして設定を適用し、スクリプトを実行します。

SAPGUI の実行環境の設定

SAPGUI Vuser スクリプトの作成と拡張を終えたら、その実行環境の設定を行い、VuGen から実行してその機能を確認します。実行環境を設定することによって、再生時の Vuser の動作を制御します。これらの設定は Vuser スクリプトを実行する前に行います。一般の実行環境と SAPGUI 固有の実行環境の両方を設定できます。

この一般設定には、実行論理、ペース設定、ログ、思考遅延時間、パフォーマンスの設定が含まれます。一般の実行環境の設定については、第 22 章「実行環境の設定」を参照してください。SAPGUI 固有の設定については、以降の項を参照してください。

実行環境の設定が完了したら、Vuser スクリプトを保存して VuGen から実行し、正しく動作することを確認します。Vuser スクリプトをスタンドアロン・テストとして実行する方法の詳細については、第 9 章「スタンドアロン・モードでの Vuser スクリプトの実行」を参照してください。

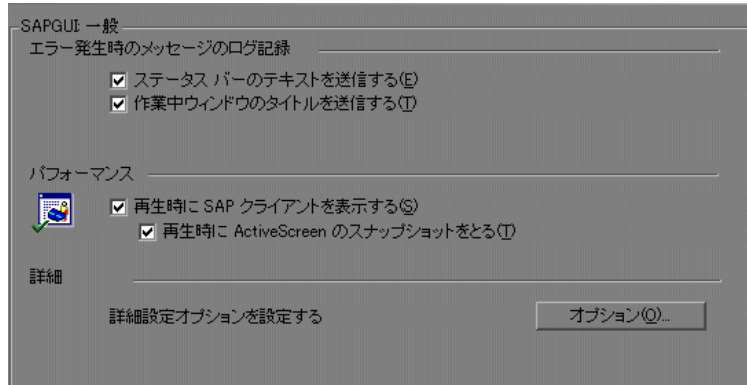
スクリプトを作成した後、スクリプトを自分の環境（LoadRunner シナリオ、Performance Center 負荷テスト、または Business Process Monitor プロファイル）に組み込みます。詳細については、**HP LoadRunner Controller**、**HP Performance Center**、または **HP Business Availability Center** のドキュメントを参照してください。

次の項目について SAPGUI 固有の実行環境を設定できます。

- ▶ SAPGUI : 実行環境の一般設定
- ▶ SAPGUI : 実行環境の詳細設定

SAPGUI : 実行環境の一般設定

実行環境の一般設定では、SAPGUI Vuser スクリプトの一般設定が行えます。VuGen ではこれらの設定がスクリプトの実行時に使用されます。



[SAPGUI : 一般] の [エラー発生時のメッセージのログ記録] では、エラーが発生するたびに Vuser が実行ログに送信する情報を指定します。

- ▶ [ステータスバーのテキストを送信する] : ステータス・バーからログ・ファイルにテキストを送信します。
- ▶ [作業中ウィンドウのタイトルを送信する] : 作業中のウィンドウのタイトル・テキストをログ・ファイルに送信します。

[SAPGUI : 一般] の [パフォーマンス] では、再生時に SAP クライアントを表示するかどうかを指定できます。

- ▶ [再生時に SAP クライアントを表示する] : 再生中に SAP クライアントにアクションのアニメーションを表示します。ユーザ・インタフェース (UI) を表示させる利点は、フォームにどのように入力が行われているかを確認でき、Vuser のアクションを詳細に追えることです。しかし、このオプションではより多くのリソースが必要になるため、負荷テストのパフォーマンスに影響を与える場合があります。

- ▶ **[再生時に ActiveScreen のスナップショットをとる]** : Control ID 情報を持つ再生スナップショットをすべてのアクティブ・オブジェクトでキャプチャします。ActiveScreen スナップショットは、通常のスナップショットと異なり、SAPGUI クライアントの中でどのオブジェクトが VuGen によって認識されているかを知ることができます。スナップショット上でマウスを動かすと、VuGen は検出したオブジェクトを強調表示します。スナップショット内からスクリプトに直接新しいステップを追加できます。また、特定のオブジェクトに対して、スナップショット内から対話的にステップを追加することもできます。詳細については、『第2巻ープロトコル』の第46章「SAPGUI プロトコル」を参照してください。

詳細設定のためのオプションを使って、[SAPfewgsvr.exe] プロセスのタイムアウトの設定、エラー発生時のスナップショットの保存、および再生中に VuGen が SAPlogon を使用する設定を行うことができます。詳細については、488 ページ「SAPGUI : 実行環境の詳細設定」を参照してください。

SAPGUI 用の実行環境を設定するには、次の手順を実行します。

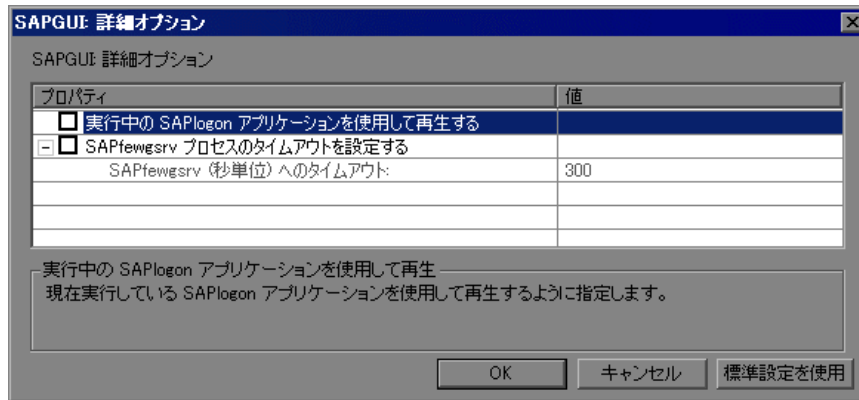


- 1 [実行環境設定] ダイアログ・ボックスを開きます。VuGen ツールバーの **[実行環境の設定]** ボタンをクリックするか、[仮想ユーザ] > [実行環境の設定] を選択して、[実行環境設定] ダイアログ・ボックスを開きます。
- 2 **[SAPGUI : 一般]** ノードを選択します。
- 3 **[エラー発生時のメッセージのログ記録]** セクションで、メッセージ・ソース **[ステータス バーのテキストを送信する]** と **[作業中ウィンドウのタイトルを送信する]** から少なくとも1つを選択します。
- 4 再生時に SAP ユーザ・インタフェースを表示するには、**[パフォーマンス]** セクションで **[再生時に SAP クライアントを表示する]** チェック・ボックスを選択します。
- 5 **[オプション]** をクリックし、**SAPfewgsvr.exe** プロセスのタイムアウトを設定します。

SAPGUI : 実行環境の詳細設定

各 Vuser はテスト実行時に、個別の **SAPfewgsrv.exe** プロセスを呼び出します。場合によっては、再生セッションの終了後もプロセスが終了しないことがあります。プロセスがアクティブかどうかを調べるには、Windows タスク・マネージャを確認します。

[SAPGUI : 詳細オプション] では、このアプリケーションのタイムアウトを設定できます。タイムアウトの時間に達した時点で、VuGen はまだ終了していない **SAPfewgsrv** プロセスを終了します。



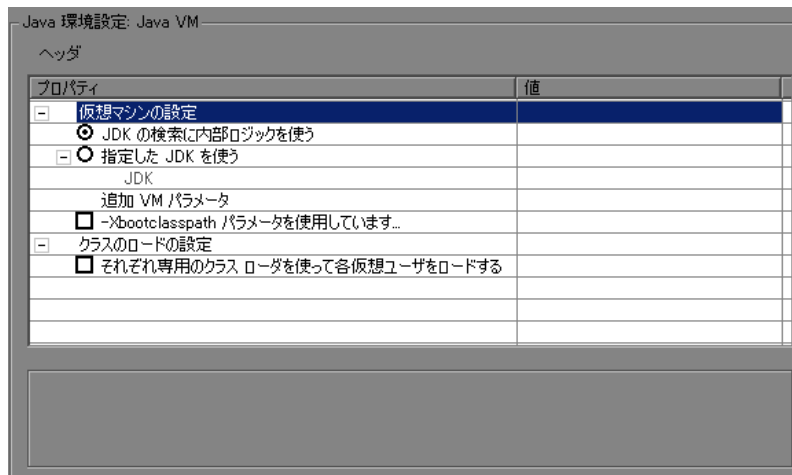
- ▶ [実行中の SAPlogon アプリケーションを使用して再生する] : Vuser が再生のために現在実行中の SAPlogon アプリケーションを使うように指定します。
- ▶ [SAPfewgsrv プロセスのタイムアウトを設定する] : SAPfewgsrv.exe プロセスのタイムアウトを修正できます。
- ▶ [SAPfewgsrv へのタイムアウト] : SAPfewgsrv.exe プロセスのタイムアウトを秒単位で指定します。標準設定値は 300 秒です。

Java および EJB の実行環境の設定

Java 関連の実行環境の設定は、[実行環境設定] ダイアログ・ボックスの [Java VM] オプションで行います。

JVM 実行環境の設定

[Java VM] セクションで、Java 仮想マシンの設定情報を入力します。



以下の設定を利用できます。

- ▶ **[仮想マシンの設定] :**
 - ▶ **[JDK の検索に内部ロジックを使う] :** PATH, レジストリ, Windows フォルダで、再生時に使用する JDK を検索します。
 - ▶ **[指定した JDK を使う] :** 再生時に、下に指定されている JDK を使用します。
- ▶ **[追加 VM パラメータ] :** 仮想マシンで使用する任意のパラメータを入力します。
 - ▶ **[-Xbootclasspath パラメータを使用しています ...] :** Xbootclasspath /p オプションを指定してスクリプトを再生します。
- ▶ **[クラスのロードの設定] :**
 - ▶ **[それぞれ専用のクラスローダを使って各仮想ユーザをロードする] :** 各 User を専用のクラス・ローダを使用してロードします。これにより、User ごとに一意の名前空間を使用して、リソースを別々に管理できるようになります。

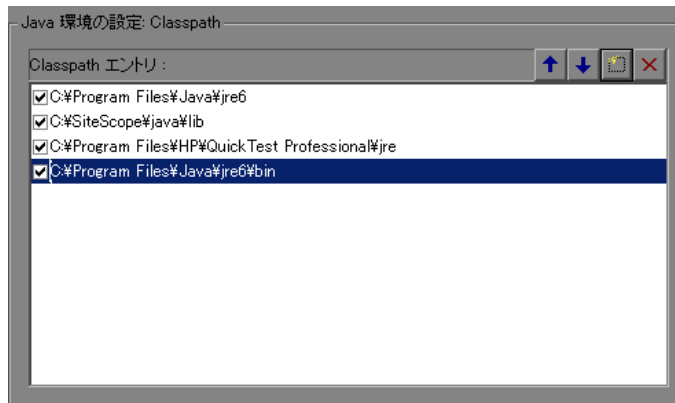
Java VM の実行環境を設定するには、次の手順を実行します。

- 1 [仮想ユーザ] > [実行環境の設定] を選択し、[実行環境の設定] ツリーの [Java 環境の設定 : Java VM] ノードを選択します。
- 2 必要な [仮想マシンの設定] を選択し、再生時に使用する JDK を指定します。
- 3 `-Xbootclasspath/p` オプションを使って再生するには、[`-Xbootclasspath パラメータを使用しています ...`] オプションを選択します。
- 4 [OK] をクリックします。

実行環境の設定のクラスパス・オプションの設定

[Classpath] セクションでは、システムのクラスパス環境変数に含まれていない追加のクラスの場所を指定できます。これらのクラスは、Java アプリケーションの実行や、正しい再生を保証するのに必要な場合があります。

コンピュータまたはネットワークから必要なクラスを検索し、特定のテストの場合にそのクラスを無効にすることができます。また、クラスの順序を変更して、クラスパスのエントリを操作することもできます。



クラスパスの実行環境を設定するには、次の手順を実行します。

- 1 [実行環境の設定] (F4) を開きます。[実行環境の設定] ツリーの [Java 環境の設定 : Classpath] ノードを選択します。
- 2 リストにクラスパスを追加します。



[**クラスパスの追加**] ボタンをクリックします。クラスパスのリストに新しい行が追加されます。

新しいクラスに対して、**jar**、**zip** またはほかのアーカイブ・ファイルのパスと名前を入力します。または、フィールドの右にある [**参照**] ボタンをクリックして、指定したいファイルを選択します。VuGen によって、クラスパスのリストに新しい場所が、ステータスが有効な状態で追加されます。



3 クラスパスのエントリを完全に削除するには、対象のエントリを選択して [**削除**] ボタンをクリックします。

4 特定のテストの場合にだけクラスパスのエントリを無効にするには、エントリの左にあるチェック・ボックスをクリアします。



5 クラスパス・エントリをリストの下方へ移動するには、対象エントリを選択して下向き矢印ボタンをクリックします。



6 クラスパス・エントリをリストの上方へ移動するには、対象エントリを選択して上向きの矢印ボタンをクリックします。

7 [**OK**] をクリックしてダイアログ・ボックスを閉じます。

RTE の実行環境の設定

ターミナル・エミュレータ関連の実行環境の設定は、[実行環境設定] ダイアログ・ボックスの [**RTE**] ノードで行います。

接続試行回数の変更

TE_connect 関数は、ホストへの接続を記録したときに VuGen によって生成されます。RTE Vuser スクリプトを再生すると、**TE_connect** 関数はターミナル・エミュレータを指定されたホストに接続します。最初の接続試行に失敗すると、Vuser は一定の回数だけ接続を試行します。接続の詳細は、**output.txt** というレポート・ファイルに記録されます。

Vuser が行う接続試行の最大回数を設定するには、RTE 実行環境の設定の [**最高接続試行回数**] ボックスに回数を入力します。

標準設定では、Vuser は接続を 5 回試行します。

TE_connect 関数の詳細については、『**Online Function Reference**』（英語版）（[ヘルプ] > [関数リファレンス]）を参照してください。

元のデバイス名の指定

特定の環境では、各セッション（Vuser）に固有のデバイス名を付ける必要があります。**TE_connect** 関数は、各 Vuser に対して 8 文字からなる一意のデバイス名を生成し、この名前を使って接続します。デバイス名（**TE_connect** 関数の **com_string** パラメータ内に含まれる）を使って接続するには、RTE 実行環境の設定の「**元のデバイス名を使用する**」オプションを選択します。

注：元のデバイス名の設定は、IBM ブロック・モード・ターミナルにのみ適用されます。

標準設定では、Vuser は元のデバイス名を使って接続します。

TE_connect 関数の詳細については、『**Online Function Reference**』（英語版）（**[ヘルプ]** > **[関数リファレンス]**）を参照してください。

入力遅延の設定

入力遅延の設定では、Vuser による **TE_type** 関数の実行方法を指定します。

Vuser が文字列の最初の文字を入力する前の待機時間を指定するには、**[第一キー]** ボックスに値（ミリ秒単位）を入力します。

Vuser がそれ以降の文字を送信するときの間隔を指定するには、**[後続キー]** ボックスに値（ミリ秒単位）を入力します。

最初のキー入力遅延とそれ以降のキー入力遅延の両方に 0 を入力すると、文字間の遅延はなくなり、Vuser は複数の文字を 1 つの文字列として送信します。

Vuser スクリプトの一部で入力遅延の設定をオーバーライドするには、**TE_typing_style** 関数を使用します。

TE_type 関数と **TE_typing_style** 関数の詳細については、『**Online Function Reference**』（英語版）（**[ヘルプ]** > **[関数リファレンス]**）を参照してください。

X SYSTEM 同期化の設定

RTE Vuser スクリプトは、**TE_wait_sync** 関数を使って同期化を行います。すべての **TE_wait_sync** 関数に適用されるタイムアウト値と安定時間値を設定できます。**TE_wait_sync** 関数の詳細については、『**Online Function Reference**』（英語版）（[ヘルプ] > [関数リファレンス]）を参照してください。

タイムアウト

TE_wait_sync 関数を再生したときに、システムが安定しないまま同期化のタイムアウトに達すると、**TE_wait_sync** 関数からエラー・コードが返されます。同期化のタイムアウトを設定するには、RTE 実行環境の設定の [タイムアウト] セクションに値（秒単位）を入力します。

標準のタイムアウト値は 60 秒です。

安定時間

Vuser は、**TE_wait_sync** 関数を実行した後、ターミナルが X SYSTEM モードに入らなくなるまで待機します。ターミナルが X SYSTEM モードから戻った後も、Vuser は少しの間システムを監視します。これによって、ターミナルが安定した（システムが X SYSTEM モードに戻らない）ことを確認します。この確認ができた場合にのみ、**TE_wait_sync** 関数が終了します。

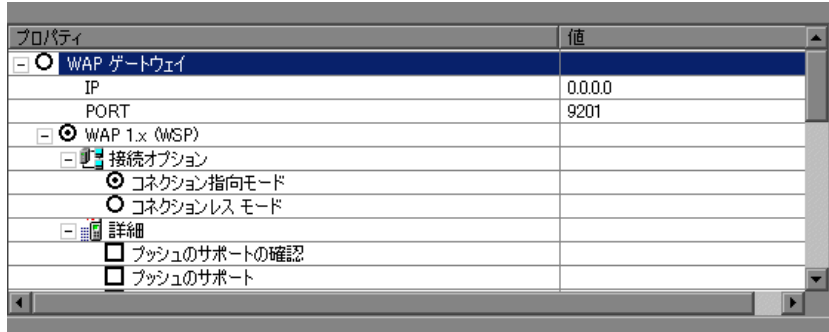
Vuser が X SYSTEM モードから戻った後のシステムを監視し続ける時間を設定するには、RTE 実行環境の設定の [Stable time] ボックスに値（ミリ秒単位）を入力します。

標準の安定時間は 1000 ミリ秒です。

WAP の実行環境の設定

ゲートウェイ・オプションの設定

[実行環境設定] ダイアログ・ボックスの [WAP : ゲートウェイ] ノードを使って、ゲートウェイの設定を行います。



接続オプション

接続オプションでは、Vuser が WAP ゲートウェイへの接続に使用する方式を指定します。

[**WAP ゲートウェイ**] : Web サーバに WAP ゲートウェイ経由でアクセスする Vuser を実行します。

[**HTTP 直接**] : Web サーバに直接アクセスする Vuser を HTTP モードで実行します。

注 : [HTTP 直接] 接続モードを選択した場合、残りの WAP ゲートウェイ・オプションは適用されません。

ゲートウェイの設定

Vuser がゲートウェイ経由で接続する場合は、IP、ポート、WAP のバージョンの各オプションによってゲートウェイ接続が指定されます。

[**IP**] : ゲートウェイの IP アドレスを指定します。

[**ポート**] : ゲートウェイのポートを指定します。WAP ゲートウェイ経由で Vuser を実行する場合は、選択したモードに応じて標準のポート番号が自動的に設定されます。ただし、設定をカスタマイズして、ユーザ定義の IP アドレスとポートを指定することもできます。

[**Wap のバージョン**] : 適切な WAP のバージョン (**1.x (WSP)** または **2.0 (HTTP プロキシ)**) を選択します。WAP 1.x (WSP) で記録した場合は、1.x (WSP) モードまたは 2.0 (HTTP プロキシ) モードで Vuser を実行できます。WAP 2.0 (HTTP プロキシ) で記録した場合は、その同じモードでのみ Vuser を実行できます。

スクリプトを WAP 1.x (WSP) で実行する場合は、いくつかの接続オプションと詳細オプションを指定できます。

ゲートウェイ接続モード

接続モードの設定は WAP バージョン 1.x (WSP) 接続に適用されます。

[**コネクション指向型**] : WSP セッションの接続モードを「コネクション指向」に設定します。

[**コネクションレス型**] : WSP セッションの接続モードを「コネクションレス」に設定します。

[**セキュリティを有効にする**] : WAP ゲートウェイへの接続のセキュリティを有効にします。

ゲートウェイの詳細設定

WAP の機能およびゲートウェイの詳細オプションを設定するには、[ゲートウェイ] ノードの [詳細設定] オプションを展開します。

| プロパティ | 値 |
|---|------|
| <input type="checkbox"/> プッシュのサポートの確認 | |
| <input type="checkbox"/> プッシュのサポート | |
| <input type="checkbox"/> CapSessionResume | |
| <input type="checkbox"/> 確認応答ヘッダ | |
| サーバ SDU バッファ サイズ | 4000 |
| クライアント SDU バッファ サイズ | 4000 |
| MethodMOR | 1 |
| PushMOR | 1 |
| Bearer Type | UDP |

- ▶ [プッシュのサポートの確認] : CO モードでプッシュ型メッセージが受信されたときに、Vuser にメッセージの受信を確認させます (標準設定では無効)。詳細については、『第 2 巻 - プロトコル』の第 58 章「ワイヤレス・プロトコル」を参照してください。
- ▶ [プッシュのサポート] : プッシュ・タイプのメッセージがゲートウェイを通過できるようにします (標準設定では無効)。
- ▶ [CAPSessionResume] : セッションの一時停止または再開の要求を可能にします。
- ▶ [確認応答ヘッダ] : ゲートウェイに情報を提供する標準ヘッダを返します (標準設定では無効)。
 - ▶ [サーバ SDU バッファ サイズ] : セッションの実行中にサーバへ送信可能な最大のトランザクション・サービス・データ・ユニット (標準設定では 4000)。
 - ▶ [クライアント SDU バッファ サイズ] : セッションの実行中にクライアントへ送信可能な最大のトランザクション・サービス・データ・ユニット (標準設定では 4000)。
 - ▶ [MethodMOR] : 同時に発生可能な未処理の方式の数。
 - ▶ [PushMOR] : 同時に発生可能な未処理のプッシュ・トランザクションの数。
 - ▶ [ベアラ タイプ] : 伝送の基盤として使用されるベアラのタイプ。

- ▶ **[メッセージ取得]** : Vuser がプッシュ型メッセージを受信すると、そのメッセージに示された URL からメッセージ・データを取得します (標準設定では無効)。
- ▶ **[クッキーをサポート]** : クッキーの保存と取得をサポートします (標準設定では無効)。
- ▶ **[WTP の分割と再組み立て]** : WTP (Wireless Transport Protocol) による分割と再組み立て (SAR) を有効にします (標準設定では有効)。
 - ▶ **[Wtls Retransmission Time]** : WTP レイヤが応答の受信に失敗して PDU を再送信する前に待機する秒数 (標準設定では 5000 秒)。
- ▶ **[Wtls 簡略ハンドシェイク]** : リダイレクト・メッセージの受信時に、完全なハンドシェイクではなく略式のハンドシェイクを使用します (標準設定では無効)。
- ▶ **[WTLS Diffie Hellman]** : WTLS (Wireless Transport Layer Security) で、標準の暗号化方式である RSA ではなく Diffie-Hellman 暗号化方式を使用します (標準設定では無効)。
 - ▶ **[WTLS Diffie Hellman 識別子]** : Diffie-Hellman 暗号化方式の識別子。この識別子は、Diffie-Hellman 暗号化方式を使用する Operwave ゲートウェイによる略式のハンドシェイクに必要です。
 - ▶ **[ネットワーク MTU サイズ]** : ネットワーク・パケットの最大バイト数 (標準設定では 4096 バイト)。

ゲートウェイのオプション設定

この項では、WAP ゲートウェイのオプションを設定する手順を示します。

WAP ゲートウェイ・オプションを設定するには、次の手順を実行します。



- 1 **[実行環境の設定]** ボタンをクリックするか、**[仮想ユーザ]** > **[実行環境の設定]** を選択して、**[実行環境設定]** ダイアログ・ボックスを表示します。**[WAP : ゲートウェイ]** ノードを選択します。
- 2 スクリプトを (HTTP ではなく) WSP モードで再生する場合は、**[WAP ゲートウェイ]** を選択します。
- 3 ゲートウェイの IP アドレスとポートを指定します。VuGen の標準設定のポートを使用することもできます。
- 4 WAP のバージョン (**WAP 1.x (WSP)** または **WAP 2.0 (HTTP)**) を選択します。

- 5 WAP 1.x (WSP) を選択した場合は、接続モード ([**コネクション指向**] または [**コネクションレス**]) を選択します。セキュリティ上安全な接続モードを指定するには、[**セキュリティを有効にする**] オプションを選択します。
- 6 WAP 1.x (WSP) を選択した場合は、[**詳細設定**] ノードを展開し、クライアントの機能とほかのゲートウェイ詳細オプションを設定します。[詳細設定] オプションの詳細については、前述を参照してください。

Radius 接続データの設定

RADIUS (Remote Authentication Dial-In User Service) は、クライアント / サーバのプロトコルとソフトウェアの組み合わせで、リモート・アクセス・サーバと中央サーバとの通信によって、ダイヤルアップ・ユーザの認証と、そうしたユーザのシステムやサービスへのアクセス要求に対する権限を付与することを可能にします。

RADIUS により、すべてのリモート・サーバから利用できる中央のデータベースにユーザ・プロファイルを維持することができます。その結果、セキュリティが向上し、集中管理を行うネットワーク上の 1 か所でポリシーを設定して適用できます。サービスを集中管理することによって、請求処理のための使用状況の追跡、ネットワークの統計情報の格納が簡単になります。

RADIUS には、次の 2 つのサブプロトコルがあります。

- ▶ [**認証**] : ユーザ・アクセス権を付与して制御します。
- ▶ [**会計**] : 請求処理を行うため、およびネットワークの統計情報を取得するために使用状況を追跡します。

VuGen では、WSP 再生の場合のみ、RADIUS のサブプロトコルである認証とアカウントの両方をサポートしています。

[実行環境設定] の [Radius] ノードで、ダイヤルアップ情報を入力します。

| プロパティ | 値 |
|-----------|--|
| ネットワークの種類 | アカウントネットワークの種類 : GPRS (General Packet Radio Service) または CSD (Circuit-Switched Data) を選択します。 |
| IP アドレス | Radius サーバの IP アドレス。 |
| 認証ポート番号 | Radius サーバの認証ポート番号。 |

| | |
|-------------------------------|---|
| アカウント ポート番号 | Radius サーバのアカウント・ポート番号。 |
| 秘密鍵 | Radius サーバの秘密鍵。 |
| 接続タイムアウト (秒) | Radius サーバが応答を待機する秒数。標準設定値は 120 秒です。 |
| 再送の再試行回数 | 伝送の失敗後に再送を試みる回数。標準設定値は 0 です。 |
| サーバから返された属性をパラメータに保存する | Vuser が、サーバから返された属性をパラメータ (後で使用可能) として保存できるようにします。標準設定は False です。 |
| Radius クライアントの IP | Radius パケットの発信元 IP。通常は、1 台の Load Generator マシン上の異なる NIC カードから送信されるパケットを区別するために使用されます。 |

WAP Radius のオプションを設定するには、次の手順を実行します。



- 1 **[実行環境の設定]** ボタンをクリックするか、**[仮想ユーザ]** > **[実行環境の設定]** を選択して、**[実行環境設定]** ダイアログ・ボックスを表示します。**[Radius]** ノードをクリックします。

WAP: Radius

設定

| プロパティ | 値 |
|------------------------|---------|
| ネットワークの種類 | CSD |
| IP アドレス | 0.0.0.0 |
| 認証ポート番号 | 1812 |
| アカウント ポート番号 | 1813 |
| 秘密鍵 | secret |
| 接続タイムアウト (秒) | 120 |
| 再送の再試行回数 | 0 |
| サーバから返された属性をパラメータに保存する | False |
| Radius クライアントの IP | default |

ネットワークのタイプ
ネットワークのタイプを 1 つ選択します。

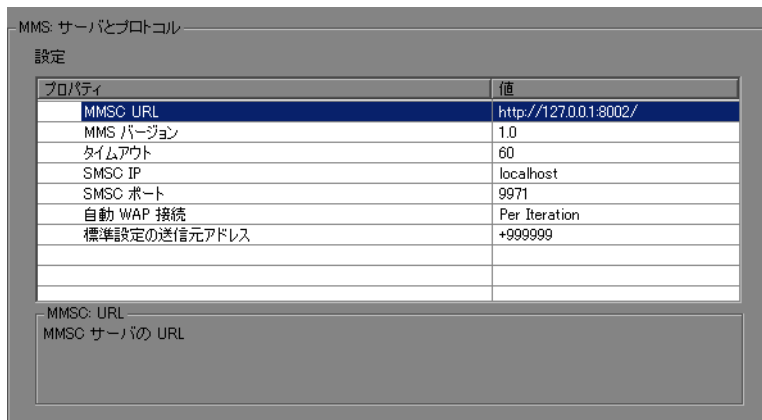
- 2 アカウント用の **[ネットワークの種類]** として、GPRS (General Packet Radio Service) または CSD (Circuit-Switched Data) を選択します。
- 3 Radius サーバの **[IP アドレス]** をドット区切りの形式で入力します。

- 4 Radius サーバの [認証ポート番号] と [アカウントポート番号] を入力します。
- 5 Radius のアカウント認証で使用する [秘密鍵] の値を入力します。
- 6 [接続タイムアウト] に値を入力します。
- 7 [再送の再試行回数] を指定します。
- 8 [サーバから返された属性をパラメータに保存する] を有効にするかどうかを指定します。
- 9 [OK] をクリックして設定を適用し、ダイアログ・ボックスを閉じます。

MMS の実行環境の設定

スクリプトを実行する前に、実行環境の設定を行って、スクリプトが実ユーザを正確にエミュレートすることができます。

次の項では、MMS（マルチメディア・メッセージング・サービス）Vuser に固有の実行環境の設定について説明します。これらの実行環境の設定により、サーバとプロトコルの設定を行うことができます。



次のオプションを設定できます。

- ▶ **MMSC URL** : MMSC（マルチメディア・メッセージング・センタ）サーバの URL です。
- ▶ **MMS バージョン** : スクリプトで使用される MMS プロトコルのバージョンです。

- ▶ **タイムアウト** : サーバが着信メッセージを待機する時間です。標準時間は 60 秒です。
- ▶ **SMSC IP** : SMPP を介して MMS 通知を送信するのに使用される SMSC サーバの IP アドレスです。
- ▶ **SMSC ポート** : SMPP を介して MMS 通知を送信するのに使用される SMSC サーバの IP ポートです。
- ▶ **自動 WAP 接続** : WAP ゲートウェイに対する接続 / 接続解除を行うタイミングを定義します。この設定は、WAP ゲートウェイを使用する場合にのみ関係しません。次の値を指定できます。
 - ▶ **Per Iteration** : 各反復の始まりに接続し、各反復の終わりに接続を解除します (標準設定)。
 - ▶ **Per Send or Receive** : 各メッセージの始まりに接続し、各メッセージの終わりに接続を解除します。
 - ▶ **None** : 自動 WAP 接続を使用しません。
- ▶ **標準設定の送信元アドレス** : Sender ヘッダで送信される標準設定のアドレスです。標準設定値は +999999 です。

MMS サーバとプロトコルの設定を行うには、次の手順を実行します。

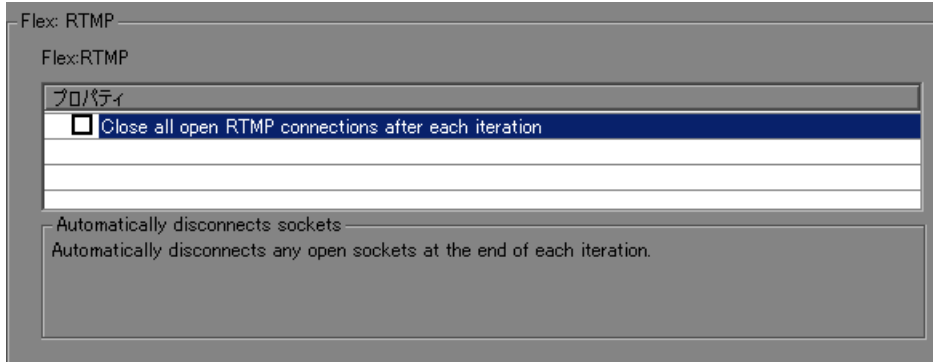


- 1 [実行環境設定] ダイアログ・ボックスを開きます。[仮想ユーザ] > [実行環境の設定] を選択するか、VuGen ツールバーの [実行環境の設定] ボタンをクリックします。
- 2 実行環境の設定ツリーから [MMS: サーバとプロトコル] ノードを選択します。
- 3 前述の説明に従って、必要な値を選択します。
- 4 実行環境の設定ツリーから [一般: その他] を選択します。
- 5 [マルチスレッド] の下の [仮想ユーザをプロセスとして実行する] を選択します。
- 6 [OK] をクリックして設定を適用し、スクリプトを実行します。

Flex の実行環境の設定

スクリプトを実行する前に、実行環境の設定を行って、スクリプトが実ユーザを正確にエミュレートするようにできます。

次の項では、Flex Vuser 専用の実行環境の設定について説明します。



次のオプションを設定できます。

- ▶ **[Close all open RTMP connections after each iteration]** : 各反復の終了時に、開いている RTMP 接続を自動的に切断します。

第 5 部

上級ユーザのための情報

第 25 章

Visual Studio による Vuser スクリプトの作成

Visual C または Visual Basic を使用して、Visual Studio により Vuser スクリプトのテンプレートを作成できます。作成したテンプレートは、C または Visual Basic プログラムと同じようにコンパイルします。

本章の内容

- ▶ Visual Studio による Vuser スクリプトの作成について (505 ページ)
- ▶ Visual C による Vuser スクリプトの作成 (507 ページ)
- ▶ Visual Basic による Vuser スクリプトの作成 (508 ページ)
- ▶ 実行環境の設定とパラメータの設定 (510 ページ)

Visual Studio による Vuser スクリプトの作成について

Vuser スクリプトを作成するには、VuGen を使用したり Visual Studio などの開発環境を使用したりする方法があります。

VuGen

VuGen の記録機能を使用したり、VuGen エディタを使用して手作業でプログラミングしたりすることにより、Windows や UNIX プラットフォームで実行する Vuser スクリプトを作成できます。Windows 環境で作成したスクリプトは Windows と UNIX 環境の両方で実行できます。記録は UNIX 環境で行うことはできません。

Visual Studio

Visual Studio を使えば、Vuser スクリプトを Visual Basic、C、C++ でプログラミングすることができます。プログラムはダイナミック・リンク・ライブラリ (dll) としてコンパイルします。

本章では、Visual C と Visual Basic の開発環境でプログラミングによって Vuser スクリプトを作成する方法について説明します。これらの環境では、開発アプリケーションに Vuser API のライブラリをインポートして Vuser スクリプトを作成します。

また、VuGen エディタ上でも、アプリケーションのライブラリやクラスを組み込んで Vuser スクリプトのプログラミングをすることもできます。VuGen では、C、Java、Visual Basic、VBScript または JavaScript のプログラミングが行えます。詳細については、『第 2 巻 – プロトコル』の「VuGen エディタでのスクリプトのプログラミング」を参照してください。

プログラミングで Vuser スクリプトを作成する場合、VuGen テンプレートを大規模な Vuser スクリプトの原形として使用できます。テンプレートで提供するものは、次のとおりです。

- ▶ 正しいプログラム構造
- ▶ Vuser API 呼び出し
- ▶ ダイナミック・リンク・ライブラリを作成するためのソース・コードとメイクファイル

テンプレートに基づいて基本となる Vuser スクリプトを作成したら、実行時の情報と統計値を提供できるようにスクリプトを拡張します。詳細については、第 6 章「Vuser スクリプトの拡張」を参照してください。

Vuser スクリプトで使用可能なオンラインの C 言語の共通関数リファレンスについては、『**Online Function Reference**』（英語版）（[ヘルプ] > [関数リファレンス]）を参照してください。

Visual C による Vuser スクリプトの作成

Visual C で Vuser スクリプトを作成する場合は、バージョン 6.0 以上の Visual C を使用してください。

Visual C を使用して Vuser スクリプトを作成するには、次の手順を実行します。

- 1 Visual C で、ダイナミック・リンク・ライブラリ (dll) を作成する新規プロジェクトを開きます。[ファイル] > [新規作成] を選択し、[プロジェクト] タブをクリックします。
- 2 [ウィザード] で [空の DLL プロジェクト] を選択します。
- 3 プロジェクトに次のファイルを追加します。
 - ▶ 新規 **cpp** ファイルに、**init**, **run**, **end** の 3 つの関数をエクスポートしたもの (関数名は変更できます)。
 - ▶ ライブラリ・ファイル **lrn50.lib** (< LoadRunner のインストール・ディレクトリ > \lib にあります)。
- 4 プロジェクトの設定で次の変更を行います。
 - ▶ [C/C++] タブを選択して、[コード生成 (カテゴリ)] > [使用するランタイム ライブラリ (リスト)] を選択し、これを [マルチスレッド (DLL)] に変更します。
 - ▶ [C/C++] タブを選択して、[プリプロセッサ (カテゴリ)] > [プリプロセッサの定義 (編集フィールド)] を選択し、_DEBUG を削除します。
- 5 これで、クライアント・アプリケーションからコードを追加したり、通常どおりプログラミングします。
- 6 Vuser API 関数を使ってスクリプトを拡張します。たとえば、メッセージを発行するには **lr_output_message**, トランザクションの開始を示すには **lr_start_transaction** を使用します。詳細については、『**Online Function Reference**』(英語版) ([ヘルプ] > [関数リファレンス]) の「一般関数」を参照してください。
- 7 プロジェクトをビルドします。DLL として出力されます。
- 8 DLL と同じ名前のディレクトリを作成し、作成した DLL をこのディレクトリにコピーします。
- 9 **Template** ディレクトリの **lrvuser.usr** ファイルを開き、USR ファイル・キー **BinVuser** を次のようにその DLL 名で上書きします。BinVuser=<DLL_name>

次の例は、`lr_output_message` 関数でどのセクションが実行されているかを示すメッセージを発行するものです。`lr_eval_string` 関数は、ユーザ名を取得します。次の例を使用する場合は、Vuser API のインクルード・ファイル `lrn.h` へのパスが正しいことを確認してください。

```
#include "c:\lrn_5\include\lrn.h"

extern "C" {
int __declspec(dllexport) Init (void *p)
{
    lr_output_message("in init");
    return 0;
}

int __declspec(dllexport) Run (void *p)
{
    const char *str = lr_eval_string("<name>");
    lr_output_message("in run and parameter is %s", str);
    return 0;
}

int __declspec(dllexport) End (void *p)
{
    lr_output_message("in end");
    return 0;
}
} //extern C end
```

Visual Basic による Vuser スクリプトの作成

Visual Basic を使用して Vuser を作成するには、次の手順を実行します。

- 1 Microsoft Visual Basic で新規プロジェクトを作成します。[ファイル] > [新しいプロジェクト] を選択します。
- 2 [LoadRunner Virtual User] を選択します。新しいプロジェクトが 1 つのクラスと Vuser 用のテンプレートで作成されます。
- 3 プログラミングを始める前に、プロジェクトを保存しておきます。[ファイル] > [プロジェクトの上書き保存] を選択します。

- 4 オブジェクト・ブラウザ ([表示] メニュー) を開きます。「LoadRunner Vuser」ライブラリを選択し、Vuser クラス・モジュールをダブルクリックしてテンプレートを開きます。テンプレートには、Vuser_Init、Vuser_Run、Vuser_End の 3 つのセクションが含まれています。

```
Option Explicit
```

```
Implements Vuser
```

```
Private Sub Vuser_Init()
```

```
' ここで Vuser の初期化コードを実装する
```

```
End Sub
```

```
Private Sub Vuser_Run()
```

```
' ここで Vuser の主なアクションを示すコードを実装する
```

```
End Sub
```

```
Private Sub Vuser_End()
```

```
' ここで Vuser の終了コードを実装する
```

```
End Sub
```

- 5 これで、クライアント・アプリケーションからコードを追加したり、通常どおりプログラミングします。
- 6 オブジェクト・ブラウザを使って、トランザクション、思考遅延時間、ランデブー、メッセージなど、使用する VuGen の要素をコードに追加します。
- 7 実行環境の設定とパラメータで、プログラムを拡張します。詳細については、510 ページ「実行環境の設定とパラメータの設定」を参照してください。
- 8 Vuser スクリプトをビルドします。[ファイル] > [project_name.dll の作成] を選択します。

プロジェクトは、Vuser スクリプト形式 (.usr) で保存されます。スクリプトは、プロジェクトと同じディレクトリに作成されます。

実行環境の設定とパラメータの設定

スクリプト用に DLL を作成したら、スクリプト・ファイル (.usr) を作成して、その設定を行います。VuGen で提供される **lrbin.bat** ユーティリティを使って、パラメータを定義し、Visual C や Visual Basic を使って作成したスクリプトの実行環境の設定を行います。このユーティリティは、製品のインストール・ディレクトリにある **bin** ディレクトリにあります。

実行環境の設定を行い、スクリプトをパラメータ化するには、次の手順を実行します。

- 1 製品の **bin** ディレクトリにある **lrbin.bat** をダブルクリックします。[Standalone Vuser Configuration] ダイアログ・ボックスが開きます。



- 2 [ファイル] > [新しいプロジェクト] を選択します。usr ファイルとなるスクリプトの名前を指定します。このスクリプト名は、DLL を保存したディレクトリの名前と同じでなくてはなりません。
- 3 [Vuser] > [Advanced] を選択し、[Advanced] ダイアログ・ボックスに DLL の名前を入力します。
- 4 [Vuser] > [Run-time Settings] を選択して、実行環境の設定を定義します。[実行環境設定] ダイアログ・ボックスは、VuGen のインターフェースに表示されるものと同じです。詳細については、第 22 章「実行環境の設定」を参照してください。
- 5 [Vuser] > [Parameter List] を選択して、スクリプトにパラメータを定義します。[パラメータ] ダイアログ・ボックスは、VuGen のインターフェースに表示されるものと同じです。詳細については、第 13 章「VuGen パラメータを使った作業」を参照してください。

スクリプトをスタンドアロン・モードで実行して、テストします。[仮想ユーザ] > [Run Vuser] を選択します。スクリプトの実行中は、Vuser の実行ウィンドウが現れます。

- 6 [File] > [Exit] を選択して、設定ユーティリティを閉じます。

第 26 章

XML API プログラミング

完全な XML 構造をサポートする Vuser スクリプトを作成できます。VuGen は、XML データの検索および操作を可能にする関数を提供します。

本章の内容

- ▶ XML API プログラミングについて (512 ページ)
- ▶ XML ドキュメントについて (513 ページ)
- ▶ XML 関数の使用方法 (514 ページ)
- ▶ XML 関数のパラメータの指定 (517 ページ)
- ▶ XML 属性での作業 (519 ページ)
- ▶ XML スクリプトの作成 (520 ページ)
- ▶ 記録されたセッションの拡張 (522 ページ)
- ▶ 結果パラメータの使用 (527 ページ)

以降の情報は、主に **Web**、**Web サービス**、およびワイヤレス **Vuser** スクリプトを対象とします。

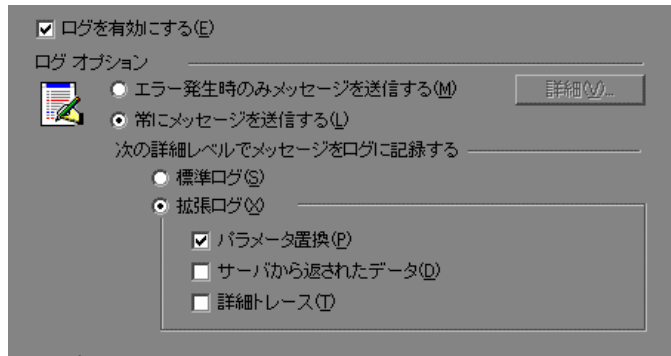
XML API プログラミングについて

VuGen の XML サポート機能により、テスト実行中に XML コードを動的に使用し、値を取得できます。効果的な XML スクリプトを作成するには、次の手順を実行します。

- ▶ 使用するプロトコル（通常 Web、Web サービス、またはワイヤレス）でスクリプトを記録します。
- ▶ XML の構造をスクリプトにコピーします。
- ▶ 動的データと XML 要素の値を取得するには、LR API の XML 関数を追加します。

LR API は、XML Path 言語である XPath を使用して、XML ドキュメントのテキストを操作します。

[実行環境設定] を使用することによって、[実行ログ] ウィンドウに XML 要素の出力値が表示されるようになります。VuGen には、行番号、一致した件数、値が表示されます。値を表示するには、パラメータ置換を有効にする必要があります。[実行環境設定] ダイアログ・ボックスで [一般: ログ] ノードを開き、[拡張ログ] を選択して、[パラメータ置換] を選択します。詳細については、第 22 章「実行環境の設定」を参照してください。



Vuser API の XML 関数はすべて、正常に検索された一致の件数か、失敗を表す 0 を返します。

XML ドキュメントについて

XML (eXtensible Markup Language) は、ユーザが独自にタグを定義できるマークアップ言語です。これらのタグを使用することによって、タグに挟まれたテキストに意味を与えます。これは、標準の HTML タグ (H1, P, DIV など) とは対照的です。標準の HTML タグはカスタマイズできず、テキストの内容を指定できません。

XML ドキュメントは、多くのノードと分岐を持つツリーで構成されます。XML ドキュメントを構成する部品を表すために、タグ、要素、属性という 3 つの用語がよく使用されます。次の例は、これらの用語を表しています。

```
<acme_org>
  <accounts_dept>
    <employee type='PT'>
      <name>John Smith</name>
      <cubicle>227</cubicle>
      <extension>2145</extension>
    </employee>
  </accounts_dept>
  <engineering_dept>
    <employee type='PT'>
      <name>Sue Jones</name>
      <extension>2375</extension>
    </employee>
  </engineering_dept>
</acme_org>
```

タグとは、左向きと右向きの山括弧の間にあるテキストです。<acme_org>, <employee>, <name> がタグの例です。タグには、開始タグ (<name> など) と終了タグ (</name> など) があります。前述の XML コードの抜粋は、John Smith と Sue Jones という 2 人の従業員がいる Acme という会社を表しています。

要素とは、開始タグおよび終了タグ、そしてそれらのタグに挟まれたすべての内容です。前述の例では、<employee> 要素には、<name>, <cubicle>, <extension> という 3 つの子要素が含まれています。

属性とは、要素の開始タグ内にある、名前と値の組み合わせです。この例では、**type='PT'** が <employee> 要素の属性です。

前述の例では、**name** というタグは **employee** の要素です。それぞれの要素には値があります。たとえば、**name** 要素の値は、「John Smith」という文字列です。

XML 関数の使用方法

以降では、XML ツリーのデータの使用方法について例を示します。いくつかの関数では情報を取得できます。またいくつかの関数では XML ツリーに情報を書き込めます。これらの例では、Acme という会社に所属する複数の従業員の名前と内線番号が入っている次の XML ツリーを使用します。

```
<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>
```

XML ツリーからの情報の読み取り

XML ツリーから情報を読み取る関数は次のとおりです。

- | | |
|--------------------------|--------------------------------|
| lr_xml_extract | XML 文字列から XML 文字列フラグメントを抽出します。 |
| lr_xml_find | XML 文字列に対するクエリを実行します。 |
| lr_xml_get_values | クエリで検出された XML 要素の値を取得します。 |

クエリを使用して特定の値を取得するには、パス形式で親ノードと子ノードのタグを指定します。

たとえば、Accounting 部門の従業員の名前を取得するには、次の文字列を使用します。

```
lr_xml_get_values("XML={XML_Input_Param}",
  "ValueParam=OutputParam",
  "Query=/acme_org/accounting_dept/employee/name",
  LAST);
```

拡張ログ機能が有効な場合、[実行ログ] ウィンドウには、この関数の出力が表示されます。

```
Output:Action.c(20): "lr_xml_get_values" was successful, 1 match processed
Action.c(25): Query result = John Smith
```

XML 構造への書き込み

XML ツリーに値を書き込む関数は次のとおりです。

| | |
|--------------------------|---|
| lr_xml_delete | XML 文字列からフラグメントを削除します。 |
| lr_xml_insert | 新しい XML フラグメントを XML 文字列に挿入します。 |
| lr_xml_replace | XML 文字列のフラグメントを置き換えます。 |
| lr_xml_set_values | クエリで検出された XML 要素の値を設定します。 |
| lr_xml_transform | XML データに XSL (Extensible Stylesheet Language) 変換を適用します。 |

最もよく使用される**書き込み**関数は **lr_xml_set_values** です。この関数は、XML 文字列の指定された要素の値を設定します。次の例では、**lr_xml_set_values** を使用して、XML 文字列の 2 つの **employee** 要素の内線番号を変更しています。

まず、**XML_Input_Param** というパラメータに XML 文字列を保存します。2 つの値を検索して置き換えます。そこで、**ExtensionParam_1** と **ExtensionParam_2** という新しい 2 つのパラメータを用意し、それらの値を新しい 2 つの内線番号 1111 および 2222 に設定します。

lr_xml_set_values には、ExtensionParam_1 および ExtensionParam_2 の値を受け取る「ValueName=ExtensionParam」という引数が含まれています。2 人の従業員の現在の内線番号が、これらのパラメータの値、1111 および 2222 で置き換えられます。その後 OutputParam の値が評価され、新しい内線番号に確かに置き換えられたことが証明されます。

```

Action() {

    int i, NumOfValues;
    char buf[64];

    lr_save_string(xml_input, "XML_Input_Param"); // 入力をパラメータとして保存する
    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++) { /* MultiParam の複数の値を出力する */

        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

```

XML 関数のパラメータの指定

ほとんどの XML API 関数では、**XML 要素**と**クエリ**を指定する必要があります。また、すべての結果を取得するか、それとも 1 つの結果を取得するか指定できます。

XML 要素の定義

検索する XML 要素を定義するには、XML 要素をそのまま文字列として指定するか、XML 要素が含まれるパラメータを指定します。次の例では、XML 入力文字列をそのまま文字列として定義する場合を示します。

```
"XML=<employee>JohnSmith</employee>"
```

また、**XML** 文字列は、XML データを含むパラメータでもかまいません。たとえば以下のようになります。

```
"XML={EmployeeNameParam}"
```

XML ツリーのクエリ

XML タグに含まれている値（従業員の内線番号など）を検索するとします。必要とする値に対するクエリを作成します。クエリには、要素の場所と、取得または設定する要素を指定します。指定したパスにより、検索範囲が特定のタグに限定されます。また、ルートの下のすべてのノードで特定の種類の要素をすべて検索することもできます。

特定のパスを指定するには、「"Query=/ < XML フル・パス名 > / < 要素名 > 」を指定します。

同じ名前前の要素をすべてのノードの下で検索するには、「"Query="// < 要素名 > 」を指定します。

VuGen における XML 関数の実装では、クエリの範囲は XML ツリー全体です。ツリー情報は、**xml** 引数の値として Vuser API 関数に送られます。

クエリの複数の検索

XML 要素に対してクエリを実行すると、標準では最初に一致したものだけが返されます。クエリで複数の値を取得するには、関数内で "SelectAll=yes" 属性を指定します。VuGen は、複数のパラメータを表すために `_<連番>` という形式のサフィックスを追加します。たとえば、**EmployeeName** という名前のパラメータを定義した場合は、**EmployeeName_1**、**EmployeeName_2**、**EmployeeName_3** などが作成されます。

```
lr_xml_set_values("XML={XML_Input_Param}",
"ResultParam=NewXmlParam", "ValueParam=ExtensionParam",
"SelectAll=yes", "Query=//extension", LAST);
```

パラメータに書き込む関数を使用すれば、書き込んだ後にパラメータの値を評価できます。たとえば次のコードでは、クエリによる複数の検索結果を取得して出力しています。

```
NumOfValues = lr_xml_get_values("Xml={XmlParam}", "Query=//name",
"SelectAll=yes", "ValueParam=EmployeeName", LAST);
```

パラメータから値を読み取る関数の場合、パラメータの値は事前に定義しておく必要があります。また、パラメータは、`<パラメータ名>_<連番>` という形式（たとえば **Param_1**、**Param_2**、**Param_3** など）を使用する必要もあります。このようなパラメータの集合をパラメータ・セットとも言います。

次の例では、`lr_xml_set_values` はパラメータ・セットから値を読み取り、その値を XPath クエリで使用しています。従業員の内線番号を表すパラメータ・セットには、`ExtensionParam` という名前が付いています。このパラメータ・セットには、`ExtensionParam_1` および `ExtensionParam_2` という 2 つのメンバがあります。`lr_xml_set_values` 関数は XML 入力文字列を検索し、最初に一致した値を 1111 に、2 番目に一致した値を 2222 に設定します。

```
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 属性での作業

VuGen では属性がサポートされています。要素を操作する場合と同じように、簡単な表現を使用して XML の要素とノードの属性を操作できます。特定の属性、または特定の値を持つ属性を変更することもできます。

次の例では、**lr_xml_delete** を使って、name 属性を持っている最初の cubicle 要素を削除しています。

```
lr_xml_delete("Xml={ParamXml}",
              "Query="//cubicle/@name",
              "ResultParam=Result",
              LAST
            );
```

次の例では、**lr_xml_delete** を使って、Paul という値の name 属性を持っている最初の cubicle 要素を削除しています。

```
lr_xml_delete("Xml={ParamXml}",
              "Query="//cubicle/@name="Paul",
              "ResultParam=Result",
              LAST
            );
```

XML スクリプトの作成

最初に、使用するプロトコルで新しいスクリプトを作成します。そのプロトコルでセッションを記録することも、また記録せずにスクリプト全体をプログラミングすることもできます。次のように、スクリプトの Actions セクションを作成します。

- ▶ XML 入力の宣言
- ▶ Actions セクション

XML 入力セクションには、入力変数として使用する XML ツリーを含めません。XML ツリーを `char` 型の変数として定義します。たとえば以下のようになります。

```
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>";
```


Action セクションには、変数の評価、および要素の値に対するクエリを含めません。次の例では、**lr_save_string** を使って XML 入力文字列を評価しています。入力変数を対象に、従業員名と内線番号を検索しています。

```
Action() {  
    /* 入力をパラメータとして保存する */  
    lr_save_string(xml_input, "XML_Input_Param");  
  
    /* クエリ 1 - 指定された要素から従業員名を取得する */  
    lr_xml_get_values("XML={XML_Input_Param}",  
        "ValueParam=OutputParam",  
        "Query=/acme_org/employee/name", LAST);  
  
    /* クエリ 2 - ルート以下のすべてのパスで内線番号を取得する */  
    lr_xml_get_values("XML={XML_Input_Param}",  
        "ValueParam=OutputParam",  
        "Query=//extension", LAST);  
  
    return 0;  
}
```

記録されたセッションの拡張

セッションを記録し、必要な XML 関数と Vuser API 関数を手作業で追加することによって、XML スクリプトを作成できます。

次の例では、記録されたセッションを Vuser API 関数を使って拡張する方法を示します。記録された関数は、太字で示した **web_submit_data** だけです。

最初のセクションには、SOAP メッセージを表す変数 SOAPTemplate が XML 入力宣言として含まれています。

```
#include "as_web.h"

// SOAP メッセージ
const char*
pSoapTemplate=
"<soap:Envelope xmlns:soap=¥\"http://schemas.xmlsoap.org/soap/envelope/¥\">"
  "<soap:Body>"
    "<SendMail xmlns=¥\"urn:EmailPortTypeInft-IEmailService¥\"/>"
  "</soap:Body>"
"</soap:Envelope>";
```

次のセクションは、ユーザのアクションを表しています。

```
Action1()
{
    // 応答の本体を取得する
    web_reg_save_param("ParamXml", "LB=", "RB=", "Search=body", LAST);

    // HTTP GET で天気をフェッチする
    web_submit_data("GetWeather",
        "Action=http://glkev.net.innerhost.com/glkev_ws/
        WeatherFetcher.aspx/GetWeather",
        "Method=GET",
        "EncType=",
        "RecContentType=text/xml",
        "Referer=http://glkev.net.innerhost.com
        /glkev_ws/WeatherFetcher.aspx?op=GetWeather",
        "Snapshot=t2.inf",
        "Mode=HTTP",
        ITEMDATA,
        "Name=zipCode", "Value=10010", ENDITEM,
        LAST);

    // City の値を取得する
    lr_xml_get_values("Xml={ParamXml}",
        "Query=City",
        "ValueParam=ParamCity",
        LAST
    );

    lr_output_message(lr_eval_string("***** City = {ParamCity} *****"));

    // State の値を取得する
    lr_xml_get_values("Xml={ParamXml}",
        "Query=State",
        "ValueParam=ParamState",
        LAST
    );

    lr_output_message(lr_eval_string("***** State = {ParamState} *****"));
}
```

```

// テンプレート
lr_xml_get_values_ex("Xml={ParamXml}") を使用して、複数の値を一度に取得する。
"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} *****"));

// 人に読める形式で予報を生成する
lr_save_string(lr_eval_string("***** Weather Forecast for {ParamCity}, {ParamState} *****"
                              "¥tTime: {ParamTime}¥r¥n"
                              "¥tTemperature: {ParamTemp} deg. Fahrenheit¥r¥n"
                              "¥tHumidity: {ParamHumid}¥r¥n"
                              "¥t{ParamCond} conditions expected¥r¥n"
                              "¥r¥n"),
              "ParamForecast"
);

// SOAP テンプレートをパラメータに保存する
lr_save_string(pSoapTemplate, "ParamSoap");

```

```

// 要求の本体を SOAP テンプレートに挿入する
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:EmailPortTypeInft-IEmailService¥"/>"
// "<FromAddress>taurus@merc-int.com</FromAddress>"
// "<ToAddress>support@merc-int.com</ToAddress>"
// "<ASubject>Weather Forecast</ASubject>"
// "<MsgBody/>"
// "</SendMail>"
// "</soap:Body>"
// "</soap:Envelope>";
//

// 実際の予報のテキストを挿入する
lr_xml_set_values("Xml={ParamRequest}",
    "ResultParam=ParamRequest",
    "Query=Body/SendMail/MsgBody",
    "ValueParam=ParamForecast",
    LAST);

```

```
// SOAP 用のヘッダを追加する
web_add_header("SOAPAction", "urn:EmailPortTypeInft-IEmailService");

// 応答の本体を取得する
web_reg_save_param("ParamXml", "LB=", "RB=", "Search=body", LAST);

// SOAP 要求を使用して予報を受信者に送信する
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);

// メールが送信されたことを確認する
lr_xml_find("Xml={ParamXml}",
    "Query=Body/SendMailResponse/return",
    "Value=0",
    LAST
);

return 0;
}
```

結果パラメータの使用

`lr_xml` 関数の中には、**ResultParam** のように結果パラメータを返すものがあります。このパラメータには、関数が実行された後の結果の XML データが含まれます。結果パラメータは、[パラメータの選択または作成] ダイアログ・ボックスのパラメータ・リストから使用できます。

たとえば、`lr_xml_insert` 関数の場合、**ResultParam** には新しい XML フラグメントの挿入の結果である完全 XML データが含まれます。

結果パラメータを Web Service 呼び出しのような他の XML 関連関数への入力として使用できます。再生時、VuGen によって結果パラメータの値がキャプチャされます。この値は、後の手順で入力引数として使用できます。

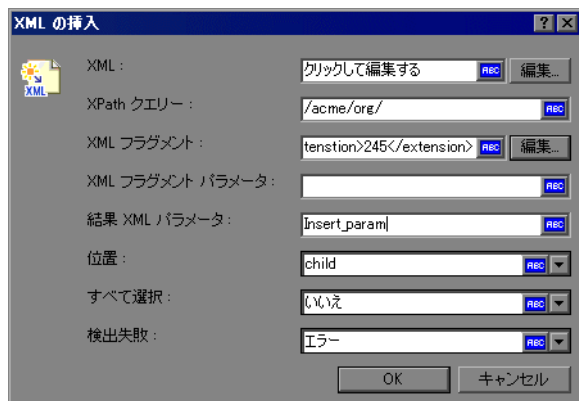
結果パラメータをサポートする関数には、`lr_xml_insert`、`lr_xml_transform`、`lr_xml_replace`、`lr_xml_delete`、and `lr_xml_set_values` があります。

次の関数は、値を `resultParam` 以外のパラメータに保存します。

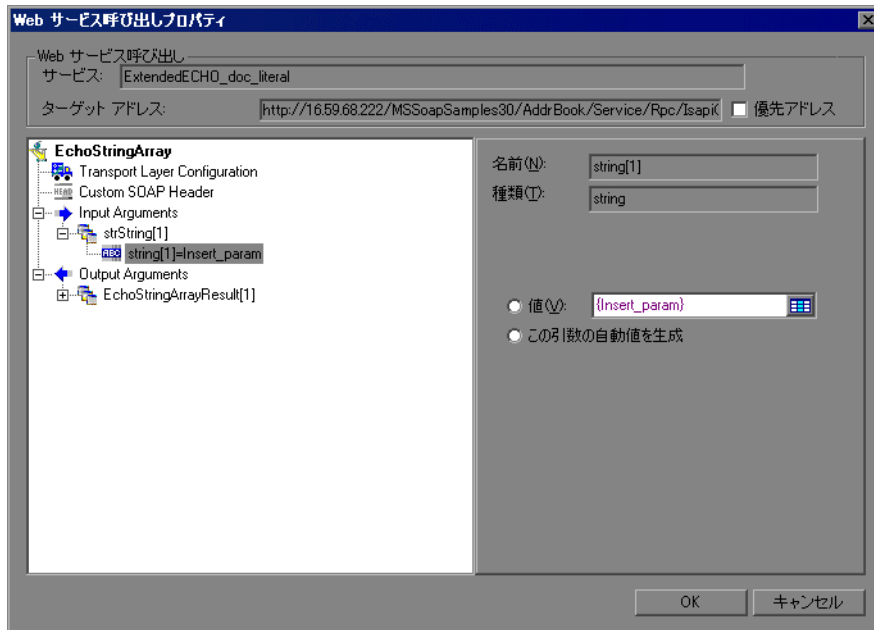
`lr_xml_get_values` は値を `ValueParam` に、`lr_xml_extract` は値を `XMLFragmentParam` に保存します。これらの値は、パラメータ置換にも使用されます。

結果パラメータを入力に使用するには、次の手順を実行します。

- 1 ツリー・ビューで、XML ステップをダブルクリックして、そのプロパティを表示します。
- 2 [結果 XML パラメータ] ボックスで、**結果 XML パラメータ** の名前（または `ValueParam` および `XMLFragmentParam`）を指定します。



3 パラメータ名を入力引数として参照します。



詳細については、99 ページ「入力引数値」を参照してください。

第 27 章

VuGen のデバッグのヒント

以下の統合および設定のヒントを使用して、エラーのない Vuser スクリプトを作成できます。

本章の内容

- ▶ 一般的なデバッグのヒント (529 ページ)
- ▶ C 関数を使用した追跡 (530 ページ)
- ▶ 付加的な C 言語のキーワードの追加 (530 ページ)
- ▶ 再生出力の検証 (531 ページ)
- ▶ データベース・アプリケーションのデバッグ (531 ページ)
- ▶ Oracle Applications を使った作業 (533 ページ)
- ▶ Oracle 2-Tier Vuser での一般的な問題の解決方法 (534 ページ)
- ▶ 2-tier データベースのスクリプト作成のヒント (539 ページ)
- ▶ PeopleSoft-Tuxedo スクリプトの実行 (548 ページ)

一般的なデバッグのヒント

VuGen は、通常のテキスト・エディタとして使用できます。VuGen で、任意のテキスト・ファイルを開いて編集できます。再生中、下の出力ウィンドウにエラー・メッセージが表示されている場合は、その上でダブルクリックすると、VuGen は問題の原因となっているテキスト行にカーソルを移動します。また、エラー・コードにカーソルを置いて F1 キーを押すと、オンライン・ヘルプのエラー・コードの説明が表示されます。

C 関数を使用した追跡

C インタプリタの追跡オプション（バージョン 230 以上）を使用して、Vuser スクリプトをデバッグできます。`ci_set_debug` ステートメントを使って、スクリプト内の特定の位置で、追跡とデバッグのオン/オフを切り替えることができます。

```
ci_set_debug(ci_this_context, int debug, int trace);
```

たとえば、スクリプトに次のステートメントを追加できます。

```
ci_set_debug(ci_this_context, 1, 1) /* 追跡とデバッグをオンにする */
ci_set_debug(ci_this_context, 0, 0) /* 追跡とデバッグをオフにする */
```

付加的な C 言語のキーワードの追加

VuGen で C スクリプトを実行すると、VuGen のパーサは組み込み C インタプリタを使ってスクリプト内の関数を解析します。標準パーサのライブラリに含まれていないキーワードを追加できます。標準設定では、インストール中に **size_t** や **DWORD** といった一般的な C++ キーワードが追加されます。リストを編集して、環境に合ったキーワードを追加します。

キーワードを追加するには、次の手順を実行します。

- 1 `vugen_extra_keywords.ini` ファイルを開きます。このファイルはお使いのコンピュータの <Windows> または <Windows>\¥¥System ディレクトリにあります。
- 2 `EXTRA_KEYWORDS_C` セクションに、C インタプリタ用のキーワードを追加します。

このファイルの形式は次のとおりです。

```
[EXTRA_KEYWORDS_C]
FILE=
size_t=
WORD=
DWORD=
LPCSTR=
```

再生出力の検証

再生出力を見ます (VuGen から、あるいは VuGen ドライバの出力を表示する **output.txt** ファイルから)。また、より詳細な再生出力を得るために、VuGen の実行環境の設定オプションで、ログの記録を拡充することもできます。

データベース・アプリケーションのデバッグ

以降のヒントは、データベース・アプリケーション (Oracle, ODBC, および Ctlib) に適用されます。

- ▶ デバッグ情報の生成
- ▶ コンパイラ情報の検証
- ▶ コード生成情報
- ▶ 前処理とコンパイルの情報

デバッグ情報の生成

注：本項で説明する情報の大部分は、VuGen のユーザ・インタフェースを使って表示するように設定できます。

VuGen には、インスペクタ「エンジン」が含まれています。

WINDOWS_DIR\vugen.ini を次のように編集して、VuGen レコーダが「インスペクタ」出力を作成するようにできます。

```
[LogMode]
EnableAscii=ASCII_LOG_ON
```

このオプションが有効になっている場合、VuGen は記録終了時に Data ディレクトリに **vuser.asc** ファイルを作成します。このオプションは、デバッグの目的に限って使うべきものです。出力ファイルが非常に大きくなり (数 MB)、マシンのパフォーマンスとディスク領域に深刻な影響を及ぼす可能性があるからです。

ODBC ベースのアプリケーションの場合は、[ODBC データ ソース アドミニストレータ] (Windows の [コントロール パネル] にあります) で同様の追跡出力を得るように設定できます。ODBC オプションを開いて、[Trace ODBC calls] を ON にします。同様に、ODBC Developer Kit には呼び出しの追跡を行うスパイ・ユーティリティが用意されています。

詳細なデバッグ情報を有効にするには、¥WINDOWS_DIR¥vugen.ini ファイルに次のセクションを追加します。

```
[INSPECTOR]
TRACE_LEVEL=3
TRACE_FILENAME=c:¥tmp¥sqltrace.txt
```

sqltrace.txt ファイルには、記録中に行われたフック呼び出しについての有用な内部情報が含まれます。trace_level は 1 から 3 までであり、3 は最も詳細なデバッグ・レベルを表します。VuGen バージョン 5.02 以上では、ユーザ・インタフェースから追跡レベルを設定できます。

コンパイラ情報の検証

コード生成、前処理、コンパイルの各段階についての情報を表示して、エラーの原因を特定できます。

コード生成情報

Data ディレクトリ内の **vuser.log** ファイルを見ます。このファイルには、コード生成段階のログが含まれており、lrd 記録 (すなわちすべてのデータベース・プロトコル) が終わるたびに、自動的に作成されます。

次にログ・ファイルの例を示します。

```
lrd_init: OK
lrd_option: OK
lrd_option: OK
lrd_option: OK
Code generation successful
lrd_option: OK
lrd_end: OK
```

いずれかのメッセージが OK (成功) ではない場合は、コード生成中に問題が生じています。

前処理とコンパイルの情報

実行中、VuGen は前処理とコンパイル処理の両方についての情報を表示します。

Oracle Applications を使った作業

Oracle Applications は、2-tier (「ファット」クライアント) パッケージのアプリケーションで、35 ものさまざまなモジュール (Oracle Human Resources, Oracle Financials など) で構成されています。

Oracle Applications 用の Vuser の記録と再生のために、いくつか知っておくべきことがあります。

- ▶ 一般的なスクリプトには、何千ものイベント、バインド、およびアサインが含まれています。
- ▶ 一般的なスクリプトには、各ユーザ・セッションに多くの db 接続が含まれています。
- ▶ スクリプトには、ほとんどの場合関連したクエリが必要です。
- ▶ Oracle Applications のクライアントは 16 ビットのみです (Oracle Developer 2000 で開発されたもの)。つまり、デバッグに際して Oracle 32 ビット・クライアントがなければ、VuGen の Force 16-bit オプションを使う必要があります。

新しいウィンドウが作成されると、アプリケーションは、表示用にファイル・システムから .xpf ファイルを取得します。VuGen はクライアント/サーバ・レベルで記録を行うため、現在はこれを考慮に入れていません。したがって、パフォーマンスの測定はかなり不正確になります。多くの場合、パフォーマンスの問題はクライアントとファイル・サーバの間のボトルネックに関するものだからです。現在、この問題の解決に向け、検討中です。

Oracle 2-Tier Vuser での一般的な問題の解決方法

本項では、Oracle Vuser を扱っているときに生じるいくつかの一般的な問題と、その解決策を示します。

ORA-20001 と ORA-06512

`lrd_stmt` に次の PL/SQL ブロックが含まれている場合、再生中にエラー ORA-20001 と ORA-06512 が発生します。

```
fnd_signon.audit_responsibility(...)
```

このステートメントが再生中に失敗するのは、新しい接続をするたびに一意のサインオン番号が割り当てられるためです。

解決策

この問題を解決するには、サインオン番号を扱う新しい関連ツールを使用する必要があります。サインオン番号は、ステートメント内で 2 番目に割り当てられる値です。

関連候補の値を検索した後、失敗したステートメントの 2 番目の `lrd_assign_bind()` の値を強調表示します。「関連クエリ」ウィンドウでは、値が実際に記録されたステートメントと同じ順番で表示されない場合があります。

置換する値が含まれるグリッドは、次の PL/SQL ブロックの含まれる `lrd_stmt` の後に現れます。

```
fnd_signon.audit_user(...)
```

注：サインオン番号は接続ごとに一意なので、記録する新しい接続のそれぞれについて関連を行う必要があります。

解決策の例

次のステートメントは、2 番目の値「1498224」がそれぞれの新しい接続に一意のサインオン番号なので、再生で失敗します。

```
lrd_stmt(Csr6, "begin fnd_signon.audit_responsibility(:s,:l,:f,:a,:r,:t,:p)"
"; end;", -1, 1, 1, 0);
lrd_assign_bind(Csr6, "s", "D", &s_D216, 0, 0, 0);
lrd_assign_bind(Csr6, "l", "1498224", &l_D217, 0, 0, 0);
lrd_assign_bind(Csr6, "f", "1", &f_D218, 0, 0, 0);
lrd_assign_bind(Csr6, "a", "810", &a_D219, 0, 0, 0);
lrd_assign_bind(Csr6, "r", "20675", &r_D220, 0, 0, 0);
lrd_assign_bind(Csr6, "t", "Windows PC", &t_D221, 0, 0, 0);
lrd_assign_bind(Csr6, "p", "", &p_D222, 0, 0, 0);
lrd_exec(Csr6, 1, 0, 0, 0, 0);
```

このサインオン番号は、lrd_stmt で「fnd_signon.audit_user」を手がかりに見つけることができます。最初のプレースホルダの値「a」はそのままにしておきます。「a」への入力値はいつも「0」ですが、出力は要求された値です。

変更後のコード :

```
lrd_stmt(Csr4, "begin fnd_signon.audit_user(:a,:l,:u,:t,:n,:p,:s); end;", -1, 1, 1, 0);
lrd_assign_bind(Csr4, "a", "0", &a_D46, 0, 0, 0);
lrd_assign_bind(Csr4, "l", "D", &l_D47, 0, 0, 0);
lrd_assign_bind(Csr4, "u", "1001", &u_D48, 0, 0, 0);
lrd_assign_bind(Csr4, "t", "Windows PC", &t_D49, 0, 0, 0);
lrd_assign_bind(Csr4, "n", "OraUser", &n_D50, 0, 0, 0);
lrd_assign_bind(Csr4, "p", "", &p_D51, 0, 0, 0);
lrd_assign_bind(Csr4, "s", "14157", &s_D52, 0, 0, 0);
lrd_exec(Csr4, 1, 0, 0, 0, 0);

lrd_save_value(&a_D46, 0, 0, "saved_a_D46");
Grid0(17);

lrd_stmt(Csr6, "begin fnd_signon.audit_responsibility(:s,:l,:f,:a,:r,:t,:p)"
"; end;", -1, 1, 1, 0);
lrd_assign_bind(Csr6, "s", "D", &s_D216, 0, 0, 0);
lrd_assign_bind(Csr6, "l", "<saved_a_D46>", &l_D217, 0, 0, 0);
lrd_assign_bind(Csr6, "f", "1", &f_D218, 0, 0, 0);
lrd_assign_bind(Csr6, "a", "810", &a_D219, 0, 0, 0);
lrd_assign_bind(Csr6, "r", "20675", &r_D220, 0, 0, 0);
lrd_assign_bind(Csr6, "t", "Windows PC", &t_D221, 0, 0, 0);
lrd_assign_bind(Csr6, "p", "", &p_D222, 0, 0, 0);
lrd_exec(Csr6, 1, 0, 0, 0, 0);
```

大きな数値の処理

大きな数（NUMBER データ型）は、GRID と ASCII ファイルでは異なる形式で表示されます。この違いにより、関連用に保存する値の検索時に数値を特定するのが困難になります。

たとえば、グリッド内に 1000003 と表示される値がある場合、これは記録ログ（ASCII ファイル）では 1e+0006 と表示されます。

解決法

再生中にエラーが生じ、関連ツールが前回の結果の中で値を見つけられない場合は、この値が別の形式で表現されているものとしてグリッドの中を探します。

ORA-00960

このエラーは、一意でないカラム名に発生することがあります。たとえば以下のようにになります。

```
Ird_stmt(Csr9,"SELECT UOM_CODE, UOM_CODE, DESCRIPTION FROM "  
"MTL_UNITS_OF_MEASURE "  
"WHERE NVL(DISABLE_DATE, SYSDATE + 1) > "  
"SYSDATE ORDER BY UOM_CODE", -1, 1, 1, 0);
```

この場合、次のエラーが返されます。

```
"Ird_o.c/fjParse: "oparse" ERROR return-code=960, oerhms=ORA-00960:  
ambiguous column naming in select list".
```

解決法

少なくとも 1 つの一意でないカラムに別名を追加し、この別名を新しい一意の名前にして使うように、ステートメントを変更します。たとえば以下のようにになります。

```
Ird_stmt(Csr9,"SELECT UOM_CODE,UOM_CODE second, DESCRIPTION  
FROM"  
"MTL_UNITS_OF_MEASURE "  
"WHERE NVL(DISABLE_DATE, SYSDATE + 1) > "  
"SYSDATE ORDER BY UOM_CODE", -1, 1, 1, 0);
```

別の対処方法：Ird ステートメントから ORDER BY を削除します。

ORA-2002

このエラーは、開いていないカーソルを使用しようとしたときに発生します。これは、複数の反復でユーザを再生し、スクリプトの複数のセクションに記録したときに生じます。

具体的には、カーソルが `vuser_init` セクションで開き、`Actions` セクションで閉じた場合に、カーソルを使用しようとするると 2 回目の反復でこのエラーが生じます。これは、カーソルが閉じられており、再び開かれていないからです。

たとえば以下のようになります。`vuser_init` セクションに `lrd_open_cursor` があり、`Actions` セクションに `lrd_close_cursor` がある場合、このユーザの再生で複数回の反復を行うと、2 回目の反復でエラーが生じます。これは、開いていないカーソルを使用しようとしたためです（カーソルを最初の反復で閉じ、2 回目の反復では開いていないためです）。

解決法

この問題を最も簡単に解決するには、問題のカーソルの `lrd_close_cursor` または `lrd_close_connection` を `vuser_end` セクションに移動します。

データベース・プロトコル (lrd)

記録された非同期操作の再生はサポートされていません。

クライアント・バージョンの誤り

実行している Oracle クライアントのバージョンが正しくない場合、次のエラーが返されます。

```
"Error: lrd_open_connection: "olog" LDA/CDA return-code_019: unable to allocate memory in the user side"
```

解決法

製品の `bin` ディレクトリにあるライブラリ情報を `lrd.ini` ファイルで修正します。このファイルには記録および再生中にどのバージョンのデータベース・サポートがロードされるかを示す設定情報が含まれています。ファイルに各タイプのホストごとのセクションがあります。たとえば、`lrd.ini` ファイル内の HP/UX 上の Oracle セクションは次のようになります。

```
[ORACLE_HPUX]
;816=liblrdhpo816.sl
;81=liblrdhpo81.sl
;80=liblrdhpo80.sl
73=liblrdhpo73.sl
72=liblrdhpo72.sl
```

この設定により、クライアントが Oracle 8.1.6 を使用していれば LoadRunner ライブラリ liblrdhpo816.sl を、Oracle 8.1.5 を使用していれば、liblrdhpo81.sl を Vuser が使用するということがわかります。

UNIX 上の再生では、lrd.ini ファイルで使用するデータベースの正しいバージョンを表示するようにします。Oracle 8.1.5 を使用して HP/UX 用 Vuser を再生するとします。その場合、Oracle のそのほかのバージョンを示す行は行の先頭を「;(セミコロン)」でコメントアウトします。すると lrd.ini ファイルは次のようになります。

```
[ORACLE_HPUX]
;816=liblrdhpo816.sl
81=liblrdhpo81.sl
;80=liblrdhpo80.sl
73=liblrdhpo73.sl
72=liblrdhpo72.sl
```

アプリケーションが lrd.ini ファイルに記されている DLL を使用していない場合、Win32 も変更します。たとえば、PowerBuilder 6.5 は Oracle 8.0.5 を使用しますが、DLL は ora803.dll を使用し、ora805.dll は使用しません。その場合、ORACLE_WINNT の項目で 805 および 804 をコメントアウトするか、805 のセクション

```
805=lrdo32.dll+ora805.dll
```

を次のように変更します。

```
805=lrdo32.dll+ora803.dll
```

2-tier データベースのスクリプト作成のヒント

本項では 2-tier データベース・スクリプトのための解決策を示します。Siebel 固有の問題の解決策については、544 ページ「Siebel 固有のスクリプト作成のヒント」を参照してください。

Question 1 : アプリケーション自体は同じ値を扱えるのに、データ駆動のスクリプトで失敗する原因は？

Answer : データ値の後に続く空白が原因である可能性があります。GUI に直接入力するデータ値ではおそらく切り詰められているとしても、データ・ファイルから手作業で除去する必要があります。タブ区切りファイルでは、後続の空白が見えにくく、問題が発見しづらくなります。一般に、カンマ区切りファイルをお勧めします。Excel を使って問題がないかどうか確かめることができます。

Question 2 : 2 回目の反復でカーソル状態が無効という SQL エラーが発生する原因は？

Answer : `lrd_close_cursor` 関数が生成されていないか、スクリプトの **action** セクションではなく、**end** セクションに生成されている可能性があります。スクリプトを反復できるようにするには、カーソル・クローズ関数を追加するか、**end** セクションから移動する必要があります。

反復のたびに新しいカーソルを開くのは資源効率の点から好ましくありません。したがって、最初の反復の **actions** セクションで 1 回だけカーソルを開くことをお勧めします。それから [Iteration Number] タイプを使用して、反復番号を文字列として含む新しいパラメータを追加します。このパラメータに

IterationNum パラメータという名前を付けます。次に **actions** セクション内で新しいカーソルを開く呼び出し

```
lrd_open_cursor(&Csr1, Con1, 0);
```

を次のように置換します。

```
if (!strcmp(lr_eval_string("<IterationNum>"), "1"))
    lrd_open_cursor(&Csr1, Con1, 0);
```

Question 3 : `vdf.h` ファイルのデータ宣言が原因で VuGen で生成したコードのコンパイルができない場合の修正方法は？

Answer : 問題は、おそらく、VuGen でサポートされていない SQL のデータ型です。Microsoft SQL では多くの場合、`vdf.h` ファイル内の未定義エラー・メッセージを「DT_SZ」（ヌル終端文字列）に置き換えれば回避できます。これは実際のデータ型ではありませんが、VuGen でそのスクリプトを正常にコンパイルできます。カスタマ・サポートに問題を通知し、元のスクリプトをお送りください。

Question 4 : LRD Error 2048 の意味は？

Answer : VuGen が失敗します。記録時の割り当てよりも長い変数をバインドしようとしているためです。`vdf.h` ファイルで変数定義を拡大して、データベースから長い文字列を受け取れるようにします。このファイルで一意的な数値識別子を検索します。その定義と長さがわかります。長さは構成要素の内 3 つ目の要素です。この長さを増やせばスクリプトを正常に再生できるようになります。

たとえばスクリプト内に次の行があるものとします。

```
lrd_assign(&_2_D354, "<ROW_ID>", 0, 0, 0);
```

`vdf.h` ファイルで `_2_D354` を検索すると次のようになっています。

```
static LRD_VAR_DESC _2_D354 = {  
    LRD_VAR_DESC_EYECAT, 1, 10, LRD_BYTYPE_ODBC,  
    {0, 0, 0}, DT_SZ, 0, 0, 15, 12};
```

これを次のように変更します。

```
static LRD_VAR_DESC _2_D354 = {  
    LRD_VAR_DESC_EYECAT, 1, 12, LRD_BYTYPE_ODBC,  
    {0,0, 0, 0}, DT_SZ, 0, 0, 15, 12};
```

LRD_VAR_DESC の定義全体は `lrd.h` ファイルにあります。これを見つけるには「`typedef struct LRD_VAR_DESC`」で検索します。

Question 5 : ODBC および Oracle の使用で UPDATE, INSERT, DELETE に よって影響を受けた行数を取得する方法は？

Answer : lrd 関数を使用して情報を取得します。ODBC には **lrd_row_count** 関数を使用します。構文は次のとおりです。

```
int rowcount;
.
.
.
lrd_row_count(Csr33, &rowcount, 0);
```

lrd_row_count 関数は関連するステートメント実行の直後に使用します。

Oracle には **lrd_exec** の 4 番目の引数を使用します。

```
lrd_exec(Csr19, 1, 0, &rowcount, 0, 0);
```

Oracle の OCI 8 を使用している場合は、**lrd_ora8_exec** の 5 番目の引数を使用します。

```
lrd_ora8_exec(OraSvc1, OraStm3, 1, 0, &uliRowsProcessed, 0, 0, 0, 0, 0);
```

Question 6 : キーの重複割り当て違反を防ぐ方法は？

Answer : 挿入を実行するときに重複キー違反が生じることがあります。問題を識別するために 2 つの記録を比較して、主キーを見つけることができます。このステートメントまたはそれ以前に出てきた UPDATE または INSERT ステートメントで相関クエリが使われていたかどうかチェックします。データ辞書を使って一意制約に違反しているカラムを見つけることができます。

Oracle では、一意制約違反が生じると、次のメッセージが表示されます。

```
ORA-00001: unique constraint (SCOTT.PK_EMP) violated
```

この例では、**SCOTT** は関連する一意インデックスの所有者で、**PK_EMP** がそのインデックス名です。SQL*Plus を使用してデータ辞書のクエリを行い、カラムを見つけます。このクエリのパターンは次のようになります。

```
select column_name from all_ind_columns where index_name = '<IndexName>'
and index_owner = '<IndexOwner>';
```

```
select column_name from all_ind_columns where index_name = 'PK_EMP' and
index_owner = 'SCOTT';
```

データベースに挿入された値が新しいため、以前のクエリでは見つかっていないかもしれませんが、以前のクエリよりも 1 つ多い戻り値として、以前のクエリの結果と関係していることがあります。

Microsoft SQL サーバでは次のいずれかのメッセージが表示されます。

Cannot insert duplicate key row in object 'newtab' with unique index 'IX_newtab'.

Violation of UNIQUE KEY constraint 'IX_Mark_Table'. Cannot insert duplicate key in object 'Mark_Table'.

Violation of PRIMARY KEY constraint 'PK_NewTab'. Cannot insert duplicate key in object 'NewTab'.

Query Analyzer を使用して、どのカラムがキーまたはインデックスで使用されているのかを検索します。このクエリのパターンは次のようになります。

```
select C.name
  from sysindexes A, sysindexkeys B, syscolumns C
  where C.colid = B.colid and C.id = B.id and
  A.id = B.id and A.indid = B.indid
  and A.name = '<IndexName>' and A.id = object_id('<TableName>')
```

```
select C.name
  from sysindexes A, sysindexkeys B, syscolumns C
  where C.colid = B.colid and C.id = B.id and
  A.id = B.id and A.indid = B.indid
  and A.name = 'IX_newtab' and A.id = object_id('newtab')
```

DB2 では次のメッセージが表示されます。

SQL0803N One or more values in the INSERT statement, UPDATE statement, or foreign key update caused by a DELETE statement are not valid because they would produce duplicate rows for a table with a primary key, unique constraint, or unique index. SQLSTATE=23505

まだ問題が続くようであれば、記録時、再生時のスクリプトの両方で更新および挿入で変更した行番号を確認します。UPDATE では、WHERE 句の条件式が間違っているために再生時に行を変更できないことがよくあります。これは直接エラーになりませんが、テーブルが正確に更新されず、後続の SELECT がクエリの相関時に間違った値を選択する原因になります。

また、マルチ・ユーザの再生中に問題がないことも確認します。特定のインスタンスでは、1 ユーザだけしか UPDATE を実行できません。この現象は Siebel で発生します。その場合には、手作業でループを書いて問題を回避する必要があります。

Question 7 : スクリプト再生後にデータベースが変更されているはずなのにされていないことがあります。

Answer : ユーザ・アプリケーションの UI からアプリケーションからアクセスできる現在のデータを調べ、それが更新された値かどうかを確認してください。値が更新されていない場合、それが変更されていないことを判定する必要があります。アプリケーションの記録中に UPDATE ステートメントが 1 つ以上の行を変更したために、再生時には変化がなかったということも考えられます。

次の項目を確認します。

- ▶ **ステートメントの検証 :** UPDATE ステートメント内の WHERE 句条件式が正しいことを確認します。
- ▶ **関連の確認 :** アプリケーションを 2 回記録して、それぞれの記録の UPDATE ステートメントを比較し、必要な相関が行われているか確認します。
- ▶ **行の総数の確認 :** UPDATE の後で変更された行数を確認します。Oracle ではこの情報は `lrd_exec` の 4 番目のパラメータに格納されています。ODBC では、`lrd_row_count` を使用して行数を調べます。スクリプトに更新された行数を出力するコードを追加できます。出力値が 0 ならば、UPDATE はデータベースの変更に失敗したことがわかります。
- ▶ **SET 句のチェック :** UPDATE ステートメントの SET 句の条件式を確認します。必要な値がすべて相関されており、ハードコードされていないかどうかチェックします。UPDATE の 2 つの記録を比べることによって判断できます。

これが問題なのは、UPDATE が単独の Vuser の再生時には動作するのに、複数の Vuser では動作しない場合です。ある Vuser の UPDATE がほかの Vuser のものと干渉していることが考えられます。各 Vuser に同じ値で更新を行うように指定する場合を除いて、各 Vuser をパラメータ化してそれぞれの Vuser が UPDATE 時に異なる値を使用するようにします。この場合、再試行論理を追加して UPDATE を再び試みます。

Question 8 : Oracle Application を使用して記録されたステートメントの再生時に一意カラム名エラーを防ぐ方法は？たとえば以下のようなステートメントがあるとします。

```
lrd_stmt(Csr9,"SELECT UOM_CODE, UOM_CODE, DESCRIPTION FROM "  
    "MTL_UNITS_OF_MEASURE "  
    "WHERE NVL(DISABLE_DATE, SYSDATE + 1) > "  
    "SYSDATE ORDER BY UOM_CODE", -1, 1, 1, 0);
```

この例では次のエラー・メッセージが発行されます。

```
"lrdo.c/fjParse: "oparse" ERROR return-code=960, oerhms=ORA-00960:  
ambiguous column naming in select list".
```

Answer : 一意でないカラムの少なくとも 1 つに別名を追加して、この新しい一意の名前にマッピングするようにステートメントを変更します。たとえば以下のようになります。

```
lrd_stmt(Csr9,"SELECT UOM_CODE,UOM_CODE second, DESCRIPTION  
FROM"  
    "MTL_UNITS_OF_MEASURE "  
    "WHERE NVL(DISABLE_DATE, SYSDATE + 1) > "  
    "SYSDATE ORDER BY UOM_CODE", -1, 1, 1, 0);
```

Siebel 固有のスクリプト作成のヒント

本項では、Siebel データベース・ユーザのための解決策を示します。前項までの記述にも、一般的なデータベース・スクリプト作成のヒントがありますので参照してください。

Question 9 : VuGen で仮想ユーザは正常に実行できるのに Controller では重複キー割り当て違反になります。

Answer : Siebel クライアントは S_SSA_ID テーブルの NEXT_SUFFIX カラムにキーを格納します。このクライアントにはサフィックス値のブロックを取得できない場合に、その状況を検出してそこから回復するためのコードがあります。

VuGen は自動的に S_SSA_ID テーブルで、NEXT_SUFFIX および MODIFICATION_NUM フィールドを相関します。UPDATE 中、MODIFICATION_NUM フィールドは 1 つ増加し、NEXT_SUFFIX フィールドは 36 ベースで 100 増加します。ただし、クライアントが新しいサフィックス値のブロックを取得できない場合、VuGen ではインスタンスにコードが追加されません。その結果、新しい値をデータベースに挿入しようとする、再生時に一意制約エラーが生じます。

1 回目の試行が失敗したときに再試行するために、スクリプトでサフィックスのブロックを取得する各箇所に、手作業でコードを追加します。スクリプト内で「SiebelPreSave」を検索して、該当箇所を見つけます。次の例に似たコードを含む **while** ループも追加する必要があります。この例は、Oracle のみに該当します。ODBC の場合は、**lrd_exec** の 4 番目の引数ではなく、**lrd_row_count** を使用します。

```
unsigned long lRowUpdated;
int nAttempt;
```

```
...
```

```
// 「next_suffix」を取得するまでループが継続します。
```

```
lRowUpdated = 0;
```

```
nAttempt=0;
```

```
while (lRowUpdated != 1) {
```

```
    nAttempt++;
```

```
    if (nAttempt > 1)
```

```
        lr_output_message (".....Next suffix retry %d", nAttempt);
```

```
    else
```

```
    {
```

```
        lrd_open_cursor(&Csr13, Con1, 0);
```

```
        lrd_stmt(Csr13, "SELECT%n T1.LAST_UPD,%n T1.CREATED_BY,%n "
```

```
            "T1.CONFLICT_ID,%n T1.CREATED,%n T1.NEXT_SUFFIX,%n "
```

```
            "T1.ROW_ID,%n T1.NEXT_PREFIX,%n T1.CORPORATE_PREFIX,%n "
```

```
            "T1.MODIFICATION_NUM,%n T1.NEXT_FILE_SUFFIX,%n "
```

```
            "T1.LAST_UPD_BY%n FROM %n SIEBEL.S_SSA_ID T1", -1, 1, 1, 0);
```

```
    }
```

```
        lrd_bind_cols(Csr13, BCInfo_D375, 0);
```

```
        lrd_exec(Csr13, 0, 0, 0, 0, 0);
```

```

SiebelPreSave_1();
Ird_fetch(Csr13, -1, 4, 0, PrintRow26, 0);
GRID(26);
SiebelPostSave_1();

if (nAttempt > 1)
{
Ird_open_cursor(&Csr14, Con1, 0);
Ird_stmt(Csr14, "¥nUPDATE SIEBEL.S_SSA_ID SET¥n LAST_UPD_BY=:1,¥n "
"NEXT_SUFFIX = :2,¥n  MODIFICATION_NUM = :3,¥n  LAST_UPD = "
":4¥n WHERE¥n ROW_ID = :5 AND MODIFICATION_NUM = :6¥n", -1, 1,
1, 0);
}
Ird_assign_bind(Csr14, "6", "<modification_num>", &_6_D376, 0,
LRD_BIND_BY_NUMBER, 0);
Ird_assign_bind(Csr14, "5", "0-11",&_5_D377,0,LRD_BIND_BY_NUMBER, 0);
strcpy (szTimeAtNewButton, Ir_eval_string("<Now>"));
sprintf (szTimeStamp, "%s %s", Ir_eval_string("<Today>"),
szTimeAtNewButton);
Ir_save_string (szTimeStamp, "DateTimeStamp");
Ird_assign_bind(Csr14, "4", "<DateTimeStamp>", &_4_D378, 0,
LRD_BIND_BY_NUMBER, 0);
Ird_assign_bind(Csr14, "3", "<next_modnum>", &_3_D379, 0,
LRD_BIND_BY_NUMBER, 0);
Ird_assign_bind(Csr14, "2", "<next_suffix_x100>", &_2_D380, 0,
LRD_BIND_BY_NUMBER, 0);
Ird_assign_bind(Csr14, "1", "1-1E1",&_1_D381,0,LRD_BIND_BY_NUMBER, 0);

// このアップデートではサフィックスを正常に取得しないかぎり行を更新しま
せん。
Ird_exec(Csr14, 1, 0, &IRowUpdated, 0, 0);
Ird_commit(0, Con1, 0);

} //while
Ird_output_message ("...Rows updated %ld", IRowUpdated);

```

Question 10 : 主キーの相関に使う正しい値を見つける方法は？

Answer : Siebel は <next_suffix> をベース 36 で数学的に処理した結果に基づいてキーの値を生成する傾向があります。いくつかの記録同士を比較して、関連性を見つけられるかどうか試してみてください。Siebel で相関を行うときには、スクリプト再生に影響しないので日付フィールドを無視できます。

Question 11 : 重複キー違反による INSERT into S_SRV_REQ の失敗を解決する方法は？

Answer : 主キーは SR_NUM です。新しいバージョンの VuGen は、S_SSA_ID テーブルの NEXT_SUFFIX 値をベース 36 からベース 10 の同等の値に変換する lrd_siebel_str2num 関数を使って自動的にこのテーブルへの挿入を相関します。旧バージョンの VuGen ではこの相関が正しく行われなことがあることがあります。

Question 12 : VuGen でスクリプトを正しく再生するために必要な相関が自動的に行われません。足りない相関を追加する方法は？

Answer : 現在のところ VuGen は S_SSA_ID テーブルの NEXT_SUFFIX および MODIFICATION_NUM カラムの値を保存し、スクリプトで使用するときにその値をパラメータで置換しているだけです。したがって手作業で相関を追加する必要があります。print.inl ファイル内の SiebelPreSave 関数と SiebelPostSave 関数の相関コードは、何を相関する必要があるかわかっている場合に、特定の値を相関する方法を示す例です。

- ▶ NEXT_FILE_SUFFIX および MODIFICATION_NUM カラムは S_SSA_ID テーブルから選択される場合があります。その場合、UPDATE ステートメントはベース 36 でこの文字列、および MODIFICATION_NUM に 1 を追加して NEXT_FILE_SUFFIX を更新します。NEXT_FILE_SUFFIX の値はテーブル内の FILE_REV_NUM フィールドに挿入されることがしばしばあります。このテーブルの名前はしばしばサフィックス **_ATT** で終わり、それが添付ファイルであることを示します。
- ▶ Siebel が UPDATE ステートメントを実行するときには必ず、値が 1 ずつ増加する MODIFICATION_NUM カラムが存在します。VuGen は S_SSA_ID テーブルに対してだけ、自動的にこの相関を行います。それ以外は手作業で相関する必要があります。

- ▶ Siebel は記録を ID 番号で参照します。Siebel は通常、特定のタイプ（たとえば契約など）のすべての記録を検索し、そのタイプの特定の記録を更新または削除しようとするときに、ID 番号を使います。意味のある負荷テストを生成するには、再生中に ID 番号をパラメータで置換する必要があります。ID 番号は 1-QPF9 のように、1 桁以上の数字とハイフンに続く 1 桁以上の英数字からなります。VuGen ではこのパラメータ化は自動的に行われないので、手作業で行います。
- ▶ ほかにも関連またはパラメータ化の不足が見つかった場合は、VuGen の Siebel のサポート向上のために、カスタマ・サポートにご連絡ください。

PeopleSoft-Tuxedo スクリプトの実行

Tuxedo 7.x で PeopleSoft-Tuxedo Vuser を実行するには、**mdrv.dat** ファイル内のライブラリ拡張子を変更する必要があります。

```
[PeopleSoft-Tuxedo]  
WINNT_EXT_LIBS=lrt7.dll
```

第 28 章

上級ユーザのために

以下の詳細情報は、再生問題の特定や Vuser スクリプトのデバッグに役立つでしょう。

本章の内容

- ▶ 記録中に生成されるファイル (550 ページ)
- ▶ 再生中に生成されるファイル (552 ページ)
- ▶ UNIX コマンド・ラインからの Vuser の実行 (554 ページ)
- ▶ Vuser の動作の指定 (555 ページ)
- ▶ コマンド・ライン・パラメータ (556 ページ)
- ▶ OLE サーバの記録 (556 ページ)
- ▶ .dat ファイルの検証 (559 ページ)
- ▶ 新規 Vuser タイプの追加 (561 ページ)

記録中に生成されるファイル

記録されたテストに「vuser」という名前を付け、それを `c:\tmp` の下に格納したとします。記録後に生成される、特に重要なファイルの一覧を以下に示します。

| | |
|--------------------|--|
| vuser.usr | 仮想ユーザに関する情報（タイプ、テスト対象アプリケーション、アクション・ファイルなど）が含まれます。 |
| vuser.bak | 最後に保存した <code>Vuser.usr</code> の 1 つ前のコピー。 |
| default.cfg | VuGen アプリケーションで定義されたすべての実行環境の設定（思考遅延時間、反復、ログ、Web）の一覧が含まれます。 |
| vuser.asc | 記録されている API 呼び出し。 |
| vuser.grd | データベース・スクリプトのグリッドのカラム・ヘッダが含まれています。 |
| default.usp | スクリプトの実行ロジック（Actions セクションの実行方法など）が含まれます。 |
| init.c | VuGen メイン・ウィンドウに表示される <code>Vuser_init</code> 関数とまったく同じもの。 |
| run.c | VuGen メイン・ウィンドウに表示される <code>Action</code> 関数とまったく同じもの。 |
| end.c | VuGen メイン・ウィンドウに表示される <code>Vuser_end</code> 関数とまったく同じもの。 |
| vdf.h | スクリプトで使用される C 変数定義のヘッダ・ファイル。 |
| ¥Data | <code>Data</code> ディレクトリには、主にバックアップ用として使用されるすべての記録データが格納されます。データはこのディレクトリに格納されると、編集したり使用したりできなくなります。たとえば、 <code>Vuser.c</code> は、 <code>run.c</code> のコピーです。 |

Vuser.usr ファイルの例

```
[General]
Type=Oracle_NCA
DefaultCfg=default.cfg
AppName=C:\PROGRA~1\Netscape\COMMUN~1\Program\netscape.exe
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
```

default.cfg ファイルの例

```
[General]
XIBridgeTimeout=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
```

再生中に生成されるファイル

本項では、Vuser を再生したときに何が起きるかを説明します。

- 1 プリプロセッサ用のコマンド・ライン・パラメータを含む **options.txt** ファイルが作成されます。
- 2 関連するすべての .c および .h ファイルに対する「includes」を含む **Vuser.c** ファイルが作成されます。
- 3 開発用ファイルからマクロ定義およびプリコンパイラ指示子などを「挿入する」ために C のプリプロセッサ **cpp.exe** が呼び出されます。

次のコマンド・ラインが使用されます。

```
cpp -foptions.txt
```

- 4 **pre_cci.c** ファイルが作成されます。これも C ファイルです (**pre_cci.c** は、**options.txt** ファイルで定義されます)。このプロセスのあらゆる出力を含む **logfile.log** ファイル (このファイルも **options.txt** で定義されます) が作成されます。**logfile.log** ファイルは、プリプロセス処理の段階で何も問題がなければ、空のはずです。このファイルが空でなければ、コンパイルの次の段階で致命的なエラーが発生し、ほぼ確実に失敗します。
- 5 仮想ユーザ・ドライバ・プログラムによって実行時に解釈される、プラットフォームに依存する疑似バイナリ・ファイル (.ci) を作成するために、C コンパイラ **cci.exe** が起動されます。**cci** は **pre_cci.c** ファイルを入力として受け取ります。
- 6 **pre_cci.ci** ファイルが次のように作成されます。

```
cci -errout c:\tmp\Vuser\logfile.log -c pre_cci.c
```
- 7 **logfile.log** ファイルは、コンパイル時の出力を格納するログ・ファイルです。
- 8 **pre_cci.ci** ファイルの名前はここで **Vuser.ci** に変わります。

コンパイル時には警告とエラーの両方が生成される可能性があり、ドライバはこのプロセスの結果を知らないので、ドライバはまず **logfile.log** ファイルにエントリがあるか調べます。**logfile.log** ファイルにエントリがある場合、ドライバは、**Vuser.ci** ファイルが生成されているかどうかを調べます。ファイル・サイズがゼロでなければ、**cci** がコンパイルに成功したということです。ファイル・サイズがゼロであれば、コンパイルは失敗しており、エラー・メッセージが表示されます。

- 9 関連するドライバが実行され、入力データとして **.usr** ファイルと **Vuser.ci** ファイルが使われます。たとえば以下のようになります。

```
mdrv.exe -usr c:¥tmp¥Vuser¥Vuser usr -out c:¥tmp¥Vuser -file
c:¥tmp¥Vuser¥Vuser.ci
```

.usr ファイルは、ドライバ・プログラムにどのデータベースが使用されているのかを知らせるために必要です。この段階で、実行のためにロードすべきライブラリがわかります。

- 10 実行中に出力されたすべてのメッセージを含む **output.txt** が（「out」変数によって定義されたパス内に）作成されます。これは、VuGen の実行時の出力ウィンドウおよび VuGen のメイン・ウィンドウの下部の表示枠に表示されるものとまったく同じです。

options.txt ファイルの例

```
-DCCI
-D_IDA_XL
-DWINNT
-lc:¥tmp¥Vuser(Vuser インクルード・ファイルの名前と場所)
-lE:¥LRUN45B2¥include(インクルード・ファイルの名前と場所)
-ec:¥tmp¥Vuser¥logfile.log(出力ログ・ファイルの名前と場所)
c:¥tmp¥Vuser¥VUSER.c(処理されるファイルの名前と場所)
```

Vuser.c ファイルの例

```
#include "E:¥LRUN45B2¥include¥lrun.h"
#include "c:¥tmp¥web¥init.c"
#include "c:¥tmp¥web¥run.c"
#include "c:¥tmp¥web¥end.c"
```

UNIX コマンド・ラインからの Vuser の実行

VuGen には、仮想ユーザと同じ操作をコマンド・ラインから自動的に実行する UNIX シェル・スクリプト・ユーティリティ **run_db_Vuser.sh** が含まれています。このユーティリティは各再生ステップを個別に実行できます。このツールは、UNIX 上で再生するテストをデバッグするのに便利です。

run_db_Vuser.sh を `$M_LROOT/bin` ディレクトリに配置します。Vuser タイプを再生するには、次のように入力します。

```
run_db_Vuser.sh Vuser.usr
```

次のコマンド・ライン・オプションを使用することもできます。

- cpp_only** このオプションは、プリプロセス処理のフェーズを開始します。この処理によって、「**Vuser.c**」が生成されます。
- cci_only** このオプションは、コンパイルのフェーズを実行します。「**Vuser.c**」ファイルは入力データとして使用されます。この処理によって、「**Vuser.ci**」ファイルが生成されます。
- exec_only** このオプションは、「**Vuser.ci**」ファイルを入力データとして受け取り、再生ドライバを介して Vuser を実行します。
- ci ci_file** このオプションを使って、実行する **.ci** ファイルの名前と場所を指定できます。2 つ目のパラメータに、**.ci** ファイルの場所を指定します。
- out output_directory** このオプションを使って、各種の処理によって作成される出力ファイルの場所を指定できます。2 つ目のパラメータに、ディレクトリの名前と場所を指定します。
- driver driver_path** このオプションを使って、Vuser の実行に使用する実際のドライバ実行可能ファイルを指定できます。標準設定では、ドライバ実行可能ファイルは VuGen の **.dat** ファイル内の設定から取得されます。

最初の 3 つのオプションは、**run_db_vuser** を実行するとき、一度にどれか 1 つだけを使用できます。

Vuser の動作の指定

VuGen は Vuser スクリプトと の動作を 2 つの独立した情報源として作成するので、たとえば待機時間、時間間隔、反復のループ、ログの記録などのユーザの動作を、Vuser スクリプトを直接参照せずに設定できます。この機能により、Vuser の設定が変更できると同時に、同じ Vuser スクリプトについて「プロファイル」を複数格納できます。

標準設定では、Vuser の動作は VuGen の [実行環境設定] ダイアログ・ボックスで指定されているとおりに、「**Vuser.cfg**」ファイルに定義されています。このファイルの、ユーザ動作ごとに異なる複数のバージョンを保存することができます。そして、関連する **.cfg** ファイルを参照する Vuser スクリプトを実行できます。

サーバ・マシンから、使用する設定ファイルを指定して Vuser スクリプトを実行できます。これを行うには、次のパラメータ指定を Vuser のコマンド・ラインに追加します。

```
-cfg c:¥tmp¥profile2.cfg
```

コマンド・ライン・パラメータについては、556 ページ「コマンド・ライン・パラメータ」を参照してください。

VuGen からは、振る舞いを定義したファイルを指定できません。VuGen は Vuser と同じ名前の **.cfg** ファイルを自動的に使用します（もちろんファイル名を「**Vuser.cfg**」に変更することもできます）。ただし、上で説明した **-cfg** パラメータをドライバのコマンド・ラインの最後に追加すれば、コマンド・ラインから手作業でファイルを指定できます。

注：UNIX 用のユーティリティ **run_db_vuser** では、このオプションはまだサポートされていません。

コマンド・ライン・パラメータ

Vuser は、起動時にコマンド・ライン・パラメータを受け付けます。Vuser API には、コマンド・ライン・パラメータを参照するための関数がいくつかあります (`lr_get_attrib_double` など)。環境内で、スクリプト・ウィンドウのコマンド・ライン・エントリにパラメータを追加することで、Vuser にコマンド・ライン・パラメータを送ることができます。

VuGen から Vuser を実行するときは、コマンド・ライン・パラメータを指定できません。ただし、Windows のコマンド・ラインでほかのすべてのドライバ・パラメータの後、つまり行の最後にパラメータを追加することでこれを手作業で指定できます。

```
mdrv.exe -usr c:%tmp%\Vuser%\Vuser usr -out c:%tmp%\vuser  
vuser_command_line_params
```

注：UNIX 用のユーティリティ `run_db_vuser` では、このオプションはまだサポートされていません。

OLE サーバの記録

VuGen では現在、OLE アプリケーションの記録はサポートされていません。OLE アプリケーションでは、実際のプロセスが標準のプロセス生成ルーチンによってではなく、OLE オートメーション・システムによって起動されます。ただし、以下に示すガイドラインに沿って、OLE アプリケーション用の Vuser スクリプトが作成できます。

OLE サーバには、実行可能ファイルと DLL の 2 つの種類があります。

DLL サーバ

サーバが DLL である場合、このサーバは最終的にアプリケーションのプロセス空間にロードされ、VuGen は `LoadLibrary` への呼び出しを記録します。この場合、ユーザはこれが OLE アプリケーションであることに気付かないかもしれません。

実行可能サーバ

サーバが実行形式の場合は、以下に示す方法で VuGen から実行ファイルを起動する必要があります。

- ▶ まず、実際に記録する必要があるプロセスを特定します。多くの場合、アプリケーションの実行可能ファイルの名前がわかっています。名前がわからない場合は、対象アプリケーションを起動し、NT のタスク・マネージャでその名前を確認します。
- ▶ 必要なプロセスを特定したら、VuGen で **〔記録開始〕** をクリックします。アプリケーション名の入力を要求されるので、OLE アプリケーションの名前と、その後ろに「/Automation」というフラグを入力します。次に、VuGen からではなく通常の方法でユーザ・プロセスを実行します。VuGen は実行中の OLE サーバを記録し、同じサーバを別に起動することはありません。VuGen で OLE サーバのアクションを記録するには、ほとんどの場合、この手順でうまくいきます。
- ▶ それでもうまく記録できない場合は、**CmdLine** プログラムを使って、直接起動されないプロセスの完全なコマンド・ラインを調べます（このプログラムは、カスタマ・サポート Web サイト <http://support.hp.com> の知識ベースの記事からダウンロードできます）。

CmdLine の使用法

次の例では、**CmdLine.exe** を使って、ほかのプロセスによって起動されるプロセス **MyOleSrv.exe** の完全なコマンド・ラインを調べています。

完全なコマンド・ラインを調べるには、次の手順を実行します。

- 1 **MyOleSrv.exe** の名前を **MyOleSrv.orig.exe** に変更します。
- 2 アプリケーションと同じディレクトリに **CmdLine.exe** を入れ、この名前を **MyOleSrv.exe** に変更します。
- 3 **MyOleSrv.exe** を起動します。この **MyOleSrv.exe** は、元のアプリケーションの完全なコマンド・ラインを含むポップアップ・メッセージ（追加情報を含む）を表示し、その情報を **c:%temp%\CmdLine.txt** に書き込みます。
- 4 それぞれを元の名前に戻し、正しいコマンド・ライン・パラメータを使って OLE サーバ **MyOleSrv.exe** を VuGen から起動します。ユーザ・アプリケーションは VuGen からではなく、通常の方法で起動します。ほとんどの場合、VuGen は正しく記録を行います。

それでもうまく記録できない場合は、次の処理を行います。

- 1 OLE サーバ名を MyOleSrv.1.exe に、CmdLine を MyOleSrv.exe に変更します。
- 2 環境変数「CmdStartNotepad」と「CmdNoPopup」を「1」に設定します。
CmdLine 環境変数については、558 ページ「CmdLine 環境変数」の一覧を参照してください。
- 3 VuGen 以外からアプリケーションを起動します。「メモ帳」が開き、完全なコマンド・ラインが表示されます。コマンド・ライン引数を調べます。アプリケーションを数回起動し、コマンド・ライン引数を比較します。何度アプリケーションを起動しても引数が同じである場合は、CmdStartNotepad 環境変数をリセットします。そうでなければ、設定を「1」のままにしておきます。
- 4 VuGen で、コマンド・ライン・パラメータを使用して（「メモ帳」のウィンドウからコピー / 貼り付けで指定してください）プログラム MyOleSrv.1.exe を起動します。
- 5 VuGen 以外からアプリケーションを起動します。

CmdLine 環境変数

以下に示す環境変数を使うことで、CmdLine の実行を制御できます。

- | | |
|------------------------|---|
| CmdNoPopup | これが設定されていると、ポップアップ・ウィンドウが現れません。 |
| CmdOutFileName | これが設定されており、空でない場合は、CmdLine は c:\temp\CmdLine.txt のかわりにこのファイルを作成しようとしています。 |
| CmdStartNotepad | これが設定されていると、出力ファイルがメモ帳に表示されます（CmdNoPopup との併用をお勧めします）。 |

.dat ファイルの検証

VuGen は、`vugen.dat` と `mdrv.dat` という 2 つの `.dat` ファイルを使用します。

vugen.dat

この `vugen.dat` ファイルは `M_LROOT\dat` ディレクトリにあり、VuGen についての一般情報が含まれています。VuGen と Controller およびコンソールの両方で使用されます。

[Templates]

`RelativeDirectory=template`

Templates セクションは、VuGen プロトコル用のテンプレートの場所を示します。標準のエントリは、これらのテンプレートが相対 **template** ディレクトリにあることを示します。各プロトコルには、**template** の下にサブディレクトリがあり、この中には、そのプロトコル用のテンプレート・ファイルが含まれています。

次のセクションは `GlobalFiles` セクションです。

```
[GlobalFiles]
main.c=main.c
@@TestName@@.usr=test usr
default.cfg=test.cfg
default.usp=test.usp
```

GlobalFiles セクションには、新規テストが作成されたときに VuGen がテスト・ディレクトリにコピーしたファイルの一覧が含まれます。たとえば、「`user1`」というテストがある場合、VuGen は `main.c`、`user1 usr`、および `user1.cfg` をテスト・ディレクトリにコピーします。

ActionFiles セクションには、`Vuser` によって実行されるアクションを含むファイルの名前と、反復を実行する `Vuser` が含まれます。

[ActionFiles]

`@@actionFile@@=action.c`

上に示した設定に加え、**vugen.dat** には、オペレーティング・システムおよびコンパイルに関連するそのほかの設定が含まれます。

mdrv.dat

mdrv.dat ファイルには、ライブラリ・ファイルとドライバの実行可能ファイルの場所を定義する、プロトコル別のセクションがあります。次の項では、新しいプロトコルを定義するためにファイルに追加すべき項目を説明します。

新規 Vuser タイプの追加

VuGen に新しい Vuser のタイプまたはプロトコルを追加するのに必要な項目は次のとおりです。

- ▶ 新しいプロトコルの設定による **mdrv.dat** ファイルの編集
- ▶ **.cfg** ファイルの追加
- ▶ **lrp** ファイルを挿入
- ▶ テンプレート・ディレクトリの作成

mdrv.dat ファイルの編集

まず、mdrv.dat ファイルを編集します。このファイルは、M_LROOT\%Ydat ディレクトリにあります。新しい Vuser タイプのためのセクションを追加します。

```
[<extension_name>]
ExtPriorityType=<{internal, protocol}>
WINNT_EXT_LIBS=<NT 用 DLL 名 >
WIN95_EXT_LIBS=<95 用 DLL 名 >
SOLARIS_EXT_LIBS=<Solaris 用 dll 名 >
LINUX_EXT_LIBS=<Linux 用 dll 名 >
HPUX_EXT_LIBS=<HP 用 dll 名 >
AIX_EXT_LIBS=<IBM 用 dll 名 >
LibCfgFunc=< 設定関数名 >
UtilityExt=< ほかの拡張子リスト >
WINNT_DLLS=< インタプリタ・コンテキストにロードする DLL (NT 用) >
WIN95_DLLS=< インタプリタ・コンテキストにロードする DLL (95 用) >
SOLARIS_DLLS=< インタプリタ・コンテキストにロードする dll (Solaris 用) >
LINUX_DLLS=< インタプリタ・コンテキストにロードする dll (Linux 用) >
HPUX_DLLS=< インタプリタ・コンテキストにロードする dll (HP 用) >
AIX_DLLS=< インタプリタ・コンテキストにロードする dll (IBM 用) >
ExtIncludeFiles=< 追加インクルード・ファイル。複数のファイルをカンマで区切って指定できる >
ExtCmdLineConc=< 追加コマンド・ライン (属性がある場合は値を連結する) >
ExtCmdLineOverwrite=< 追加コマンド・ライン (属性がある場合は値を上書きする) >
CallActionByNameFunc=< インタプリタ exec_action 関数 >
GetFuncAddress=< インタプリタ get_location 関数 >
RunLogicInitFunc=<action_logic init 関数 >
RunLogicRunFunc=<action_logic run 関数 >
RunLogicEndFunc=<action_logic end 関数 >
```

たとえば、Oracle NCA Vuser タイプは、以下で表されます。

```
[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 は、コードに変更を加えずに新しい Vuser タイプを処理できるように設計されています。ただし、特別なビューを追加しなければならない場合があります。

VuGen では汎用のドライバは提供されていませんが、既存のドライバをカスタマイズできます。カスタマイズしたドライバを使用するには、**mdrv.dat** を変更します。プラットフォームと既存のドライバの行を追加した後、カスタマイズしたドライバの名前の行を <プラットフォーム>_DLLS=<再生用 DLL 名> の形式で追加します。たとえば、SAP の再生用 DLL が SAPPLAY32.DLL という名前の場合は、次の 2 行を **mdrv.dat** の [sap] セクションに追加します。

```
WINNT=sapdrv32.exe
WINNT_DLLS=sapplay32.dll
```

CFG ファイルの追加

プロトコルに標準の [実行環境の設定] を設定するために、設定ファイルを任意で指定できます。これを行うには、`mdrv.dat` ファイル内の `LibCfgFunc` 変数で定義するか、テンプレートの下の新しいプロトコル・サブディレクトリに `default.cfg` というファイルをおきます。サンプルの `.cfg` は次のとおりです。

```
[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
```

LRP ファイルの挿入

dat¥protocols ディレクトリに、プロトコルを定義する **lrp** ファイルを挿入します。このファイルには、Protocol, Template, VuGen, API というセクションにプロトコルの設定情報が含まれています。プロトコルの中には、追加の実行環境設定オプションに応じて、これ以外のセクションがあるものもあります。

Protocol セクションには、プロトコルの名前、カテゴリ、説明、ビットマップの場所などが記述されています。

```
[Protocol]
Name=WAP
CommonName=WAP
Category=Wireless
Description=Wireless Application Protocol - used for Web-based, wireless
communication between mobile devices and content providers.
Icon=bitmaps¥wap.bmp
Hidden=0
Single=1
Multi=0
```

Template セクションには、スクリプトのさまざまなセクションの名前と標準のテスト名が記載されています。

```
[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
```

VuGen セクションには、記録と再生エンジン、必要な DLL と実行時ファイルに関する情報が記述されています。

API セクションには、プロトコルのスクリプト API 関数についての情報が記述されています。

プロトコル・ディレクトリ内の任意の lrp ファイルを新しいプロトコルのひな形として使用できます。

テンプレートの指定

lrp ファイルを追加したら、**M_LROOT**¥template の下にサブディレクトリを作成し、**lrp** ファイルで定義したプロトコル名に対応する名前を付けます。このサブディレクトリに、一般設定および実行環境設定のための標準の設定を収めた **default.cfg** ファイルをおきます。

新しいプロトコルのすべてのスクリプトでグローバルなヘッダ・ファイルを使用する場合には、**globals.h** という名前のファイルを追加します。このファイルには、新しいプロトコルのためのヘッダ・ファイルを指す **include** ステートメントを含めておきます。たとえば、**template**¥http サブディレクトリには、**globals.h** というファイルがあり、**include** ディレクトリにある **as_web.h** ファイルをインクルードしています。

```
#include #as_web.h"
```


第 6 部

付録

付録 29

外部関数の呼び出し

VuGen 使用時に、外部 DLL で定義されている関数を呼び出せます。スクリプトから外部関数を呼び出すことにより、スクリプトと実行環境全体で必要とするメモリを減らせます。

外部関数を呼び出すには、その関数が定義されている DLL をロードします。

DLL は次のようにしてロードできます。

- ▶ ローカル：1つのスクリプトにロードする場合には、`lr_load_dll` 関数を使用します。
- ▶ グローバル：すべてのスクリプトにロードする場合には、`vugen.dat` ファイルにステートメントを追加します。

本章の内容

- ▶ DLL のローカル・ロード (570 ページ)
- ▶ DLL のグローバル・ロード (572 ページ)

DLL のローカル・ロード

`lr_load_dll` 関数を使用して、DLL を Vuser スクリプトにロードします。一度 DLL をロードすれば、その DLL で定義されている任意の関数をスクリプト内で宣言せずに呼び出せます。

DLL 内で定義されている関数を呼び出すは、次の手順を実行します。

- 1 `lr_load_dll` 関数を使用して、スクリプトの先頭で DLL を読み込みます。このステートメントを `vuser_init` セクションの先頭に置きます。`ci_load_dll` 関数が `lr_load_dll` 関数に置き換えられます。

次の構文を使用します。

```
lr_load_dll(library_name);
```

UNIX プラットフォームでは、DLL は共有ライブラリと呼ばれています。ライブラリの拡張子はプラットフォームによって異なります。

- 2 DLL 内で定義されている関数をスクリプト内の適切な場所で呼び出します。

次の例では、Test_1 テーブル作成後、`oracl.dll` で定義されている `insert_vals` 関数を呼び出します。

```
int LR_FUNC Actions(LR_PARAM p)
{
  lr_load_dll("oracl.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);
  /* insert_vals 関数を呼び出し、値をテーブルに挿入する。*/insert_vals();
  lrd_stmt(Csr1, "select * from Test_1¥n", -1, 1 /*Deferred*/, 1 /*Dflt Ora Ver*/, 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);
  ...
}
```

注 : DLL の完全パス名を指定できます。 `lr_load_library` 関数は、パスが指定されなかった場合、Windows プラットフォームでは C++ 言語関数 `LoadLibrary` で使用される標準的な検索順序に従って DLL を検索します。UNIX プラットフォーム（または同等のプラットフォーム）では、`LD_LIBRARY_PATH` 環境変数を設定できます。 `lr_load_dll` 関数は、`dlopen` と同じ検索規則を使用します。詳細については、`dlopen` の main ページなどを参照してください。

DLL のグローバル・ロード

DLL をグローバルにロードして、その関数をすべての Vuser スクリプトで利用できます。一度 DLL をロードすれば、その DLL で定義されている任意の関数をスクリプト内で宣言せずに呼び出せます。

DLL 内で定義されている関数を呼び出すは、次の手順を実行します。

- 1 (アプリケーションの **dat** ディレクトリにある) **mdrv.dat** ファイルの適切なセクションに、ロードする DLL のリストを追加します。

次の構文を使用します。

PLATFORM_DLLS=my_dll1.dll, my_dll2.dll, ...

文字列 **PLATFORM** を使用するプラットフォームに置き換えます。プラットフォームのリストについては、**mdrv.dat** ファイルの最初のセクションを参照してください。

たとえば、NT プラットフォーム上の WinSock Vuser の DLL をロードするには、**mdrv.dat** ファイルに次の文を追加します。

```
[WinSock]
ExtPriorityType=protocol
WINNT_EXT_LIBS=wrun32.dll
WIN95_EXT_LIBS=wrun32.dll
LINUX_EXT_LIBS=liblrs.so
SOLARIS_EXT_LIBS=liblrs.so
HPUX_EXT_LIBS=liblrs.sl
AIX_EXT_LIBS=liblrs.so
LibCfgFunc=winsoc_exten_conf
UtilityExt=lrun_api
ExtMessageQueue=0
ExtCmdLineOverwrite=-WinInet No
ExtCmdLineConc=-UsingWinInet No
WINNT_DLLS=user_dll1.dll, user_dll2.dll, ...
```

- 2 DLL 内で定義されている関数をスクリプト内の適切な場所で呼び出します。

付録 30

多国語を使った作業

VuGen では多国語環境がサポートされており、スクリプトを作成、実行する際に自国語のマシン上で英語その他の言語を使用できます。

本章の内容

- ▶ 多国語を使った作業について (574 ページ)
- ▶ 手作業による文字列エンコーディングの変換 (574 ページ)
- ▶ パラメータ・ファイル内の文字列エンコーディングの変換 (575 ページ)
- ▶ Web 記録および再生のための文字列エンコーディングの設定 (577 ページ)
- ▶ Accept-Language ヘッダの言語の指定 (580 ページ)
- ▶ プロトコルに関する制限 (581 ページ)
- ▶ Quality Center の統合 (582 ページ)

多国語を使った作業について

英語以外の言語で作業を行うときは、記録や再生時に **VuGen** がテキストのエンコーディングを認識していることを確認することが、主な問題になります。エンコーディングは、スクリプトで使用するすべてのテキストに適用されます。これには、**Web Vuser** の場合の **HTTP** ヘッダ内のテキストや **HTML** ページ、パラメータ・ファイル内のデータなどが含まれます。

Windows 2000 以降では、**ANSI**、**Unicode**、**Unicode ビッグ・エンディアン**、**UTF-8** など、特定のエンコーディングがされたテキスト・ファイルを、メモ帳から直接保存できます。

標準設定では、**VuGen** はローカル・マシンのエンコーディング (**ANSI**) で動作します。多国語で作業を行っているサーバによっては、**UTF-8** エンコーディングでの作業が必要になることがあります。このようなサーバに対して作業を行うためには、詳細記録オプションの中で、スクリプトが **UTF-8** エンコーディングであるよう指定する必要があります。

手作業による文字列エンコーディングの変換

lr_convert_string_encoding 関数を使用すると、文字列のエンコーディング (**UTF-8**、**Unicode**、またはローカル・マシン・エンコーディング) を手作業で変換できます。関数の構文は次のとおりです。

```
lr_convert_string_encoding(char * sourceString, char * fromEncoding, char * toEncoding, char * paramName)
```

この関数は、結果の文字列 (終端の **NULL** を含む) を第 3 引数 **paramName** に保存します。変換に成功した場合は **0** を返し、失敗した場合は **-1** を返します。

fromEncoding 引数および **toEncoding** 引数の形式は次のとおりです。

| | |
|-----------------------------|----------------|
| LR_ENC_SYSTEM_LOCALE | NULL |
| LR_ENC_UTF8 | "utf-8" |
| LR_ENC_UNICODE | "ucs-2" |

次の例では、`lr_convert_string_encoding` によって「Hello world」をシステム・ロケールから Unicode に変換しています。

```

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,
"stringInUnicode");
    if (rc<0)
    {
        // エラー
    }
    return 0;
}

```

実行ログでは、出力ウィンドウに次の情報が表示されます。

```

Output:
Starting action Action.
Action.c(7): Notify: Saving Parameter "stringInUnicode = H¥x00e¥x00l¥x00l¥x00o¥x00
¥x00w¥x00o¥x00r¥x00l¥x00d¥x00¥x00¥x00"
Ending action Action.

```

変換の結果は `paramName` 引数に保存されます。

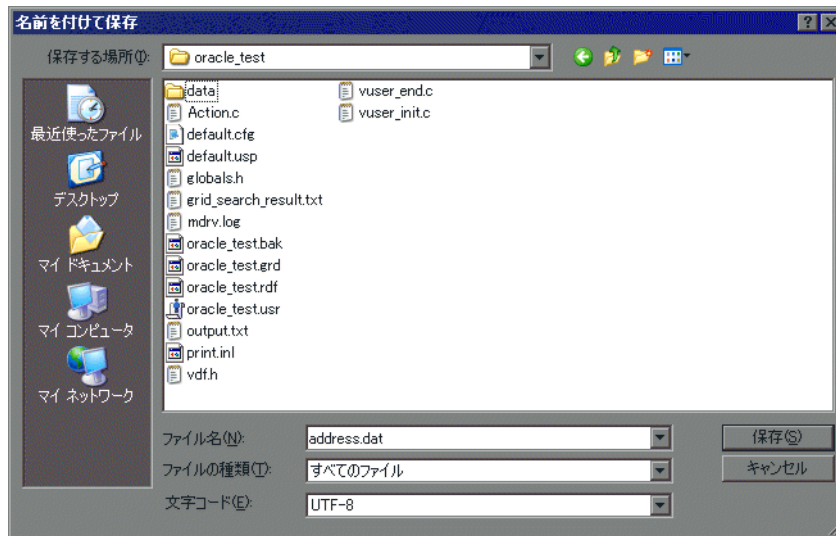
パラメータ・ファイル内の文字列エンコーディングの変換

パラメータ・ファイルには、スクリプトの中で定義されたパラメータに対応するデータが収められています。このファイルはスクリプトのディレクトリに格納され、拡張子 `*.dat` が付けられます。スクリプトを実行するとき、Vuserはこのデータを使用して、アクションをさまざまな値で実行します。

標準設定では、パラメータ・ファイルはマシンのエンコーディングを使用して保存されます。しかし、英語以外の言語で作業を行うときは、サーバにおいて文字列を UTF-8 で受信することが前提となっている場合に、パラメータ・ファイルを UTF-8 に変換しなければならないことがあります。Windows 2000 以降で作業を行っていれば、メモ帳から直接この変換を実行できます。

UTF-8 エンコーディングをパラメータ・ファイルに適用するには、次の手順を実行します。

- 1 [仮想ユーザ] > [パラメータ リスト] を選択し、パラメータ・プロパティを表示します。
- 2 右側の表示枠にある [ファイルパス] ボックスで、パラメータ・ファイルを探します。
- 3 パラメータ・テーブルを表示した状態で、[メモ帳で編集] をクリックします。メモ帳が開き、パラメータ・ファイルが csv 形式で表示されます。
- 4 [Save as type] ボックスで、[All Files] を選択します。
[Encoding] ボックスで、エンコーディングのタイプとして [UTF-8] を選択します。



- 5 [保存] をクリックします。既存のパラメータ・ファイルを上書きしてもよいかどうかの確認を求められます。[はい] をクリックします。

これで、パラメータ・ファイルが VuGen によって UTF-8 テキストとして認識されるようになります。ただし、表示上は通常の文字で表示されます。

Web 記録および再生のための文字列エンコーディングの設定

Web またはその他のインターネット・プロトコルを使って作業を行うときは、Web ページ・テキストの記録用のエンコーディングを指定できます。記録するサイトの言語がオペレーティング・システムの言語と一致している必要があります。1 つの記録の中で複数のエンコーディングを組み合わせることはできません。たとえば、UTF-8 を ISO-8859-1 や shift_jis と一緒に使用することはできません。

本項では、次について説明します。

- ▶ [エンコーディング] 記録オプション
- ▶ 手作業によるエンコーディングの有効化
- ▶ ブラウザの設定

[エンコーディング] 記録オプション

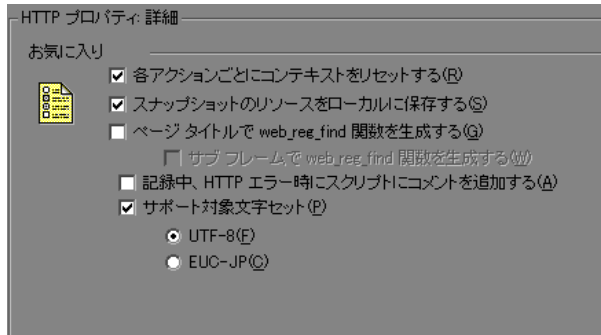
英語以外の言語の Web ページとして認識させるためには、ページの HTTP ヘッダまたは HTML メタ・タグの中で文字セットを指定する必要があります。そうしないと、EUC-JP エンコーディングが VuGen によって認識されず、Web サイトが正しく記録されません。英語以外の言語の要求を **EUC-JP** または **UTF-8** として記録するよう VuGen に指示するには、該当するオプションを [記録オプション] ダイアログ・ボックスの [HTTP プロパティ: 詳細] ノードで選択します。

- ▶ [UTF-8]: UTF-8 エンコーディングのサポートを有効にします。この設定を有効にすると、非 ASCII の UTF-8 文字がマシンのロケールで使用されているロケールに変換され、VuGen に正しく表示されます。
- ▶ [EUC-JP]: 日本語版の Windows を利用するユーザは、このオプションを選択することで、EUC-JP 文字エンコーディングを使用する Web サイトに対するサポートを有効にできます。この設定を有効にすると、EUC-JP 文字がマシンのロケールで使用されているロケールに変換され、VuGen に正しく表示されます。VuGen によって、すべての EUC-JP (日本語 UNIX) 文字列をマシンのロケールに合わせて Shift-JIS (日本語版 Windows) エンコーディングに変換し、スクリプトに `web_sjis_to_euc_param` 関数を追加します (漢字のみ)。

[記録オプション] で [EUC-JP] または [UTF-8] を選択すると、Web ページで異なるエンコーディングが使用されている場合でも、選択したエンコーディングによって強制的に Web ページが記録されます。たとえば、非 EUC でエンコードされた Web ページを EUC-JP として記録した場合、スクリプトでは正しく再生されません。

文字セット・エンコーディングを有効にするには、次の手順を実行します。

- 1 [記録オプション] を開き (Ctrl+F7), [Advanced] ノードを選択します。



- 2 [サポート対象文字セット] を選択します。必要な文字エンコーディングとして、[UTF-8] または [EUC-JP] (漢字対応のオペレーティング・システムでのみ有効) を選択します。
- 3 [OK] をクリックします。

これらの設定の詳細については、『第 2 巻 - プロトコル』の「選択したプロトコルの記録オプション」を参照してください。

手作業によるエンコーディングの有効化

`web_sjis_to_euc_param` 関数を使用すると、EUC-JP でエンコードされている HTML ページについて、その記録と再生を行うためのすべてのサポートを手作業で追加できます。これにより、EUC エンコードされた日本語文字を Vuser スクリプトで正しく表示できるようになります。

`web_sjis_to_euc_param` を使用すると、パラメータの値が実行ログで EUC-JP エンコーディングを使用して表示されます。たとえば、`web_find` 関数を再生するとき、VuGen ではエンコードされた値が表示されます。これらの値には、`web_sjis_to_euc_param` 関数によって EUC に変換された文字列値や、[実行環境設定] > [ログ] > [拡張ログ] で有効になっているときのパラメータ置換が含まれます。

ブラウザの設定

記録中、スクリプト内の非英語の文字がエスケープされた 16 進数として表示されることがあります (たとえば文字列 "Ü&" が "%DC%26" になるなど)。そのような場合は、URL を UTF-8 エンコーディングで送信しないようブラウザを設定することによって修正できます。Internet Explorer では、[ツール] > [インターネット オプション] を選択し、[詳細設定] タブをクリックします。そして、[ブラウザ] セクションにある [常に UTF-8 として URL を送信する] オプションをクリアします。

`web_sjis_to_euc_param` の詳細については、『オンライン関数リファレンス』を参照してください。

Accept-Language ヘッダの言語の指定

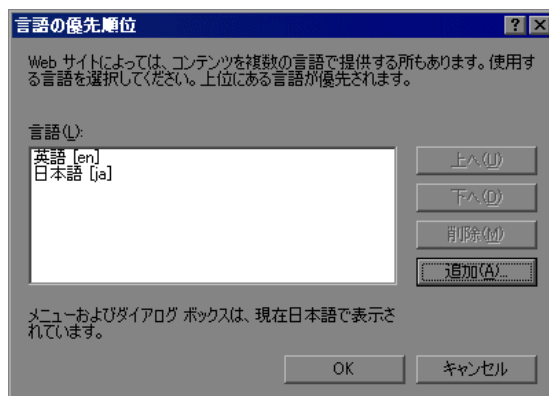
Web スクリプトを実行する前に、現在使用している言語に合わせてページのリクエスト・ヘッダを設定しておくことができます。[実行環境設定] の [インターネットプロトコル] で、**Accept-Language** リクエスト・ヘッダを設定します。このヘッダを通じて、許容されるすべての言語のリストがサーバに提供されます。

Accept-Language ヘッダを設定するには、次の手順を実行します。

- 1 [実行環境設定] ダイアログ・ボックス (F4) を開き、[インターネットプロトコル: プリファレンス] ノードを選択します。
- 2 [詳細] セクションで、[オプション] をクリックして [詳細オプション] ダイアログ・ボックスを開きます。



- 3 **[Accept-Language リクエスト ヘッダ]** オプションを探します。**[値]** カラムで、一覧から必要な言語を選択します。この一覧は、使用しているブラウザの**[インターネット オプション]** の**[言語]** 設定に基づくものです。



これらの設定の詳細については、『**第 2 巻** — プロトコル』の「選択したプロトコルの実行環境の設定」を参照してください。

プロトコルに関する制限

SMTP プロトコル

SMTP プロトコルを MS Outlook や MS Outlook Express を通じて使用している場合は、Vuser スクリプトに記録される日本語テキストが正しく表示されません。ただし、スクリプトの記録と再生は正しく実行されます。

スクリプト名の長さ

COM, FTP, IMAP, SMTP, POP3, REAL, または VBA の VB モードで記録するときは、スクリプト名の長さがマルチバイト文字で 10 文字 (21 バイト) に制限されます。

Quality Center の統合

Quality Center プロジェクトに保存されているスクリプトを VuGen から開く場合や、Quality Center プロジェクトに保存されているシナリオを Controller から開く場合は、「Default」（英語）という名前の新しいテスト・セットを Quality Center プロジェクトに追加します。

付録 31

UNIX プラットフォームでのスクリプトのプログラミング

UNIX プラットフォームを利用する Vuser は、プログラミングによって UNIX プラットフォーム向けのスクリプトを作成できます。プログラミングによってスクリプトを作成するには、テンプレートを使用します。

本章の内容

- ▶ UNIX プラットフォームで実行可能な Vuser スクリプトのプログラミングについて (584 ページ)
- ▶ テンプレートの生成 (585 ページ)
- ▶ Vuser のアクションのプログラミング (586 ページ)
- ▶ Vuser の実行環境設定 (587 ページ)
- ▶ トランザクションとランデブー・ポイントの定義 (592 ページ)
- ▶ スクリプトのコンパイル (593 ページ)

UNIX プラットフォームで実行可能な Vuser スクリプトのプログラミングについて

UNIX プラットフォームで実行可能な Vuser スクリプトを作成する方法は 2 つあります。1 つは VuGen を使用する方法で、もう 1 つはプログラミングを行う方法です。

VuGen VuGen を使って、UNIX プラットフォームで実行可能な Vuser スクリプトを作成できます。Windows 環境でアプリケーションを記録して、それを UNIX で実行できます（記録は UNIX ではサポートされていません）。

プログラミング UNIX だけの環境を使用している場合は、Vuser スクリプトをプログラミングによって作成できます。スクリプトは C 言語または C++ 言語でプログラミングして、ダイナミック・ライブラリとしてコンパイルする必要があります。

本付録では、プログラミングによって Vuser スクリプトを作成する方法について説明します。

プログラミングによってスクリプトを作成するには、Vuser のテンプレートを土台に、より大きな Vuser スクリプトを作成します。テンプレートで提供するものは、次のとおりです。

- ▶ 正しいプログラム構造
- ▶ Vuser API 呼び出し
- ▶ ダイナミック・リンク・ライブラリを作成するためのソース・コードとメイクファイル

テンプレートに基づいて基本となるスクリプトを作成したら、実行時の Vuser 情報と統計値を提供できるようにスクリプトを拡張します。詳細については、第 6 章「Vuser スクリプトの拡張」を参照してください。

テンプレートの生成

VuGen には、テンプレートを作業ディレクトリにコピーするユーティリティが含まれています。このユーティリティは `mkdbtest` と呼ばれているもので、`$M_LROOT¥bin` にあります。このユーティリティを実行するには、次のように入力します。

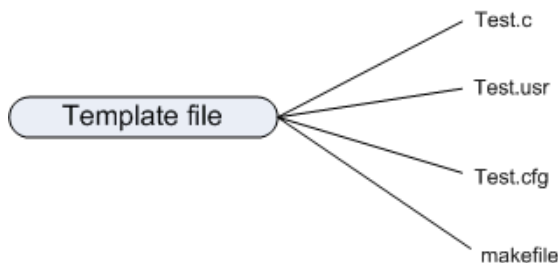
```
mkdbtest name
```

`mkdbtest` を実行すると、`mkdbtest` はテンプレート・ファイル `name.c` を格納するためのディレクトリ `name` を作成します。たとえば、次のように入力するとします。

```
mkdbtest test1
```

`mkdbtest` は、テンプレート・スクリプト `test1.c` を格納するためのディレクトリ `test1` を作成します。

`mkdbtest` ユーティリティを実行すると、4つのファイル (`test.c`, `test.usr`, `test.cfg`, `Makefile`) を格納するためのディレクトリが作成されます。これらのファイルの「`test`」の部分には、`mkdbtest` で指定したテスト名が入ります。



Vuser のアクションのプログラミング

Vuser スクリプトの `test.c`, `test.usr`, `test.cfg` ファイルは、使用する Vuser に応じてカスタマイズできます。

実際の Vuser のアクションをプログラミングして、`test.c` ファイルに挿入します。このファイルには、プログラミングされる Vuser スクリプトに必要な構造になっています。Vuser スクリプトには、`vuser_init`, `Actions`, `vuser_end` の 3 つのセクションが含まれます。

C++ のユーザに対しては、テンプレートは `extern C` を定義します。エクスポートされる関数が不用意に変更されないように、すべての C++ ユーザに対してこの定義を行う必要があります。

```
#include "lrunit.h"
#ifdef __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;
}
#ifdef __cplusplus || defined(cplusplus)
#endif
```

Vuser のアクションをプログラミングして空のスクリプトに直接挿入します。場所は、各セクションの `lr_message` 関数の前です。

`vuser_init` セクションは、初期化中に最初に実行されます。このセクションには、接続情報とログオンの処理を挿入します。`vuser_init` セクションは、スクリプトの実行時に一度だけ実行されます。

Actions セクションは初期化の後に実行されます。このセクションには、Vuser によって実行される実際の操作を挿入します。Actions セクションを繰り返すように Vuser を設定できます (**test.cfg** ファイルを使います)。

vuser_end セクションは、最後、つまり Vuser のすべてのアクションの実行後に実行されます。このセクションには、クリーンアップとログオフの処理を挿入します。**vuser_end** セクションは、スクリプトの実行時に一度だけ実行されます。

注 : LoadRunner は、UNIX のシグナル、SIGHUP、SIGUSR1、SIGUSR2 を送信することによって Vuser を制御します。Vuser プログラムでは、これらのシグナルを使用しないでください。

Vuser の実行環境設定

Vuser の実行環境を設定するには、スクリプトとともに作成する **default.cfg** と **default.usp** に変更を加えます。このような実行環境の設定は、VuGen の実行環境の設定に相当します。詳細については、第 22 章「実行環境の設定」を参照してください。**default.cfg** ファイルには、一般設定、思考遅延時間、およびログに関する設定が含まれています。**default.usp** ファイルには、実行論理とペース設定のための設定が含まれています。

一般オプション

UNIX Vuser スクリプト用の一般オプションが 1 つあります。

ContinueOnError は、エラーが発生しても実行を継続するように Vuser に指示します。このオプションを有効にするには、値を 1 にします。このオプションを無効にするには、値を 0 にします。

次の例では、Vuser はエラーが発生しても実行を継続します。

```
[General]
ContinueOnError=1
```

思考遅延時間のオプション

思考遅延時間のオプションを設定して、スクリプト実行時の Vuser による思考遅延時間の使用法を制御できます。次の図に基づいて、パラメータ Options, Factor, LimitFlag, Limit のパラメータを設定します。

| オプション | Options | Factor | LimitFlag | Limit |
|--------------------|-----------------------|--------------------|-----------|-------|
| 思考遅延時間を無視する | NOTHINK | なし | なし | なし |
| 記録された思考遅延時間を使用 | RECORDED | 1.000 | なし | なし |
| 記録された思考遅延時間を乗じる値 | MULTIPLY | 数値 | なし | なし |
| 記録された思考遅延時間の乱数率を使用 | RANDOM | 範囲 | 最小の割合 | 最大の割合 |
| 記録された思考遅延時間の上限 | RECORDED/ MULTIPLY | 数値 (MULTIPLY 用) | 1 | 秒単位の値 |

実行時に使用する思考遅延時間を制限するには、LimitFlag 変数を 1 に設定し、Limit により思考遅延時間の上限を秒単位で指定します。

次の例では、思考遅延時間を 50% ~ 150% のランダムな割合で乗じるように Vuser に指定しています。

```
[ThinkTime]
Options=RANDOM
Factor=1
LimitFlag=0
Limit=0
ThinkTimeRandomLow=50
ThinkTimeRandomHigh=150
```

ログ・オプション

ログ・オプションは、スクリプトの実行中に簡略または詳細ログ・ファイルを作成するために設定できます。

```
[Log]
LogOptions=LogBrief
MsgClassData=0
MsgClassParameters=0
MsgClassFull=0
```

次の図に基づいて、パラメータ `LogOptions`、`MsgClassData`、`MsgClassParameters`、`MsgClassFull` の変数を設定します。

| ログの種類 | LogOptions | MsgClassData | MsgClassParameters | MsgClassFull |
|-----------------------|-------------|--------------|--------------------|--------------|
| ログを無効にする | LogDisabled | なし | なし | なし |
| 標準ログ | LogBrief | なし | なし | なし |
| パラメータの置換 (のみ) | LogExtended | 0 | 1 | 0 |
| サーバから返された データ (のみ) | LogExtended | 1 | 0 | 0 |
| 詳細トレース (のみ) | LogExtended | 0 | 0 | 1 |
| すべて | LogExtended | 1 | 1 | 1 |

次の例では、`Vuser` はサーバによって返されたすべてのデータと、置換に使用されたパラメータのログを記録します。

```
[Log]
LogOptions=LogExtended
MsgClassData=1
MsgClassParameters=1
MsgClassFull=0
```

反復と実行論理

反復のオプションを設定して、反復を複数回実行したり、反復のペースを制御したりできます。また、アクションの順番と重み付けを手作業で設定することもできます。スクリプトの実行論理と反復の設定を変更するには、**default.usp** ファイルを編集する必要があります。

Actions セクションを複数回反復するように Vuser に指示するには、反復の回数を `RunLogicNumOfIterations` の値として設定します。

反復の間隔（ペース）を指定するには、次の表に従って `RunLogicPaceType` 変数と関連する値を設定します。

| ペースの設定 | RunLogicPaceType | 関連変数 |
|--------------------------|------------------|---|
| すぐに次の反復を開始する | Asap | なし |
| 次の反復を開始する前に、指定した時間だけ待機する | Const | RunLogicPaceConstTime |
| 反復の間隔をランダムにする | ランダム | RunLogicRandomPaceMin, RunLogicRandomPaceMax |
| 反復の後、一定の時間待機する | ConstAfter | RunLogicPaceConstAfterTime |
| 反復の後、ランダムな時間だけ待機する | After | RunLogicAfterPaceMin, RunLogicAfterPaceMax |

次の例は、反復を 4 回実行し、反復の間隔はランダムな長さにするよう Vuser に対して指示する設定です。ランダムな間隔の範囲は 60 ~ 90 秒です。

```
[RunLogicRunRoot]
MerClniTreeFather=""
MerClniTreeSectionName="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"
MerClniTreeSons="Action,Action2,Action3"
RunLogicPaceConstAfterTime="30.000"
```

トランザクションとランデブー・ポイントの定義

VuGen を使用せずに Vuser スクリプトをプログラミングするときには、トランザクションとランデブーを有効にするために、Vuser ファイルを手作業で設定する必要があります。これらの設定は、test.usr ファイルに含まれています。

```
[General]
Type=any
DefaultCfg=Test.cfg
BinVuser=libtest.libsuffix
RunType=Binary

[Actions]
vuser_init=
Actions=
vuser_end=

[Transactions]
transaction1=

[Rendezvous]
Meeting=
```

各トランザクションとランデブーは、usr ファイルに定義する必要があります。トランザクション名を Transactions セクションに追加します（トランザクション名に続けて "=" を指定します）。各ランデブー名を Rendezvous セクションに追加します（ランデブー名に続けて "=" を指定します）。セクションがない場合は、前述のように usr ファイルにセクションを追加します。

スクリプトのコンパイル

テンプレートを修正したら、スクリプト・ディレクトリの中で適切な **Makefile** を使用してスクリプトをコンパイルします。C++ でコンパイルをするときは、gnu コンパイラではなく、ネイティブ・コンパイラを使用する必要があります。コンパイラは、次のダイナミック・ライブラリを作成します。

- ▶ libtest.so (solaris)
- ▶ libtest.a (AIX)
- ▶ libtest.sl (HP)

Makefile の適切なセクションを変更することで、ほかのコンパイラ・フラグやライブラリを指定できます。

汎用のテンプレートを使った作業の場合には、アプリケーションのライブラリとヘッダ・ファイルをインクルードする必要があります。たとえば、アプリケーションで **testlib** というライブラリを使用している場合は、そのライブラリを **LIBS** セクションに指定します。

```
LIBS    = ¥
        -testlib ¥
        -lrun50 ¥
        -lm
```

Makefile の変更後、作業ディレクトリのコマンド・ラインで「**Make**」と入力して、Vuser スクリプト用のダイナミック・ライブラリ・ファイルを作成します。

スクリプトを作成したら、コマンド・ラインでその機能を確認します。

UNIX コマンド・ラインから Vuser スクリプトを実行するには、次のように入力します。

```
mdrv -usr 'pwd' test.usr
```

ここで、**pwd** は Vuser スクリプトが格納されているディレクトリへのフル・パスで、**test.usr** は Vuser ファイル名です。スクリプトがサーバと通信し、要求されているすべてのタスクを実行することを確認します。

スクリプトが正常に機能することを確認した後、スクリプトを自分の環境 (LoadRunner シナリオ, Performance Center 負荷テスト, または Business Process Monitor プロファイル) に組み込みます。詳細については、**HP LoadRunner Controller**, **HP Performance Center**, または **HP Business Availability Center** のドキュメントを参照してください。

付録 32

キーボード・ショートカットの使用

以下は、Virtual User Generator で使用できるキーボード・ショートカットのリストです。

| | |
|-----------|-------------------------------|
| ALT + F8 | 現在のスナップショットを比較 (Web Vuser のみ) |
| ALT + INS | 新規ステップの作成 |
| CTRL + A | すべて選択 |
| CTRL + C | コピー |
| CTRL + F | 検索 |
| CTRL + G | 指定行へ移動 |
| CTRL + H | 置換 |
| CTRL + N | 新規 |
| CTRL + O | 開く |
| CTRL + P | 印刷 |
| CTRL + S | 保存 |
| CTRL + V | 貼り付け |
| CTRL + X | 切り取り |
| CTRL + Y | やり直し |
| CTRL + Z | 元に戻す |
| CTRL + F7 | 記録オプション |
| CTRL + F8 | 関連を検索 |

付録 32・キーボード・ショートカットの使用

| | |
|----------------------|--------------------------------------|
| CTRL + SHIFT + SPACE | 関数の構文を表示 (Intellisense) |
| CTRL + SPACE | Complete ウィザード (関数名を完成) |
| F1 | ヘルプ |
| F3 | 次を検索 (下方) |
| SHIFT + F3 | 次を検索 (上方) |
| F4 | 実行環境設定 |
| F5 | Vuser を実行 |
| F6 | 表示枠間を移動 |
| F7 | EBCDIC 変換ダイアログを表示 (WinSock スクリプトの場合) |
| F9 | ブレークポイントの切り替え |
| F10 | Vuser をステップごとに実行 |

索引

記号

.NET 内のフィルタ
利点 312

A

ABC アイコン 223
Accept-Language リクエスト・ヘッダ 452, 580
Action
 セクション 80
Adobe Reader 15
AMF Vuser スクリプト
 記録モードの設定 305
ANSI 320
AUT 設定 479

C

Citrix, 『第 II 巻 - プロトコル』を参照 21
Citrix Vuser スクリプト
 記録オプション 300
 実行環境の設定 475
Citrix 実行環境の設定
 同期化設定 477
COM, 『第 II 巻 - プロトコル』を参照 21
Controller
 シナリオ 162
[CORBA 記録オプション] ダイアログ・ボック
 クス 369
CtLib
 オプション 303
 サーバ・メッセージのログの記録 426
CtLib, 『第 II 巻 - プロトコル』を参照 21
C 関数
 Vuser スクリプトでの使用 53
 追加キーワード 530
 デバッグ用 530
C 言語サポート
 インタプリタ 54

D

Date/Time (日時), パラメータ値 267
DBCS 320
DLL, Vuser スクリプトからの呼び出し 569
DLL のロード
 概要 569
 グローバル 572
 ローカル 570
DNS, 『第 II 巻 - プロトコル』を参照 21
DNS キャッシング
 Web 453
DOM のメモリ割り当て 457
DOM メモリ割り当て 457
DSL 439

E

EJB
 コード生成オプション 288
EUC-JP エンコーディング
 記録オプション 332
 設定 577
EUC エンコードされた Web ページ 392

F

FTP, 『第 II 巻 - プロトコル』を参照

G

GUI Vuser スクリプト
 ツール 26
gzip 461

H

history オブジェクト, サポート 456
HP ソフトウェア Web サイト 18
HP ソフトウェア・サポート Web サイト 17
HTML タグ・オブジェクト 348

HTML ビュー (Web スナップショット) 42
[HTML ファイルへエクスポート] ダイアログ・ボックス 181

HTML ベース・モード 380

HTML 用の圧縮 (gzip) 461

HTTP

バッファ・サイズ (Web) 454

HTTP 記録モード, WAP 494

I

If-Modified-Since ヘッダ
Web 447

IMAP, S 『第 II 巻 - プロトコル』を参照 21

IntelliSense 55

ISDN 439

i モード 『第 II 巻 - プロトコル』を参照 21

J

Jacada, 『第 II 巻 - プロトコル』を参照 21

Java, 『第 II 巻 - プロトコル』を参照 21

Java Vuser

記録オプション, Java VM 359

記録オプション, シリアル化 364

記録オプション, 相関 366

クラスパスの実行環境の設定 490

実行環境の設定 -Java VM 489

Java Vuser スクリプト

記録オプション (CORBA オプション) 369

シリアル化オプション 364

相関オプション 366

デバッグ・オプション 367

レコーダ・オプション 362

Java Vuser (すべて)

実行環境の設定 489, 491

Java Vuser, 『第 II 巻 - プロトコル』を参照 21

Java 仮想マシン

記録オプション 359

実行環境の設定 489

Jscript 373

K

Keep-Alive 接続, Web 451

Kerebros

認証 455

L

LDAP, 『第 II 巻 - プロトコル』を参照

Load Generator 名, パラメータ値 270

LoadRunner Analysis ユーザーズ・ガイド 16

LoadRunner Monitor Reference 16

LoadRunner インストール・ガイド 16

lrbat ユーティリティ 510

M

MAPI, 『第 II 巻 - プロトコル』を参照 21

Media Player, 『第 II 巻 - プロトコル』を参照 21

META 更新 454

Microsoft .NET Vuser スクリプトの記録

記録オプション 312

実行環境の設定 479

mkdbtimpl スクリプト (UNIX) 585

MMS Vuser スクリプト

実行環境の設定 500

MS Query 244

N

NTLM

セキュリティ 454

認証 101

O

Oracle, 『第 II 巻 - プロトコル』を参照 21

Oracle NCA Vuser スクリプト

実行環境の設定 482

Oracle Web Applications 11i Vuser スクリプト

GUI ベースの詳細オプション 342

Oracle アプリケーションのデバッグ 533

P

PeopleSoft, 『第 II 巻 - プロトコル』を参照

PeopleSoft Enterprise Vuser スクリプト

GUI ベースの詳細オプション 342

PeopleSoft-Tuxedo Vuser 548

Performance Center

Vuser スクリプトの管理 208

接続 209

POP3, 『第 II 巻 - プロトコル』を参照 21

Q

Quality Center

Vuser スクリプトの管理 189

Vuser スクリプトを開く 194

スクリプトの管理 189

スクリプトの保存 196

接続 190

切断 193

バージョン管理 197

バージョン・コントロール 197

プロジェクトへの接続 183

Quality Center からの切断 193

Quality Center プロジェクトからスクリプトを開く 194

Quality Center への接続 183

R

Radius

サポート 498

実行環境の設定 (WAP) 498

Random (ランダム) 方式でのパラメータの割り当て 251

RDP, 『第 II 巻 - プロトコル』を参照

RDP Vuser スクリプト

記録オプション 289

実行環境の設定 470

RDP エージェント

記録オプション 293

RTE, 『第 II 巻 - プロトコル』を参照 21

RTE Vuser スクリプト

実行環境の設定 491

run_db_vuser シェル・スクリプト 159

S

S_SSA_ID テーブル 547

SAP, 『第 II 巻 - プロトコル』を参照 21

SAPGUI Vuser スクリプト

共通記録オプション 325

記録オプションの設定 325

コード生成記録オプション 326

自動ログオン記録オプション 327

SAP-Web Vuser スクリプト

記録オプション 328

Sequential (順次) 方式でのパラメータの割り当て 250

Service Test 28

setInterval, setTimeout のしきい値 456

Siebel

2 層タイプのスクリプト作成ヒント 544

ベース 36 キーの値 547

Siebel, 『第 II 巻 - プロトコル』を参照 21

Siebel のサフィックス値 544

SJIS (Shift Japan Industry Standard) 392

SMTP, 『第 II 巻 - プロトコル』を参照 21

SQL ステートメント 245

T

TestDirector, Quality Center を参照 189

U

Unique (一意) の値のパラメータの割り当て 251

UNIX

コマンド・ライン 159

URL ベース・モード 380

UTF-8 変換

記録オプション 332

実行環境の設定 453

設定 577

V

VBA 実行環境の設定 435

VBA リファレンス 436

verify_generator 159

Virtual User Generator, VuGen を参照

Visual Basic

Vuser スクリプト 505

スクリプト・オプション 373

Visual Studio 505

Visual C, Visual Studio の使用 505

VM (仮想マシン) 359

VuGen

Vuser スクリプトの記録 79

Vuser スクリプトの実行 51

開始 30

環境オプション 34

紹介 29

ツールバー 96

Vuser

紹介 22

vuser_init, vuser_end セクション 80

Vuser ID, パラメータ値 275

Vuser 関数

- lr (C 関数) 52
- 一般 (C) 52
- オンライン関数リファレンスを参照
- 外部, ユーザ定義 569
- 構文 56
- 単語の自動補完 55
- ヘルプ 54

Vuser 情報の取得 115

Vuser スクリプト

- Performance Center のアップロード/ダウンロード 207
 - Performance Center へのアップロード 210
 - Quality Center の統合 189
 - UNIX, コンパイル 593
 - UNIX での作成 583-594
 - UNIX での実行 159
 - Zip からのインポート 91
 - Zip からの作業 91
 - 拡張 107
 - 関数の追加 107
 - コマンド・プロンプトからの実行 158
 - コメント, 挿入 114
 - 再生成 103
 - 実行 141
 - 実行環境の設定 411, 469
 - 実行環境の設定 -Java 489-491
 - シナリオへの統合 161
 - 生成言語の選択 371
 - セクション 80
 - タイプ, Vuser のタイプを参照
 - デバッグ機能 149
 - トランザクション 110
 - バージョンの履歴 202
 - パラメータ化 219
 - 表示 36
 - 開く 91
 - プログラミング 583-594
 - ランデブー・ポイント 113
- Vuser スクリプトの記録
- 概要 79
- Vuser スクリプトの実行
- VuGen の使用 141
 - ステップごと 149
 - 表示実行モード 143
- Vuser スクリプトのセクション 80

Vuser スクリプトのデータベース 第2巻 - プロトコルを参照してください。

Vuser スクリプトの同期化
ランデブー 122

Vuser タイプ, 『第II巻 - プロトコル』を参照 21

Vuser の再生成
全プロトコル 103

Vuser のタイプ
一覧 24

W

WAP, 『第II巻 - プロトコル』を参照 21

WAP Vuser スクリプト
デバッグ情報 460

Wdiff 138

Web (Click and Script) Vuser スクリプト

- 結果サマリ・レポート 165
- 実行時ビューア 154
- デバッグ機能の有効化 145
- デバッグ・ツール 154
- ログ表示オプション 145

Web Vuser スクリプト

- 結果サマリ・レポート 165
- 記録オプション 379, 287
- 実行環境の設定 438
- 実行時ビューア 154
- デバッグ機能の有効化 145
- デバッグ・ツール 154
- 内容のフィルタリング 335
- ユーザ定義ヘッダ 333
- ログ表示オプション 145
- ログ表示オプションの設定 154

web_set_user 関数の生成 101

Web (Click and Script)

- 一般オプション 455
- タイマ 456
- ナビゲータのプロパティ 457
- メモリ管理 457
- 履歴オプション 456

Web (Click and Script) Vuser スクリプト

- GUI ベースの詳細オプション 342
- Web ページのコンテンツ・チェック 462
- 記録オプションの設定 341
- 記録済み関数の表示 340
- 記録レベルの選択 380
- 内容のフィルタリング 335

ユーザ定義ヘッダ 333
 Web Vuser スクリプト, 『第 II 巻 - プロトコル』
 を参照 21
 Web イベント記録の設定 345-356
 カスタマイズ 348-356
 標準 346-347
 [Web イベント記録の定義] ダイアログ・ボッ
 クス 347
 Web サービス, 『第 II 巻 - プロトコル』を参照 21
 Web サービス Vuser スクリプト
 レポート・ツール 184
 Web パフォーマンス・グラフ
 Web Vuser の生成 450
 WinInet エンジン (インターネット・プロトコ
 ル) 450
 WinSock, 『第 II 巻 - プロトコル』を参照 21

X

XML
 属性, での作業 519
 XML API
 プログラミング 511
 XML パラメータ
 作成 255
 XML ビューア 50

Z

Zip ファイル
 エクスポート 97
 作業 91
 使用 99
 zlib ヘッダ 452

あ

アクション
 インポート 100
 オープン・モードの設定 45
 順序の並べ替え 101
 マルチ・アクションの記録 94
 アクションの分割 374
 アセンブリを無視 317
 値セット
 オプションの要素 257
 作成 257
 圧縮ヘッダ, 要求 452
 アップデート, ドキュメント 18

アップロード
 スクリプトを Performance Center へ 210

い

一意カラム名 544
 一意の数, パラメータ値 272
 一時停止, Vuser 147
 一致する次の文字列を置換コマンド 231
 一般オプション
 環境タブ 34
 再生タブ 144
 すべての Vuser 236
 ダイアログ・ボックス 237
 パラメータ化タブ 235
 表示タブ (Web のみ) 146
 移動コマンド 154
 イベント記録設定のカスタマイズ 348-356
 手順 349
 リッスン・イベントの追加 351
 リッスン条件の指定 352
 イベント記録の設定 345-356
 設定レベル 346
 リセット 355
 レベルのカスタマイズ 348
 [印刷] ダイアログ・ボックス, テスト結果
 ウィンドウ 178
 [印刷プレビュー] ダイアログ・ボックス 179
 インポート
 アクション 100
 データベースからのデータ 243

う

ウィザード, ワークフロー 65
 ウィンドウ名, Citrix 295
 ウィンドウを並べて表示 155

え

エージェント実行環境の設定 (RDP) 474
 エディタのフォント 34
 エディタ, フォントの設定 34
 エラー時も処理を継続する
 ローカル 121
 エラー処理
 グローバルまたはローカルの設定 121
 実行環境の設定 430
 エラー, スナップショットを生成する 431

索引

- エラーでスナップショットを生成する 431
- エラーでも処理を継続する
 - グローバル 430
- エラーの一致, 制限 454
- [エラーメッセージ] ダイアログ・ボックス 120
- エンコード, EUC 394
- エンコード方式
 - EUC 392
 - パスワード 126
- エンジン, 記録 332

- お
- オプション
 - CtLib 303
 - lrd ログ 302
 - 記録 (RTE) 321
- オプション・タブ 33
- オプションの要素
 - 除外 257
- オンライン・ドキュメント 15
- オンライン・ブラウザ 154
- オンライン・リソース 17

- か
- 外部オブジェクト 309, 310
- 外部関数 569
- 拡張結果設定 303
- 拡張ログ・オプション 424
- 仮想ユーザ, 定義 22
- 画面のプロパティ, 実行環境の設定 457
- 環境タブ 34
- 環境の実行環境の設定, .NET 479
- 関数
 - lr (C 関数) 52
 - 構文 56
 - 単語の自動補完 55
 - ヘルプ 54
- 関数構文の表示 56
- 関数の表示 55
- 管理
 - Quality Center のスクリプトのバージョン 197

- き
- キーボード・タブ 33
- キーボードのショートカット
 - 記録オプション 377
 - 実行環境の設定 412
 - ショートカット・リスト 595
- キーワード, 追加 530
- 既存のパラメータで置換コマンド 231
- 基本コードの生成オプション
 - RDP 290
- 基本のイベント記録設定レベル 346
- キャッシュ
 - 更新バージョンのチェック 447
 - 反復ごとにクリア 447
- 行カウント, 取得 541
- 記録
 - ステータス, オプション 353
- 記録エンジン 332
- 記録オプション
 - .NET Vuser 312
 - .NET の記録 313
 - CORBA オプション 369
 - Java 言語 357–368
 - RDP 289
 - RDP エージェント 293
 - RDP ログイン 289
 - RTE 321
 - RTE の設定 320
 - SAPGUI 326
 - Web 287, 379
 - キーボードのショートカット 377
 - 基本コードの生成, RDP 290
 - 記録 (Web) 393
 - 高度なコード生成, RDP 292
 - コード生成, AMF 309
 - コード生成, Citrix 296
 - コード生成, Flex 310
 - 詳細 (Web, ワイヤレス) 331
 - スクリプト (FTP, COM, Mail) 373
 - データベース 301
 - デバッグ (Java) 367
 - ポートの割り当て 396
 - レコーダ (Java) 362
 - ログイン (Citrix) 297
- 記録オプション『第 II 巻 - プロトコル』を参照
- [記録開始] ダイアログ・ボックス 92
- 記録ボタン 92
- 記録ログ・タブ 98

く

区切り文字, カラム
 データ・テーブル内 248
 データ・ファイル内 247
 クライアント・エミュレーション実行環境の
 設定 482
 クラスパス
 実行環境の設定 490
 グラフ
 Web Vuser スクリプトでの有効化 450
 グリッド
 非表示 155
 グループ名, パラメータ値 269
 グローバル・ディレクトリ 236

け

形式
 パラメータ化用 277
 ゲートウェイの設定 (WAP) 494
 結果サマリ・レポート
 ツリーの分岐つりのふんき 167
 Web サービス Vuser 184
 概要 165
 ユーザ定義メッセージの送信 187
 結果詳細タブ, テスト結果ウィンドウ 169
 [結果ディレクトリの指定] ダイアログ・ボッ
 クス 144
 結果パラメータ, XML の保存 527
 言語エンコーディング 577
 言語ヘッダ 580
 検索
 ファイル内の検索 156
 [検索] ダイアログ・ボックス
 テスト結果 177
 [検索と置換] ダイアログ・ボックス 231
 検証チェック
 Web 449

こ

高位のイベント記録設定レベル 347
 更新方法
 パラメータの使用方法 278
 パラメータの割り当て 252
 高度なコード生成オプション
 RDP 292
 構文, 関数の表示 56

コード生成オプション

AMF 309
 Citrix 296
 EJB 288
 Flex 310
 SAPGUI 326
 コマンド・タブ ([ユーザ定義] ダイアログ・
 ボックス) 31
 コマンド・プロンプト 158
 コマンド・ライン引数
 C Vuser スクリプトでの読み取り 124
 UNIX Vuser スクリプト 159
 コメント
 Vuser スクリプトへの挿入 114
 画面ヘッダ・コメント (RTE) 323
 [コメントの挿入] ダイアログ・ボックス 114
 コメントを挿入ボタン 114
 コンテンツ・チェック
 エラーの制限 454
 設定 (Web) 462
 [コンテンツ・タイプのフィルタ] ダイアロ
 グ・ボックス 336
 コンテンツ・タイプのフィルタリング (Web) 335
 コンパイル
 UNIX の Vuser スクリプト 593
 コンポーネント
 実行結果, 実行結果を参照

さ

サーバ・サイド圧縮を受け付ける 452
 最近使用したテストのリストからテストを開く 195
 再生タブ, [一般オプション] ダイアログ・
 ボックス 144

サムネイル

注釈の追加 48
 名前の変更 48
 表示 44
 ワークフロー・ウィザード 46

し

思考遅延時間
 しきい値, データベース 301
 実行環境の設定 427
 挿入 123
 定義 427
 [思考遅延時間] ダイアログ・ボックス 123

- 持続的な接続, Web 451
- 実行環境の設定
 - .NET 環境 479
 - Java 489-491
 - MMS 500
 - Oracle NCA 482
 - Radius (WAP) 498
 - RDP 470
 - VBA(Visual Basic Apps) 435
 - エージェント (RDP) 474
 - 画面のプロパティ 457
 - キーボードのショートカット 412
 - クライアントのエミュレーション (Oracle NCA) 482
 - ゲートウェイ・ノード (WAP) 494
 - コンテンツ・チェック・ノード (Web) 462
 - 思考遅延時間 427
 - 実行論理 413
 - 手動で設定 587
 - 詳細 (RDP) 472
 - ショートカット 412
 - 設定 (RDP) 470
 - 全プロトコル 411, 469
 - 速度のシミュレーション 438
 - その他 430
 - その他の属性 429
 - ダイアログ・ボックス 412
 - デバッグ情報 (WAP) 460
 - 同期化 (RDP) 471
 - ネットワーク 437
 - ブラウザ・エミュレーション・ノード 444
 - プリファレンス - 詳細 450
 - プリファレンス (インターネット・プロトコル) 449
 - プロキシ (インターネット・プロトコル) 439
 - ペースの設定ノード 419
 - ログ・ノード 422
- 実行環境の設定, 『第 II 巻 - プロトコル』を参照 21
- 実行環境の追加属性の設定 429
- 実行結果 165
 - HTML ヘエクスポート 181
 - 印刷 178
 - 印刷前のプレビュー 179
 - 検索 176, 177
 - スキーマ 183
 - テスト結果ウィンドウ 167
 - 特定の実行の表示 171
 - 表示のカスタマイズ 183
 - フィルタリング 175
 - 実行結果の分析, 実行結果を参照
 - 結果 実行結果を参照
 - 実行コマンド 146
 - 実行時の完全トレース 424
 - 実行時ビューア
 - VuGen で有効化 154
 - 表示オプション 145
 - 実行セッション
 - 結果の印刷 178
 - 生成ログ・タブ 98
 - 実行論理の設定 413
 - 自動回復 34
 - 自動検出, プロトコル 402
 - 自動トランザクション
 - トランザクション名 450
 - 有効化 434
 - 自動プロキシ設定スクリプト 441
 - シナリオ
 - VuGen で作成 162
 - Vuser スクリプトの統合 161
 - パラメータ・シミュレーション 282
 - 修飾キー 290
 - 出力ウィンドウ
 - 再生タブ 147
 - 実行時データ・タブ 148
 - 非表示 147
 - 表示 / 非表示 155
 - 出力パラメータ
 - XML 527
 - 選択 233
 - [出力メッセージ] ダイアログ・ボックス 120
 - [詳細 GUI] ダイアログ・ボックス 342
 - 詳細記録オプション 331
 - 詳細実行環境の設定 (RDP) 472
 - 消失したスクリプトの回復 34
 - シリアル化オプション 365
 - [新規カラムの追加] ダイアログ・ボックス 242
 - 診断
 - VuGen で有効化 435

す

- スキーマ, 実行結果の 183
- スクリプト生成言語 371
- スクリプト生成レベル, RDP 290, 292
- スクリプトのエクスポート
 - Word ファイルに 129
 - Zip ファイル 96
- スクリプトのディレクトリ, 開く 97
- スクリプト・ビュー
 - 表示 36
 - 開く 45
- スクリプト・フォルダ, 開く 97
- スクリプト・モード 45
- スクリプト・レベル, RDP 290, 292
- ステップ・ボタン 149
- スナップショット
 - Citrix の記録での保存 296
 - Web ページ 39
 - エラーで生成する 431
 - 再生時のスナップショットをローカルに保存する 451
 - テスト結果ウィンドウ 166
- すべて閉じるコマンド 155

せ

生成

- 自動同期化ポイント (RDP) 291
- マウス移動呼び出し 291

接続

- VuGen から Quality Center 190
- 接続試行回数, 変更 (RTE) 491
- 接続プール 480
- 設定実行環境の設定 (RDP) 470
- 設定レベル
 - カスタマイズ 348–356
 - 標準 346–347

そ

相関

- Siebel のデバッグに関するヒント 544
- 概要 133
- 関数 (C) 135
- 関数 (Java) 137
- 既存パラメータの変更 140
- 記録オプション -Java 366
- スクリプト言語オプション 376

相関オプション, スクリプト記録オプション 376

- 相関タブ, 記録オプション 329
- 速度のシミュレーション設定 438
- その他の実行環境の設定 430

た

タイムアウト

- Citrix 接続 477
- HTTP 要求 452
- WAP 接続 452
- タイム・スタンプ (データベース) 303
- ダウンロード
 - Performance Center から VuGen への 213
 - ダウンロード・フィルタ 458
 - 多国語のサポート 573–582
 - パラメータ・ファイル 575
- タスク表示枠 66
- 単語の補完 55

ち

- チェックアウト操作, 取り消し 205
- チェックイン・コマンド 200
- 中位のイベント記録設定レベル 346
- 重複キー違反
 - Oracle, MSSQL 541
 - Siebel 544

つ

- ツール・タブ 32
- ツールバー・タブ 31
- ツリー・ビュー
 - ステップの挿入 39
 - すべての Vuser 38

て

- 定義, パラメータのプロパティ
 - 一般 228
 - データ・ファイル 246
 - テーブル 248

停止, Vuser 147

- データ・ウィザード, SQL ステートメント 245
- データ・グリッド
 - 有効化 155
- データ・テーブルのパラメータ
 - 行とカラムの追加 242

- データ・ウィザードを使用したデー
タ・ソースのインポート 241
- データ・ソースの作成 240
- データ・ソースの選択 240
- データベースからのデータのインポート 243
- 編集 242
- データの割り当て方法, パラメータ化 250
- データ・ファイル
 - パラメータ化用 226
- データ・ファイルのパラメータ
 - 行とカラムの追加 242
- データ・ウィザードを使用したデー
タ・ソースのインポート 241
- データ・ソースの作成 240
- データ・ソースの選択 240
- データベースからのデータのインポー
ト 243
- 編集 242
- データベース Vuser スクリプト
 - ヒント 539
- データベース記録オプション 301
- [データベースクエリウィザード] ダイアロ
グ・ボックス 243
- テーブル
 - パラメータ化用 227
- テーブル・アイコン 225
- テキストの暗号化 125
- テキストの同期化
 - Citrix 296
- テキストの復号 125
- テスト
 - 実行結果を参照
- テスト結果
 - Web サービス Vuser 184
 - 表示 170
- テスト結果, Web サービス 184
- テスト結果ウィンドウ 167
 - 結果詳細タブ 169
 - 実行結果ツールバー 169
 - 実行結果ツリー 168
 - テーマ 169
 - ルック・アンド・フィール 169
- テスト結果ツールバー, テスト結果ウィンド
ウ 169
- テスト結果ツリー 168
- テスト結果レポート

- ユーザ定義メッセージの送信 187
- デバイス名 (RTE) 492
- デバッグ
 - Oracle アプリケーション 533
 - Web Vuser での有効化 145
 - 再生中 149
 - 情報の取得 (WAP) 460
 - データベース・アプリケーション 531
 - デバッグ機能の有効化 145
 - デバッグ・レベルの設定 425
- デバッグ記録設定 (Java) 367
- [デバッグメッセージ] ダイアログ・ボックス 119
- テンプレート
 - C 言語を使ったプログラミング, UNIX 585
 - Visual Basic を使用するプログラミング 508
 - Visual C で開く 507
 - 新規作成 90
 - ユーザ定義 88

と

- 同期化実行環境の設定 (RDP) 471
- 同期関数
 - Citrix テキストの生成 296
- 動作, DHTML 352
- 同時実行の設定, Web サービス 480
- ドキュメント, オンライン 15
- ドキュメントのアップデート 18
- トラフィックの転送 401
- トラブルシューティング
 - 2 層データベース 539
 - Oracle アプリケーション 533
 - Siebel Vuser 544
 - VuGen 529
- トラブルシューティング & ナレッジベース 17
- トランザクション
 - Oracle DB のブレイクダウンの制限 81
 - Web Vuser 434
 - 入れ子 75, 112
 - ウィザード・ワークフロー 70
 - エディタ 70
 - 自動, Web Vuser スクリプト 434
 - 出力ログ内 147
 - 挿入 110
- トランザクション・エディタ 46, 70
- [トランザクション開始] ダイアログ・ボックス 110

な

内部データ, パラメータ化 227, 267
ナレッジベース 17

に

認証

Quality Center への接続 183
認証, 記録時 101
認証の再試行時の思考遅延時間 454

ね

ネットワーク設定 439
ネットワークのバッファ・サイズ (インター
ネット) 454

は

バージョン・コントロール 197
 テストのチェックイン 200
バージョン・コントロールのためのスクリプ
トのチェックイン 200
バージョンの履歴 202
[バージョンの履歴] ダイアログ・ボックスの
使用方法 202

配列

XML での複製 260
[パスワードエンコーダ] ダイアログ・ボック
ス 126

パスワード, エンコーディング 126

バブリング, Web イベント 353

パラメータ

削除 234
スクリプト・ビューでの作成 223
ツリー・ビューでの作成 223
パラメータ・リストを使用した作成 234
変更 234

パラメータ化

COM, .NET, VB 222
Java 223
UTF-8 エンコードされた 575
XML 227
一意の値を使った更新 251
概要 220
括弧のスタイル 225
既存パラメータの変更 140
既定値を復元 232

グローバル・ディレクトリ 236

シードによる乱数シーケンス 251

シミュレーション 279

新規パラメータの作成 223

スクリプト内の括弧 235

制限 221

データ・ファイル 226

データ・ファイルのプロパティ設定 246

テーブル 227

テーブルのプロパティ設定 248

内部データ・タイプの形式 277

内部データの使用 227

内部データのプロパティ 267

名前, パラメータ 224

パラメータ値の更新 278

パラメータ・リスト 233

ファイルおよびテーブルからの値の割
り当て 250

プロパティの定義 228

元に戻す (Web) 232

ユーザ定義関数 228

ユーザ定義関数の使用 276

パラメータ化オプション 235

パラメータ化で使用する括弧 235

パラメータ化, 長い文字列の置換 375

パラメータ形式

削除 278

追加 277

元に戻す 278

パラメータ・タイプ

Date/Time (日時) 267

Load Generator 名 270

User ID 275

XML 227

一意の数 272

グループ名 269

出力 233

データ・テーブル 266

データ・ファイル 226, 266

テーブル 227

内部データ 266, 267

内部データ・タイプ 227

反復回数 270

ユーザ定義関数, 概要 266

ユーザ定義関数, 形式 228

ユーザ定義関数, プロパティ 276

索引

- 乱数 271
- 理解 226
- パラメータの既定値の復元 232
- [パラメータの選択または作成] ダイアログ・ボックス 223
- [パラメータのプロパティ] ダイアログ・ボックス 228
- パラメータを元に戻すコマンド 232
- 半径, 同期化 291
- ハンドラ 352
- 反復
 - 実行環境の設定 419
 - 反復ごとのパラメータ更新 278
- 反復回数, パラメータ値 270
- ひ
- 比較
 - Vuser 138
- 比較ツール, 設定 35
- ビューア, XML 50
- 表示実行
 - 定義 143
 - 有効化 144
- 表示タブ, 一般オプション 146
- 標準的なイベント記録設定 346-347
- 標準ログ・オプション 424
- 標準ログの実行環境の設定 424
- ヒント
 - Siebel 固有 544
 - データベース関連 539
- ふ
- ファイル, スクリプトへの追加 127
- フィルタリング
 - .NET Vuser 312
 - ダウンロードされたリソース 458
 - 内容タイプ (Web, ワイヤレス) 335
- プール, 接続 480
- ブックマーク
 - Vuser スクリプト 153
- ブラウザ・キャッシュ (Web およびワイヤレス) 447
- ブラウザのエミュレーション設定, Web 444
- プラグマ・モード 482
- プリファレンス, 実行環境の設定 449
- ブレークポイント 149
- ブレークポイント・マネージャ 151
- プロキシ・サーバ
 - 実行環境の設定 (インターネット) 439
- プログラミング
 - Visual C テンプレートの使用 507, 508
 - Visual Studio による 505
 - Vuser アクション 586
 - テンプレートの使用 585
- ブロック・サイズ, Vuser の値の割り当て 247, 262
- プロトコル
 - アドバイザ 59
 - プロトコル『第 II 巻 - プロトコル』を参照 21
 - プロトコル・アドバイザ 59
 - プロトコル検出 59
 - プロトコル, サポートされている一覧 87
 - プロパティ, パラメータ
 - データ・ファイルに対する定義 246
 - テーブルのための定義 248
 - 定義 228
- へ
- ページ・ビュー (Web スナップショット) 42
- ペースの設定 419
- ヘッダ
 - ユーザ定義 333
 - リスクキー 333
- ヘッダ・ファイル 56
- ほ
- ポートの割り当て設定 396
- ホスト・サフィックス, フィルタリング 458
- ま
- マルチ・アクション 82
- マルチ・スレッド 433
- マルチ・プロトコル 82
- み
- 未処理のキーボードおよびマウス呼び出し (RDP) 291
- め
- メール・サービス『第 II 巻 - プロトコル』を参照 21

メッセージ

出力への送信 116

メモリ管理 457

も

文字エンコーディング 577

文字セット, RTE 320

文字列, 長い文字列をパラメータで置換 375

モデム速度, 実行環境の設定 439

ゆ

ユーザ・エージェント・ブラウザのエミュレーション 444

[ユーザ定義 Web イベント記録の設定] ダイアログ・ボックス 349

ユーザ定義関数のパラメータ

形式 228

プロパティ 276

[ユーザ定義] ダイアログ・ボックス

オプション・タブ 33

キーボード・タブ 33

コマンド・タブ 31

ツール・タブ 32

ツールバー・タブ 31

ユーザ定義の要求 392

ユーザ定義ヘッダ, Webおよびワイヤレス用 333

ら

ライセンス情報 25

ライブラリ, スクリプトの作成 436

乱数, パラメータ値 271

ランデブー

[ランデブー] ダイアログ・ボックス 113

ランデブー・ポイント, 挿入 113

り

リソース

Web 記録で除外 337

リソース以外 337

れ

レポート・ツリー, 結果サマリ (Web) 167

ろ

ログ

詳細レベルの設定 - PC 423

詳細レベルの設定 - UNIX 589

ログ記録オプションの無効化 423

ログ, 実行環境の設定 422

ログのキャッシュ・サイズ 423

ログ表示オプション (Web) 145

[ログメッセージ] ダイアログ・ボックス 118

わ

ワークフロー・ウィザード 65

ワークフロー・ウィザード, 復元 77

ワイヤレス, 『第 II 巻 - プロトコル』を参照

ワイヤレス Vuser スクリプト

ユーザ定義ヘッダ 333

割り当て, Vuser の値

データ・テーブル 250

データ・ファイル 247, 262

