# LoadRunner

## 创建 Vuser 脚本

7.8 版

LoadRunner 创建 Vuser 脚本 （Windows 和 UNIX）， 7.8 版

本手册及附带的软件和其他文档受美国和国际版权法保护，并且只能依据附带的许可协议使用。软件及 Mercury Interactive Corporation 其他产品和服务的功能包含于下列一项或多项专利中：美国专利号 5,701,139、 5,657,438、 5,511,185、 5,870,559、 5,958,008、 5,974,572、 6,138,157、 6,144,962、 6,205,122、 6,237,006、 6,341,310、 6,360,332, 6,449,739、 6,470,383、 6,477,483、 6,560,564 和 6,564,342。以及其他待批准的专利。保留所有权利。

ActiveTest、ActiveTune、Astra、FastTrack、Global SiteReliance、LoadRunner、Mercury Interactive、Mercury Interactive 徽标、 Open Test Architecture、 Optane、 POPs on Demand、 ProTune、 QuickTest、 RapidTest、 SiteReliance、 SiteRunner、 SiteScope、 SiteSeer、 TestCenter、 TestDirector、 TestSuite、 Topaz、 Topaz AIMS、 Topaz Business Process Monitor、 Topaz Client Monitor、 Topaz Console、 Topaz Delta、 Topaz Diagnostics、 Topaz Global Monitoring Service、 Topaz Managed Services、 Topaz Open DataSource、 Topaz Real User Monitor、 Topaz WeatherMap、 TurboLoad、 Twinlook、 Visual Testing、 Visual Web Display、 WebTest、 WebTrace、 WinRunner 和 XRunner 是 Mercury Interactive Corporation 或其完全控股的 Mercury Interactive (Israel) Ltd. 在美国和 / 或其他国家 （地区）的商标或注册商标。

所有其他的公司、品牌和产品名均为其各自所有者的注册商标或商标。 Mercury Interactive Corporation 不承担说明各商标所有者的责任。

Mercury Interactive 上海代表处
地址：上海市南京西路 1266 号恒隆广场 35 层 3501 室 (200040)
电话：8621 - 62882525
传真：8621 - 62883030

© 2003 Mercury Interactive Corporation，保留所有权利

如果您对此文档有任何意见或建议，请通过电子邮件发送至 documentation@merc-int.com。

LRDBUG7.8SC/01

# 目录


**欢迎使用 LoadRunner** ..... xix
联机资源 ..... xix
LoadRunner 文档集 ..... xx
使用 LoadRunner 文档集 ..... xx
版式约定 ..... xxii

**第 I 部分：介绍 VUSER 脚本**

**第 1 章：开发 Vuser 脚本** ..... 3
介绍 Vuser ..... 3
查看 Vuser 类型 ..... 5
开发 Vuser 脚本 ..... 6
使用本指南 ..... 8

**第 II 部分：使用 VUGEN**

**第 2 章：介绍 VuGen** ..... 11
关于 VuGen ..... 11
使用 VuGen 录制 Vuser 脚本 ..... 12
启动 VuGen ..... 13
了解 VuGen 环境选项 ..... 14
设置环境选项 ..... 15
使用 VuGen 运行 Vuser 脚本 ..... 15
了解 VuGen 代码 ..... 16
使用 C Vuser 函数 ..... 18
获得函数帮助 ..... 21

**第 3 章：使用 VuGen 进行录制** ..... 25
关于使用 VuGen 进行录制 ..... 25
Vuser 脚本部分 ..... 26
新建虚拟用户脚本 ..... 28
添加和删除协议 ..... 31
选择虚拟用户类别 ..... 32
导入操作 ..... 38
重新生成 Vuser 脚本 ..... 39
**第 4 章：设置脚本生成首选项** ..... 43
关于设置脚本生成首选项 ..... 43
选择脚本语言 ..... 44
应用基本选项 ..... 44
了解关联选项 ..... 45
设置脚本录制选项 ..... 46
**第 5 章：配置端口映射** ..... 47
关于配置端口映射 ..... 47
定义端口映射 ..... 48
添加新的服务器项 ..... 49
设置自动检测选项 ..... 51
设置端口映射录制选项 ..... 52
**第 6 章：增强 Vuser 脚本** ..... 55
关于增强 Vuser 脚本 ..... 56
将事务插入到 Vuser 脚本 ..... 57
将集合点插入到 Vuser 脚本 ..... 58
将注释插入到 Vuser 脚本 ..... 60
获取 Vuser 信息 ..... 61
将消息发送到输出 ..... 61
处理 Vuser 脚本在执行期间出现的错误 ..... 64
同步 Vuser 脚本 ..... 66
模拟用户思考时间 ..... 66
处理命令行参数 ..... 67
加密文本 ..... 68
在 Vuser 脚本中使用 C 函数 ..... 69

**第 7 章：定义参数** ..... 71
关于定义参数 ..... 72
了解参数限制 ..... 73
创建参数 ..... 74
定义参数属性 ..... 77
了解参数类型 ..... 78
分配内部数据 ..... 78
指定参数格式 ..... 85
选择一个文件作为参数值的源 ..... 87
从现有数据库中导入数据 ..... 92
用户定义的函数 ..... 95
参数化选项 ..... 98

**第 8 章：关联语句** ..... 101
关于关联语句 ..... 101
使用 C Vuser 的关联函数 ..... 103
使用 Java Vuser 的关联函数 ..... 104
使用 Wdiff 比较 Vuser 脚本 ..... 105
修改保存的参数 ..... 106

**第 9 章：配置运行时设置** ..... 107
关于运行时设置 ..... 107
配置运行逻辑运行时设置 （多操作） ..... 109
步运行时设置 ..... 112
配置步运行时设置 （多操作） ..... 113
设置 “步” 和 “运行逻辑” 选项 （单操作） ..... 114
配置日志运行时设置 ..... 116
配置思考时间设置 ..... 120
配置其他运行时设置 ..... 122
设置 VB 运行时设置 ..... 126

**第 10 章：配置网络运行时设置** ..... 127
关于网络运行时设置 ..... 127
设置网路速度 ..... 128

**第 11 章：在独立模式下运行 Vuser 脚本** ..... 129
关于在独立模式下运行 Vuser 脚本 ..... 129
在 VuGen 中运行 Vuser 脚本 ..... 130
输出窗口：执行日志 ..... 132
使用 VuGen 的调试功能 ..... 133
使用 VuGen 的 Web Vuser 脚本调试功能 ..... 134
使用 VuGen 窗口 ..... 136
从命令提示符运行 Vuser 脚本 ..... 137
从 UNIX 命令行运行 Vuser 脚本 ..... 137
将 Vuser 脚本集成到场景中 ..... 139

**第 12 章：使用 TestDirector 管理脚本** ..... 141
关于使用 TestDirector 管理脚本 ..... 141
与 TestDirector 连接以及与其断开连接 ..... 141
打开 TestDirector 项目中的脚本 ..... 144
将脚本保存到 TestDirector 项目 ..... 146

**第 III 部分：使用 JAVA 语言协议**

**第 13 章：录制 Java 语言 Vuser 脚本** ..... 149
关于录制 Java 语言 Vuser 脚本 ..... 149
录制入门 ..... 150
了解 Java 语言 Vuser 脚本 ..... 151
作为程序包的一部分运行脚本 ..... 152
查看 Java 方法 ..... 153
手动插入 Java 方法 ..... 154
配置脚本生成设置 ..... 157

**第 14 章：设置 Java 录制选项** ..... 161
关于设置 Java 录制选项 ..... 161
Java 虚拟机 (JVM) 录制选项 ..... 162
设置类路径录制选项 ..... 163
录制器选项 ..... 164
序列化选项 ..... 168
关联选项 ..... 169
调试选项 ..... 170
CORBA 选项 ..... 173

**第 15 章：关联 Java 脚本** ..... 175
关于关联 Java 脚本 ..... 176
标准关联 ..... 176
高级关联 ..... 177
字符串关联 ..... 179
使用序列化机制 ..... 180

**第 16 章：配置 Java 运行时设置** ..... 187
关于 Java 运行时设置 ..... 187
指定 JVM 运行时设置 ..... 188
设置运行时类路径选项 ..... 189

**第 IV 部分：应用程序部署解决方案协议**

**第 17 章：开发 Citrix Vuser 脚本** ..... 193
关于录制 Citrix Vuser 脚本 ..... 193
Citrix Vuser 脚本入门 ..... 194
了解 Citrix 录制选项 ..... 195
设置 Citrix 录制选项 ..... 198
同步回放 ..... 199
使用 Citrix 函数 ..... 202
查看 Citrix Vuser 脚本 ..... 204
设置 Citrix 显示设置 ..... 206
设置 Citrix 运行时设置 ..... 207
了解 ICA 文件 ..... 210
断开与 Citrix 服务器的连接 ..... 211
Citrix Vuser 脚本使用提示 ..... 213

**第 V 部分：客户端服务器协议**

**第 18 章：开发数据库 Vuser 脚本** ..... 217
关于录制数据库 Vuser 脚本 ..... 217
介绍数据库 Vuser ..... 218
了解数据库 Vuser 技术 ..... 219
数据库 Vuser 脚本入门 ..... 220
设置数据库录制选项 ..... 221
数据库高级录制选项 ..... 222
使用 LRD 函数 ..... 224
了解数据库 Vuser 脚本 ..... 228
评估错误代码 ..... 231
处理错误 ..... 232

**第 19 章：关联数据库 Vuser 脚本** ..... 235
关于关联数据库 Vuser 脚本 ..... 235
在脚本中扫描关联 ..... 236
关联已知值 ..... 238
数据库关联函数 ..... 240

**第 20 章：开发 DNS Vuser 脚本** ..... 241
关于开发 DNS Vuser 脚本 ..... 241
使用 DNS 函数 ..... 242

**第 21 章：开发 WinSock Vuser 脚本** ....243
关于录制 Windows 套接字 Vuser 脚本 ....243
Windows 套接字 Vuser 脚本入门 ....244
设置 WinSock 录制选项 ....246
使用 LRS 函数 ....248
在树视图和脚本视图之间切换 ....252

**第 22 章：使用 Window 套接字数据** ....255
关于使用 Windows 套接字数据 ....255
查看快照窗口中的数据 ....256
在数据中导航 ....257
修改缓冲区数据 ....261
修改缓冲区名称 ....266
在脚本视图中查看 Windows 套接字数据 ....267
了解数据文件格式 ....268
查看十六进制格式的缓冲区数据 ....270
设置显示格式 ....272
调试提示 ....274
手动关联 WinSock 脚本 ....275

**第 VI 部分：自定义 VUSER 脚本**

**第 23 章：创建自定义 Vuser 脚本** ....281
关于创建自定义 Vuser 脚本 ....281
C Vuser ....282
Java Vuser ....284
VB Vuser ....285
VBScript Vuser ....286
JavaScript Vuser ....287

**第 24 章：编写 Java 脚本** ....289
关于编写 Java 脚本 ....289
创建 Java Vuser ....290
编辑 Java Vuser 脚本 ....291
LoadRunner 的 Java API ....293
使用 Java Vuser 函数 ....295
设置 Java 环境 ....301
运行 Java Vuser 脚本 ....301
将脚本作为包的一部分编译和运行 ....302
编写提示 ....303

## 第 VII 部分：分布式组件协议

**第 25 章：录制 COM Vuser 脚本** ..... 307
关于录制 COM Vuser 脚本 ..... 307
COM 概述 ..... 308
COM Vuser 入门 ..... 309
选择要录制的 COM 对象 ..... 310
设置 COM 录制选项 ..... 312

**第 26 章：了解 COM Vuser 脚本** ..... 321
关于 COM Vuser 脚本 ..... 321
了解 VuGen COM 脚本结构 ..... 322
检查 VuGen COM 脚本示例 ..... 323
在脚本中扫描关联 ..... 328
关联已知值 ..... 330

**第 27 章：了解 COM Vuser 函数** ..... 333
关于 COM Vuser 函数 ..... 334
创建实例 ..... 334
IDispatch 接口调用方法 ..... 334
类型分配函数 ..... 335
变量类型 ..... 336
从引用到变量的分配 ..... 337
参数化函数 ..... 338
从变量中提取 ..... 339
将数组分配给变量 ..... 339
数组类型和函数 ..... 340
字节数组函数 ..... 341
ADO 记录集函数 ..... 341
调试函数 ..... 342
VB 集合支持 ..... 342

**第 28 章：开发 Corba-Java Vuser 脚本** ..... 343
关于 Corba-Java Vuser 脚本 ..... 343
录制 Corba-Java Vuser ..... 344
使用 Corba-Java Vuser 脚本 ..... 347
在 Windows XP 和 Windows 2000 Server 上录制 ..... 349

**第 29 章：开发 RMI-Java Vuser 脚本** ..... 351
关于开发 RMI-Java Vuser 脚本 ..... 351
录制 RMI over IIOP ..... 352
录制 RMI Vuser ..... 352
使用 RMI Vuser 脚本 ..... 355

## 第 VIII 部分：电子商务协议

**第 30 章：开发 FTP Vuser 脚本** ..... 359
关于开发 FTP Vuser 脚本 ..... 359
使用 FTP 函数 ..... 360

**第 31 章：开发 LDAP Vuser 脚本** ..... 363
关于开发 LDAP Vuser 脚本 ..... 363
使用 LDAP 函数 ..... 364
定义可分辨名称项 ..... 366

**第 32 章：创建 Web Vuser 脚本** ..... 369
关于开发 Web Vuser 脚本 ..... 369
介绍 Web Vuser ..... 370
了解 Web Vuser 技术 ..... 371
Web Vuser 脚本入门 ..... 372
录制 Web 会话 ..... 373
将 Web Vuser 脚本转换为 Java ..... 374

**第 33 章：使用 Web Vuser 函数** ..... 375
关于 Web Vuser 函数 ..... 375
添加和编辑函数 ..... 376
Web 函数列表 ..... 377
在树视图中查看脚本 ..... 383
在脚本视图中查看 Vuser 脚本 ..... 385

**第 34 章：录制 Web/WinSock 和 SOAP Vuser 脚本** ..... 387
关于录制 Web/WinSock Vuser 脚本 ..... 388
Web/WinSock Vuser 脚本入门 ..... 389
设置浏览器和代理服务器录制选项 ..... 390
设置 Web 捕获录制选项 ..... 393
录制 Web/WinSock 会话 ..... 394
录制 Palm 应用程序 ..... 396

**第 35 章：为 Internet 协议设置录制选项** ..... 399
关于为 Internet 协议设置录制选项 ..... 399
使用代理服务器设置 ..... 400
设置高级录制选项 ..... 402
设置录制场景 ..... 404

**第 36 章：为 Web Vuser 设置录制选项** ..... 409
关于设置录制选项 ..... 409
指定用于录制的浏览器 ..... 410
选择录制级别 ..... 411

**第 37 章：配置 Internet 运行时设置**.....423
关于 Internet 运行时设置.....423
设置代理服务器选项.....424
设置浏览器仿真属性.....428
设置 Internet 首选项.....432
获取调试信息.....435
执行 HTML 压缩.....436

**第 38 章：检查网页内容**.....437
关于检查网页内容.....437
设置内容检查运行时设置.....438

**第 39 章：在负载下验证网页**.....443
关于负载下的验证.....444
添加文本检查.....446
使用其他文本检查方法.....449
添加图像检查.....451
定义其他属性.....454
使用常规表达式.....455

**第 40 章：修改 Web 和无线 Vuser 脚本**.....457
关于修改 Web 和无线 Vuser 脚本.....457
向 Vuser 脚本中添加步骤.....458
从 Vuser 脚本中删除步骤.....459
修改操作步骤.....460
修改控制步骤.....473
修改服务步骤.....476
修改 Web 检查 （仅限于 Web）.....477

**第 41 章：配置 Web Vuser 脚本的关联规则**.....479
关于关联语句.....480
了解关联方法.....481
使用 VuGen 的关联规则.....481
测试规则.....486
设置关联录制选项.....487

**第 42 章：录制后关联 Vuser 脚本**.....491
关于与快照关联.....491
了解快照.....492
设置 VuGen 关联.....497
执行扫描关联.....500
执行手动关联.....502
定义动态字符串的边界.....507

**第 43 章：测试 XML 页面** ..... 509
关于测试 XML 页 ..... 509
将 XML 视为 URL 步骤 ..... 510
将 XML 作为自定义请求插入 ..... 511
查看 XML 自定义请求步骤 ..... 512

**第 44 章：使用报告调试 Vuser 脚本** ..... 515
关于使用报告调试 Vuser 脚本 ..... 516
了解 “结果概要” 报告 ..... 517
筛选报告信息 ..... 519
搜索结果 ..... 520
管理执行结果 ..... 520

**第 45 章：有关 Web Vuser 的 Power User 提示** ..... 523
安全性问题 ..... 523
处理 Cookie ..... 526
运行时查看器 （联机浏览器） ..... 528
浏览器 ..... 529
配置问题 ..... 532
兼容问题 ..... 532

**第 IX 部分：ENTERPRISE JAVA BEAN 协议**

**第 46 章：执行 EJB 测试** ..... 535
关于 EJB 测试 ..... 535
使用 EJB 检测器 ..... 536
创建 EJB 测试 Vuser ..... 540
设置 EJB 录制选项 ..... 543
了解 EJB Vuser 脚本 ..... 544
运行 EJB Vuser 脚本 ..... 549

**第 X 部分：ERP/CRM 协议**

**第 47 章：创建 Oracle NCA Vuser 脚本** ..... 555
关于创建 Oracle NCA Vuser 脚本 ..... 556
Oracle NCA Vuser 入门 ..... 557
录制指南 ..... 558
启用按名称录制对象的功能 ..... 559
通过个人主页启动 Oracle 应用程序 ..... 562
使用 Oracle NCA Vuser 函数 ..... 563
了解 Oracle NCA Vuser ..... 567
在树视图和脚本视图之间切换 ..... 568
配置运行时设置 ..... 569

测试 Oracle NCA 应用程序 ..... 571
关联 Oracle NCA 语句以实现负载平衡 ..... 573
其他建议的关联 ..... 575
以 Pragma 模式录制 ..... 577

**第 48 章：开发 SAPGUI Vuser 脚本** ..... 579
关于开发 SAPGUI Vuser 脚本 ..... 580
检查 SAPGUI Vuser 的环境 ..... 581
创建 SAPGUI Vuser 脚本 ..... 591
设置 SAPGUI 录制选项 ..... 592
了解 SAPGUI Vuser 脚本 ..... 594
增强 SAPGUI Vuser 脚本 ..... 597
回放 SAPGUI Vuser 脚本 ..... 600
设置 SAPGUI 运行时设置 ..... 601
SAPGUI 函数 ..... 603
SAPGUI Vuser 脚本提示 ..... 609
SAPGUI Vuser 脚本疑难解答 ..... 612
其他资源 ..... 614

**第 49 章：开发 SAP-Web Vuser 脚本** ..... 617
关于开发 SAP-Web Vuser 脚本 ..... 618
创建 SAP-Web Vuser 脚本 ..... 618
设置 SAP-Web 录制选项 ..... 619
了解 SAP-Web Vuser 脚本 ..... 620
回放 SAP-Web Vuser 脚本 ..... 622

**第 50 章：开发 Siebel-Web Vuser 脚本** ..... 623
关于开发 Siebel-Web Vuser 脚本 ..... 623
录制 Siebel-Web 会话 ..... 624
关联 Siebel-Web 脚本 ..... 624
Siebel-Web Vuser 脚本疑难解答 ..... 626

**第 51 章：创建 Baan Vuser 脚本** ..... 629
关于开发 Baan Vuser 脚本 ..... 629
Baan Vuser 脚本入门 ..... 630
Baan Vuser 函数 ..... 630
创建 Baan Vuser 脚本 ..... 634
了解 Baan Vuser 脚本 ..... 635
自定义 Baan Vuser 脚本 ..... 636

## 第 XI 部分：传统协议

**第 52 章：介绍 RTE Vuser 脚本** ..... 641
关于开发 RTE Vuser 脚本 ..... 641
介绍 RTE Vuser ..... 642
了解 RTE Vuser 技术 ..... 642
RTE Vuser 脚本入门 ..... 643
使用 TE 函数 ..... 644
将终端键映射到 PC 键盘键 ..... 645

**第 53 章：录制 RTE Vuser 脚本** ..... 647
关于录制 RTE Vuser 脚本 ..... 647
新建 RTE Vuser 脚本 ..... 648
录制终端设置和连接过程 ..... 649
录制典型的用户操作 ..... 652
录制注销过程 ..... 653
设置 RTE 录制选项 ..... 653
在终端仿真器中键入输入 ..... 656
生成唯一设备名 ..... 658
设置字段分界字符 ..... 659

**第 54 章：配置 RTE 运行时设置** ..... 661
关于终端仿真器运行时设置 ..... 661
修改连接尝试 ..... 662
指定原始设备名 ..... 663
设置键入延迟 ..... 663
配置 X-System 同步 ..... 664

**第 55 章：同步 RTE Vuser 脚本** ..... 665
关于同步 Vuser 脚本 ..... 665
同步块模式 (IBM) 终端 ..... 666
同步字符模式 (TV) 终端 ..... 669

**第 56 章：读取终端屏幕中的文本** ..... 673
关于读取终端屏幕中的文本 ..... 673
搜索屏幕中的文本 ..... 673
读取屏幕中的文本 ..... 674

**第 XII 部分：邮件服务协议**

**第 57 章：开发用于邮件服务的 Vuser 脚本** ..... 677
关于开发用于邮件服务的 Vuser 脚本 ..... 677
邮件服务 Vuser 脚本入门 ..... 678
使用 IMAP 函数 ..... 679
使用 MAPI 函数 ..... 681
使用 POP3 函数 ..... 682
使用 SMTP 函数 ..... 683

**第 XIII 部分：中间件协议**

**第 58 章：开发 Jacada Vuser 脚本** ..... 687
关于 Jacada Vuser 脚本 ..... 687
Jacada Vuser 入门 ..... 688
录制 Jacada Vuser ..... 689
回放 Jacada Vuser ..... 691
了解 Jacada Vuser 脚本 ..... 691
使用 Jacada Vuser 脚本 ..... 692

**第 59 章：开发 Tuxedo Vuser 脚本** ..... 695
关于 Tuxedo Vuser 脚本 ..... 695
Tuxedo Vuser 脚本入门 ..... 696
使用 LRT 函数 ..... 697
了解 Tuxedo Vuser 脚本 ..... 701
查看 Tuxedo 缓冲区数据 ..... 703
定义 Tuxedo Vuser 的环境设置 ..... 704
调试 Tuxedo 应用程序 ..... 705
关联 Tuxedo 脚本 ..... 705

**第 XIV 部分：流数据协议**

**第 60 章：开发流式数据 Vuser 脚本** ..... 713
关于录制流式数据虚拟用户脚本 ..... 713
流式数据 Vuser 脚本入门 ..... 714
使用 RealPlayer LREAL 函数 ..... 715
使用 Media Player MMS 函数 ..... 716

## 第 XV 部分：无线协议

**第 61 章：介绍无线 Vuser** ..... 721
关于无线 Vuser ..... 721
了解 WAP 协议 ..... 722
了解 i-mode 系统 ..... 723
i-mode 与 WAP 之比较 ..... 724
了解 VoiceXML ..... 724

**第 62 章：录制无线 Vuser 脚本** ..... 727
关于录制无线 Vuser 脚本 ..... 727
无线 Vuser 脚本入门 ..... 728
使用无线 Vuser 函数 ..... 729
无线 Vuser 脚本疑难解答 ..... 731

**第 63 章：使用 WAP Vuser 脚本** ..... 733
关于 WAP Vuser ..... 733
通过电话录制 ..... 734
载体支持 ..... 735
RADIUS 支持 ..... 735
推支持 ..... 736
LoadRunner 推支持 ..... 737
MMS 支持 ..... 738

**第 64 章：设置无线 Vuser 录制选项** ..... 739
关于设置录制选项 ..... 739
指定录制模式（仅限于 WAP） ..... 740
指定要录制的信息（i-mode 和 VoiceXML） ..... 741
指定工具包 ..... 743

**第 65 章：配置 WAP 运行时设置** ..... 745
关于 WAP 运行时设置 ..... 745
配置网关选项 ..... 746
配置载荷信息 ..... 750
配置 RADIUS 连接数据 ..... 752

## 第 XVI 部分：GUI VUSER 脚本

**第 66 章：开发 GUI Vuser 脚本** ..... 757
关于开发 GUI Vuser 脚本 ..... 758
GUI Vuser 简介 ..... 758
了解 GUI Vuser 技术 ..... 759
GUI Vuser 入门 ..... 760
使用 WinRunner 创建 GUI Vuser 脚本 ..... 761
度量服务器性能：事务 ..... 762
生成较重的用户负载：集合点 ..... 762
了解 GUI Vuser 脚本 ..... 763
在 GUI Vuser 脚本中使用 Vuser 函数 ..... 764
向 Controller 发送消息 ..... 765
获取关于 Vuser 和负载生成器的信息 ..... 766

## 第 XVII 部分：高级用户信息

**第 67 章：在 Visual Studio 中创建 Vuser 脚本** ..... 769
关于在 Visual Studio 中创建 Vuser 脚本 ..... 769
用 Visual C 创建 Vuser 脚本 ..... 770
用 Visual Basic 创建 Vuser 脚本 ..... 772
配置运行时设置和参数 ..... 773

**第 68 章：通过 XML API 编程** ..... 775
关于通过 XML API 编程 ..... 775
了解 XML 文档 ..... 776
使用 XML 函数 ..... 777
指定 XML 函数参数 ..... 780
使用 XML 属性 ..... 781
构建 XML 脚本 ..... 782
增强已录制的会话 ..... 783

**第 69 章：VuGen 调试提示** ..... 789
常规调试提示 ..... 789
使用 C 函数进行跟踪 ..... 790
添加附加 C 语言关键字 ..... 790
检查回放输出 ..... 791
调试数据库应用程序 ..... 791
使用 Oracle 应用程序 ..... 793
解决 Oracle 应用程序的常见问题 ..... 793
两层数据库脚本提示 ..... 798
运行 PeopleSoft-Tuxedo 脚本 ..... 805

**第 70 章：高级主题** ..... 807
录制期间生成的文件 ..... 807
回放期间生成的文件 ..... 809
从 Unix 命令行运行 Vuser ..... 811
指定 Vuser 行为 ..... 812
命令行参数 ..... 812
录制 OLE 服务器 ..... 813
检查 .dat 文件 ..... 815
添加新的 Vuser 类型 ..... 816

**第 XVIII 部分：附录**

**附录 A：Java 环境：综合指南** ..... 823
关于 Java 环境 ..... 823
术语 ..... 824
JDK 版本 ..... 826
浏览器 ..... 830
Java 插件 ..... 833
其他环境 ..... 834
常见问题 ..... 835

**附录 B：EJB 体系结构和测试** ..... 839
EJB 简介 ..... 839
EJB 体系结构 ..... 839
EJB 结构和机制 ..... 840
在常见应用程序服务器中部署 ..... 842
EJB 单元测试 ..... 843

**附录 C：调用外部函数** ..... 845
加载 DLL - 本地 ..... 845
全局加载 DLL ..... 847

**附录 D：在 UNIX 平台上编写脚本** ..... 849
关于编写要在 UNIX 平台上运行的 Vuser 脚本 ..... 849
生成模板 ..... 850
将 Vuser 操作编入脚本 ..... 851
配置 Vuser 运行时设置 ..... 852
定义事务和集合点 ..... 857
编译脚本 ..... 857

**附录 E：使用键盘快捷键** ..... 859

**索引** ..... 861

# 欢迎使用 LoadRunner

欢迎使用 Mercury Interactive 用于测试应用程序性能的工具：LoadRunner。LoadRunner 对应用程序进行压力测试，以隔离并标识潜在的客户端、网络和服务器瓶颈。

通过 LoadRunner，可以在可控制的峰值负载条件下测试系统。要生成负载，LoadRunner 需要运行分布在网络中的数千个虚拟用户 (*Vuser*)。通过使用最少的硬件资源，这些 Vuser 提供一致的、可重复并可度量的负载，像实际用户一样使用您的应用程序。 LoadRunner 深入的报告和图可以提供评估应用程序性能所需的信息。

## 联机资源

LoadRunner 包括下列联机工具：

**自述文件**提供有关 LoadRunner 的最新新闻和信息。

**联机图书**显示 PDF 格式的文档集。可以使用安装包中包括的 Adobe Acrobat Reader 来阅读和打印联机图书。请检查 Mercury Interactive 的客户支持网站是否有 LoadRunner 联机图书更新。该网站的 URL 是 *http://support.mercuryinteractive.com*。

**LoadRunner 函数参考**使您能够联机访问创建 Vuser 脚本时可以使用的 LoadRunner 的所有函数，包括如何使用函数的示例。请检查 Mercury Interactive 的客户支持网站 *http://support.mercuryinteractive.com* 是否有在线《LoadRunner Online Function Reference》更新。

**LoadRunner 上下文相关帮助**对使用 LoadRunner 中出现的问题提供即时答案。它描述对话框，并介绍如何执行 LoadRunner 任务。要激活该帮助，请在窗口中单击，然后按 F1 键。请检查 Mercury Interactive 的客户支持网站 *http://support.mercuryinteractive.com* 是否有更新。

**联机技术支持**使用默认的 Web 浏览器打开 Mercury Interactive 的客户支持网站。该网站的 URL 是 *http://support.mercuryinteractive.com*。

**支持信息**显示 Mercury Interactive 的客户支持网站、主页以及 Mercury Interactive 全球办事处的列表。

**网上 Mercury Interactive** 使用默认的 Web 浏览器打开 Mercury Interactive 的主页。该网站的 URL 是 *http://www.mercuryinteractive.com*。

## LoadRunner 文档集

LoadRunner 附带有描述下列操作的一套文档：

- 安装 LoadRunner
- 创建 Vuser 脚本
- 使用 LoadRunner Controller
- 使用 LoadRunner Analysis

## 使用 LoadRunner 文档集

LoadRunner 文档集包括安装指南、Controller 用户指南、Analysis 用户指南和两份创建 Vuser 脚本的指南。

### 安装指南

有关安装 LoadRunner 的说明，请参阅 《LoadRunner 安装指南》。该安装指南介绍如何安装：

- LoadRunner Controller – 在基于 Windows 的计算机上
- 虚拟用户组件 – 应用于 Windows 和 UNIX 平台

### Controller 用户指南

LoadRunner 文档包中包括一份 Controller 用户指南：

《LoadRunner Controller 用户指南 (Windows)》描述如何在 Windows 环境中使用 LoadRunner Controller 来创建和运行 LoadRunner 场景。Vuser 可以在 UNIX 和 Windows 平台上运行。 Controller 用户指南概述 LoadRunner 测试过程。

### Analysis 用户指南

LoadRunner 文档包中包括一份 Analysis 用户指南：

《LoadRunner Analysis 用户指南》描述如何在运行场景之后使用 LoadRunner Analysis 图和报告来分析系统性能。

### 创建 Vuser 脚本指南

LoadRunner 文档包中包括一份 VuGen 用户指南：

《LoadRunner 创建 Vuser 脚本指南》描述如何使用 VuGen 来创建 Vuser 脚本。如有必要，请使用联机 《LoadRunner Function Reference》和用于创建 GUI Vuser 脚本的 《WinRunner User's Guide》来补充此文档。

| 有关以下信息 | 参考以下指南 |
|---|---|
| 安装 LoadRunner | 《LoadRunner 安装指南》 |
| LoadRunner 测试过程 | 《LoadRunner Controller 用户指南 (Windows)》 |
| 创建 Vuser 脚本 | 《LoadRunner 创建 Vuser 脚本指南》 |
| 创建并运行场景 | 《LoadRunner Controller 用户指南 (Windows)》 |
| 分析测试结果 | 《LoadRunner Analysis 用户指南》 |

# 版式约定

本书使用下列版式约定：

| | |
|---|---|
| **1, 2, 3** | 粗体数字指示过程中的步骤。 |
| ➤ | 点符指示选项和功能。 |
| > | 大于号分隔菜单级别 （例如 “文件” > “打开”）。 |
| **Stone Sans** | **Stone Sans** 字体指示您对其执行操作的界面元素的名称（例如，单击 “运行” 按钮 )。 |
| **粗体** | **粗体**文本指示方法或函数名 |
| *斜体* | *斜体*文本指示方法或函数参数、文件名或路径以及书名。 |
| Arial | Arial 字体用于要按字面逐字键入的示例和文本。 |
| <> | 尖括号包含各不相同的文件路径或 URL 地址部分 （例如，< 产品安装文件夹 >/bin）。 |
| [ ] | 方括号包含可选的参数。 |
| { } | 花括号指示必须将括起来的某个值分配给当前参数。 |
| … | 在语法行中，省略号指示可能包括更多相同格式的项目。 |

# 第 I 部分

## 介绍 Vuser 脚本

# 1

# 开发 Vuser 脚本

LoadRunner 可模拟一个数千用户同时使用客户端 / 服务器系统的环境。为执行此操作，LoadRunner 用“虚拟用户 (Vuser)”代替实际用户。Vuser 执行的操作是用 Vuser 脚本描述的。 LoadRunner 提供各种帮助您开发 Vuser 脚本的工具。

本章讨论：

- 介绍 Vuser
- 查看 Vuser 类型
- 开发 Vuser 脚本
- 使用本指南

## 介绍 Vuser

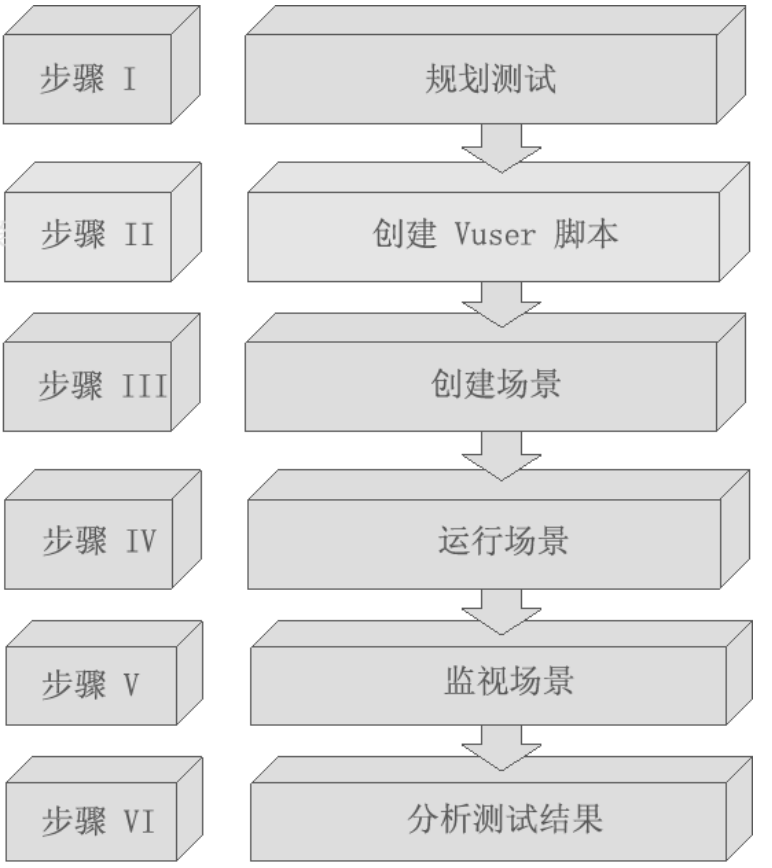
LoadRunner 用虚拟用户或 Vuser 代替实际用户。 Vuser 通过执行典型业务流程模拟实际用户的操作。对于 Vuser 执行的每个操作， LoadRunner 向服务器或类似的企业系统提交输入信息。增加 Vuser 的数量可以增大系统上的负载。一台工作站只能容纳一个实际用户，而多个 Vuser 可以同时运行在同一台工作站上。

要模拟较重用户负载的情形，请创建大量执行一系列任务的 Vuser。例如，可以观察一百个 Vuser 同时从银行的 ATM 提取现金时服务器的行为。

通过使用 LoadRunner，可以将您的客户端 / 服务器性能测试需求划分为多个场景。场景定义每个测试会话中发生的事件。这样，例如，场景会定义并控制要模拟的用户的数量和他们执行的操作，以及运行模拟操作所用的计算机。

LoadRunner 拥有各种 Vuser 类型，每一类型都适合于特定的负载测试环境。这样就能够使用 Vuser 精确模拟真实世界的情形。 Vuser 在场景中执行的操作是用 Vuser 脚本描述的。 Vuser 脚本中包括在场景中度量并录制服务器性能的函数。每个 Vuser 类型都需要特定类型的 Vuser 脚本。创建场景所需的 Vuser 脚本是 LoadRunner 测试过程的一部分。

下面的图表显示 LoadRunner 测试过程的六个步骤。本指南描述步骤 II：“创建 Vuser 脚本。”有关其他步骤的详细信息，请参阅 《LoadRunner Controller 用户指南》。



# 查看 Vuser 类型

LoadRunner 提供了多种 Vuser 技术，通过这些技术可以在使用不同类型的客户端 / 服务器体系结构时生成服务器负载。每种 Vuser 技术都适合于特定体系结构并产生特定的 Vuser 类型。例如，可以使用 Web Vuser 模拟用户操作 Web 浏览器、使用 Tuxedo Vuser 模拟 Tuxedo 客户端与 Tuxedo 应用程序服务器之间的通信、使用 RTE Vuser 操作终端仿真器。各种 Vuser 技术既可单独使用，又可一起使用，以创建有效的负载测试场景。

Vuser 类型分为下列几种：

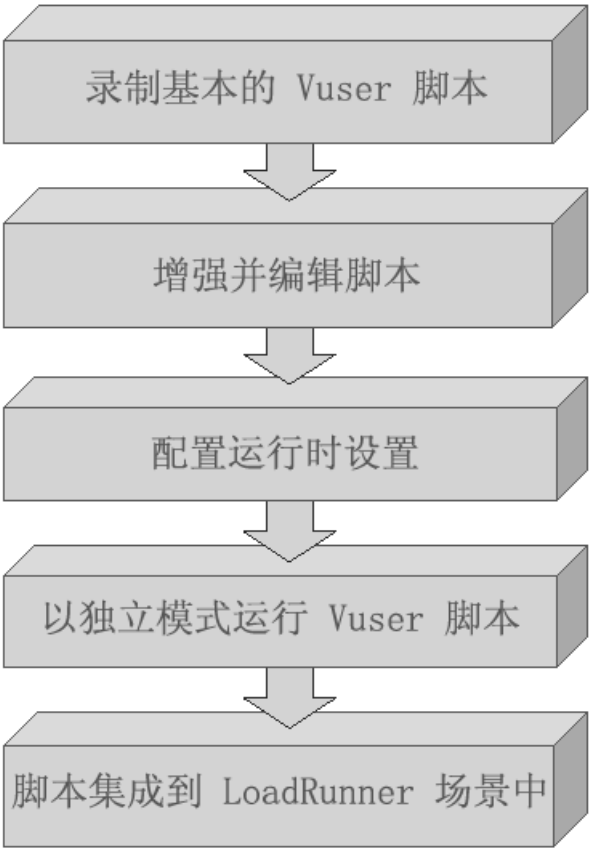
- **应用程序部署解决方案：** Citrix ICA
- **客户端 / 服务器：** DB2 CLI、DNS、Informix、MS SQL Server、ODBC、Oracle（2 层）、 Sybase Ctlib、 Sybase Dblib 和 Windows Sockets 协议。
- **自定义：** C 模板、Visual Basic 模板、Java 模板、Javascript 和 VBScript 类型的脚本。
- **分布式组件：** 适用于 COM/DCOM、 Corba-Java 和 Rmi-Java 协议。
- **电子商务：** FTP、 LDAP、 Palm、 PeopleSoft 8 mulit-lingual、 SOAP、 Web (HTTP/HTML) 和双 Web/WinSocket 协议。
- **Enterprise Java Bean：** EJB 测试和 Rmi-Java 协议。
- **ERP/CRM：** Baan、 Oracle NCA、 Peoplesoft-Tuxedo、 SAP-Web、 SAPGUI、 Siebel-DB2 CLI、 Siebel-MSSQL、 Siebel-Web 和 Siebel-Oracle 协议。
- **传统：** 终端仿真 (RTE)。
- **邮件服务：** Internet 邮件访问协议 (IMAP)、 MS Exchange (MAPI)、 POP3 和 SMTP。
- **中间件：** Jacada 和 Tuxedo （6、 7）协议。
- **流数据：** Media Player (MMS) 和 Real 协议。
- **无线：** i-Mode、 VoiceXML 和 WAP 协议。

要按字母顺序查看所有支持的协议的列表，请选择 “文件” > “新建”，然后在 “协议类型”列表框中选择 “所有协议”。

# 开发 Vuser 脚本

Vuser 脚本的结构和内容因 Vuser 类型的不同而不同。例如，数据库 Vuser 脚本总是包含三部分，是在一段类似 C 语言并且包括对数据库服务器的 SQL 调用的代码中编写的。相反， GUI Vuser 脚本只有一个部分，并且是用 TSL（测试脚本语言）编写的。

下图概述开发 Vuser 脚本的过程。



开发 Vuser 脚本的过程开始于录制一个基本的脚本。 LoadRunner 为您提供了大量录制 Vuser 脚本的工具 （有关工具列表请参阅下表）。您可以通过将控制流结构和其他 LoadRunner API 添加到脚本中来增强该基本脚本。然后配置运行时设置。运行时设置包括循环、日志和计时信息，以及定义 Vuser 在执行 Vuser 脚本时的行为。要验证脚本是否能正确运行，请以单独模式运行该脚本。如果脚本运行正确，则将其合并到 LoadRunner 场景中。

| 可以使用下列 **LoadRunner** 工具开发 **Vuser** 脚本： | |
|---|---|
| *VuGen* | LoadRunner 开发 Vuser 脚本的主要工具。 Vuser 脚本生成器 （也称为 *VuGen*）是一种基于 Windows 的应用程序，通过它可以开发多种 Vuser 脚本。 VuGen 不仅录制脚本，也可以运行这些脚本。 VuGen 生成的许多脚本可以同时在 Windows 和 UNIX 平台上运行 |
| *WinRunner* | Mercury Interactive 的基于 Windows 的 GUI 应用程序自动测试工具。增强 WinRunner 生成的测试脚本可以产生运行在 Windows 平台上的 GUI Vuser 脚本 |
| *Astra LoadTest 虚拟用户录制器* | Mercury Interactive 的基于 Web 的应用程序负载测试工具。 LoadRunner 的一个简化版。 LoadRunner 可以运行使用 Astra LoadTest 虚拟用户录制器创建的 Vuser 脚本。此产品包括在 LoadRunner 中。详细信息，请参阅 《安装指南》 |
| *QuickTest Professional* | Mercury Interactive 用于测试复杂 Web 环境 （Java 小程序、Flash 等）的基于图标的工具。 LoadRunner 可以运行使用 QuickTest Professional 创建的 Vuser 脚本。注意，此产品必须单独购买 |

## 使用本指南

本指南分为下列几个部分：

- 第 I 部分 “介绍 Vuser 脚本”适用于所有类型的 Vuser 脚本。
- 第 II 部分 “使用 VuGen”仅适用于那些使用 VuGen 录制和 （或）运行的 Vuser 脚本。第 II 部分不适用于开发 GUI Vuser 脚本。
- 第 III 到 XVI 部分适用于特定的 Vuser 脚本类型。请参阅目录，找出描述 Vuser 类型的部分。
- 第 XVII 部分包含高级用户的信息。它提供一些常规调试提示，描述 VuGen 生成的文件并解释如何用 Visual C 和 Visual Basic 编写脚本。
- 第 XVIII 部分包含几个技术概述附录。它描述 Java 环境、 EJB 体系结构、调用外部函数、在 UNIX 中编程以及键盘快捷键。

---

**注意**：要开发 GUI Vuser 脚本，需要参阅 《WinRunner Use’s Guide》。

---

# 第 II 部分

## 使用 VuGen

# 2

# 介绍 VuGen

虚拟用户生成器 （也称为 *VuGen*）可用于针对各种应用程序类型和通信协议开发 Vuser 脚本。

本章描述：

- 使用 VuGen 录制 Vuser 脚本
- 启动 VuGen
- 了解 VuGen 环境选项
- 使用 VuGen 运行 Vuser 脚本
- 了解 VuGen 代码
- 使用 C Vuser 函数
- 获得函数帮助

***以下信息适用于所有类型 （GUI 除外）的 Vuser 脚本。***

## 关于 VuGen

Vuser 脚本生成器 （也称为 *VuGen*）是 LoadRunner 用于开发 Vuser 脚本的主要工具。

VuGen 不仅能够录制 Vuser 脚本，还可以运行这些脚本。进行调试时，从 VuGen 运行脚本很有用。通过运行脚本，可以模拟 Vuser 脚本在作为负载测试场景的一部分执行时，将如何运行。

---

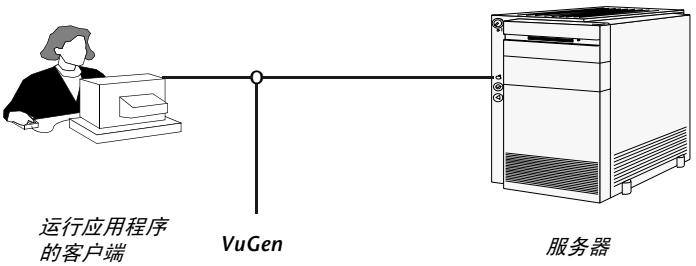
**注意：**VuGen 仅能录制 Windows 平台上的会话。但是，录制的 Vuser 脚本既可以在 Windows 平台上运行，也可以在 UNIX 平台上运行。

---

录制 Vuser 脚本时， VuGen 会生成各种函数，来定义您在录制会话过程中执行的操作。 VuGen 将这些函数插入到 VuGen 编辑器中，以创建基础 Vuser 脚本。

## 使用 VuGen 录制 Vuser 脚本

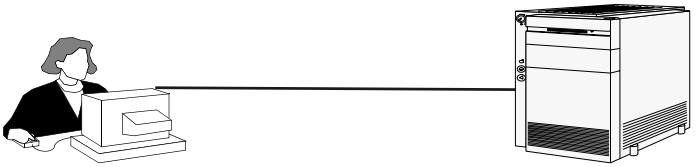
使用 VuGen，可通过录制用户在客户端应用程序中执行的典型业务流程来开发 Vuser 脚本。 VuGen 通过录制客户端和服务器之间的活动来创建脚本。例如，在数据库应用程序中， VuGen 会监控数据库的客户端，并跟踪发送到数据库服务器和从数据库服务器接收的所有请求。




不需要通过编写应用程序对服务器的 API 函数调用来手动创建 Vuser 脚本，而可以使用 VuGen 完成以下任务：

- 监控应用程序和服务器之间的通信
- 生成需要的函数调用
- 将生成的函数调用插入到 Vuser 脚本中

用 VuGen 创建的每个 Vuser 脚本都可以通过执行对服务器 API 的调用来直接与服务器通信，而不需要依赖客户端软件。这样，便可以使用 Vuser 来检查服务器性能 （甚至在客户端软件的用户界面完全开发好之前）。




此外，当 Vuser 与服务器直接通信时，不需要在用户界面中耗费系统资源。这样就可以在一个工作站中同时运行大量 Vuser，进而可以使用很少的测试计算机来模拟非常大的服务器负载。

## 启动 VuGen

可以从 “开始”菜单中启动 VuGen。选择 “开始” > “程序” > “LoadRunner” > “虚拟用户生成器”。每次调用 VuGen 时，都会打开 “启动”窗口。要禁止自动打开该屏幕，请在 “启动”对话框中选择 “以后不再显示启动对话框”选项。

要在 VuGen 中启用或禁用 “启动”对话框，请选择 “工具” > “常规选项”，然后选择 “环境”选项卡。在对话框的底部，选中或清除 “显示 ‘启动’对话框”选项。

“启动”对话框提供以下功能的快捷方式：

**新建单协议脚本**：创建单协议 Vuser 脚本。这是 “启动”对话框打开时的默认选项。您可以查看所有协议或按类别查看可用协议。

**新建多协议脚本**：创建多协议 Vuser 脚本。 VuGen 显示所有可用协议，并允许您指定要录制的协议。

**使用最近使用过的协议新建脚本**：列出最近用于新建 Vuser 脚本的协议。

**打开脚本**：打开文件系统、网络或 TestDirector 数据库中的现有脚本。

**最近打开过的脚本**：列出最近在 VuGen 中打开过的脚本。

# 了解 VuGen 环境选项

可以设置您的 VuGen 工作环境，以便自定义自动恢复设置、VuGen 编辑器和启动首选项。可以在 VuGen “常规选项”的 “环境”选项卡中设置这些选项。

### 自动恢复

通过自动恢复选项可以在发生崩溃或掉电的情况时还原 Vuser 脚本设置。要启用自动恢复，请选中 “保存自动恢复信息”复选框并指定保存时间间隔 （分钟）。

### 编辑器

可以设置编辑器选项，以选择字体并启用 VuGen 的 Intellisense 功能 （该功能可以自动填写文字和函数语法）。

**自动显示函数语法：**键入函数的左括号时， VuGen 将显示函数的语法及其参数和原型。要全局启用显示语法的功能，请选中该选项旁边的复选框。要禁用此功能，请清除 “自动显示函数语法”选项旁边的复选框。如果全局禁用 “显示函数语法”，仍可以通过按 Ctrl + Shift + 空格键或者通过在编辑器中键入左括号后选择 “编辑” > “显示函数语法”来显示函数语法。

**自动完成字：**键入函数的第一个下划线时， VuGen 会打开一个函数列表，通过它可以选择确切的函数，而无须手动键入整个函数。要全局启用文字完成功能，请选中该选项旁边的复选框。要禁用此功能，请清除 “自动完成字”选项旁边的复选框。如果全局禁用此选项，当在编辑器中键入内容时，仍可以通过按 Ctrl + 空格键或者选择 “编辑” > “完成字”来显示函数列表框。

**选择字体**：要设置编辑器字体，请单击 “选择字体”。将打开 “字体”对话框。选择需要的字体、字形和字号，然后单击 “确定”。注意，只能使用固定大小的字体 （Courier、 Lucida Console、 FixedSys 等）。

#### “启动”对话框

打开 VuGen 时，将显示 “启动”对话框。“启动”对话框包含用于新建脚本、打开现有脚本或查看最近使用的脚本的快速链接。如果禁用此选项， VuGen 在打开时将显示一个空白屏幕。

#### 默认的环境设置

默认情况下，“显示函数语法”和 “自动完成字”被全局启用。“自动恢复”被设置为 10 秒。 VuGen 在打开时会显示 “启动”对话框。

## 设置环境选项

#### 设置与环境相关的选项

1. 选择 “工具” > “常规选项”，然后单击 “环境”选项卡。
2. 要保存当前的 Vuser 脚本信息以便自动恢复，请选择 “保存自动恢复信息”选项，并指定保存时间间隔 （分钟）。
3. 要设置编辑器字体，请单击 “选择字体”。将打开 “字体”对话框。选择需要的字体、字形和字号，然后单击 “确定”。注意，只能使用固定大小的字体 （Courier、 Lucida Console、 FixedSys 等）。
4. 要指示每次打开 VuGen 时都显示 “启动”对话框，请选择 “‘启动’对话框”部分中的 “显示 ‘启动’对话框”。
5. 单击 “确定”接受设置并关闭 “常规选项”对话框。

## 使用 VuGen 运行 Vuser 脚本

要使用 Vuser 脚本执行负载测试，需要将脚本集成到 LoadRunner 场景中。执行此操作之前，可以通过从 VuGen 运行脚本来检查脚本的功能。如果脚本成功执行，便可以随后将其集成到场景中。有关 LoadRunner 场景的详细信息，请参阅《LoadRunner Controller 用户指南》。

运行 Vuser 脚本之前，可以修改脚本的运行时设置。这些设置包括 Vuser 执行的循环次数，以及将在脚本运行时应用于 Vuser 的步长和思考时间。有关配置运行时设置的详细信息，请参阅第 9 章 “配置运行时设置”。

运行 Vuser 脚本时，脚本将先由解释器进行处理，然后执行，无需编译脚本。如果修改了脚本，则解释器将记录引入脚本中的任何语法错误。还可以从脚本中调用可以由解释器识别并执行的外部函数。详细信息，请参阅附录 C “调用外部函数”。

高级用户可以编译录制的脚本以创建可执行程序。详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

# 了解 VuGen 代码

录制 Vuser 脚本时， VuGen 将生成 Vuser 函数并将它们插入到脚本中。有两种类型的 Vuser 函数：

- 通用 Vuser 函数
- 特定于协议的 Vuser 函数

通用 Vuser 函数和特定于协议的函数，它们共同构成了 LoadRunner API，并使 Vuser 能够直接与服务器通信。新建脚本时， VuGen 将显示支持的所有协议的列表。有关所有 Vuser 函数的语法信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

## 通用 Vuser 函数

通用 Vuser 函数也称为 LR 函数，因为每个 LR 函数都有一个 **lr** 前缀。LR 函数可以在任何类型的 Vuser 脚本中使用。通过 LR 函数可以：

- 获得有关 Vuser、其 Vuser 组和主机的运行时信息。
- 将事务和同步点添加到 Vuser 脚本中。例如， **lr_start_transaction** （在 Java 中为 **lr.start_transaction**）函数标记事务的开始，**lr_end_transaction** （在 Java 中为 **lr.end_transaction**）函数标记事务的结束。详细信息，请参阅第 6 章 “增强 Vuser 脚本”。
- 向输出设备发送表示错误或警告的消息。

有关 LR 函数的列表，请参阅第 18 页的 “使用 C Vuser 函数”；详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

## 特定于协议的 Vuser 函数

除了通用 Vuser 函数以外， VuGen 还会在录制时生成特定于协议的函数，并将它们插入到 Vuser 脚本中。

特定于协议的函数是专门针对要录制的 Vuser 的类型而生成的。例如， VuGen 会将 LRD 函数插入到数据库脚本中，将 LRT 函数插入到 TUXEDO 脚本中，将 LRS 函数插入到 Windows 套接字脚本中。

默认情况下， VuGen 的自动脚本生成器使用 C 为大多数协议创建 Vuser 脚本，而使用 Java 创建 Corba-Java/Rmi-Java Vuser 脚本。您可以指示 VuGen 使用 Visual Basic 或 Javascript 来生成代码。详细信息，请参阅第 4 章 “设置脚本生成首选项”。

所有标准约定都适用于脚本，包括控制流和语法。可以像在其他编程语言中那样向脚本添加注释和条件语句。

以下 Web Vuser 脚本段显示了 VuGen 在脚本中录制和生成的几个函数：

```
#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");
…
```

有关在 Vuser 脚本中使用 C 函数的详细信息，请参阅第 6 章“增强 Vuser 脚本”。有关修改 Java 脚本的详细信息，请参阅第 24 章 “编写 Java 脚本”。

---

**注意**：用于运行 Vuser 脚本的 C 解释器仅支持 ANSI C 语言。它不支持 Microsoft 对 ANSI C 的任何扩展。

---

## 使用 C Vuser 函数

可以将 *C* Vuser 函数添加到任何 Vuser 脚本中以便增强脚本。在录制时，VuGen 仅生成少数几个通用 Vuser 函数。如果需要，可以将其余函数手动写到脚本中。有关通用 Vuser 函数的详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

下面的列表显示了 ANSI C 脚本的通用 LoadRunner 函数。其中包括所有协议，但 Java 和 GUI 除外。有关 Java 函数的列表，请参阅第 24 章 “编写 Java 脚本”。

**事务函数**：

| | |
|---|---|
| **lr_end_sub_transaction** | 标记子事务的结束以便进行性能分析 |
| **lr_end_transaction** | 标记 LoadRunner 事务的结束 |
| **lr_end_transaction_instance** | 标记事务实例的结束以便进行性能分析 |
| **lr_fail_trans_with_error** | 将打开事务的状态设置为 LR_FAIL 并发送错误消息 |
| **lr_get_trans_instance_duration** | 获取事务实例的持续时间 （由它的句柄指定） |
| **lr_get_trans_instance_wasted_time** | 获取事务实例浪费的时间 （由它的句柄指定） |
| **lr_get_transaction_duration** | 获取事务的持续时间 （按事务的名称） |
| **lr_get_transaction_think_time** | 获取事务的思考时间 （按事务的名称） |
| **lr_get_transaction_wasted_time** | 获取事务浪费的时间 （按事务的名称） |
| **lr_resume_transaction** | 继续收集事务数据以便进行性能分析 |
| **lr_resume_transaction_instance** | 继续收集事务实例数据以便进行性能分析 |

| | |
|---|---|
| **lr_set_transaction_instance_status** | 设置事务实例的状态 |
| **lr_set_transaction_status** | 设置打开事务的状态 |
| **lr_set_transaction_status_by_name** | 设置事务的状态 |
| **lr_start_sub_transaction** | 标记子事务的开始 |
| **lr_start_transaction** | 标记事务的开始 |
| **lr_start_transaction_instance** | 启动嵌套事务 （由它的父事务的句柄指定） |
| **lr_stop_transaction** | 停止事务数据的收集 |
| **lr_stop_transaction_instance** | 停止事务 （由它的句柄指定）数据的收集 |
| **lr_wasted_time** | 消除所有打开事务浪费的时间 |

**命令行分析函数**

| | |
|---|---|
| **lr_get_attrib_double** | 检索脚本命令行中使用的 *double* 类型变量 |
| **lr_get_attrib_long** | 检索脚本命令行中使用的 *long* 类型变量 |
| **lr_get_attrib_string** | 检索脚本命令行中使用的字符串 |

**信息性函数**

| | |
|---|---|
| **lr_user_data_point** | 记录用户定义的数据示例 |
| **lr_whoami** | 将有关 Vuser 脚本的信息返回给 Vuser 脚本 |
| **lr_get_host_name** | 返回执行 Vuser 脚本的主机名 |
| **lr_get_master_host_name** | 返回运行 LoadRunner Controller 的计算机名 |

**字符串函数**

| | |
|---|---|
| **lr_eval_string** | 用参数的当前值替换参数 |
| **lr_save_string** | 将以 NULL 结尾的字符串保存到参数中 |
| **lr_save_var** | 将变长字符串保存到参数中 |
| **lr_save_datetime** | 将当前日期和时间保存到参数中 |
| **lr _advance_param** | 前进到下一个可用参数 |

| | |
|---|---|
| **lr _decrypt** | 解密已编码的字符串 |
| **lr_eval_string_ext** | 检索指向包含参数数据的缓冲区的指针 |
| **lr_eval_string_ext_free** | 释放由 **lr_eval_string_ext** 分配的指针 |
| **lr_save_searched_string** | 在缓冲区中搜索字符串实例，并相对于该字符串实例，将该缓冲区的一部分保存到参数中 |

### 消息函数

| | |
|---|---|
| **lr_debug_message** | 将调试信息发送到输出窗口 |
| **lr_error_message** | 将错误消息发送到输出窗口 |
| **lr_get_debug_message** | 检索当前消息类 |
| **lr_log_message** | 将消息发送到日志文件 |
| **lr_output_message** | 将消息发送到输出窗口 |
| **lr_set_debug_message** | 设置调试消息类 |
| **lr_vuser_status_message** | 生成带格式的输出，并将其写到 Controller Vuser 状态区域 |
| **lr_message** | 将消息发送到 Vuser 日志和输出窗口 |

### 运行时函数

| | |
|---|---|
| **lr_load_dll** | 加载外部 DLL |
| **lr_peek_events** | 指明可以暂停 Vuser 脚本执行的位置 |
| **lr_think_time** | 暂停脚本的执行，以模拟思考时间 （实际用户在操作之间暂停以进行思考的时间） |
| **lr_continue_on_error** | 指定处理错误的方法 |
| **lr_rendezvous** | 在 Vuser 脚本中设置集合点 |

# 获得函数帮助

可以通过以下方法获得 LoadRunner 函数的帮助：

- LoadRunner 函数参考
- IntelliSense
- 标头文件

## LoadRunner 函数参考

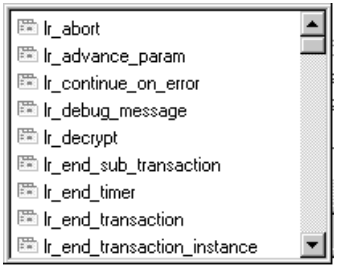
联机的 《Online Function Reference》包含有关所有 LoadRunner 函数的详细语法信息。它还提供了函数示例。可以按名称搜索函数，也可以通过分类列表或按字母顺序排列的列表来搜索函数。

要打开 《Online Function Reference》，请从 VuGen 界面中选择 “帮助” > “函数参考”。然后，选择协议并选择需要的类别。

要获得已经包含在脚本中的特定函数的信息，请将光标放在 VuGen 编辑器中的相应函数上，然后按 F1 键。

## IntelliSense

VuGen 编辑器现在集成了 *Intellisense* （也称为 “字完成”）。开始键入函数时，Intellisense 功能会打开一个列表框，其中显示了函数前缀的所有可用匹配项。




要使用某个函数，请将其选中。 VuGen 随后会将其放在光标位置。要关闭该列表框，请按 Esc 键。

要指示 VuGen 全局使用此功能，请选择 “工具” > “常规选项”，然后选择 “环境”选项卡。选中 “自动完成字”选项旁边的复选框。当键入第一个下划线时，会打开此列表框。默认情况下，字完成功能被全局启用。

要禁用字完成，请选择 “工具” > “常规选项”，然后选择 “环境”选项卡。清除 “自动完成字”选项旁边的复选框。如果全局禁用字完成功能，当在编辑器中键入内容时，仍可以通过按 Ctrl + 空格键或者选择 “编辑” > “完成字”来显示函数列表框。

Intellisense 的另一个功能是“显示函数语法”。当键入函数的左括号时，VuGen 将显示函数的语法及其参数和原型。

要指示 VuGen 全局使用此功能，请选择 “工具” > “常规选项”，然后选择 “环境”选项卡。选中 “自动显示函数语法”选项旁边的复选框。默认情况下，“显示函数语法”被全局启用。

要禁用此功能，请选择 “工具” > “常规选项”，然后选择 “环境”选项卡。清除 “自动显示函数语法”选项旁边的复选框。如果全局禁用 “显示函数语法”，当在编辑器中键入左括号后，仍可以通过按 Ctrl + Shift + 空格键或者选择 “编辑” > “显示函数语法”来显示语法。

### 标头文件

库标头文件中列出了所有函数原型。标头文件位于 LoadRunner 安装的 include 目录中。它们包含详细的语法信息和返回值，还包含常量定义、可用性和 “函数参考”中可能未包含的其他高级信息。

在大多数情况下，标头文件的名称与协议的前缀相对应。例如，以 **lrd** 前缀开头的数据库函数列在 *lrd.h* 文件中。下表显示了与最常用的协议相关联的标头文件：

| 协议 | 文件 |
|---|---|
| Citrix | *ctrxfuncs.h* |
| COM/DCOM | *lrc.h* |
| 数据库 | *lrd.h* |
| FTP | *mic_ftp.h* |
| 通用 C 函数 | *lrun.h* |
| IMAP | *mic_imap.h* |
| LDAP | *mic_mldap.h* |
| MAPI | *mic_mapi.h* |
| MediaPlayer | *mic_media.h* |

| 协议 | 文件 |
| --- | --- |
| Oracle NCA | *orafuncs.h* |
| POP3 | *mic_pop3.h* |
| RealPlayer | *lreal.h* |
| SAPGUI | *as_sap.gui.h* |
| Siebel | *lrdsiebel.h* |
| SMTP | *mic_smtp.h* |
| 终端仿真器 | *lrrte.h* |
| Tuxedo | *lrt.h* |
| Web | *as_web.h* |
| Windows 套接字 | *lrs.h* |

# 3

# 使用 VuGen 进行录制

VuGen 通过录制客户端应用程序与服务器之间的通信来创建 Vuser 脚本。

本章描述：

- Vuser 脚本部分
- 新建虚拟用户脚本
- 添加和删除协议
- 导入操作
- 重新生成 Vuser 脚本

***以下信息适用于除 GUI 以外的所有类型的 Vuser 脚本。***

## 关于使用 VuGen 进行录制

VuGen 通过录制对客户端应用程序执行的操作来创建 Vuser 脚本。运行录制的脚本时，生成的 Vuser 将模拟客户端与服务器之间的用户活动。

创建的每个 Vuser 脚本都至少包含三部分：*vuser_init*、一个或多个 *Actions* 及 *vuser_end*。录制期间，可以选择脚本中 VuGen 要插入已录制函数的部分。通常情况下，可以将登录到服务器的活动录制到 *vuser_init* 部分中、将客户端活动录制到 *Actions* 部分中，并将注销过程录制到 *vuser_end* 部分中。

创建测试后，可以将其保存到 zip 存档文件中，并作为电子邮件附件发送出去。

录制时，可以在脚本中插入事务、注释和集合。详细信息，请参阅第 6 章“增强 Vuser 脚本”。

# Vuser 脚本部分

每个 Vuser 脚本都至少包含三部分：*vuser_init*、一个或多个 *Actions* 及 *vuser_end*。录制前和录制期间，可以选择脚本中 VuGen 要插入已录制函数的部分。下表显示了要在每一部分录制的内容以及执行每一部分的时间。

| 脚本部分 | 录制内容 | 执行时间 |
|---|---|---|
| *vuser_init* | 登录到服务器 | 初始化 Vuser （已加载） |
| *Actions* | 客户端活动 | Vuser 处于运行状态 |
| *vuser_end* | 注销过程 | Vuser 完成或停止 |

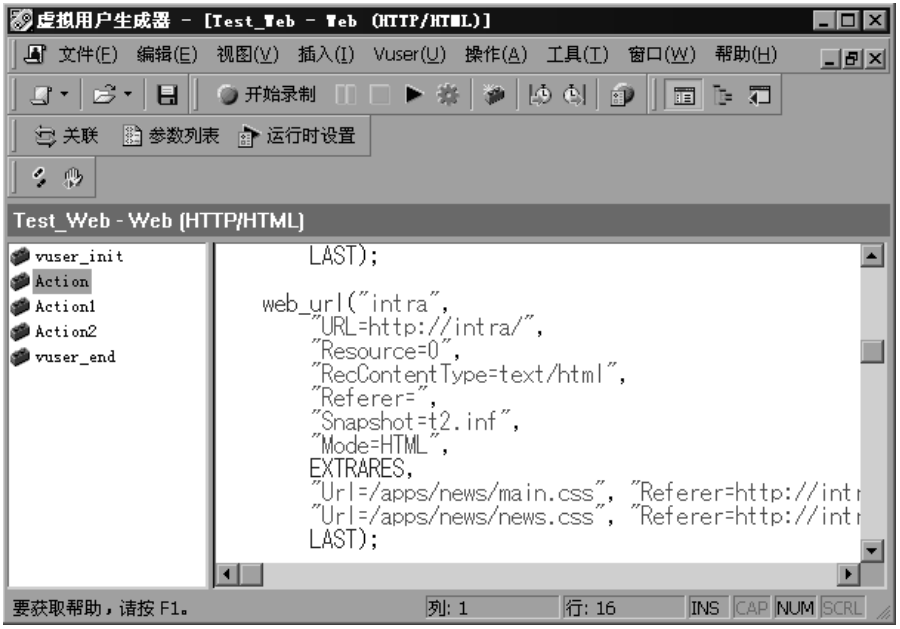
运行多次循环的 Vuser 脚本时，只有脚本的 *Actions* 部分重复，而 *vuser_init* 和 *vuser_end* 部分将不重复。有关循环设置的详细信息，请参阅第 9 章 “配置运行时设置”。

可以使用 VuGen 脚本编辑器来显示并编辑每个脚本部分的内容。但一次只能显示一个部分的内容。要显示某一部分，请在左窗格中突出显示该部分的名称。

在处理使用 Java 类的 Vuser 脚本时，可以将所有代码都置于 *Actions* 类中。Actions 类包含三个方法：*init*、*action* 和 *end*。这些方法对应于脚本中使用其他协议开发的部分，您可以在 *init* 方法中插入初始化例程、在 *action* 方法中插入客户端操作，并在 *end* 方法中插入注销过程。详细信息，请参阅第 24 章 “编写 Java 脚本”。

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

在下例中， VuGen 脚本编辑器将显示 Web Vuser 脚本的 *Action1* 部分。



# 新建虚拟用户脚本

通过 VuGen，可以采用单协议或多协议模式进行录制，从而新建脚本。

单击 “新建”时，将打开 “新建虚拟用户”窗口。该对话框可提供下列选项的快捷方式：

**新建单协议脚本**：创建单协议 Vuser 脚本。这是 “启动”对话框打开时的默认选项。要创建单协议脚本，请从 “选择虚拟用户类别”中进行选择并在该窗口的右窗格中选择协议。

**新建多协议脚本**：创建多协议 Vuser 脚本。 VuGen 将显示所有可用的协议，并可以指定要录制的协议。要创建多协议脚本，请选择一个协议并单击右箭头将其移入 “选定的协议” 部分中。

**使用最近使用过的协议新建脚本：**列出最近用于新建 Vuser 脚本的协议，并指示它们是单协议还是多协议。从列表中选择协议并单击 “确定”，为该协议新建脚本。

录制单协议时， VuGen 将只录制指定的协议。采用多协议模式进行录制时，VuGen 将录制多个协议中的操作。下列协议支持多协议脚本：COM、 FTP、IMAP、Oracle NCA、POP3、RealPlayer、Window Sockets（原始）、SMTP 和 Web。“双协议 Web/W”的引擎使用一种不同的机制，因此应视为单协议（不能与其他多协议类型结合使用）。

各种 Vuser 类型之间的另一个区别是多操作支持功能。大多数协议都可支持多个操作部分。目前，下列协议支持多操作：Oracle NCA、 Web、 RTE、General (C Vusers)、 WAP、 i-Mode 和 VoiceXML。

对于大多数 Vuser 类型，在每次录制时都会新建一个 Vuser 脚本，您不能在现有脚本中进行录制。但是，在录制 Java、CORBA-Java、RMI-Java、Web、WAP、i-mode、Voice XML、Oracle NCA 或 RTE Vuser 脚本时，可以同时在现有脚本中进行录制。

由于 LoadRunner 支持多种协议，因此下面所述的某些录制步骤仅适用于特定协议。

对于所有 Java 语言的 Vuser（CORBA、RMI、Jacada 和 EJB），请参阅第 13 章“录制 Java 语言 Vuser 脚本”。了解有关录制的详细信息，或者请参阅讨论特定协议的章节。

## 添加和删除协议

录制多协议会话之前，可以使用 VuGen 来修改要在录制会话过程中生成代码的协议列表。如果在创建脚本时指定了某些协议，则可以使用 “协议录制”选项启用或禁用这些协议。

要打开录制选项，请选择 “工具” > “录制选项”，或者按下 CTRL+F7。选择“常规 : 协议”节点。

选中要在下一个录制会话中录制的协议旁边的复选框。

清除不希望在下一个录制会话中录制的协议旁边的复选框。

# 选择虚拟用户类别

Vuser 类型分为下列类别：

- **所有协议**：按字母顺序排列的所有可支持协议的列表。
- **应用程序部署解决方案**：适用于 Citrix 协议。
- **客户端 / 服务器**：适用于 MS SQL、 ODBC、 Oracle （两层）、 DB2 CLI、 Sybase Ctlib、 Sybase Dblib、 Windows Sockets 和 DNS 协议。
- **自定义**：适用于 C 模板、 Visual Basic 模板、 Java 模板、 Javascript 和 VBscript 类型脚本。
- **分布式组件**：适用于 COM/DCOM、 Corba-Java 和 Rmi -Java 协议。
- **电子商务**：适用于 FTP、 LDAP、 Palm、 SOAP、 Web (HTTP/HTML) 和双 Web/Winsocket 协议。
- **Enterprise Java Bean**：适用于 EJB 测试和 Rmi-Java 协议。
- **ERP/CRM**：适用于 Baan、Oracle NCA、Peoplesoft-Tuxedo、Peoplesoft 8 Web 多语言、 SAPGUI、 SAP-Web 和 Siebel （Siebel-DB2CLI、 Siebel-MSSQL、 Siebel-Web 和 Siebel-Oracle）协议。
- **传统**：适用于终端仿真 (RTE)。
- **邮件服务**：Internet 邮件访问协议 (IMAP)、 MS Exchange (MAPI)、 POP3 和 SMTP。
- **中间件**：Jacada 和 Tuxedo （6、 7）协议。
- **流**：适用于 MediaPlayer 和 RealPlayer 协议。
- **无线**：适用于 i-Mode、 VoiceXML 和 WAP 协议。

## 新建脚本

**要新建 Vuser 脚本，请执行下列操作：**

1 选择 “开始” > “程序” > “LoadRunner” > “虚拟用户生成器”，启动 VuGen。将打开启动屏幕 （除非上次打开 VuGen 时禁用了该屏幕）。

2 要创建单协议脚本，请从 “类别”列表中进行选择，然后选择一个协议。

3 要创建多协议脚本，以便在单个录制会话中录制两个或两个以上的协议，请单击左窗格中的 “新建多协议脚本”按钮，启用 “协议选择”窗口。

从 “可用协议”列表中选择所需的协议。单击右箭头，将所选内容移入 “选定的协议”列表中。对所需的所有协议重复该步骤。

---

**注意：**录制某些 Oracle NCA 应用程序时，只需选择 “Oracle NCA”，而不是 “Web 协议”。详细信息，请参阅第 47 章 “创建 Oracle NCA Vuser 脚本”。对于 SOAP，请使用 “单协议”列表中的 Web/WinSock 双协议。

---

4 要在下次打开 VuGen 时不显示该启动窗口，请选择 “以后不再显示启动对话框”选项。要再次启用该窗口，请选择 “工具” > “常规选项”，然后选择 “环境”选项卡上的 “显示 ‘启动’对话框”。

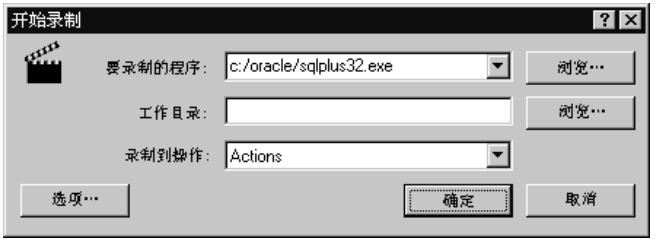
5 单击 “确定”关闭对话框并开始生成 Vuser 脚本。

## 录制脚本

对于大多数 Vuser 脚本类型，在新建脚本时， VuGen 会自动打开 “开始录制”对话框。

1 如果未打开 “开始录制”对话框，请单击 “开始录制”按钮。将打开 “开始录制”对话框。根据录制的协议，该对话框会有所不同。

2 对于大多数客户端 / 服务器协议，将打开以下对话框：



输入要录制的程序、工作目录（可选）和操作。如果必要，请单击“选项”设置录制选项。

3 对于非 Internet 应用程序，请选择此处的两种应用程序类型：Win32 应用程序或 Internet 应用程序。例如， Web 和 Oracle NCA 脚本将录制 Internet 应用程序，而 Windows Socket Vuser 则录制 Win32 应用程序。

4 对于 Internet 应用程序，请填写下列相关信息：

**要录制的程序：**选择要录制的浏览器或 Internet 应用程序。对于 Citrix，请在 bin 目录中查找 *rundlg.exe* 文件。

**URL 地址：**指定起始 URL 地址。

**工作目录：**对于要求指定工作目录的应用程序，请在此处指定工作目录。根据 Vuser 脚本的类型，所需的信息将有所不同。

5 对于 Win32 应用程序，请填写下列相关信息：

**要录制的程序**：输入要录制的 Win 32 应用程序。

**程序参数**：为上面指定的可执行程序指定命令行参数。例如，如果在指定 *plus32.exe* 时使用命令行选项 peter@neptune，则在启动 *plus32.exe* 时它会将用户 *Peter* 连接至服务器 *Neptune*。

**工作目录**：对于要求指定工作目录的应用程序，请在此处指定工作目录。

**6** 在“录制到操作”框中，选择要录制到的部分。最初，可以使用 *vuser_init*、*Action1* 和 *vuser_end* 部分。对于支持多个操作（Oracle NCA、Web、RTE、C Vusers、WAP、i-Mode 和 VoiceXML）的单协议 Vuser 脚本，可以通过选择“操作”>“新建操作”来添加新的部分，并指定新的操作名。

**7** 要录制应用程序启动，请选择“录制应用程序启动”（不适用于 Java 类型的 Vuser 脚本）。要指示 VuGen 不录制应用程序启动，请清除该复选框。在下列实际情形中，最好不要录制启动：

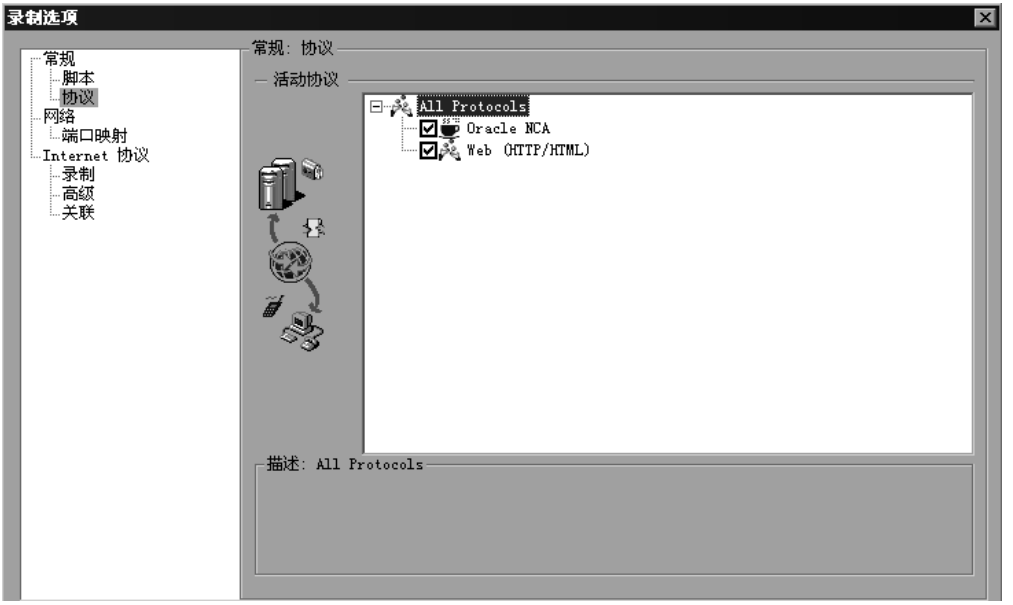
- 如果正在录制多个操作，只需在一个操作中录制启动。
- 开始录制前希望导航到应用程序中的特定位置。
- 正在现有脚本中录制。

**8** 单击“选项”或“录制选项”按钮，打开“录制选项”对话框并设置录制选项。根据录制的协议，可用选项可能有所不同。详细信息，请参阅相应的章节。

**9** 要选择生成代码所使用的语言并设置脚本选项，请单击“脚本”选项卡。详细信息，请参阅第 4 章“设置脚本生成首选项”。

**10** 要指定端口信息，请单击“端口映射”选项卡。如果要在非标准端口上录制 SSL 应用程序，该信息会非常有用。查看端口列表。如果要使用的端口不在列表中，则可以使用“端口映射”选项指定该信息。详细信息，请参阅第 5 章“配置端口映射”。

**11** 对于多协议录制：要修改希望录制的协议列表，请单击 “协议”选项卡。展开节点并选择所需的协议。




此时，便可开始录制了。

**12** 单击 “确定”关闭对话框并开始录制。

**13** 如果清除了 “录制应用程序启动”复选框，则会显示 “录制已暂停”对话框。当到达要开始录制的位置时，请单击 “录制”。如果决定不录制，请单击 “放弃”。

**14** VuGen 将启动应用程序，然后会显示 “录制”工具栏。

在应用程序中执行典型的操作。 VuGen 将同时填充 Vuser 脚本中的选定操作部分。在录制过程中，可以使用浮动工具栏在各个部分之间切换。

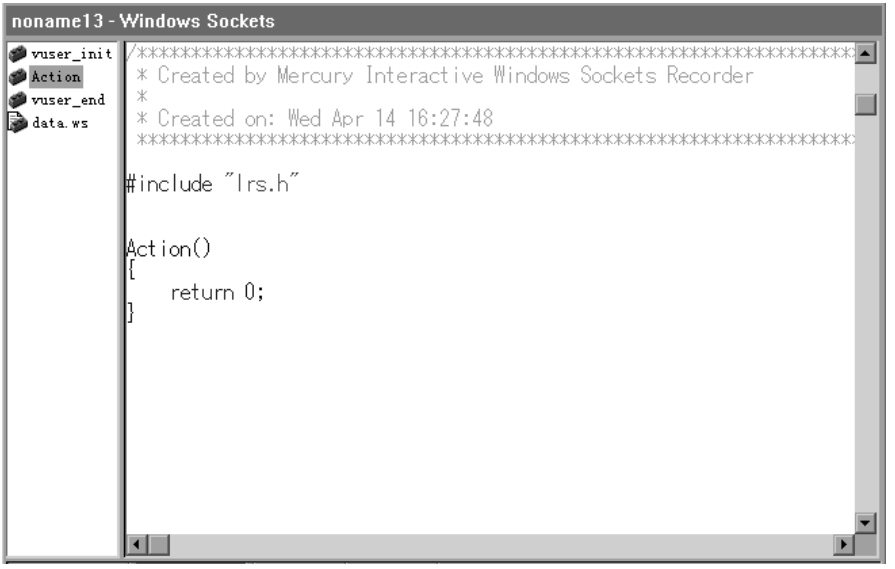
### 结束录制会话

录制完典型的业务流程后，可以通过执行业务流程的结束步骤并保存 Vuser 脚本来完成录制会话。

要完成录制，请执行下列操作：

1 切换到浮动工具栏中的 *vuser_end* 部分，并执行注销或清除过程。

2 单击 “录制”工具栏上的 “停止录制”按钮。 VuGen 编辑器将显示录制的所有语句。



3 单击 “保存”以保存录制的会话。将打开 “保存测试”对话框 （仅限新的 Vuser 脚本）。指定脚本名。**注意：** 切勿命名脚本中的 *init*、 *run* 或 *end*，因为这些名称将由 VuGen 使用。

4 要将整个脚本目录另存为 zip 文件，请选择 “文件” > “导出到 Zip 文件”。

指定要保存的文件。要仅保存运行时文件，请选择 “要压缩的文件”部分中的 “运行时文件”。默认情况下， VuGen 会将所有文件保存到存档文件中。

选择压缩比例：最大值、普通、快速、最快或无。压缩比例越大， VuGen 创建存档文件的时间越长。

单击 “确定”。

5 要创建 zip 文件并作为电子邮件附件发送出去，请选择 “文件” > “Zip 并发送邮件”。

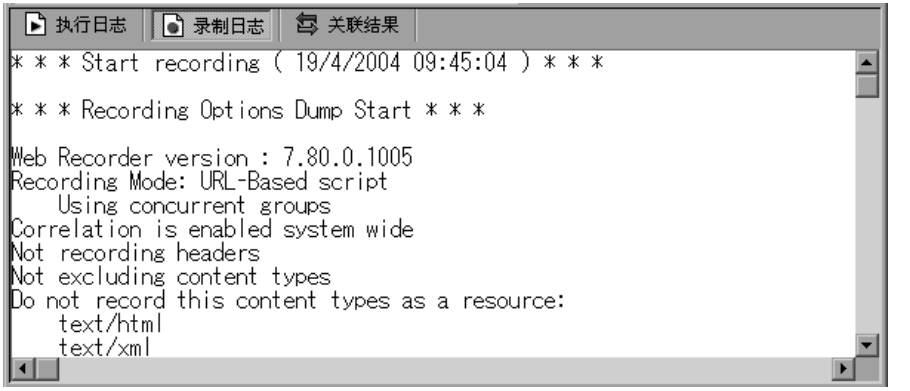
单击 “确定”。将打开电子邮件撰写窗体。

输入电子邮件地址并发送电子邮件。

### 查看录制日志

录制后，可以在 VuGen 脚本编辑器中查看 *vuser_init*、*Actions* 和 *vuser_end* 部分的内容。要显示操作，请在左窗格中选择操作名。

要查看录制期间发出的消息的日志，请选择 “视图” > “输出窗口”，然后选择 “录制日志”选项卡。可以在 “录制选项”的 “高级”选项卡中设置该日志的详细级别。




录制时， VuGen 会创建一系列配置、数据和源代码文件。这些文件包含 Vuser 运行时和设置信息。 VuGen 会将这些文件连同脚本一起进行保存。

可以在 VuGen 编辑器中手动编辑脚本。对于支持多个操作的协议，可以随时录制其他操作。

## 导入操作

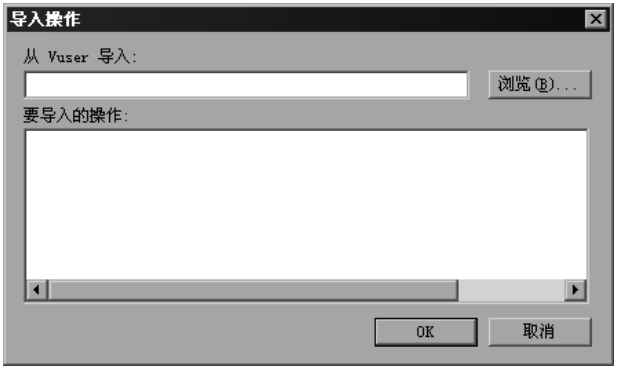
对于支持多个操作的 Vuser 类型，可以将其他 Vuser 脚本中的操作导入到当前脚本中。只能从相同类型的 Vuser 中导入操作。注意，与导入的操作相关的任何参数都将与脚本合并。可用选项有：

**从 Vuser 导入：**输入或浏览要从中导入操作的 Vuser 脚本。

**要导入的操作：**选择要导入的操作。

要将操作导入到当前脚本中，请执行下列操作：

1 选择 “操作” > “从其他 Vuser 中导入”。将打开 “导入操作”对话框。




2 单击 “浏览”选择 Vuser 脚本。该脚本中的操作列表将显示在 “要导入的操作”部分中。

3 突出显示某个操作并单击 “确定”。该操作将显示在脚本中。

4 要重新排列操作的顺序，必须首先启用 action 重新排序。右键单击任意操作，并选择 “启用 action 重新排序”。然后根据需要拖动操作，使其排列在所需的位置上。注意，在 VuGen 的左窗格中对操作进行重新排序时，不会影响执行这些操作的顺序。要更改执行顺序，请按照第 9 章 “配置运行时设置”。中的说明使用 “运行时设置”中的 “步”节点。

## 重新生成 Vuser 脚本

录制脚本后，可以通过添加事务、集合、消息或注释来增强该脚本。详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

此外，还可以对该脚本进行参数化，并关联其中的变量。详细信息，请参阅第 7 章 “定义参数”。

如果需要还原为最初录制的脚本，则可以重新生成该脚本。对于调试或修复已损坏的脚本，该功能非常有用。重新生成脚本时，该功能将删除在录制的操作中手动添加的所有增强内容。注意，如果已在脚本中添加了参数，则 VuGen 会将其还原为原始值。但不会删除参数列表；您可以重新插入先前创建的参数。注意，重新生成过程只会清除录制的操作，而不会清除手动添加的操作。

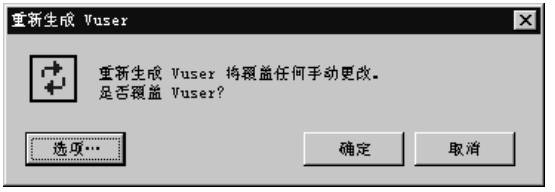
“重新生成 Vuser”对话框中包含下列按钮。

**确定**：从原始的录制日志中重新生成 Vuser 脚本。重新生成过程将删除在脚本中手动执行的所有关联和参数化。

**选项**：使用多协议脚本时，可以指定要重新生成哪些协议。要自定义重新生成，请单击“重新生成 Vuser”对话框中的“选项”按钮，打开录制选项。选择“协议”选项卡并指定要重新生成哪些协议以及要保留哪些协议。选中要重新生成的协议所对应的复选框。清除不希望重新生成的协议所对应的复选框。

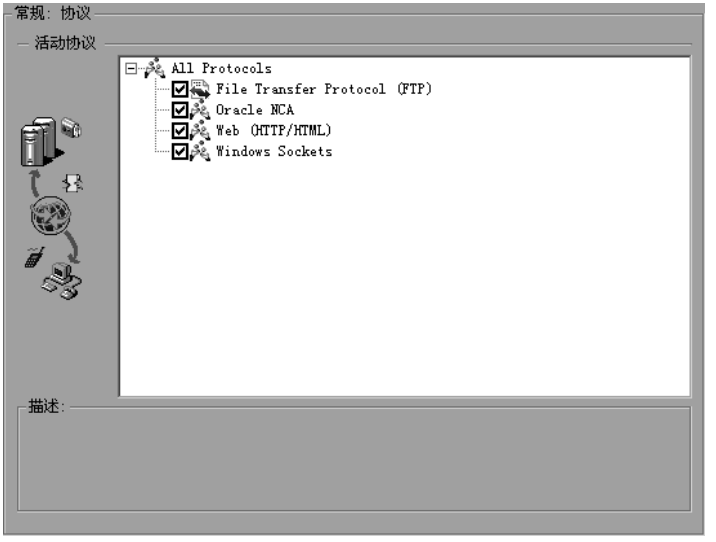
**要重新生成多协议 Vuser 脚本，请执行下列操作：**

1 选择“工具”>“重新生成 Vuser”。 VuGen 将发出一条警告，指示所有手动进行的更改都会被覆盖。

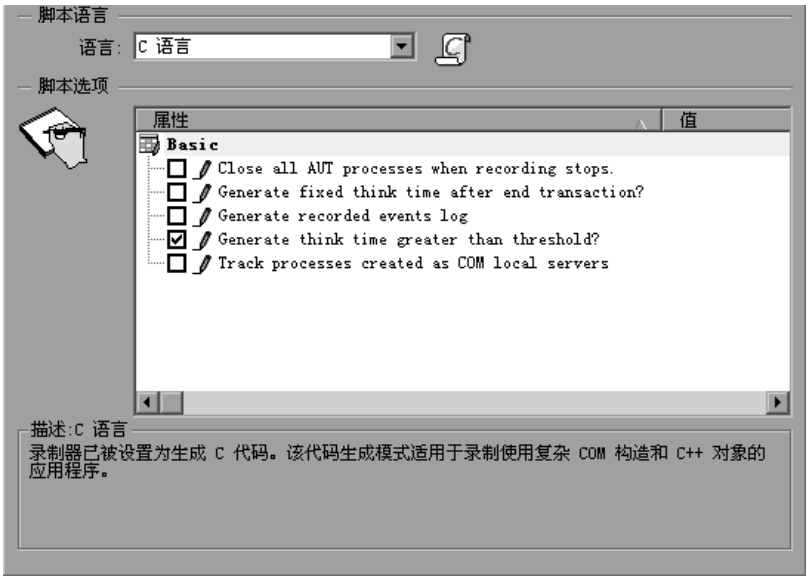



2 单击“选项”打开“重新生成选项”对话框。

3 选择“常规：协议”节点。指定要重新生成哪些协议以及要保留哪些协议。选中要重新生成的协议所对应的复选框。选中要保持不变的协议所对应的复选框。



**4** 要更改脚本选项，请选择 “常规 : 脚本”节点并选中或清除相应的复选框。



# 4

# 设置脚本生成首选项

使用 VuGen 录制脚本之前，应该指明需要的脚本语言：C、 Visual Basic、 VB Script 或 Javascript。本章描述适用于众多所支持的协议的脚本语言录制选项。

- 选择脚本语言
- 应用基本选项
- 了解关联选项
- 设置脚本录制选项

***以下信息适用于支持多协议录制的所有 Vuser 脚本。***

## 关于设置脚本生成首选项

录制会话之前，可以设置几个录制选项，以指示录制器在脚本中包括哪些内容以及如何生成脚本。

如果您要录制的协议中至少有一种具有多协议功能，则可以使用 “脚本”选项。唯一的例外是当您将 HTTP 或 WinSock 录制为单一协议脚本时。在这种情况下，“脚本”选项不可用。

## 选择脚本语言

录制会话时，默认情况下 VuGen 会创建一个模拟您的操作的脚本。默认的脚本生成语言为 C。对于 FTP、 COM/DCOM 和邮件协议 （IMAP、 POP3 和 SMTP），VuGen 还可以使用 Visual Basic、 VB Script 和 Javascript 来生成脚本。

**C 语言** – 用于那些使用复杂的 COM 构造和 C++ 对象的录制应用程序。

**Visual Basic for Applications** – 用于基于 VB 的应用程序，这些应用程序使用 VB 的完整功能 （与 VBScript 不同）。

**Visual Basic Scripting** – 用于基于 VBscript 的应用程序，例如 ASP。

**Java Scripting** – 用于基于 Javascript 的应用程序，例如 *js* 文件和动态 HTML 应用程序。

录制会话后，可以使用常规的 C、 Visual Basic、 VB Script 或 Javascript 代码或控制流语句来修改脚本。

以下几个部分描述了脚本选项。对于所有脚本，请参阅第 44 页的 “应用基本选项”。要为非 C 脚本设置关联选项，请参阅第 45 页的 “了解关联选项”。

详细说明，请参阅第 46 页的 “设置脚本录制选项”。

## 应用基本选项

“基本”脚本选项适用于所有生成语言。通过这些选项可以控制生成的脚本中的详细信息级别。

**录制停止时关闭所有 AUT 进程**：VuGen 停止录制时，自动关闭所有正在测试的应用程序 (AUT) 的进程。（默认情况下禁用）。

**显式变量声明** – 显式声明变量类型，以便处理 *ByRef* 变量。（仅限于 Visual Basic for Applications，默认情况下禁用）。

**结束事务后生成固定的思考时间**：在每个事务结束后添加固定的思考时间 （以秒为单位）。如果启用此选项，则可以指定思考时间的值。默认值为 3 秒。（默认情况下禁用）。

**生成录制事件日志**：生成在录制过程中发生的所有事件的日志。（默认情况下禁用）。

**思考时间大于阈值时生成：**使用思考时间的阈值。如果录制思考时间小于该阈值，VuGen 将不会生成思考时间语句。您也可以指定阈值。默认值为 3 – 如果思考时间少于 3 秒，VuGen 将不会生成思考时间语句。如果禁用此选项，VuGen 将不会生成任何思考时间。（默认情况下启用）

**插入预调用信息 –** 每次调用消息之前插入信息性日志记录消息。（仅限 C，默认情况下为启用）

**插入后调用信息 –** 每次调用消息之后插入信息性日志记录消息。（仅限于 C，默认情况下启用）

**跟踪创建为 COM 本地服务器的进程：**如果录制的应用程序的某个子进程被创建为 COM 本地服务器，则跟踪该应用程序的活动。（默认情况下禁用）

**使用数组帮助器 –** 使用帮助器功能来提取变量数组中的成分。（仅限于 Java 和 VB Scripting，默认情况下禁用）

**使用对象帮助器 –** 使用帮助器功能从作为函数参数传递的变量中提取对象引用。（仅限于 Java 和 VB Scripting，默认情况下禁用）

详细说明，请参阅第 46 页的 “设置脚本录制选项”。

## 了解关联选项

通过关联可在测试执行过程中保存动态值。使用这些设置可以配置 VuGen 在录制过程中执行的自动关联的程度。所有关联选项在默认情况下都被禁用。关联选项仅适用于 VBScript 和 JScript 语言。

**关联小数 –** 关联短数据类型，例如字节、字符和短整型。（默认情况下禁用）

**关联大数 –** 关联长数据类型，例如整型、长整型、 64 位字符、浮点型和双精度型。（默认情况下禁用）

**关联简单字符串 –** 关联简单的非数组字符串和短语。（默认情况下启用）

**关联数组 –** 跟踪并关联所有数据类型的数组，例如字符串、结构、数字等。（默认情况下禁用）

**关联结构 –** 跟踪并关联复杂的结构。（默认情况下禁用）

详细说明，请参阅第 46 页的 “设置脚本录制选项”。

# 设置脚本录制选项

您应该在初次进行与脚本相关的录制之前设置录制选项。可用选项的数量取决于脚本生成语言。

### 设置脚本录制选项

1 打开录制选项。从主菜单中选择 “工具” > “录制选项”，或者在 “开始录制” 对话框中单击 “选项 ...” 对话框。将打开 “录制选项” 对话框。

2 选择 “常规：脚本” 节点。

3 在 “选择脚本语言” 框中，选择代码生成模式 – “C 语言” 或 “Visual Basic for Applications”。使用 C 来录制那些使用复杂构造和 C++ 代码的应用程序。使用 Visual Basic 来录制基于脚本的应用程序。

4 在 “脚本选项” 部分中，通过选中所需选项旁边的复选框来启用相应的选项。前面的部分中解释了这些选项。

5 单击 “确定” 保存设置并关闭对话框。

# 5

# 配置端口映射

使用在套接字级别录制网络流量的协议时，您可以指明要将通信映射到的端口。

本章描述：

- 定义端口映射
- 添加新的服务器项
- 设置自动检测选项
- 设置端口映射录制选项

***下列信息适用于在套接字级别进行录制的所有 Vuser 脚本：HTTP、SMTP、POP3、IMAP、Oracle NCA 和 WinSocket。***

## 关于配置端口映射

当录制用于在套接字级别（HTTP、SMTP、POP3、FTP、IMAP、Oracle NCA 和 WinSocket）录制网络流量的 Vuser 脚本时，您可以设置“端口映射”选项。使用这些选项，可以将来自特定服务器 : 端口组合的流量映射到需要的通信协议。

可以映射到的可用通信协议包括 FTP、HTTP、IMAP、NCA、POP3、SMTP 和 SOCKET。创建映射的方法是指定服务器名、端口号或完整的服务器 : 端口组合。例如，您可以指明来自服务器 *twilight* 的端口 25 的所有流量都应该作为 SMTP 进行处理。还可以指定来自名为 *viper* 的服务器的所有流量都应该映射到 FTP 协议，而无论端口是什么。此外，还可以将端口 23 上的所有流量都映射到 SMTP，而无论服务器名是什么。

以多协议模式录制时，如果至少有一种协议在套接字级别进行录制，则“端口映射”选项将可用。唯一的例外是当您将 HTTP 或 WinSock 录制为单一协议脚本时。在这种情况下，“端口映射”选项不可用。

# 定义端口映射

VuGen 使用 “端口映射”设置，通过特定的服务器 : 端口组合将流量定向到需要的通信协议。

**网络级服务器地址映射：**指定每个协议的映射。例如，要仅显示 FTP 映射，则选择 FTP。

**新建项：**打开 “服务器项”对话框，可以添加新映射。请参阅第 49 页的 “添加新的服务器项”。

**编辑项：**打开 “服务器项”对话框，可以编辑选定的项。

**删除项：**删除选定的项。

**选项 :** 打开 “高级设置”对话框，启用通信协议和 SSL 级别的自动检测。请参阅第 51 页的 “设置自动检测选项”。

如果未指定全部端口和服务器名， VuGen 在将数据分配给服务时将使用以下优先顺序：

| 优先顺序 | 端口 | 服务器 |
|---|---|---|
| 1 | 指定 | 指定 |
| 2 | 未指定 < 全部 > | 指定 |
| 3 | 指定 | 未指定 < 全部 > |
| 4 | 未指定 < 全部 > | 未指定 < 全部 > |

优先级高的映射项不会被优先级较低的项覆盖。例如，如果指定将服务器 *twilight* 端口 25 上的流量作为 SMTP 处理，然后指定将所有服务器的端口 25 上的流量作为 HTTP 处理，则数据将被作为 SMTP 处理。

此外，还适用下列规则：

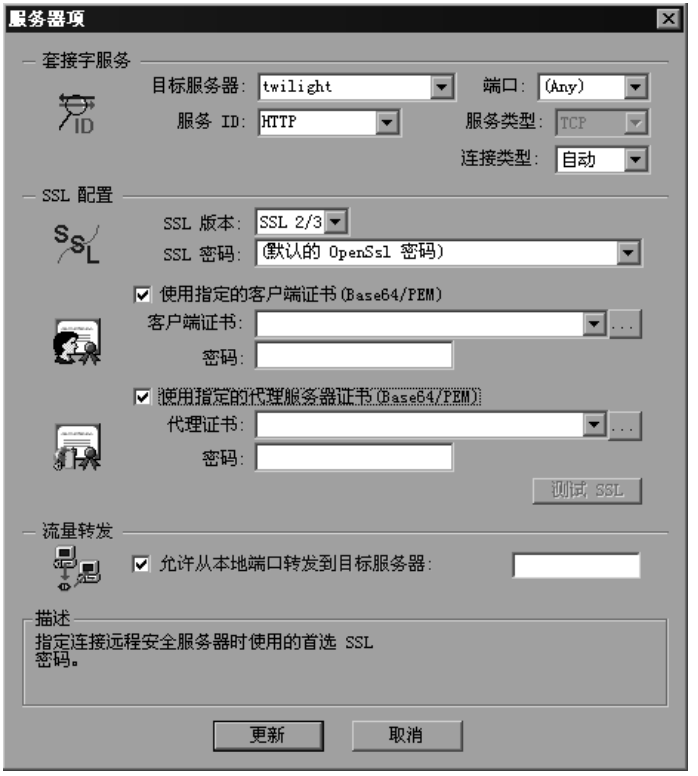
- **端口 0:** 端口号 0 指明任意端口。
- **强制映射:** 如果您为某个端口号、服务器名或服务器 : 端口组合指定映射，VuGen 将强制网络流量使用该服务。例如，如果您要指定 < 任何 > 服务器的端口 80 使用 FTP，则 VuGen 将使用 FTP 协议录制该通信，即使实际通信可能为 HTTP。在上例中， Vuser 脚本可能为空。

定义端口映射后，该映射将显示在 “端口映射”列表中。您可以通过清除任意项旁边的复选框来暂时禁用该项。禁用某一项后，VuGen 将忽略所有流向该服务器 : 端口组合的流量。数据无关或者协议不受支持时，应禁用端口项。

详细说明，请参阅第 52 页的 “设置端口映射录制选项”。

## 添加新的服务器项

使用 “服务器项”对话框在端口映射列表中新建项。




### 套接字服务

**目标服务器：**该项适用的目标服务器的 IP 地址或主机名。默认值为所有服务器。

**端口：**该项适用的目标服务器的端口。端口 0 表示所有端口。

**服务 ID**：录制器用于标识连接类型的协议或服务名。（即 HTTP、 FTP 等）。您还可以指定一个新名称。该名称不能超过 8 个字符。

**服务类型**：服务的类型，当前设置为 TCP。

**连接类型**：连接的安全级别："普通"（不安全）、"SSL"或 "自动"。如果选择 "自动"，录制器将检查前 4 个字节寻找 SSL 签名。如果检测到 SSL 签名，将假定该 SSL 正在使用。

### SSL 配置

如果选择 "SSL"或 "自动"作为连接类型，请配置本部分中的相关 SSL 设置。这些设置仅适用于新项。您应该仅当明确了解应用程序的 SSL 编码时，才指定它们。否则，将接受默认值。

**SSL 版本**：与客户端应用程序和服务器通信时使用的首选 SSL 版本。默认情况下使用 SSL 2/3。但是，某些服务要求仅使用 SSL 3.0 或 SSL 2.0。某些新的无线应用程序要求使用 TLS 1.0 – 一种不同的安全算法。

**SSL 密码**：与远程安全服务器连接时使用的首选 SSL 密码。

**使用指定的客户端证书**：连接远程服务器时使用的默认客户端证书。指定或浏览 *txt*、 *crt* 或 *pem* 格式的证书文件，并提供一个密码。

**使用指定的代理服务器证书**：提供给请求服务器证书的客户端应用程序的默认服务器证书。指定或浏览 *txt*、 *crt* 或 *pem* 格式的证书文件，并提供一个密码。单击 "测试 SSL"针对服务器检查身份验证信息。

### 流量转发

**允许从本地端口转发到目标服务器**：此选项将来自特定端口的所有流量转发到另一个服务器。在 VuGen 无法在客户端 （例如唯一的 UNIX 计算机）上正常运行，或者无法通过 VuGen 启动应用程序服务器的情况下，此选项特别有用。我们将 VuGen 配置为侦听来自有问题的客户端计算机的流量，并将其传递到服务器。这样， VuGen 便可以处理数据并为操作生成代码。

例如，如果您在名为 *host1* 的 UNIX 客户端上工作，该客户端与服务器 *server1* 通过端口 8080 通信，您可以为 *server1*、端口 8080 创建一个 "端口映射"项。在 "服务器项"对话框的 "流量转发"部分中，通过选中 "允许从本地端口转发到目标服务器"复选框启用流量转发。在本例中，将想要从其转发流量的端口指定为 8080。

然后，将客户端 *host1* 连接到运行 VuGen 的计算机，而不是连接到 *server1*。VuGen 从客户机接收通信，并通过本地端口 8080 将其转发到服务器。由于流量流经 VuGen，因此 VuGen 可以分析流量并生成相应的代码。

详细说明，请参阅第 52 页的 "设置端口映射录制选项"。

## 设置自动检测选项

默认情况下没有定义任何映射，并且 VuGen 采用 “自动检测”。 VuGen 的自动检测功能可以分析发送给服务器的数据。 VuGen 检查数据寻找签名。签名是数据内容中的一种模式，用于标识协议。为了检测签名，将组合返回第一个接收缓冲区之前的所有发送缓冲区。所有在接收缓冲区返回之前发送的发送缓冲区都被视为单次数据转换。在某些协议中， VuGen 通过一次转换便可确定类型 （例如 HTTP）。其他网络协议则需要数次转换才能确定类型。出于此目的， VuGen 为每个服务器 – 端口组合创建一个临时缓冲区。如果 VuGen 无法通过读取第一个转换缓冲区来确定协议类型，将把数据存储在临时缓冲区中。然后，它将继续读取传入的缓冲区，直到检测到特定协议的签名。

默认情况下，VuGen 允许进行 4 次转换，并使用一个大小为 2048 字节的临时缓冲区检测协议签名。如果 VuGen 在到达最大转换次数或者到达最大缓冲区大小后仍未确定类型，将把数据分配给 WinSock 协议。如果您未指示 VuGen 录制 WinSock 协议 （在多协议选择中）， VuGen 将丢弃数据。

您可以更改希望 VuGen 为检测到协议类型而读取的最大缓冲区数量。还可以指定临时缓冲区的大小。如果第一个发送缓冲区中的数据量大于临时缓冲区的大小， VuGen 将无法自动检测协议类型。在这种情况下，您应该增加临时缓冲区的大小。

**允许自动检测 SSL**：自动检测 SSL 通信。指定您希望检测的版本和默认密码。注意，此功能仅适用于在 “连接类型” 框中定义为 “自动” 或者根本未定义的端口映射。如果服务器、端口或服务器 : 端口组合被定义为 “普通” 或 “SSL”，则无法使用自动 SSL 检测。

**允许自动检测基于套接字通信的类型**：自动检测通信类型。如果需要，请增加最大转换次数，每次增加 1，直到 VuGen 成功地检测到协议。您还可以逐渐增加最大缓冲区大小 （每次增加 1024 字节，即 1 KB），直到 VuGen 成功地检测到协议。这样可使 VuGen 查看更大数量的数据以便找到签名。

**更新**：接受自动检测选项并关闭对话框。

使用上述网络级别协议时，建议您允许 VuGen 使用自动检测来确定协议类型。在大多数情况下， VuGen 的录制器能够识别这些协议的签名。然后， VuGen 可以根据协议规范自动处理它们。但是，在某些情况下， VuGen 可能无法识别协议。例如：

- 协议签名与某个现有协议几乎相同，使得处理结果出现误差。
- 协议没有唯一的签名。
- 协议使用 SSL 加密，因此无法在 WinSock 级别进行识别。

在上述所有情况中，您可以提供信息唯一地标识承载协议的服务器和端口。

详细说明，请参阅第 52 页的 “设置端口映射录制选项”。

## 设置端口映射录制选项

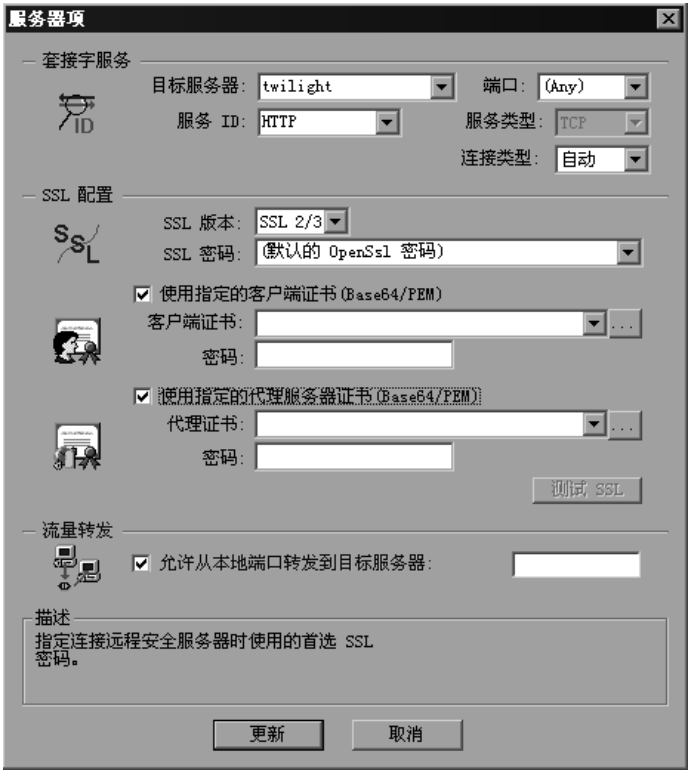
注意，您可以通过下列几种方法打开 “录制选项”对话框：

- 工具栏按钮：
- 键盘快捷键：Ctrl+F7
- “工具”菜单：选择 “工具” > “录制选项”

**要设置端口映射录制选项，请执行下列操作：**

**1** 打开 “录制选项”并选择 “网络 : 端口映射”节点。

**2** 要新建服务器 : 端口映射，请单击 “新建项”。将打开 “服务器项”对话框。




**3** 在 “套接字服务”部分输入 “服务 ID”、“服务类型”、“目标服务器”、“目标端口”和 “连接类型”：

**4** 如果选择了 “SSL”或 “自动”作为连接类型，请在 “SSL 配置”部分中配置相关的 SSL 设置。这些设置仅应用于新项。您应该仅当明确了解应用程序的 SSL 编码时，才指定它们。否则，请接受默认值。

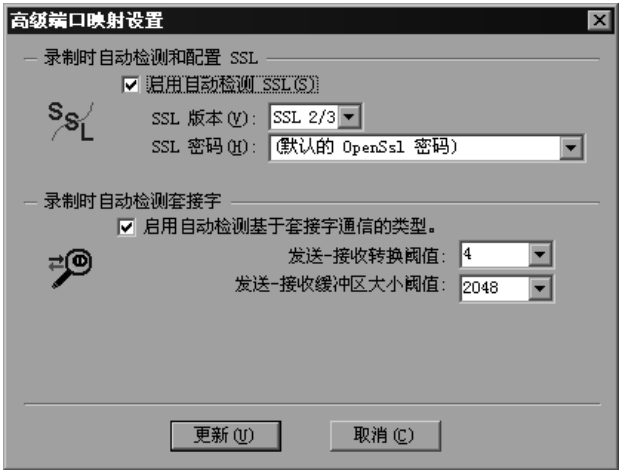
指定 “SSL 版本”、“SSL 密码”。要使用证书，请选择 “使用指定的客户端证书”或 “使用指定的代理服务器证书”并指定用户信息。

单击 “测试 SSL”针对服务器检查身份验证信息。

5 要允许进行流量转发，请选择 “允许从本地端口转发到目标服务器”，并指定端口号。注意，仅当 “目标服务器”和 “目标端口”唯一 （不是 < 任意 >）时，才会启用此选项。

6 单击 “更新”保存映射并关闭 “服务器项”对话框。

7 要设置自动检测功能，请单击 “选项”。将打开 “高级端口映射设置”对话框。




要自动检测 SSL 通信，请选择 “允许自动检测 SSL”并指定版本和密码信息。

要自动检测通信类型，请选择 “允许自动检测基于套接字通信的类型”，如果需要，还应增加最大转换次数。

单击 “更新”接受自动检测选项并关闭对话框。

8 要查看所有项，请在 “网络级服务器地址映射”框中选择 “所有 ID”。

9 要修改现有项，请选择该项并单击 “编辑项”。注意，您无法更改项的服务器名和端口号。只能更改连接类型和安全设置。

10 要永久删除某个映射，请从列表中选择相应的项并单击 “删除项”。要暂时禁用特定项的映射设置，请清除该项旁边的复选框。要启用映射，请选中相应的复选框。

11 单击 “确定”。

# 6

# 增强 Vuser 脚本

可以在录制期间或者录制后，通过添加常规 Vuser 函数、协议特定的 Vuser 函数和标准 ANSI C 函数来增强 Vuser 脚本。

本章描述：

- 将事务插入到 Vuser 脚本
- 将集合点插入到 Vuser 脚本
- 将注释插入到 Vuser 脚本
- 获取 Vuser 信息
- 将消息发送到输出
- 处理 Vuser 脚本在执行期间出现的错误
- 同步 Vuser 脚本
- 模拟用户思考时间
- 处理命令行参数
- 在 Vuser 脚本中使用 C 函数

***下列信息适用于除 GUI 和 Java 外的所有类型的 Vuser 脚本。***

# 关于增强 Vuser 脚本

可以在录制 Vuser 脚本期间或在录制后，通过添加下列函数类型来增强其功能：

- 常规 Vuser 函数
- 协议特定的 Vuser 函数
- 标准 ANSI C 函数

## 常规 Vuser 函数

常规 Vuser 函数大大增强了所有 Vuser 脚本的功能。例如，可使用常规 Vuser 函数来度量服务器性能、控制服务器负载、添加调试代码，或者检索关于参与场景的 Vuser 的运行时信息。

可在任何类型的 Vuser 脚本中使用常规 Vuser 函数。所有常规 Vuser 函数都具有 LR 前缀。 VuGen 将在录制过程中生成某些常规 Vuser 函数，并将它们插入到 Vuser 脚本中。要使用非自动生成的其他函数，请在 VuGen 的主窗口中选择“插入”>“新建步骤”，然后选择所需的函数。

本章仅讨论最常用的常规 Vuser 函数的用法。有关 Vuser 函数的其他信息，请参阅 《LoadRunner Function Reference》（“帮助”>“函数参考”）。

## 协议特定的 Vuser 函数

有多个函数库可以用于增强 Vuser 脚本。每个库都特定于一种 Vuser 类型。例如，可以在 Windows Sockets Vuser 脚本中使用 LRS 函数，在 TUXEDO Vuser 脚本中使用 LRT 函数。有关协议特定的 Vuser 函数的详细信息，请参阅《LoadRunner Function Reference》（“帮助”>“函数参考”）。

## 标准 ANSI C 函数

可以通过添加标准 ANSI C 函数来增强 Vuser 脚本。通过 ANSI C 函数可以向 Vuser 脚本中添加注释、控制流语句、条件语句等。可以向任意类型的 Vuser 脚本中添加标准 ANSI C 函数。详细信息，请参阅第 69 页的 “在 Vuser 脚本中使用 C 函数”。

# 将事务插入到 Vuser 脚本

可以定义事务以度量服务器的性能。每个事务度量服务器响应指定的 Vuser 请求所用的时间。这些请求可以是简单任务 （例如等待对单个查询的响应），也可以是复杂任务 （例如提交多个查询和生成报告）。

要度量事务，需要插入 Vuser 函数以标记任务的开始和结束。在脚本内，可以标记的事务不受数量限制，每个事务的名称都不同。

在场景执行期间， Controller 将度量执行每个事务所用的时间。场景运行后，可使用 LoadRunner 的图和报告来分析各个事务的服务器性能。

## 标记事务的开始

可将业务流程作为事务来标记其开始。

**要标记事务的开始，请执行下列操作：**

**1** 在录制 Vuser 脚本时，单击 “录制”工具栏上的 “开始事务”按钮。将打开 “开始事务”对话框。

**2** 在 “事务名”框中键入事务的名称。事务名必须以字母或数字开始，可以包含字母、数字或者下列字符：!、$、%、&、'、-、[、^、_、`、<、>、{、}、| 或 ~。

单击 “确定”接受该事务名称。 VuGen 将 **lr_start_transaction** 语句插入到 Vuser 脚本。例如，以下函数指示 *trans1* 事务的开始：

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

---

**注意：**完成录制会话后，可以选择 “插入” > “开始事务”标记事务的开始，选择 “插入” > “结束事务”标记事务的结束，从而将事务插入到脚本。

---

### 标记事务的结束

可通过结束事务语句来标记业务流程的结束。

**要标记事务的结束，请执行下列操作：**

**1** 在录制脚本时，单击 “录制”工具栏上的 “结束事务”按钮。将打开 “结束事务”对话框。

**2** 单击箭头获得打开事务的列表。选择要关闭的事务。

单击 “确定”接受该事务名称。 VuGen 将 **lr_end_transaction** 语句插入到 Vuser 脚本。例如，以下函数指示 *trans1* 事务的结束：

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

## 将集合点插入到 Vuser 脚本

要在系统上模拟较重的用户负载，需要同步各个 Vuser 以便在同一时刻执行任务。通过创建集合点，可以确保多个 Vuser 同时执行操作。当某个 Vuser 到达该集合点时， Controller 会将其保留，直到参与该集合的全部 Vuser 都到达。当满足集合条件时， Controller 将释放 Vuser。

可通过将集合点插入到 Vuser 脚本来指定会合位置。在 Vuser 执行脚本并遇到集合点时，脚本将暂停执行， Vuser 将等待 Controller 允许继续执行。 Vuser 被从集合释放后，将执行脚本中的下一个任务。

---

**注意：**只能向 *Action* 部分 （而不是 *init* 或 *end* 部分）添加集合。

---

**要插入集合点，请执行下列操作：**

**1** 在录制 Vuser 脚本时，单击 “录制” 工具栏上的 “集合” 按钮。将打开 “集合” 对话框。

**2** 在 “集合名” 框中键入集合点的名称。

单击 “确定” 接受该集合名称。 VuGen 将 **lr_rendezvous** 语句插入到 Vuser 脚本。例如，以下函数将定义名为 *rendezvous1* 的集合点：

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

---

**注意**：完成录制会话后，可以选择 VuGen 菜单上的 “插入” > “集合”，将集合点插入到脚本。

---

# 将注释插入到 Vuser 脚本

通过 VuGen 可以在 Vuser 活动之间插入注释。可插入注释以描述活动或提供关于特定操作的信息。例如，如果正在录制数据库操作，则可插入注释以标记第一个查询，如 “这是第一个查询”。

**要插入注释，请执行下列操作：**

**1** 在录制脚本时，单击 “录制” 工具栏上的 “注释” 按钮。将打开 “插入注释” 对话框。

**2** 将注释键入到文本框。

**3** 单击 “确定” 插入注释并关闭对话框。文本将置于脚本的当前点，并用注释标记将其括起来。以下脚本段显示了注释在 Vuser 脚本中的显示方式：

```
/*
* 这是第一个查询
*/
```

---

**注意：**完成录制会话后，可以选择 VuGen 菜单上的 “插入” > “注释”，将注释插入到脚本。

---

## 获取 Vuser 信息

可以向 Vuser 脚本中添加下列函数以检索 Vuser 信息：

| | |
|---|---|
| **lr_get_attrib_string** | 返回一个字符串，其中包含命令行参数值或运行时信息，例如 Vuser ID 或负载生成器的名称 |
| **lr_get_host_name** | 返回 Vuser 负载生成器的名称 |
| **lr_get_master_host_name** | 返回 LoadRunner Controller 负载生成器的名称 |
| **lr_whoami** | 返回 Vuser 的 Vuser ID、 Vuser 组和场景 ID |

在下例中， **lr_get_host_name** 函数检索正在运行 Vuser 的计算机的名称。

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

有关上述函数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

## 将消息发送到输出

运行场景时， Controller 的输出窗口将显示有关脚本执行的信息。可在 Vuser 脚本中包括相应语句，将错误和通知消息发送到 Controller。 Controller 将在输出窗口中显示这些消息。例如，可以插入一条消息，显示客户端应用程序的当前状态。还可将这些消息保存到文件。

---

**注意**：不要从事务内部发送消息，因为这可能使事务执行时间变长，并扭曲事务结果。

---

可以在 Vuser 脚本中使用下列消息函数：

| | |
|---|---|
| **lr_debug_message** | 将调试消息发送到输出窗口。 |
| **lr_error_message** | 将错误消息发送到输出窗口。 |
| **lr_get_debug_message** | 检索当前的消息类。 |

| | |
|---|---|
| **lr_log_message** | 将输出消息直接发送到 *output.txt* 文件，此文件位于 Vuser 脚本目录中。该函数有助于防止输出消息干扰 TCP/IP 通信。 |
| **lr_output_message** | 将消息发送到输出窗口。 |
| **lr_set_debug_message** | 为输出消息设置消息类。 |
| **lr_vuser_status_message** | 生成格式化输出并将其打印到 Controller Vuser 状态区域。 |
| **lr_message** | 将消息发送到 Vuser 日志和输出窗口。 |

---

**注意**：如果使用日志运行时设置修改脚本的调试级别，则 **lr_message**、**lr_output_message** 和 **lr_log_message** 函数的行为将不会更改，它们将继续发送消息。

---

## 日志消息

可使用 VuGen 来生成 **lr_log_message** 函数并将其插入到 Vuser 脚本。例如，如果正在录制数据库操作，则可插入一条消息以表示第一个查询，如 “这是第一个查询”。

**要插入 lr_log_message 函数，请执行下列操作：**

1 选择 “插入” > “日志消息”。将打开 “日志消息”对话框。

2 将消息键入到 “消息文本”框。

3 单击 “确定”插入消息并关闭对话框。 **lr_log_message** 函数将插入到脚本的当前点。

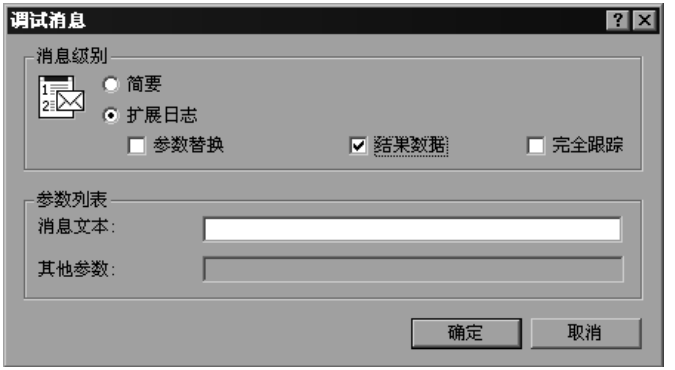
## 调试消息

可以使用 VuGen 的用户界面添加调试或错误消息。对于调试消息，可以指示文本消息的级别 （只有当指定的级别与消息类匹配时才发出该消息）。使用 **lr_set_debug_message** 可以设置消息类。

**要插入调试函数，请执行下列操作：**

1 选择 “插入” > “新建步骤”。将打开 “添加步骤”对话框。

2 选择 “调试消息”步骤，单击 “确定”。将打开 “调试消息”对话框。




3 选择消息级别：“简要”或 “扩展日志”。如果选择 “扩展日志”，请指示要记录的信息类型：参数替换、结果数据或完全跟踪。

4 将消息键入到 “消息文本”框。

5 单击 “确定”插入消息并关闭对话框。 **lr_debug_message** 函数将插入到脚本的当前点。

## 错误消息和输出消息

对于用树视图表示脚本的协议 （如 Web、 Winsock 和 Oracle NCA）而言，可以使用用户界面添加错误消息或输出消息。通常使用该函数来插入条件语句，如果检测到错误条件，则发出消息。

要插入错误消息或输出消息函数，请执行下列操作：

1 选择 “插入” > “新建步骤”。将打开 “添加步骤”对话框。

2 选择 “错误消息”或 “输出消息”步骤，单击 “确定”。将打开 “错误消息”或 “输出消息”对话框。

3 将消息键入到 “消息文本”框。

4 单击 “确定”插入消息并关闭对话框。 **lr_error_message** 或 **lr_output_message** 函数将插入到脚本的当前点。

有关消息函数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

## 处理 Vuser 脚本在执行期间出现的错误

可以指定 Vuser 如何处理在脚本执行期间出现的错误。默认情况下，当检测到错误时， Vuser 将停止执行脚本。可以指示 Vuser 在发生错误时使用下列方法之一继续下一个循环：

➤ 使用运行时设置。可以指定 “出现错误时仍继续”运行时设置。“出现错误时仍继续”运行时设置适用于整个 Vuser 脚本。可以使用 **lr_continue_on_error** 函数覆盖部分脚本的 “出现错误时仍继续”运行时设置。详细信息，请参阅第 122 页的 “错误处理”。

➤ 使用 **lr_continue_on_error** 函数。通过 **lr_continue_on_error** 函数可以控制 Vuser 脚本特定段的错误处理。要标记该段，请用 **lr_continue_on_error(1);** 和 **lr_continue_on_error(0);** 语句将其括起来。新的错误设置适用于括起来的 Vuser 脚本段。详细信息，请参阅下面的段落。

例如，如果启用了 “出现错误时仍继续”运行时设置，而且在回放以下脚本段时 Vuser 遇到了错误，则 Vuser 将继续执行该脚本。

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

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

要针对特定脚本段指示 Vuser 在出现错误时仍继续，请用相应的 **lr_continue_on_error** 语句将该段括起来：

```
    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);
```

# 同步 Vuser 脚本

可以添加同步函数将 Vuser 脚本的执行与应用程序的输出同步。同步仅适用于 RTE Vuser 脚本。

以下是可用同步函数的列表：

| | |
|---|---|
| **TE_wait_cursor** | 等待光标出现在终端窗口的指定位置 |
| **TE_wait_silent** | 等待客户端应用程序在指定秒数内处于静默状态 |
| **TE_wait_sync** | 等待系统从 X-SYSTEM 或输入禁止模式返回 |
| **TE_wait_text** | 等待字符串出现在指定位置 |
| **TE_wait_sync_transaction** | 记录系统在最近的 X SYSTEM 模式下保持的时间 |

有关在 RTE Vuser 脚本中使用同步函数的详细信息，请参阅第 55 章“同步 RTE Vuser 脚本”。

# 模拟用户思考时间

用户在执行两个连续操作期间等待的时间称为“思考时间”。 Vuser 使用 **lr_think_time** 函数模拟用户思考时间。录制 Vuser 脚本时， VuGen 将录制实际的思考时间并将相应的 **lr_think_time** 语句插入到 Vuser 脚本。可以编辑已录制的 **lr_think_time** 语句，而且可以向 Vuser 脚本中手动添加更多的 **lr_think_time** 语句。

**要手动添加思考时间语句，请执行下列操作：**

1. 将光标置于所需的位置。
2. 选择“插入”>“添加步骤”。将打开“添加步骤”对话框。
3. 选择“思考时间”，单击“确定”。将打开“思考时间”对话框。

4 以秒为单位指定所需的思考时间，然后单击 “确定”。

---

**注意**：录制 Java Vuser 脚本时， Vuser 脚本中将不生成 **lr_think_time** 语句。

---

执行 Vuser 脚本时，可以使用思考时间设置来影响 **lr_think_time** 语句的运行方式。要访问思考时间设置，请在 VuGen 主菜单上选择 “Vuser” > “运行时设置”，然后单击 “思考时间”选项卡。详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

## 处理命令行参数

运行脚本时，可以通过指定命令行参数在运行时将值传递到 Vuser 脚本。通过下列三个函数可以读取命令行参数，然后将值传递到 Vuser 脚本：

| | |
|---|---|
| **lr_get_attrib_double** | 检索双精度浮点类型参数 |
| **lr_get_attrib_long** | 检索长整型参数 |
| **lr_get_attrib_string** | 检索字符串 |

命令行应该包含下列两种格式之一，在这两种格式中，参数及其值在脚本名后面成对列出：

```
script_name   -argument argument_value   -argument argument_value
```

```
script_name   /argument argument_value   /argument argument_value
```

下例显示的命令行字符串用于在负载生成器 pc4 上重复 *script1* 五次：

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

有关命令行分析函数的详细信息，或者有关在命令行上包括参数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

# 加密文本

可对脚本内的文本进行加密，以保护密码和其他保密的文本字符串。可以自动从用户界面执行加密，也可通过编程方式手动执行加密。加密字符串时，该字符串将作为编码字符串出现在脚本中。为了使脚本使用加密的字符串，必须通过 **lr_decrypt** 来加密该字符串。

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

可在任何时候还原字符串以确定其原始值。

**要加密字符串，请执行下列操作：**

1. 对于具有树视图的协议，请在脚本视图中查看脚本。选择“视图”>“脚本视图”。
2. 选择要加密的文本。
3. 从右键单击菜单中选择“加密字符串 (*string*)”。

**要还原已加密的字符串，请执行下列操作：**

1. 对于具有树视图的协议，请在脚本视图中查看脚本。选择“视图”>“脚本视图”。
2. 选择要还原的字符串。
3. 从右键单击菜单中选择“还原加密字符串 (*string*)”。

有关 **lr_decrypt** 函数的详细信息，请参阅《LoadRunner Function Reference》（“帮助”>“函数参考”）。

# 在 Vuser 脚本中使用 C 函数

VuGen 在 C 中生成 Vuser 脚本。所有标准的 ANSI C 约定都适用于该脚本，包括控制流和语法。与在其他 C 程序中的操作类似，可以向该脚本中添加注释和条件语句。可以使用 ANSI C 约定声明和定义变量。

《LoadRunner Function Reference》（“帮助” > “函数参考”）包含 C 参考，以及常用 C 函数的语法和示例。

用于运行 Vuser 脚本的 C 解释器接受标准的 ANSI C 语言。它不支持对 ANSI C 的任何 Microsoft 扩展。

向 Vuser 脚本中添加任何 C 函数前，请注意下列限制：

- Vuser 脚本无法将其中某个函数的地址作为回调传递给库函数。
- Vuser 脚本不支持 *stdargs*、 *longjmp* 和 *alloca* 函数。
- Vuser 脚本不支持结构参数或返回类型。支持指向结构的指针。
- 在 Vuser 脚本中，字符串为只读。任何写入字符串的尝试都将生成访问冲突。

## 调用 libc 函数

在 Vuser 脚本中，可以调用 *libc* 函数。不过，由于用于运行 Vuser 脚本的解释器不支持任何对 ANSI C 的 Microsoft 扩展，因此无法使用 Microsoft 的包含文件。可以在必要时编写自己的原型，也可要求 Mercury Interactive 客户支持部门给您发送包含 *libc* 函数原型的兼容 ANSI 的包含文件。

## 链接模式

用于运行 Vuser 脚本的 C 解释器使用 “惰性”链接模式，因为不需要在运行开始就定义函数，只要在使用函数前对其进行定义即可。例如：

```
lr_load_dll("mydll.dll");
    myfun();  /* 在 mydll.dll 中定义 – 可以在
                    加载 myfun.dll 后立即直接调用。 */
```

# 7

# 定义参数

录制业务流程时， VuGen 生成一个包含录制期间用到的实际值的脚本。假设您要使用不同于录制内容的值执行该脚本的操作 （查询、提交等）。要实现这一点，可以用参数替换已录制的值。这被称为脚本参数化。

本章描述：

- 了解参数限制
- 创建参数
- 定义参数属性
- 了解参数类型
- 分配内部数据
- 指定参数格式
- 选择一个文件作为参数值的源
- 从现有数据库中导入数据
- 用户定义的函数
- 参数化选项

***下列信息适用于除 GUI 之外所有类型的 Vuser 脚本。***

# 关于定义参数

录制业务流程时， VuGen 生成一个由函数构成的 Vuser 脚本。函数中参数的值是录制期间使用的实际值。

例如，假设您在操作 Web 应用程序时录制了一个 Vuser 脚本。 VuGen 生成下列语句，在库的数据库中搜索标题 “UNIX”：

```
 web_submit_form("db2net.exe",
          ITEMDATA,
          "name=library.TITLE",
          "value=UNIX",
          ENDITEM,
          "name=library.AUTHOR",
          "value=",
          ENDITEM,
          "name=library.SUBJECT",
          "value=",
          ENDITEM,
          LAST);
;
```

使用多个 Vuser 和循环来回放该脚本时，如果您不想重复使用相同的值 UNIX，那么，可以用参数来替换该常量值：

```
web_submit_form("db2net.exe",
          ITEMDATA,
          "name=library.TITLE",
          "value={Book_Title}",
          ENDITEM,
          "name=library.AUTHOR",
          "value=",
          ENDITEM,
          "name=library.SUBJECT",
          "value=",
          ENDITEM,
          LAST);
```

然后，生成的 Vuser 使用指定的数据源中的值来替换参数。该数据源可以是一个文件或者内部生成的变量。有关数据源的详细信息，请参阅第 78 页的 “了解参数类型”。

对 Vuser 脚本进行参数化有两个好处：

- 减小脚本的大小。
- 提供了使用不同的值测试脚本的能力。例如，如果要在库的数据库中搜索几个标题，您只须写一次提交函数。将使用参数，而不是指示 Vuser 搜索一个特定的项目。回放期间， VuGen 用不同的值替换该参数。

参数化涉及下列两个任务：

- 用参数替换 Vuser 脚本中的常量值
- 为参数设置属性和数据源

## 了解参数限制

您只能对函数中的参数进行参数化。不是函数参数的文本字符串，不能参数化。此外，不是所有的函数参数都可以参数化。有关可以参数化的参数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

以 **lrd_stmt** 函数为例。该函数语法如下：

```
lrd_stmt (LRD_CURSOR FAR *mptCursor, char FAR *mpcText, long
mliTextLen, LRDOS_INT4 mjOpt1, LRDOS_INT4 mjOpt2, int
miDBErrorSeverity);
```

《LoadRunner Function Reference》指明只能参数化 *mpcText* 参数。

录制的 **lrd_stmt** 函数如下所示：

```
lrd_stmt(Csr4, "select name from sysobjects where name =\"Kim\" ", -1,
148, -99999, 0);
```

可以将录制的函数参数化为如下所示的形式：

```
lrd_stmt(Csr4, "select name from sysobjects where name =\"<name>\" ",
-1, 148, -99999, 0);
```

---

**注意：**您可以使用 **lr_eval_string** 函数来 “参数化” 不能用标准参数化方法来参数化的函数参数。此外，您可以使用 **lr_eval_string** 函数来 “参数化” Vuser 脚本中的任何字符串。有关 **lr_eval_string** 函数的详细信息，请参阅《LoadRunner Function Reference》。

---

## 创建参数

可以通过指定名称和类型来创建参数。在 Vuser 脚本中不限制可以创建的参数的数量。

**要创建参数，请执行下列操作：**

1 **在脚本视图中：**选择一个字符串并从右键单击菜单中选择 “替换为新参数”。

2 **在树视图中：**选择要参数化的步骤，并从右键单击菜单中选择 “属性”。将打开相应的 “属性” 对话框。

单击要参数化的参数旁边的 “ABC” 图标。

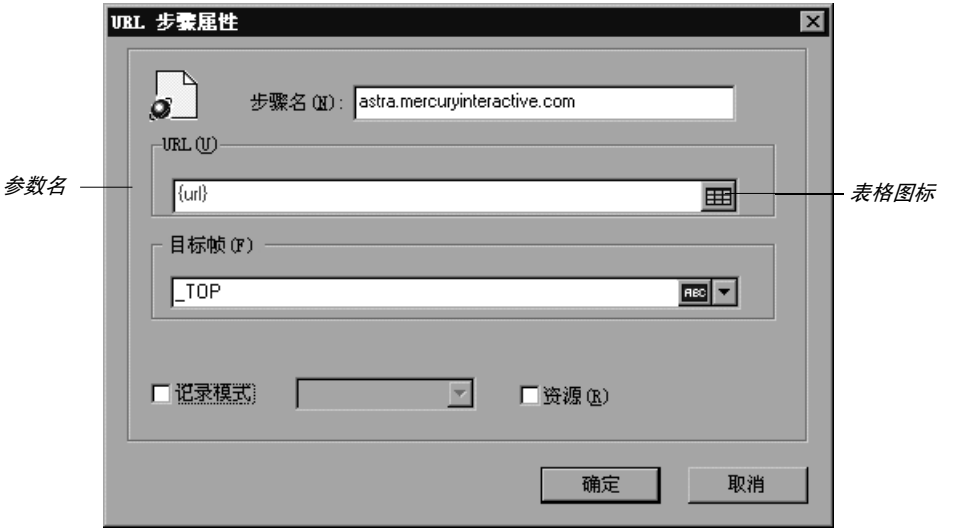
将打开 “选择或创建参数” 对话框。

3 在 “参数名” 框中键入参数的名称，或从列表中选择一个现有的参数名。

4 从 “参数类型” 列表中选择一个参数类型。可用的类型有日期 / 时间、文件、组名、循环编号、负载生成器名、随机编号、唯一编号、用户定义的函数或 Vuser ID。有关参数类型的信息，请参阅第 78 页的 “了解参数类型”。

5 单击 “确定” 关闭 “选择或创建参数” 对话框。 VuGen 用参数名替换脚本中选定的字符串，并用括号括起来。

在树视图中， VuGen 用表格图标替换 “ABC” 图标。在下例中，原始 URL 值 “http://www.merc-int.com/” 被参数 *{url}* 替换。




注意，参数化 CORBA 或 General-Java Vuser 脚本时，必须参数化整个字符串，而不是字符串的一部分。

---

**注意：**默认的参数大括号是花括号还是尖括号，取决于协议类型。可以从 “常规选项” 对话框中的 “参数化” 选项卡中更改参数大括号 （选择 “工具” > “常规选项”）。详细信息，请参阅第 98 页的 “参数化选项”。

---

### 替换多次出现的位置

您可以用新参数替换选定的或所有出现该字符串的位置。

**要用同一参数替换多个出现该字符串的位置，请执行下列操作：**

1 选择该参数并从右键单击菜单中选择 “替换其他出现位置”。将打开 “搜索和替换”对话框。

“查找内容”框显示要替换的值。“替换为”框显示括号中的参数名。

2 选中适当的复选框来匹配整词或部分。要用正则表达式（.、!、？等）搜索，请选中 “正则表达式”复选框。详细信息，请参阅第 455 页的 “使用常规表达式”。

3 单击 “替换”或 “全部替换”。

---

**注意：**请小心使用 “全部替换”，尤其在替换数字字符串时。 VuGen 将更改所有出现的该字符串。

---

4 要用先前定义的参数替换字符串，请进入脚本视图。在该字符串上右键单击并选择 “使用现有参数”。将打开 “使用现有参数”子菜单。

从 “使用现有参数”子菜单中选择一个参数，或选择 “从参数列表中选择”打开 “参数列表”对话框。

---

**注意：**要用先前定义的参数替换一个字符串，并同时查看或修改该参数的属性时，使用 “参数列表”会很方便。有关使用参数列表的详细信息，请参阅第 96 页的 “使用参数列表”。

---

### 还原初始字符串

通过 VuGen，可以还原最初录制的字符串从而撤消参数化。

**要将参数还原为初始值，请执行下列操作：**

1 在脚本视图中，右键单击该参数并选择 “还原初始值”。

2 在树视图中，右键单击该步骤并单击表格图标，然后从弹出菜单中选择 “撤消参数”。这样初始值将被还原。

## 定义参数属性

创建参数后，要定义它的属性。参数的属性定义脚本执行期间该参数的数据源。

**要定义参数的属性，请执行下列操作：**

1 在脚本视图中，选择该参数。从右键单击菜单中选择 “参数属性”。

2 在树视图中，右键单击包含要定义其属性的参数的步骤，然后选择 “属性”。将打开选定步骤的 “属性”对话框。

3 单击要定义其属性的参数旁边的表格图标，然后从弹出菜单中选择 “参数属性”。

将打开 “参数属性”对话框，并显示当前参数类型的属性。在下例中，将显示一个文件类型参数的属性。

4 输入该参数的属性。详细信息，请参阅第 78 页的 “了解参数类型”。

5 单击 “关闭”关闭 “参数属性”对话框。

## 了解参数类型

定义参数的属性时，需要指定该参数的数据源。您可以指定下列数据源类型中的任何一个：

**分配内部数据** Vuser 内部生成的数据。这包括日期 / 时间、组名、循环编号、负载生成器名、随机编号、唯一编号和 Vuser ID

**数据文件** 文件 （现有文件或者用 VuGen 或 MS Query 创建的文件）中包含的数据

**用户定义的函数** 使用外部 DLL 函数生成的数据。有关用户定义的函数的详细信息，请参阅第 95 页的 “用户定义的函数”

## 分配内部数据

内部数据是 Vuser 运行时自动生成的。下列部分描述内部数据的各种类型。

### 日期 / 时间

“日期 / 时间”用当前的日期和 / 或时间替换参数。要指定日期 / 时间的格式，可以从菜单列表中选择一个格式，或者指定您自己的格式。该格式应与脚本中录制的日期 / 时间格式相对应。要新建格式，请在 “日期 / 时间格式”框中输入该格式，然后单击 “添加格式”。要删除格式，请选择该格式并单击 “删除格式”。要还原格式，请单击 “重设格式”。

下表描述日期 / 时间符号：

| 符号 | 描述 |
|---|---|
| c | 用数字表示的完整日期和时间 |
| #c | 完整的日期 （以字符串表示）和时间 |
| H | 小时 （24 小时制） |

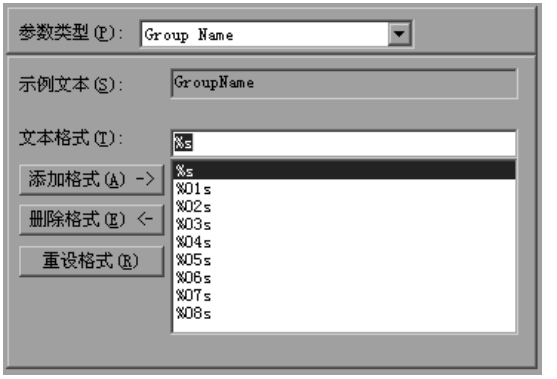
| 符号 | 描述 |
|---|---|
| I | 小时 （12 小时制） |
| M | 分钟 |
| S | 秒 |
| p | AM 或 PM |
| d | 日期 |
| m | 数字形式的月份 (01-12) |
| b | 字符串形式的月份 – 短格式 （例如， Dec） |
| B | 字符串形式的月份 – 长格式 （例如， December） |
| y | 短格式的年份 （例如， 03） |
| Y | 长格式的年份 （例如， 2003） |

**要设置 “日期 / 时间”类型参数的属性，请执行下列操作：**

1. 从现有的日期 / 时间格式中选择或者新建一种格式。在 “示例值”框中，您可以查看 VuGen 如何显示该值的示例。
2. 选择一种更新方法，指示 Vuser 何时更新参数值，选项包括 “每次出现”、“每次循环”和 “一次”。详细信息，请参阅第 86 页的 “选择更新方法”。
3. 单击 “关闭”接受设置并关闭 “参数属性”对话框。

### 组名

“组名”用 Vuser 组的名称替换参数。创建场景时，要指定 Vuser 组的名称。运行 VuGen 的脚本时，组名始终为 “无”。

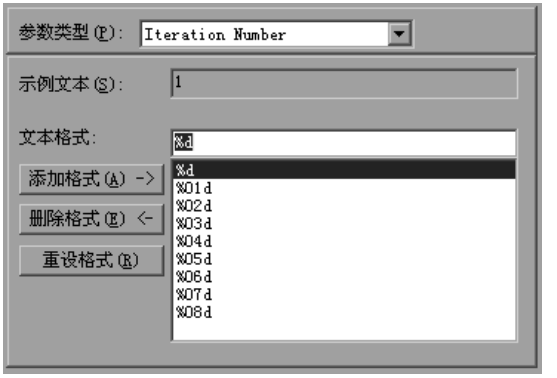



**要设置 “组名”参数类型的属性，请执行下列操作：**

1. 从可用格式中选择其中一种或者新建一种格式。选择一种格式来指定参数字符串的长度。详细信息，请参阅第 85 页的 “指定参数格式”。
2. 单击 “关闭”接受设置并关闭 “参数属性”对话框。

### 循环编号

“循环编号”用当前的循环编号替换参数。



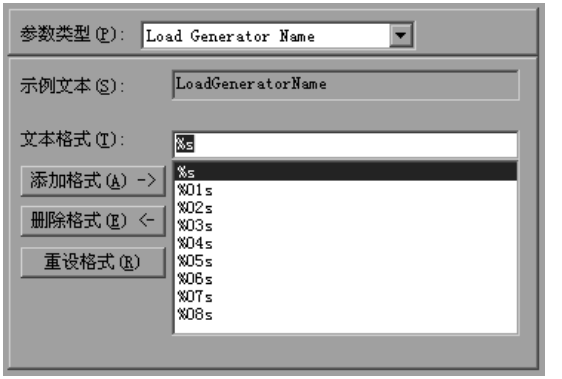
要设置 “循环编号”参数类型的属性，请执行下列操作：

1 从可用格式中选择其中一种或者新建一种格式。选择一种格式来指定参数字符串的长度。详细信息，请参阅第 85 页的 “指定参数格式”。

2 单击 “关闭”保存设置并关闭该对话框。

## 负载生成器名

“负载生成器名”用 Vuser 脚本的负载生成器名替换参数。负载生成器是运行 Vuser 的计算机。




要设置 “负载生成器名”参数类型的属性，请执行下列操作：

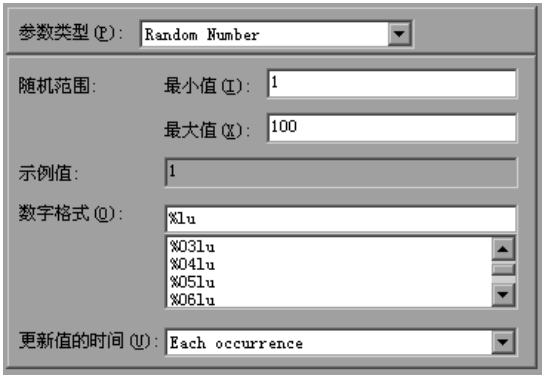
1 从可用格式中选择其中一种或者新建一种格式。选择一种格式来指定参数字符串的长度。详细信息，请参阅第 85 页的 “指定参数格式”。

2 单击 “关闭”保存设置并关闭 “参数属性”对话框。

## 随机编号

“随机编号”用一个随机编号替换参数。通过指定最小和最大值，设置随机编号的范围。

您可以使用 “随机编号”参数类型在一个可能的值域内对系统的行为进行抽样。例如，要对 50 名员工 （员工的 ID 编号范围从 1 到 1000）进行查询，请创建 50 个 Vuser 并设置其最小值为 1，最大值为 1000。每个 Vuser 都接收到一个 1 到 1000 范围内的随机编号。




**要设置 “随机编号”类型参数的属性，请执行下列操作：**

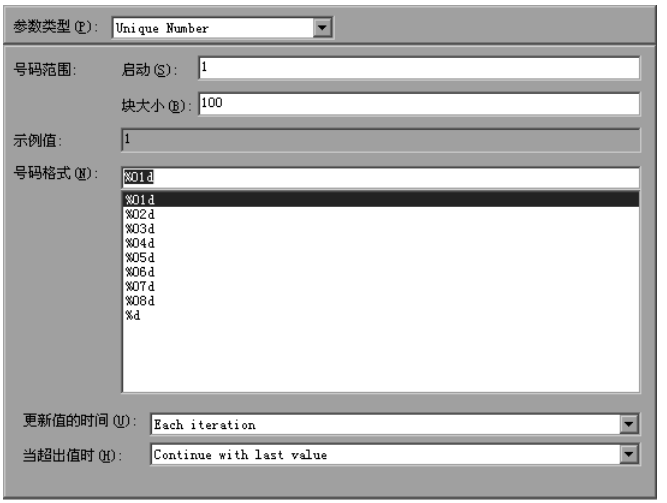
1. 输入一个范围来定义可能的参数值集。指定随机编号范围的最小和最大值。
2. 选择 “号码格式”，指明唯一编号的长度。指定 %01lu 表示一位数， %01lu 表示两位数等。在 “示例值”框中，可以查看 VuGen 如何显示该值的示例。
3. 选择一种更新方法，指示 Vuser 何时更新参数值，选项包括 “每次出现”、“每次循环”和 “一次”。详细信息，请参阅第 86 页的 “选择更新方法”。
4. 单击 “关闭”接受设置并关闭 “参数属性”对话框。

## 唯一编号

“唯一编号”用一个唯一编号替换参数。您可以指定起始编号和块大小。

使用 “唯一编号”参数类型时，指定起始编号和块大小。块大小指明分配给每个 Vuser 的编号块的大小。每个 Vuser 都从其范围的下限开始，在每次循环时递增该参数值。例如，如果设置起始编号为 1 并且块大小为 500，则在其第一次循环中，第一个 Vuser 使用值 1，下一个 Vuser 使用值 501。

唯一编号字符串中的数位与块大小共同确定循环和 Vuser 的数量。例如，如果限制为五位数并使用大小为 500 的块，则只有 100,000 个数 (0-99,999) 是可用的。因此，只能运行 200 个 Vuser，并且每个 Vuser 运行 500 次循环。




您还可以指示当块中不再有唯一编号时所执行的操作：“中断 Vuser”、“以循环方式继续”或者 “使用最后的值继续”（默认值）

您可以使用 “唯一编号”参数类型检查所有可能的参数值的系统行为。例如，要对所有的员工 （他们的 ID 编号范围是 100 到 199）进行查询，请创建 100 个 Vuser 并且设置起始编号为 100，块大小为 100。每个 Vuser 都接收到一个唯一编号，该唯一编号从 100 开始到 199 结束。

---

**注意**：LoadRunner 仅创建一个 “唯一编号”类型参数的实例。如果定义多个参数并且为它们指定唯一参数类型，那么这些值将不会重复。例如，如果使用大小为 100 的块为 5 次循环定义两个参数，则第一组中的 Vuser 使用 1、 101、 201、 301 和 401。第二组中的 Vuser 使用 501、 601、 701、 801 和 901。

---

要设置 “唯一编号”类型参数的属性，请执行下列操作：

1 输入起始编号和所需的块大小。例如，如果需要从 1 开始的 500 个编号，请在 “开始”框中指定 1，并指定块大小为 500。

2 选择 “号码格式”，指明唯一编号的长度。指定 %01d 表示一位数， %01d 表示两位数等。在 “示例值”框中，可以查看 VuGen 如何显示该值的示例。

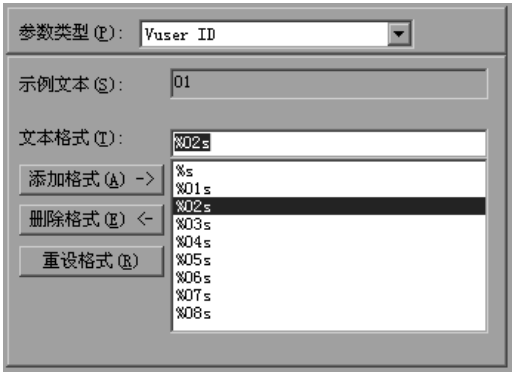
3 选择一种更新方法，指示 Vuser 何时更新参数值，选项包括 “每次出现”、“每次循环”和 “一次”。详细信息，请参阅第 86 页的 “选择更新方法”。

4 在 “当溢出框时”中，指示当不再有唯一编号时所执行的操作：“中断 Vuser”、“以循环方式继续”或者 “使用最后的值继续”

5 单击 “关闭”接受设置并关闭 “参数属性”对话框。

## Vuser ID

“Vuser ID”用分配给该 Vuser 的 ID 来替换参数，此 ID 是场景运行期间 Controller 分配给 Vuser 的。注意，该 ID 并不是在 Vuser 窗口中显示的 ID，而是在运行时生成的唯一的 ID 编号。运行 VuGen 的脚本时， Vuser ID 始终为 “-1”。




要设置 “Vuser ID”参数类型的属性，请执行下列操作：

1 从可用格式中选择其中一种或者新建一种格式。选择一种格式来指定参数字符串的长度和结构。详细信息，请参阅第 85 页的 “指定参数格式”。

2 单击 “关闭”接受设置并关闭 “参数属性”对话框。

### 数据文件

数据文件保存脚本执行期间 Vuser 访问的数据。数据可以被存储在本地或全局文件中。您可以指定一个现有的 ASC II 文件，使用 VuGen 新建一个文件，或者导入数据库。如果拥有大量已知参数值，数据文件是非常有用的。

数据文件中的数据以表格的形式存储。一个文件可以包含大量参数值。每一列都保存一个参数的数据。列的分隔由分隔符标记，例如逗号。

在下例中，数据文件包含 ID 编号和名：

```
id,first_name
120,John
121,Bill
122,Tom
```

有关为数据文件设置参数属性的详细信息，请参阅第 87 页的 “选择一个文件作为参数值的源”。有关为数据库设置参数属性的详细信息，请参阅第 92 页的 “从现有数据库中导入数据”。

## 指定参数格式

对于大多数的数据类型，可以通过选择一种现有的或者指定一种新格式来指定参数的格式。应设法让参数格式与录制的值相匹配。如果参数的格式与初始录制的值的格式不一致，那么脚本可能无法正常运行。

该格式指定所得到的参数字符串的长度和结构。所得到的参数字符串是实际的参数值和伴随该参数的所有文本。例如，如果您指定格式 “%05s”，则 5 的 Vuser ID 就会显示为 “00005”，即在个位数前添加四个零。要在编号前添加空格，请指定不带 “0” 的空格数。例如， %4s 在 Vuser ID 前添加空格，以便所得到的参数字符串长度为 4 个字符。

您可以在实际参数值前后指定一个文本字符串。例如，如果您指定一种格式 “Vuser No:%03s”，则 1 的 Vuser ID 就会显示为 “Vuser No: 001”。

**添加格式：** 在可编辑的框内输入该格式的符号，并单击 “添加格式”。向列表中添加格式时， VuGen 使用 Vuser 来保存它，这样该格式在将来也可用。

**删除格式：** 选择一种现有格式并单击 “删除格式”。

**还原初始格式：** 单击 “重设格式”。

### 选择更新方法

使用 “日期 / 时间”、“随机”、“唯一” 和 “用户定义的函数” 参数类型时，通过 VuGen 可以指定参数的更新方法。要设置更新方法，请从 “更新值的时间” 列表中选择一种方法。可用的参数更新方法有：

- 每次出现
- 每次循环
- 一次

#### 每次出现

“每次出现” 方法指示 Vuser 在每次参数出现时使用新值。当使用同一个参数的几个语句不相关时，该方法非常有用。例如，对于随机数据，在该参数每次出现时都使用新值可能是非常有用的。

#### 每次循环

“每次循环” 方法指示 Vuser 为每次脚本循环使用新值。如果一个参数在脚本中出现了若干次，则 Vuser 为整个循环中所有出现的该参数使用同一个值。当使用同一个参数的几个语句相关时，该方法非常有用。

---

**注意**：如果用来创建操作块的参数使用它自己的循环计数，这时如果指示 VuGen 在每次循环时更新它们的值，则它指的是全局循环而不是块循环。有关操作块的详细信息，请参阅第 110 页的 “创建操作块”。

---

#### 一次

“一次” 方法指示 Vuser 在场景运行期间仅对参数值更新一次。 Vuser 为所有出现的该参数和所有循环使用同一个参数值。当使用日期和时间时，该类型可能会非常有用。

# 选择一个文件作为参数值的源

一个常见的使用参数的方法是指示 Vuser 从外部文件中取值。请执行下列步骤：

➤ 选择或者创建数据文件

➤ 设置 “文件”类型参数的属性

## 选择或者创建数据文件

如果参数类型是文件，打开 “参数属性”对话框时，将会显示文件属性设置。

仅显示数据的前 100 行。要查看所有的数据，请单击 “编辑”并在记事本中查看该数据。

**要选择参数的源文件，请执行下列操作：**

**1** 在 “文件路径”框中键入数据文件的名称，或者单击 “浏览”指定现有数据文件的文件位置。默认情况下，所有新数据文件均被命名为 *parameter_name.dat* 并且存储在脚本的目录中。注意，现有数据文件的扩展名必须是 *.dat*。

您也可以指定一个全局目录。注意，提供全局目录仅为了向后兼容 LoadRunner 早期版本。详细信息，请参阅第 99 页的 “全局目录”。

2 单击 “编辑”。将打开记事本，第一行是参数名，第二行是它的初始值。用表格的形式将其他列名和值输入该文件。使用分隔符 （例如逗号或者制表符）来指示列分隔。对表格中的每个新行都要另起一行开始 （例如，对于每个新的数据行）。

---

**注意**：要在不启动记事本的情况下向数据文件中添加一列，请在 “参数属性”对话框中单击 “添加列”。将打开 “添加新列”对话框。在 “列名”框中键入新列的名称，然后单击 “确定”。 VuGen 向表格中添加了新的一列，并且第一行是参数的初始值。

---

### 设置 “文件”类型参数的属性

在选择了数据源之后，您可以设置分配属性。这些属性指示 VuGen 如何使用数据。例如，它们指示了哪些列，隔多久使用新值，以及当没有唯一的值时做什么。

**要设置 “文件”类型参数的属性，请执行下列操作：**

1 指定表格中包含参数值的列。在 “选择列”部分中，指定列的编号或者名称。

要指定列的编号，请选择 “按编号”和列的编号。列的编号就是包含数据的列的索引。例如，如果参数数据在表格的第一列，则选择 1。

要指定列名，请选择 “按名称”并从列表中选择列名。该列是每列的第一行 （行 0）。如果列编号可能会更改或者没有标头，则使用列名来选择一列。

2 在 “文件格式”部分的 “列分隔符”框中，输入列分隔符 （用来分隔表格中的列的字符）。您可以指定逗号、制表符或者空格。

3 在 “文件格式”部分的 “第一个数据行”框中，选择要在 Vuser 脚本执行期间使用的数据的第一行。标头是行 0。要从标头后的第一行开始，请指定 1。如果没有标头，则指定 0。

4 从 “选择下一行” 列表中选择一个选项，指示 Vuser 在 Vuser 脚本执行期间如何选择表格数据。选项包括：“顺序”、“随机” 和 “唯一”。详细信息，请参阅第 89 页的 “为 “文件” 类型参数选择分配方法”。

5 从 “更新值的时间” 列表中选择更新选项。选项包括 “每次循环”、“每次出现” 和 “一次”。详细信息，请参阅第 86 页的 “选择更新方法”。

6 如果您选择 “唯一” 作为 “选择下一行” 选项：
**当超出值时**：指定当不再有唯一的数据时做什么：“中断 Vuser”、“以循环方式继续” 或者 “使用最后的值继续”。
**在 Controller 中分配 Vuser 值**：指示是否要手动为 Vuser 分配数据块：“自动分配块大小” 或者 “为每个 Vuser 分配 x 值”。指定要分配的值的个数。

---

**注意**：也可以从 “参数列表” 对话框中为 “文件” 参数类型设置属性。如果您正在新建一个 “文件” 参数， VuGen 就会提示您创建数据文件。单击 “创建”。“编辑” 按钮将替换 “创建” 按钮。单击 “编辑” 打开记事本并填入数据，如上所述。

---

## 为 “文件” 类型参数选择分配方法

当使用文件中的值时，通过 VuGen 可以指定向参数分配值的方法。可用的方法包括：

- 顺序
- 随机
- 唯一

### 顺序

“顺序” 方法顺序地向 Vuser 分配参数值。当正在运行的 Vuser 访问数据表格时，它将会提取下一个可用的数据行。

如果在 “更新值的时间” 列表框中指定了 “每次循环”，则 Vuser 将为每一次循环从数据表格中提取下一个值。

如果在 “更新值的时间” 列表框中指定了 “每次出现”，则 Vuser 将为每一次参数的出现从数据表格中提取下一个值，即使它在同一次循环中。

如果在 “更新值的时间”列表框中指定了 “一次”，则第一次循环中分配的值就会在每个 Vuser 接下来所有的循环中使用。

| 名 |
| --- |
| Kim |
| David |
| Michael |
| Jane |
| Ron |
| Alice |
| Ken |
| Julie |

例如，假定您的表格拥有左侧表格中显示的值。

如果选择了 “每次循环”，则所有 Vuser 就会在第一次循环中使用 Kim，第二次循环中使用 David，第三次循环中使用 Michael，等等。

如果选择了 “每次出现”，则所有 Vuser 就会在第一次出现时使用 Kim，第二次出现时使用 David，第三次出现时使用 Michael，等等。

如果选择了 “一次”，则第一个 Vuser 为所有的循环提取 Kim，第二个 Vuser 为所有的循环提取 David，等等。

如果在数据表格中没有足够的值，则 VuGen 返回到表格中的第一个值，循环继续直到测试结束。

### 随机

当测试开始运行时，“随机”方法为每个 Vuser 分配一个数据表格中的随机值。

如果在 “更新值的时间”列表框中指定了 “每次循环”，则 Vuser 将会为每一次循环从数据表格中提取一个新的随机值。

如果在 “更新值的时间”列表框中指定了 “每次出现”，则 Vuser 将会为每一次参数的出现从数据表格中提取一个新的随机值，即使它在同一次循环中。

如果在 “更新值的时间”列表框中指定了 “一次”，则第一次循环中分配的随机值就会在该 Vuser 的所有循环中使用。

当从 LoadRunner Controller 运行一个场景时，您可以指定随机顺序的种子数。每个种子值代表用于测试执行的一个随机值顺序。每当使用该种子值时，都会将相同顺序的值分配给场景中的 Vuser。如果在测试执行中发现问题，并且要使用相同的随机值顺序重复该测试，请启用该选项。

详细信息，请参阅 《LoadRunner Controller 用户指南 (Windows)》。

### 唯一

“唯一”方法为每一个 Vuser 的参数分配一个唯一的顺序值。

如果在 “更新值的时间”列表框中指定了 “每次循环”，则 Vuser 将为每一次循环从数据表格中提取下一个唯一值。

如果在 “更新值的时间”列表框中指定了 “一次”，则第一次循环中分配的唯一值就会在每个 Vuser 的所有接下来的循环中使用。

如果在 “更新值的时间”列表框中指定了 “每次出现”，则 Vuser 将会为每一次参数的出现从数据表格中提取一个新的唯一值，即使它在同一次循环中。

| 名 |
| --- |
| Kim |
| David |
| Michael |
| Jane |
| Ron |
| Alice |
| Ken |
| Julie |
| Fred |

例如，假定您的表格拥有左侧表格中显示的值。

如果指定了 “每次循环”，对于一个有 3 次循环的测试运行，第一个 Vuser 在第一次循环中提取 Kim，第二次循环中提取 David，第三次循环中提取 Michael。第二个 Vuser 提取 Jane、 Ron 和 Alice。第三个 Vuser 提取 Ken、 Julie 和 Fred。

如果指定了 “一次”，则第一个 Vuser 为所有的循环提取 Kim，第二个 Vuser 为所有的循环提取 David，等等。

确保表格中的数据对所有的 Vuser 和它们的循环来说是充足的。如果您拥有 20 个 Vuser，并且要运行 5 次循环，则您的表格中必须至少包含有 100 个唯一值。

如果数据表格中没有足够的值，您可以指示 VuGen 如何继续运行：“中断 Vuser”、“以循环方式继续”或者 “使用最后的值继续”。如果选择使用最后的值继续运行，则 Vuser 将会为所有接下来的循环使用表格中最后一行的数据。

假如您想要给每一个 Vuser 都分配值，并且不想让那些值在 Vuser 之间共享，要实现该目标，可以在 “在 Controller 中分配 Vuser 值”部分中指示 VuGen 为每个 Vuser 的值分配一个特定的编号。默认情况下， VuGen 自动为 Vuser 的值分配必需的编号。

要跟踪这次出现，请启用日志运行时设置中的 “扩展日志” > “参数替换”选项。当没有足够的数据时， VuGen 将会向 Vuser 日志中写入警告消息 “在表 < *表名* > 中，此参数没有其他唯一值”。

# 从现有数据库中导入数据

通过 LoadRunner，可以从数据库中导入数据以用于参数化。您可以用下列两种方法中的一种导入数据：

- 新建查询
- 指定 SQL 语句

VuGen 提供一个指导您完成从数据库中导入数据的过程的向导。在该向导中，您可以指定如何导入数据 （通过 MS Query 新建查询或者指定 SQL 语句）。导入数据之后，它被另存为一个扩展名为 *.dat* 的文件，并且存储为常规参数文件。

要开始导入数据库的过程，请在 “参数列表”对话框中单击 “数据向导”（“Vuser” > “参数列表”）。将打开数据库查询向导。

### 新建查询

使用 Microsoft 的数据库查询向导新建查询。这需要在您的系统上安装 MS Query。

**要新建查询，请执行下列操作：**

1 选择 “新建查询”。如果您需要有关 Microsoft Query 的说明，请选择 “显示如何使用 Microsoft Query”。

2 单击 “完成”。如果您的计算机上没有安装 Microsoft Query， LoadRunner 会发出一条指示它不可用的消息。先从 Microsoft Office 中安装 MS Query，然后再继续。

3 遵循向导中的说明，导入所需的表格和列。

4 当完成导入数据时，选择 “退出并返回到虚拟用户生成器”并单击 “完成”。数据库记录将会作为数据文件显示在 “参数属性”框中。

要编辑并查看 MS Query 中的数据，请选择“查看数据或者在 Microsoft Query 中编辑”。

5 设置数据分配属性。请参阅第 88 页的 “设置 “文件”类型参数的属性”。

### 指定 SQL 语句

**要指定数据库连接和 SQL 语句，请执行下列操作：**

1. 选择 “指定 SQL 语句”。单击 “下一步”。
2. 单击 “创建”指定一个新的连接字符串。将打开 “选择数据源”窗口。
3. 选择一个数据源，或者单击 “新建”新建一个数据源。该向导会指导您完成创建 ODBC 数据源的过程。当完成时，连接字符串将显示在 “连接字符串”框中。
4. 在 “SQL”框中，键入或者粘贴一个 SQL 语句。

5. 单击 “完成”处理 SQL 语句并导入数据。数据库记录将会作为数据文件显示在 “参数属性”框中。
6. 设置数据分配属性。请参阅第 88 页的 “设置 “文件”类型参数的属性”。

# 用户定义的函数

用户定义的函数用从外部 DLL 函数中返回的值替换参数。

在将用户定义的函数指派为参数之前，需要创建一个包含该函数的外部库 (DLL)。该函数应该有以下格式：

```
__declspec(dllexport) char *< 函数名 >(char *, char *)
```

发送到该函数的参数都是 NULL。

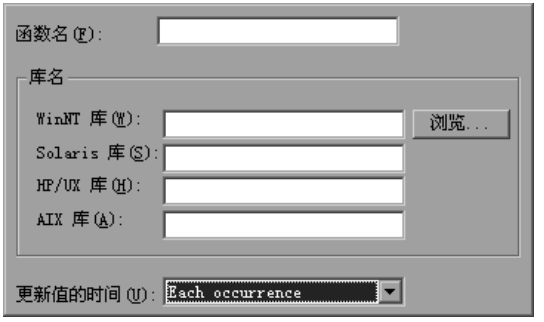
创建该库时，建议使用默认的动态库路径。这样，您就不必为库输入一个完整的路径名，仅输入库名就可以了。虚拟用户生成器 bin 目录就在默认的动态库路径上。您可以向该目录中添加库。

以下是用户定义的函数的示例：

```
__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;}
```

当选择 “用户定义的函数”类型时，将打开 “用户定义的函数属性”选项卡。



要设置用户定义的函数的属性，请执行下列操作：

1 在 “函数名”框中指定函数名。使用在 DLL 文件中显示的函数名。

2 在 “库名”部分中相关的 “库”框内指定一个库。如有必要，使用 “浏览”命令查找该文件。

3 选择一种更新值的方法。有关用户定义的函数更新方法的详细信息，请参阅第 86 页的 “选择更新方法”。

## 使用参数列表

使用参数列表可以检查所有的参数、新建参数、删除参数或者更改现有参数的属性。

要使用参数列表，请执行下列操作：

1 单击 “参数列表”按钮或者选择 “Vuser” > “参数列表”。选择一个参数以显示其属性。在下例中，将显示一个 “日期 / 时间”类型参数的属性。

2 要新建参数，请单击 “新建”。新参数以一个临时名显示在参数树中。

为新参数键入一个名称，然后按 **Enter** 键。

---

**注意**：不要将参数命名为 *unique*，因为该名已被 VuGen 使用。

---

设置参数的类型和属性，然后单击 “确定”关闭 “参数列表”对话框。

---

**注意**：VuGen 新建参数，但不会自动替换任何在脚本中选定的字符串。

---

3 要删除某个现有参数，请从参数树中选中该参数，单击 “删除”，然后确认您的操作。

4 要修改某个现有参数，请从参数树中选中该参数，然后编辑该参数的类型和属性。

有关设置参数属性的详细信息，请参阅第 78 页的 “了解参数类型”。

# 参数化选项

在 VuGen 中，您可以设置下列参数化选项：

- 参数大括号
- 全局目录

## 参数大括号

向 Vuser 脚本中插入参数时， VuGen 会在参数名的两边放置参数大括号。 Web 或者 WAP 脚本的默认大括号是花括号，例如，

```
web_submit_form("db2net.exe",
        ITEMDATA,
        "name=library.TITLE",
        "value={Book_Title}",
        ENDITEM,
        "name=library.AUTHOR",
        "value=",
        ENDITEM,
        "name=library.SUBJECT",
        "value=",
        ENDITEM,
        LAST);
```

通过指定一个或多个字符的字符串，可以更改参数大括号的样式。除了空格外，所有字符都是有效的。

---

**注意**：默认的参数大括号是尖括号还是花括号，取决于 Vuser 类型。

---

要更改参数大括号的样式，请执行下列操作：

1 在 VuGen 中选择 “工具” > “常规选项”。将打开 “常规选项”对话框。

2 选择 “参数化”选项卡并输入所需的大括号。

3 单击 “确定”接受设置并关闭对话框。

## 全局目录

提供该选项仅为了向后兼容 LoadRunner 早期版本。在早期版本 （4.51 和更低版本）中，新建数据表格时，您应指定它是本地表格还是全局表格。本地表格保存在当前的 Vuser 脚本目录中，并且仅对运行该脚本的 Vuser 可用。全局表格对所有 Vuser 脚本都是可用的。全局目录可以在本地或者网络驱动器上。确保全局目录对所有运行该脚本的计算机都是可用的。通过 “常规选项”对话框，可以随时更改全局表格的位置。

在 LoadRunner 的新版本中，可以在 “参数属性”或者 “参数列表”对话框中指定数据表格的位置。 LoadRunner 可以检索您指定的任何位置 （默认的脚本目录或者网络上的其他目录）的数据。详细信息，请参阅第 85 页的 “数据文件”。

要设置全局目录，请执行下列操作：

1 在 VuGen 中选择 “工具” > “常规选项”。将打开 “常规选项”对话框。

2 选择 “参数化”选项卡。

3 选择 “定义全局数据表目录”复选框，并指定包含您的全局数据表的目录。

4 单击 “确定”接受设置并关闭对话框。

# 8

# 关联语句

可以通过关联语句优化 Vuser 脚本。使用 VuGen 的关联查询功能可以通过将一个语句的结果用作另一个语句的输入来链接语句。

本章描述：

- 使用 C Vuser 的关联函数
- 使用 Java Vuser 的关联函数
- 使用 Wdiff 比较 Vuser 脚本
- 修改保存的参数

***下列信息适用于除 GUI 以外所有类型的 Vuser 脚本。***

## 关于关联语句

关联语句的主要原因是：

- 简化或优化代码

  例如，如果依次执行一系列的相关查询，您的代码可能会变得很长。为了减小代码的大小，可以嵌套查询，但这样会失去准确性，而且代码会变得复杂而难以理解。通过关联语句，不需要嵌套就可以链接查询。

- 用于动态数据

  许多应用程序和网站按当前日期和时间标识会话。如果尝试回放脚本，该操作将会失败，因为当前时间与录制时间不同。通过关联数据可以保存动态数据，并且在整个场景运行过程中使用这些数据。

➤ 容纳唯一数据记录

某些应用程序 （例如数据库）要求使用唯一的值。录制期间唯一的值在脚本执行时将不再唯一。例如，假设您录制了打开一个新的银行帐户的过程。每个新帐户都分配了用户未知的唯一编号。录制期间，此帐号将插入具有唯一关键字约束的表中。如果尝试按所录制的那样运行脚本，该操作将尝试用录制的编号而不是新的唯一编号创建帐户。结果将产生错误，因为该帐号已经存在。

如果在运行脚本时遇到了错误，请在发生错误的位置检查脚本。大多数情况下，通过关联查询功能可以解决该问题，关联查询将一个语句的结果用作另一个语句的输入。

关联脚本时的主要步骤是：

**1 确定要关联的值。**

对于大多数协议，可以在执行日志中查看有问题的语句。双击错误消息可以直接跳至出错的位置。

或者，您还可以使用随 VuGen 分发的 WDiff 实用程序来确定脚本内的不一致性。详细信息，请参阅请参阅第 105 页的 “使用 Wdiff 比较 Vuser 脚本”。

**2 保存结果。**

可以使用适当的函数将查询的值保存为变量。这些关联函数是特定于协议的函数。关联函数名通常包含字符串 *save_param*，例如 **web_reg_save_param** 和 **lrs_save_param**。有关如何执行关联的解释，请参阅特定协议章。在某些协议中 （如数据库和 Web）， VuGen 会将这些函数自动插入到脚本中。

**3 引用保存的值。**

将查询或语句中的常量替换为保存的变量。

一些协议具有内置的自动或部分自动的关联：

➤ 对于 Java 语言 Vuser，请参阅第 15 章 “关联 Java 脚本”。

➤ 对于数据库 Vuser，请参阅第 19 章 “关联数据库 Vuser 脚本”。

➤ 对于 Web Vuser，请参阅第 41 章 “配置 Web Vuser 脚本的关联规则”。

➤ 对于 COM Vuser，请参阅第 26 章 “了解 COM Vuser 脚本”。

# 使用 C Vuser 的关联函数

要关联没有特定函数的协议的语句，可以使用 C Vuser 关联函数。这些函数可以用于所有 C 类型 Vuser，以便将字符串保存到参数，并在需要时检索它。有关 Java、 Corba-Java 或 RMI-Java Vuser 的类似函数的信息，请参阅第 104 页的“使用 Java Vuser 的关联函数”。

| | |
|---|---|
| **lr_eval_string** | 将所有出现的参数替换为其当前值 |
| **lr_save_string** | 将以 NULL 结尾的字符串保存到参数 |
| **lr_save_var** | 将可变长度字符串保存到参数 |

有关这些函数的语法的其他信息，请参阅 《LoadRunner Function Reference》。

## 使用 lr_eval_string

在下例中， **lr_eval_string** 将参数 *row_cnt* 替换为其当前值。该值将通过 **lr_output_message** 发送到输出窗口中。

```
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>"));
```

## 使用 lr_save_string

要将以 NULL 结尾的字符串保存到参数，请使用 **lr_save_string**。要保存可变长度字符串，请使用 **lr_save_var** 并指定要保存的字符串的长度。

在下例中， **lr_save_string** 将 777 分配给参数 *emp_id*。该参数随后将用于另一查询中，或供进一步处理。

```
lrd_stmt(Csr1, "select id from employees where name='John'",...);
lrd_bind_col(Csr1,1,&ID_D1,...);
lrd_exec(Csr1, ...);
lrd_fetch(Csr1, 1, ...);
/* 显示返回值 “777” 的 GRID */
lr_save_string("777", "emp_id");
```

# 使用 Java Vuser 的关联函数

要关联 Java、 CORBA-Java 和 RMI-Java Vuser 的语句，可以使用 Java Vuser 关联函数。这些函数可以用于所有 Java 类型 Vuser，以便将字符串保存到参数，并在需要时检索它。

| | |
|---|---|
| **lr.eval_string** | 将参数替换为其当前值 |
| **lr.eval_data** | 将参数替换为字节值 |
| **lr.eval_int** | 将参数替换为整型值 |
| **lr.eval_string** | 将参数替换为字符串 |
| **lr.save_data** | 将字节保存为参数 |
| **lr.save_int** | 将整型值保存为参数 |
| **lr.save_string** | 将以 NULL 结尾的字符串保存到参数 |

录制 CORBA-Java 或 RMI-Java 脚本时， VuGen 将在内部执行关联。详细信息，请参阅第 15 章 “关联 Java 脚本”。

## 使用 Java 字符串函数

编写 Java Vuser 脚本时，可以使用 Java Vuser 字符串函数来关联脚本。

在下例中， **lr.eval_int** 函数将变量 *ID_num* 替换为脚本中先前就已为该变量定义的值。

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

在下例中， **lr.save_string** 将 John Doe 分配给参数 Student。该参数随后将用在输出消息中。

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

# 使用 Wdiff 比较 Vuser 脚本

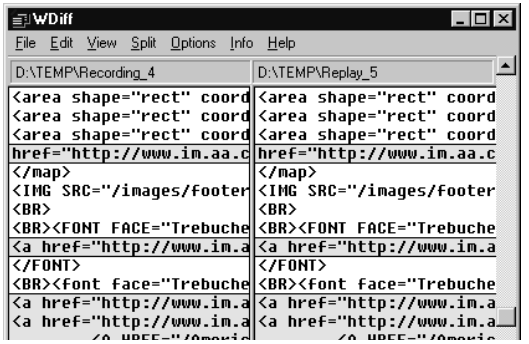
*WDiff* 是用于确定要关联的值的一种有用工具。通过该实用程序可以对录制的脚本和结果进行比较，以确定需要关联的值。

如果使用的是其他协议，您可以查看 “执行日志”以确定脚本失败的位置，然后使用 *WDiff* 实用程序帮助您查找需要关联的值。

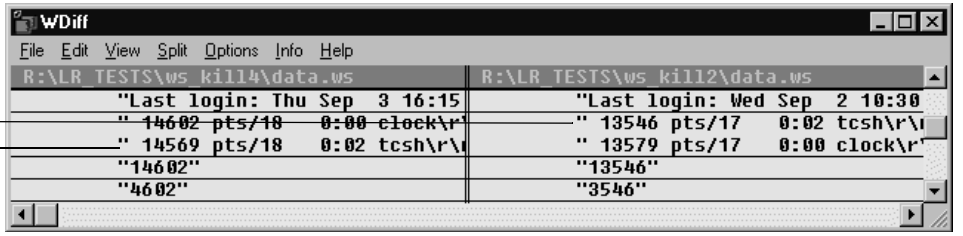
要有效地使用 Wdiff，需要录制两次同样的操作，然后比较脚本 （或 TUXEDO、WinSock 和 Jolt 的数据文件）。 WDiff 将以黄色显示差异。注意，并不是所有差异都表示要关联的值。例如，某些指明执行时间的接收缓冲区并不需要关联。

**要使用 Wdiff 搜索关联，请执行下列操作：**

1. 录制脚本并保存它。
2. 新建脚本并录制同样的操作。保存脚本。
3. 选择要比较的部分 （Actions、 data.ws 等）。
4. 选择 “工具”> “与 Vuser 比较”。将打开 “打开测试”框。
5. 指定要比较的 Vuser 脚本 （除了当前 VuGen 窗口中的脚本）并单击 “确定”。将打开 Wdiff， Vuser 脚本之间的差异将用黄色突出显示。




6. 要仅显示差异，请在 Wdiff 窗口中双击。



**7** 确定需要关联的值。

注意，在上例中， WDiff 比较的是两个 Winsock Vuser 脚本中的 *data.ws*。在该实例中，要关联的值是两个录制之间不同的 *clock* 进程的 PID。

要继续关联，请参阅相应的节：

- 对于 Java 语言 Vuser，请参阅第 15 章 “关联 Java 脚本”。
- 对于数据库 Vuser，请参阅第 19 章 “关联数据库 Vuser 脚本”。
- 对于 Web Vuser，请参阅第 41 章 “配置 Web Vuser 脚本的关联规则”。
- 对于 COM Vuser，请参阅第 26 章 “了解 COM Vuser 脚本”。
- 对于 WinSock Vuser，请参阅第 22 章 “使用 Window 套接字数据”。
- 对于 Tuxedo Vuser，请参阅第 59 章 “开发 Tuxedo Vuser 脚本”。

## 修改保存的参数

将某个值保存到参数后，在脚本中使用该参数之前可能还需要对其进行修改。如果需要对参数执行算术运算，必须使用 **atoi** 或 **atol** C 函数将其从字符串型更改为整型。将该值修改为整型之后，必须将其转换回字符串，以便在脚本中使用新变量。

在下面的 WinSock 示例中，位于偏移量 67 的数据被保存到参数 *param1*。使用 **atol**，VuGen 将该字符串转换为了长整型。在将 *param1* 增加了 1 之后，VuGen 使用 **sprintf** 将其转换回了字符串，并将其保存为新字符串 *new_param1*。参数的值使用 **lr_output_message** 显示。该新值可以稍后在脚本中使用。

```
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"));
```

# 9

# 配置运行时设置

录制 Vuser 脚本后，需要配置该脚本的运行时设置。这些设置指定脚本运行时的行为方式。

本章描述：

- 配置运行逻辑运行时设置 （多操作）
- 步运行时设置
- 配置步运行时设置 （多操作）
- 设置 “步”和 “运行逻辑”选项 （单操作）
- 配置日志运行时设置
- 配置思考时间设置
- 配置其他运行时设置
- 设置 VB 运行时设置

***下列信息适用于除 GUI 以外所有类型的 Vuser 脚本。***

## 关于运行时设置

录制 Vuser 脚本后，可以配置其运行时设置。运行时设置定义脚本运行的方式。这些设置存储在位于 Vuser 脚本目录下的文件 *default.cfg* 中。使用 VuGen 或 Controller 运行脚本时，会将运行时设置应用于 Vuser。

通过配置运行时设置，可以模拟各种不同的用户活动。例如，您可以模拟对服务器的输出立即做出响应的用户，也可以模拟在每次做出响应之前先停下来思考的用户。您还可以配置运行时设置指定 Vuser 应重复每组操作的次数。

使用 “运行时设置”对话框显示和配置运行时设置树。可以用下列方式之一打开这些设置：

- 单击 VuGen 工具栏上的 “运行时设置”按钮。
- 按键盘快捷键 F4。
- 选择 “Vuser” > “运行时设置”。

还可以从 LoadRunner Controller 中修改运行时设置。单击 “设计”选项卡，再单击 “运行时设置”按钮。

---

**注意**：对于 VuGen 和 Controller， Vuser 脚本具有单独的运行时设置默认值，支持 VuGen 的调试环境和 Controller 的负载测试环境。

下列是 VuGen 和 Controller 中 Vuser 脚本的默认设置：

**思考时间**：在 VuGen 中为关闭，而在 Controller 中为按录制参数回放。

**日志**：在 VuGen 中为标准，而在 Controller 中为关闭。

**下载非 HTML 资源**：在 VuGen 和 Controller 中都为已启用。

---

本章中描述的常规运行时设置适用于所有类型的 Vuser 脚本。其中包含：

- 运行逻辑 （循环）
- 步
- 日志
- 思考时间
- 其他

对于不支持多操作的协议，例如 WinSocket 和数据库 （Oracle 2-tier、 Sybase、 MSSQL 等），“循环”和 “步”选项都是在 “步”选项卡中处理的。许多协议都具有其他运行时设置。有关这些协议的特定运行时设置的信息，请参阅相应部分。

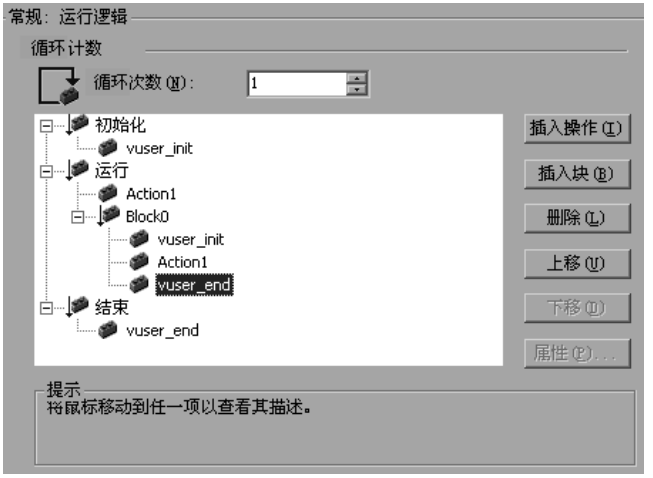
# 配置运行逻辑运行时设置 （多操作）

**注意**：以下部分仅适用于使用多操作的协议。如果运行设置中存在 “运行逻辑” 节点，则该协议为多操作协议。有关单操作协议的信息，请参阅第 114 页的 “设置 “步” 和 “运行逻辑” 选项 （单操作）”。

每个 Vuser 脚本包含三部分：*vuser_init*、*Run (Actions)* 和 *vuser_end*。可以指示 Vuser 在运行脚本时重复执行 *Run* 部分。每次重复称为一次循环。

运行多次循环时，将不会重复 Vuser 脚本的 *vuser_init* 和 *vuser_end* 部分。

打开 “运行时设置” 对话框，然后选择 “常规 : 运行逻辑” 节点。




**循环次数**：循环的次数。 LoadRunner 将按指定的次数重复执行所有 Actions。

如果在 Controller 的计划设置中指定了场景持续时间，则持续时间设置将覆盖 Vuser 循环设置。这意味着，如果将持续时间设置为五分钟 （默认设置）， Vuser 将在五分钟内按照需要继续运行任意多次循环，即使运行时设置仅指定一次循环。

运行多操作脚本时，可以指明执行操作的方法以及 Vuser 执行这些操作的方法：

**操作块**：操作块是脚本内的操作组。您可以单独设置每个块的属性 – 其顺序、循环和权重。

**顺序**：可以设置脚本内的操作顺序。还可以指明是按顺序还是随机执行操作。

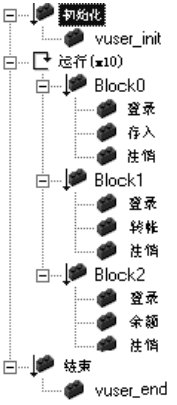
**循环**：除了为整个 *Run* 部分设置循环次数以外，还可以设置单个操作或操作块的循环。该设置非常有用，例如，可以用来模拟一个商业站点，您在该站点执行了多个查询查找某个产品，但只进行了一次采购。

**权重**：对于随机运行其操作的操作块，您可以设置块内每个操作的权重或百分比。

## 创建操作块

操作块是 Vuser 脚本内的操作组。可以为操作组创建单独的操作块，将同一操作添加到几个块中。可以指示 VuGen 按顺序或随机执行操作块或单个操作。在默认的连续模式中， Vuser 按照在循环树视图中出现的顺序执行块或操作。

在下例中， *Block0* 执行储蓄， *Block1* 执行转帐，而 *Block2* 提交结算请求。“登录”和 “注销”操作是这三个块所共有的。




单独配置每个块 – 其顺序和循环。

**要配置操作和操作块，请执行下列操作：**

1. 通过录制或编程创建所有必需的操作。
2. 打开 “运行时”设置。选择 “常规 : 运行逻辑”节点。
3. 添加新操作块。单击 “插入块”。 VuGen 将用下一个可用的索引 (*Block0*、*Block1*、 *Block2*) 在插入点插入新的操作块。

4 向块中添加操作。单击 “插入操作”。将打开 “选择操作”列表。

5 选择要向块中添加的操作，然后单击 “确定”。 VuGen 会将新操作插入到当前块或部分中。

6 对于要向块中添加的每个操作，重复执行步骤 3。

7 要删除某个操作或操作块，请选择它并单击 “删除”。

8 单击 “上移”或 “下移”修改项的位置。

9 单击 “属性”设置循环次数和操作的运行逻辑。将打开 “运行属性”对话框。

10 从 “运行逻辑”列表中选择 “顺序”或 “随机”，指明 VuGen 是按顺序还是随机运行操作。

11 在 “循环”框中指定循环次数。注意，如果在操作块中定义参数，并且指示 VuGen 每次循环都更新参数值，则此处的循环是指全局循环，而不是指单个块循环。

12 单击 “确定”。

**13** 对于使用 “随机”运行逻辑的块，请设置每个操作的权重。右键单击某个操作，然后选择 “属性”。将打开 “操作属性”对话框。

为选定的块或操作指定所需百分比。在 “随机百分比”框中，指定当前操作的百分比。所有百分比的总和必须等于 100。

**14** 对于要设置其属性的每个元素，重复执行上述步骤。

## 步运行时设置

---

**注意：**以下部分仅适用于使用多操作的协议。如果运行时设置中存在 “运行逻辑”节点，则该协议为多操作协议。有关单操作协议的信息，请参阅第 114 页的 “设置 “步”和 “运行逻辑”选项 （单操作）”。

---

通过步运行时设置，可以控制循环之间的时间。间隔将指示 Vuser 在操作循环之间等待多长时间。使用下列方法之一，可指示 Vuser 开始执行每次循环：

- 在前一次循环结束后立即开始。
- 在前一次循环结束并等待固定 / 随机的 ... 延迟后开始。
- 按固定 / 随机的时间间隔，每 ... 秒 / 最多 ... 秒。

**在前一次循环结束后立即开始：**前一次循环结束后，新的循环立即开始。

**在前一次循环结束并等待固定或随机的 ... 延迟后开始：**在前一次循环结束后，等待一段指定的时间再开始每次新的循环。请指定准确的秒数或时间范围。例如，可以指定在前一次循环结束后 60 秒到 90 秒之间的任意时间开始新的循环。

运行脚本时，将在执行日志中显示 Vuser 在一次循环结束与下一次循环开始之间等待的实际时间。

**按固定或随机的时间间隔，每 ... 秒或最多 ... 秒**：指定循环之间的时间 - 从前一次循环开始后固定的秒数或一个时间范围 （以秒计）。例如，可以指定每 30 秒开始新的循环，或按照自前一次循环开始后 30 秒到 45 秒的随机速率开始新的循环。仅当前一次循环完成时才开始执行每次计划的循环。

运行脚本时，将在执行日志中显示 Vuser 在一次循环结束与下一次循环开始之间等待的实际时间。仅当前一次循环完成时才开始执行每次计划的循环。

例如，假设您指定每四秒开始一次新的循环：

- 如果第一次循环用了三秒钟，则 Vuser 将等待一秒钟。
- 如果第一次循环用了两秒钟才完成，则 Vuser 将等待两秒钟。
- 如果第一次循环用了 8 秒钟才完成，则第二次循环将在第一次循环开始 8 秒钟后开始。 LoadRunner 将在执行日志中显示消息，指明未能完成的循环步。

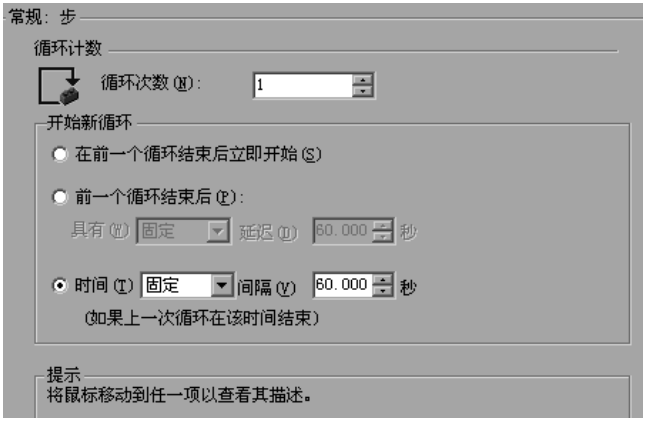
有关设置 “步”选项的进一步说明，请参阅第 113 页的 “配置步运行时设置（多操作）”。

## 配置步运行时设置 （多操作）

使用 “间隔”选项，可通过设置循环之间的时间间隔对操作进行协调。

**要设置循环之间的步，请执行下列操作：**

**1** 打开 “运行时设置”，然后选择 “常规 : 步”节点。



2 在 “开始新循环”部分中，选择下列选项之一：

- ➤ 在前一次循环结束后立即开始。
- ➤ 前一次循环结束后。
- ➤ 按固定或随机时间间隔。

3 对于 “前一次循环结束后”选项：

- ➤ 选择延迟类型：“固定”或 “随机”。
- ➤ 对于固定延迟，请指定一个值；对于随机延迟，请指定值的范围。

4 对于 “按 … 间隔”选项：

- ➤ 选择间隔类型：“固定”或 “随机”。
- ➤ 对于固定间隔，请指定一个值；对于随机间隔，请指定值的范围。

5 单击 “确定”。

## 设置 “步”和 “运行逻辑”选项 （单操作）

---

**注意**：以下部分仅适用于使用单操作的协议，而不适用于使用多操作的协议。如果常规运行时设置下是 “步”节点而不是 “运行逻辑”节点，则该协议为单操作协议。

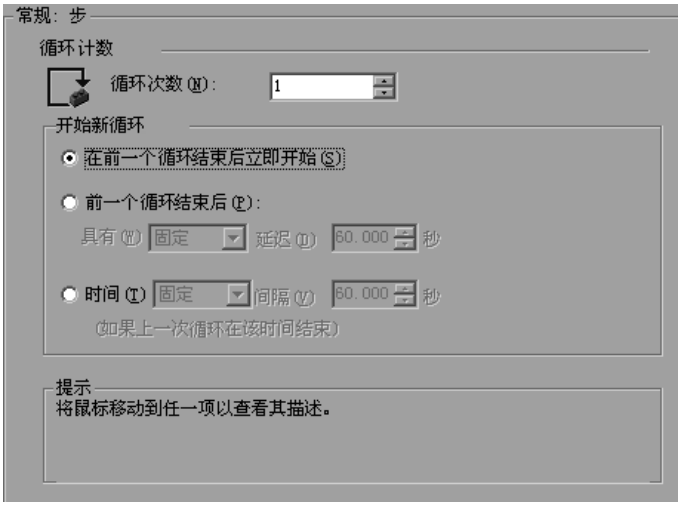
---

可以指示 Vuser 在运行脚本时重复执行 *Action* 部分。每次重复称为一次循环。运行多次循环时将不会重复执行 Vuser 脚本的 *vuser_init* 和 *vuser_end* 部分。

如果在 Controller 的计划设置中指定了场景持续时间，则持续时间设置将覆盖 Vuser 循环设置。这意味着，如果将持续时间设置为五分钟 （默认设置）， Vuser 将在五分钟内按照需要继续运行任意多次循环，即使运行时设置仅指定一次循环。

**要设置循环和步首选项，请执行下列操作：**

**1** 单击 VuGen 工具栏中的 “运行时设置”按钮，或选择 “Vuser” > “运行时设置”。单击 “步”节点显示循环和步选项。




**2** 在 “循环计数”框中指定循环次数。LoadRunner 将按指定的次数重复执行所有 Actions。

**3** 在 “开始新循环”部分中，选择下列选项之一：

- 在前一次循环结束后立即开始。
- 前一次循环结束后
- 按固定或随机时间间隔

**4** 对于 “前一次循环结束后”选项：

- 选择延迟类型：“固定”或 “随机”。
- 对于固定延迟，请指定一个值；对于随机延迟，请指定值的范围。

**5** 对于 “按 ... 间隔”选项：

- 选择间隔类型：“固定”或 “随机”。
- 对于固定间隔，请指定一个值；对于随机间隔，请指定值的范围。

**6** 单击 “确定”。

有关步选项的概述，请参阅第 112 页的 “步运行时设置”。

# 配置日志运行时设置

执行期间，Vuser 会记录有关它本身及其与服务器之间通信的信息。在 Windows 环境中，该信息将存储在脚本目录下名为 *output.txt* 的文件中。在 UNIX 环境中，该信息将直接发送到标准输出中。调试时，该日志信息非常有用。

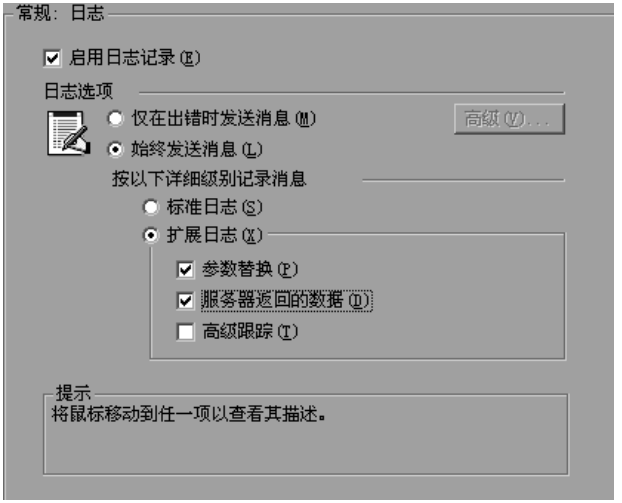
通过日志运行时设置，可以确定记录到输出中的信息量。可以选择 “标准”或 “扩展”日志，也可以完全禁用日志记录。使用多个 Vuser 时，禁用日志非常有用。如果让数十个或数百个 Vuser 将其运行时信息记录到磁盘中，系统的运行速度可能要比正常情况下慢。开发期间，可启用日志记录以便获得有关回放的信息。仅在验证了脚本能够正常工作后，才应禁用日志记录。

---

**注意**：可以编写一个 Vuser 脚本，使用 **lr_error_message** 和 **lr_output_message** 函数将消息发送到输出中。

---

单击 “运行时设置”按钮，或者选择 “Vuser” > “运行时设置”，显示 “运行时设置”对话框。选择 “常规 : 日志”节点显示日志选项。



### 启用日志记录

该选项可启用在回放期间自动进行日志记录的功能 – VuGen 将写入日志消息，您可以在执行日志中查看这些消息。该选项仅影响自动日志记录和通过 **lr_log_message** 发出的日志消息。仍会发出使用 **lr_message**、**lr_output_message** 和 **lr_error_message** 手动发送的消息。

### 日志选项

通过日志运行时设置，可以根据开发阶段调整日志记录级别。

可以指明将日志消息发送到日志的时间：“仅在出错时发送消息”或“始终发送消息”。在开发过程中，可以启用所有日志记录功能。调试脚本并验证其能够正常工作后，可以仅为错误启用日志记录功能。

如果选择“仅在出错时发送消息”（也称为 JIT （实时）消息传递），则可以设置高级选项，指明日志高速缓存的大小。请参阅第 118 页的“设置日志高速缓存大小”。

### 设置日志详细级别

可以指定所记录的信息的类型，也可以完全禁用日志记录功能。

---

**注意**：如果在常规运行时设置文件夹中将“错误处理”设置为“出现错误时仍继续”，则错误消息仍将被发送到输出窗口。

如果修改了脚本的“日志详细级别”， **lr_message**、**lr_output_message** 和 **lr_log_message** 函数的行为将不会更改 – 这些函数将继续发送消息。

---

**标准日志**：创建在脚本执行期间所发送的函数和消息的标准日志，供调试时使用。请为大型负载测试场景禁用该选项。

如果将日志记录级别设置为标准，则在将日志记录模式添加到场景中时，会自动将其设置为 JIT 日志记录。

**扩展日志**：创建扩展日志，包括警告和其他消息。请为大型负载测试场景禁用该选项。如果禁用了日志记录功能，或者如果将日志级别设置为扩展，则将日志记录模式添加到场景中并不会影响日志设置。

可以使用 “扩展日志”选项指定应向扩展日志中添加哪些其他信息：

- **参数替换**：选择该选项可记录指定给脚本的所有参数及其相应的值。有关参数的详细信息，请参阅第 7 章 “定义参数”。
- **服务器返回的数据**：选择该选项可记录服务器返回的所有数据。
- **高级跟踪**：选择该选项可记录 Vuser 在会话期间发送的所有函数和消息。调试 Vuser 脚本时，该选项非常有用。

VuGen 日志事件 （标准、参数替换等）的等级也称为消息类。有五个消息类：简要、扩展、参数、结果数据和完全跟踪。

可以使用 **lr_set_debug_message** 函数手动设置脚本内的消息类。如果您希望仅接收一小部分脚本的相关调试信息，该设置非常有用。

例如，假设将日志运行时设置设为 “标准日志”，并且希望获得脚本的特定部分的扩展日志。您将使用 **lr_set_debug_message** 函数在脚本中的所需位置设置 “扩展”消息类。必须再次调用该函数，指定是何种类型的扩展模式 （“参数”、“结果数据”或 “完全跟踪”）。通过调用 **lr_set_debug_message** 函数并指定 “简要”模式，可返回到 “标准日志”模式。有关设置消息类的详细信息，请参阅《Online Function Reference》（“帮助” > “函数参考”）。

如果将日志记录模式设置为 “标准日志”，在将脚本复制到场景中时， VuGen 会自动将日志记录模式设置为实时日志记录。如果将日志记录模式设置为 “扩展”，或者禁用了日志记录功能，则将脚本复制到场景中不会影响其日志记录设置。

### 设置日志高速缓存大小

通过日志运行时设置的 “高级”选项，可以指明日志高速缓存的大小。日志高速缓存用于存储有关测试执行的原始数据，供发生错误时使用。当该高速缓存的内容超过指定的大小时，将删除最旧的一些项。默认大小为 1KB。

**以下是日志记录的顺序：**

1. 通过选择 “仅在出错时发送消息”，指示 VuGen 仅当出错时记录消息。
2. VuGen 将有关测试执行的信息存储在日志高速缓存中，而不是将其写入文件。如果这些信息超过了 1 KB，将覆盖最旧的数据。“执行日志”选项卡也将保留为空，因为它是日志文件内容的转储。
3. 发生错误 （内部错误或使用 **lr_error_message** 函数编写的错误）时， VuGen 会将高速缓存中的内容放到日志文件和 “执行日志”选项卡中。这样您就可以查看导致错误的事件。

如果发生了错误并且 VuGen 将其存储的高速缓存内容转储到日志文件中，则实际文件的大小将大于高速缓存的大小。例如，如果高速缓存的大小为 1KB，则日志文件的大小可能为 50 KB。这种情况是正常的，它仅反映了将原始数据格式化为有意义的句子所需要的系统开销。

注意，在 JIT 模式下，如果发生错误时 **lr_message** 和 **lr_log_message** 函数的输出已经位于日志高速缓存中，则仅将这些函数的输出发送到输出窗口或日志文件中。请检查指定消息字符串的执行日志。

## 记录 CtLib 服务器消息

运行 CtLib Vuser 脚本 （在 “客户端服务器”类型协议下为 Sybase CtLib）时，由 CtLib 客户端生成的所有消息都将被记录到标准日志和输出文件中。默认情况下，不会记录服务器消息。要启用服务器消息的日志记录功能 （以便进行调试），请将以下行插入到 Vuser 脚本中：

```
LRD_CTLIB_DB_SERVER_MSG_LOG;
```

VuGen 将在标准日志中记录所有服务器消息。

要将服务器消息发送到 Controller 输出窗口中 （除了将其发送到标准日志以外），请键入：

```
LRD_CTLIB_DB_SERVER_MSG_ERR;
```

要返回到默认的不记录服务器错误的模式，请将以下行键入到 Vuser 脚本中：

```
LRD_CTLIB_DB_SERVER_MSG_NONE;
```

---

**注意**：因为生成的服务器消息很长，而且日志记录会降低系统的运行速度，所以请仅为脚本中特定的代码块激活服务器消息日志记录功能。

---

# 配置思考时间设置

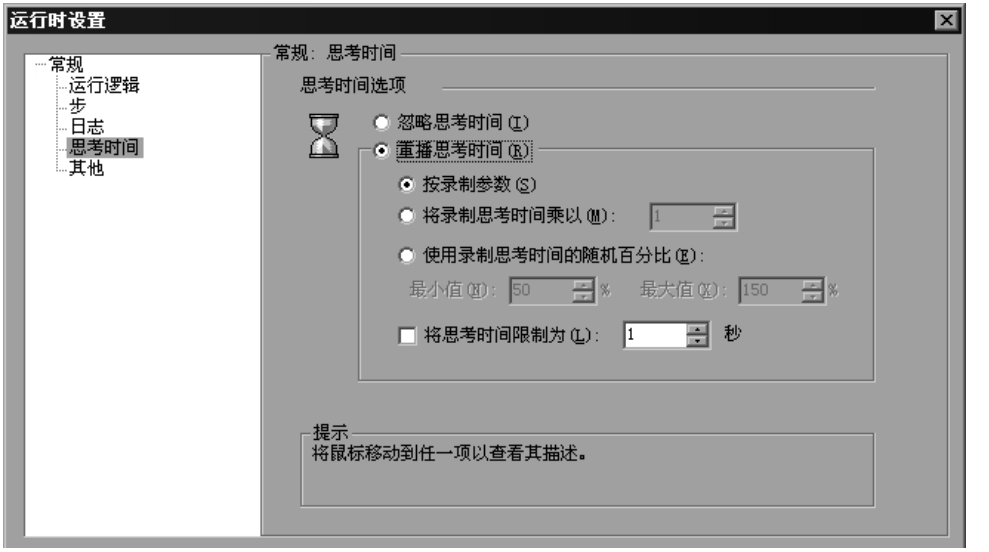
Vuser 思考时间模拟实际用户在不同操作之间等待的时间。例如，当从服务器接收数据时，用户可能等待几秒钟查看数据，然后再做出响应。这种延迟就称为“思考时间”。VuGen 使用 **lr_think_time** 函数将思考时间值录制到 Vuser 脚本中。以下录制的函数指明用户等待了 8 秒钟才执行下一个操作：

```
lr_think_time(8);
```

运行 Vuser 脚本并且 Vuser 遇到了上述 **lr_think_time** 语句时，默认情况下，Vuser 将等待 8 秒钟后再执行下一个操作。通过思考时间运行时设置，可以影响运行脚本时 Vuser 使用录制思考时间的方式。

有关 **lr_think_time** 函数以及如何手动修改该函数的详细信息，请参阅《Online Function Reference》（“帮助” > “函数参考”）。

单击 VuGen 工具栏中的 “运行时设置”按钮，或选择 “Vuser” > “运行时设置”。选择 “常规 : 思考时间”节点显示 “思考时间”选项：



## 思考时间选项

默认情况下，当您运行 Vuser 脚本时， Vuser 将使用在录制会话期间录制到脚本中的思考时间值。通过 VuGen 可以使用录制思考时间、忽略思考时间或使用与录制时间相关的值：

**忽略思考时间**：忽略录制思考时间 – 回放脚本时忽略所有 **lr_think_time** 函数。

**回放思考时间**：通过第二组思考时间选项，可以使用录制思考时间：

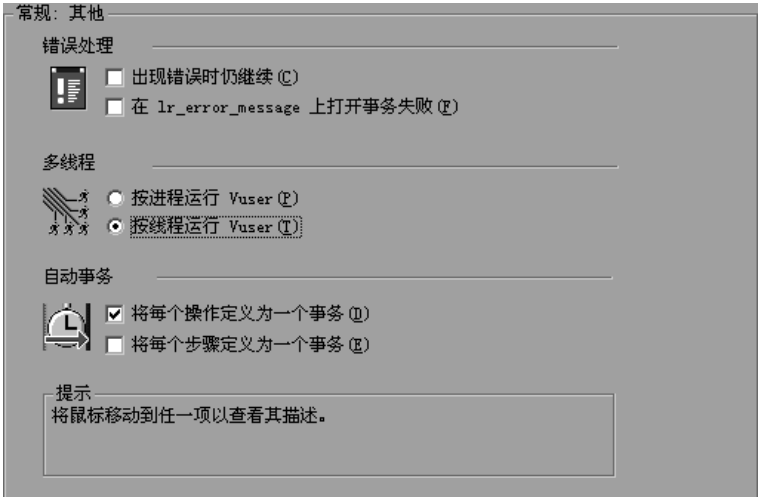
- **按录制参数**：回放期间，使用 **lr_think_time** 函数中显示的参数。例如，**lr_think_time(10)** 将等待十秒钟。
- **将录制思考时间乘以**：回放期间，使用录制思考时间的倍数。这种方法可以增加或减少在回放期间应用的思考时间。例如，如果录制思考时间为四秒，则可以指示 Vuser 用二乘以该值，即总共为八秒。要将思考时间减少至两秒，可以用 0.5 乘以录制时间。
- **使用录制思考时间的随机百分比**：使用录制思考时间的随机百分比。通过指定思考时间的范围，可设置思考时间值的范围。例如，如果思考时间参数为 4，并且您指定最小值为该值的 50%，而最大值为该值的 150%，则思考时间的最低值为二 (50%)，而最高值为六 (150%)。
- **将思考时间限制为**：限制思考时间的最大值。

# 配置其他运行时设置

可以为 Vuser 脚本设置下列其他运行时选项：

- 错误处理
- 多线程
- 自动事务

单击 “运行时设置”按钮或者选择 “Vuser” > “运行时设置”，显示 “运行时设置”对话框。从左窗格的树中选择 “常规 : 其他”节点。




其他设置适用于所有 Vuser 脚本类型。

### 错误处理

可以指定脚本执行期间 Vuser 处理错误的方法。默认情况下，当 Vuser 检测到错误时，将退出执行。可以使用运行时设置指示 Vuser 在出现错误时仍继续执行脚本。要完成该操作，请在其他运行时设置中选择 “出现错误时仍继续”复选框。

还可以指示 VuGen 标记在其中发出 **lr_error_message** 函数的所有事务，如 “失败”。遇到特定情况时，将通过 *If* 编程语句发出 **lr_error_message** 函数。

### 数据库 Vuser 的错误处理

使用数据库协议 (LRD) 时，可以控制特定脚本段的错误处理。要标记某段，请用 LRD_ON_ERROR_CONTINUE 和 LRD_ON_ERROR_EXIT 语句括住该段。Vuser 将对整个段应用新的错误设置。如果指定 “出现错误时仍继续”，则 VuGen 将发出一条消息，指明它遇到了错误并将忽略该错误。

例如，如果启用了 “出现错误时仍继续”功能，并且 Vuser 在回放以下脚本段期间遇到了错误，则 Vuser 将继续执行该脚本。

```
lrd_stmt(Csr1, "select..."...);
lrd_exec(...);
```

要指示 Vuser 在出现错误时仍继续执行除特定段以外的整个脚本，请选择 “出现错误时仍继续”选项，同时用 LRD_ON_ERROR_EXIT 和 LRD_ON_ERROR_CONTINUE 语句括住该段：

```
LRD_ON_ERROR_EXIT;
lrd_stmt(Csr1, "select..."...);
lrd_exec(...);
LRD_ON_ERROR_CONTINUE;
```

除了 LRD_ON_ERROR 语句以外，还可以使用严重级别控制错误处理。LRD_ON_ERROR 语句检测所有类型的错误 - 与数据库相关的错误、无效参数等。如果希望 Vuser 仅当出现数据库操作错误 （错误代码为 2009）时才终止，则可以设置函数的严重级别。所有执行数据库操作的函数都使用严重级别，严重级别由函数的最后一个参数 *miDBErrorSeverity* 指明。

VuGen 支持下列严重级别：

| 定义 | 含义 | 值 |
|---|---|---|
| LRD_DB_ERROR_SEVERITY_ERROR | 出现数据库访问错误时终止脚本执行。（默认设置） | 0 |
| LRD_DB_ERROR_SEVERITY_WARNING | 出现数据库访问错误时继续执行脚本，但会发出警告 | 1 |

例如，如果以下数据库语句失败 （例如表不存在），将终止脚本执行。

```
lrd_stmt(Csr1, "insert into EMP values ('Smith',301)\n", -1, 1, 1, 0);
```

要指示 VuGen 即使在发生数据库错误时仍继续执行脚本，请将语句的严重级别从 0 改为 1。

```
lrd_stmt(Csr1, "insert into EMP values ('Smith',301)\n", -1, 1, 1, 1);
```

---

**注意**：启用 “出现错误时仍继续”功能时，将覆盖 0 严重级别；即使发生数据库错误，也将继续执行脚本。然而，如果禁用了 “出现错误时仍继续”功能，但将严重级别指定为 1，则当发生数据库错误时仍将继续执行脚本。

---

### RTE Vuser 的错误处理

使用 RTE Vuser 时，可以控制特定函数的错误处理。在要更改其行为的函数前插入 **lr_continue_on_error(0);** 语句。 Vuser 将使用新设置，直到脚本执行结束，或者直到发出另一个 **lr_continue_on_error** 语句。

例如，如果启用了 “出现错误时仍继续”功能，并且 Vuser 在回放以下脚本段期间遇到了错误，则 Vuser 将继续执行该脚本。

```
TE_wait_sync();
TE_type(...);
```

要指示 Vuser 在出现错误时仍继续执行除以下段以外的整个脚本，请选择 “出现错误时仍继续”选项，并且用 **lr_continue_on_error** 语句括住该段，同时使用 0 关闭 “出现错误时仍继续”功能，使用 1 重新打开该功能：

```
lr_continue_on_error(0);
TE_wait_sync();
lr_continue_on_error(1);
....
```

## 多线程

Vusers 支持多线程环境。多线程环境的主要优势是每个负载生成器能运行多个 Vuser。只有线程安全协议才能作为线程运行。

---

**注意**：下列协议不是线程安全协议：Sybase-Ctlib、 Sybase-Dblib、 Informix、 Tuxedo 和 PeopleSoft-Tuxedo。

---

- 要启用多线程，请单击 “按线程运行 Vuser”。
- 要禁用多线程并按单独的进程运行每个 Vuser，请单击 “按进程运行 Vuser”。

Controller 将使用驱动程序 （例如 mdrv.exe、 r3vuser.exe）运行 Vuser。如果按进程运行每个 Vuser，则对于每个 Vuser 实例，都将反复启动同一驱动程序并将其加载到内存中。将同一驱动程序加载到内存中会占用大量 RAM （随机存取存储器）及其他系统资源。这就限制了可以在任一负载生成器上运行的 Vuser 的数量。

或者，如果按线程运行每个 Vuser， Controller 为每 50 个 Vuser（默认情况下）仅启动驱动程序 （如 mdrv.exe）的一个实例。该驱动进程 / 程序将启动几个 Vuser，每个 Vuser 都按线程运行。这些线程 Vuser 将共享父驱动进程的内存段。这就消除了多次重新加载驱动程序 / 进程的需要，节省了大量内存空间，从而可以在一个负载生成器上运行更多的 Vuser。

## 自动事务

可以指示 LoadRunner 按事务处理 Vuser 脚本中的每个步骤或操作。这称为 “使用自动事务”。 LoadRunner 将步骤名或操作名指定为事务名。默认情况下，将启用按操作使用自动事务的功能。

- 要禁用按操作使用自动事务的功能，请清除 “将每个操作定义为一个事务” 复选框。（默认情况下启用）
- 要启用按步骤使用自动事务的功能，请选中 “将每个步骤定义为一个事务” 复选框。（默认情况下禁用）

如果禁用了自动事务，仍可以在录制期间或录制之后手动插入事务。有关手动插入事务的详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

# 设置 VB 运行时设置

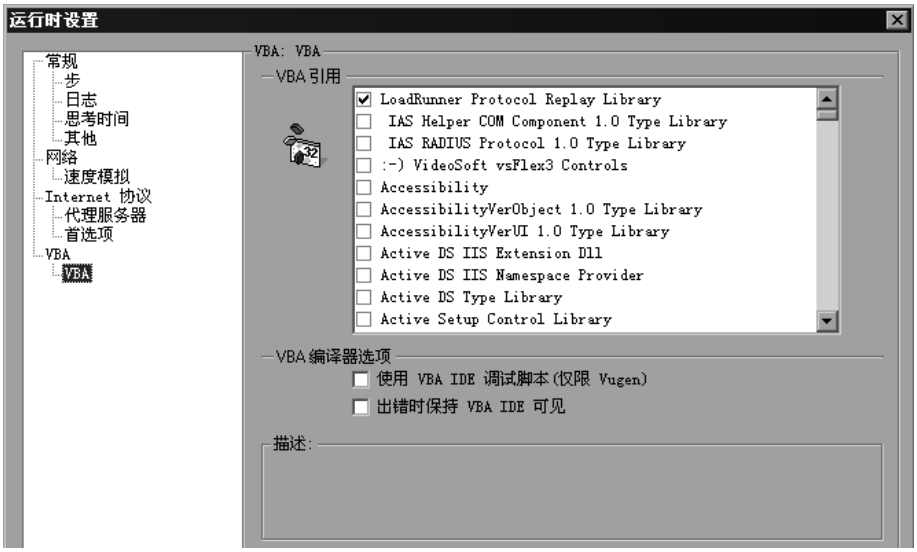
运行 Visual Basic 脚本之前，需要指明回放期间要引用的库。

使用 “运行时设置”对话框显示和配置运行时设置。要显示 “运行时设置”对话框，请单击 VuGen 工具栏上的 “运行时设置”按钮。

您还可以从 LoadRunner Controller 中修改运行时设置。单击 “设计”选项卡，再单击 “运行时设置”按钮。

**要设置 VBA 运行时设置，请执行下列操作：**

**1** 打开 “运行时设置”对话框，然后选择 “VBA：VBA”节点。




**2** 在 “VBA 引用”部分中，选择运行脚本时要使用的引用库。选择某个库后，将在对话框的底部显示该库的描述和版本。

**3** 选择适当的编译器选项：

选择 “通过 VBA IDE 调试脚本”启用通过 Visual Basic IDE （集成开发环境）进行调试的功能。

选择 “出错时保持 VBA IDE 可见”在脚本执行期间保持 Visual Basic IDE 可见。

**4** 选择 “确定”应用运行时设置。

# 10

# 配置网络运行时设置

要模拟网络速度，请配置网络运行时设置。

本章描述：

➤ 设置网路速度

***下列信息适用于所有“Internet 协议”Vuser 类型，Oracle NCA 以及 WinSock。***

有关适用于所有 Vuser 的常规运行时设置的信息，请参阅第 9 章 “配置运行时设置”。

## 关于网络运行时设置

开发 Internet 协议 Vuser 脚本之后，需要设置运行时设置。通过这些设置可以配置 Internet 环境，以便 Vuser 可以准确模拟实际用户。可以为 “代理服务器”、“浏览器”、“速度模拟”以及其他高级首选项设置 Internet 运行时设置。

可以从 “运行时设置”对话框中设置与 Internet 相关的运行时设置。单击相应的节点可指定所需的设置。

**要显示 “运行时设置”对话框，请执行下列操作：**

➤ 单击 VuGen 工具栏上的 “运行时设置”按钮。

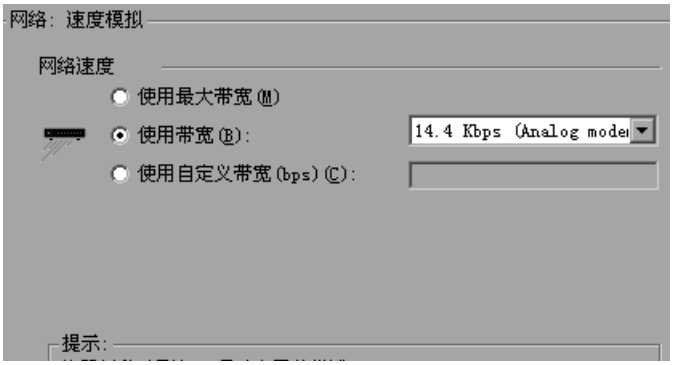
➤ 按键盘快捷键 F4。

➤ 选择 “Vuser” > “运行时设置”。

注意，还可以从 LoadRunner Controller 中修改运行时设置。在 “Controller”窗口中，单击 “设计”选项卡，再单击 “运行时设置”按钮。

# 设置网路速度

使用 “运行时设置” 树中的 “网络 : 速度模拟” 节点，可设置测试环境的调制解调器仿真。




### 速度模拟

使用 “速度模拟” 设置，可以选择能够最好地模拟所测试的环境的带宽。可以使用下列选项：

**使用最大带宽**：默认情况下会禁用带宽仿真，并且 Vuser 按网络上的最大可用带宽运行。

**使用带宽**：指明 Vuser 要模拟的特定带宽级别。可以选择从 14.4 至 512 Kbps 范围内的速度，以便模拟模拟调制解调器、 ISDN 或 DSL。

**使用自定义带宽**：指明 Vuser 进行模拟的带宽限制。以位为单位指定带宽，其中 1 K=1024。

# 11

# 在独立模式下运行 Vuser 脚本

开发 Vuser 脚本并设置其运行时设置之后，可以通过在独立模式下运行该 Vuser 脚本来对其测试。

本章描述：

- 在 VuGen 中运行 Vuser 脚本
- 使用 VuGen 的调试功能
- 使用 VuGen 的 Web Vuser 脚本调试功能
- 使用 VuGen 窗口
- 从命令提示符运行 Vuser 脚本
- 从 UNIX 命令行运行 Vuser 脚本
- 将 Vuser 脚本集成到场景中

***以下信息适用于除 GUI 以外所有类型的 Vuser 脚本。***

## 关于在独立模式下运行 Vuser 脚本

要对 Vuser 脚本执行负载测试，可以使用 Controller 将该脚本集成到场景中。将脚本集成到负载测试场景之前，请在独立模式下运行该脚本来检查其功能。

在独立模式下运行脚本意味着不使用 Controller 运行该脚本。该操作的目的是要确定从 Controller 运行脚本时，该脚本将如何运行。要在独立模式下运行 GUI Vuser，请使用 WinRunner。详细信息，请参阅第 66 章 “开发 GUI Vuser 脚本”。对于其他所有基于 Windows 的脚本，可以使用 VuGen 在独立模式下运行这些脚本。如果脚本是基于 UNIX 的，则可以从 UNIX 命令行运行该脚本。

独立运行成功后，可以将脚本合并到场景中。有关场景的详细信息，请参阅《LoadRunner Controller 用户指南》。

在独立模式下运行脚本之前，可以执行下列操作：

➤ 使用 Vuser 函数来增强该脚本 （请参阅第 6 章 “增强 Vuser 脚本”。）

➤ 参数化该脚本 （请参阅第 7 章 “定义参数”。）

➤ 关联该脚本中的查询 （请参阅第 8 章 “关联语句”。）

上述步骤是可选的，可能并不适用于所有脚本。

# 在 VuGen 中运行 Vuser 脚本

开发 Vuser 脚本之后，可以使用 VuGen 来运行该脚本，以确保脚本能够正确运行。可以设置几个回放选项。

---

**注意**：VuGen 仅在 Windows 平台上运行 Vuser 脚本。要运行基于 UNIX 的 Vuser 脚本，请参阅第 137 页的 “从 UNIX 命令行运行 Vuser 脚本”。

---

## 配置回放选项

可以在动态模式或非动态模式下运行 Vuser 脚本。在动态模式下运行时，VuGen 将突出显示 Vuser 脚本中当前正在执行的行。可以为该模式设置一个延迟，以便可以更好地查看每一步的效果。在非动态模式下运行时， VuGen 将执行 Vuser 脚本，但并不指明正在执行的行。

**动态运行延迟**：命令之间的时间延迟 （以毫秒为单位）。默认延迟值为 0。

**仅激活 “Actions” 部分中的函数**：仅激活 Action 部分而不是 *init* 或 *end* 部分的内容。

要启用动画并设置其属性，请执行下列操作：

1 选择 “视图” > “动态运行”来以动态模式运行。 VuGen 将在 “逐行显示运行”菜单选项旁放置复选标记，以启用动态模式。

2 要设置动态运行延迟，请选择 “工具” > “常规选项”。将打开 “常规选项”对话框。

3 选择 “回放”选项卡。

4 在 “动态运行延迟”框中，指定一个延迟时间 （以毫秒为单位），然后单击 “确定”。

5 选择 “仅激活 ‘Actions’部分中的函数”来仅激活 “Actions”部分中的内容。

### 设置显示选项

如果正在运行 Web Vuser 脚本，则可以设置 “显示”选项 （“工具” > “常规选项”）。这些选项用于指定是否显示 VuGen 运行时查看器、是否要在脚本执行期间生成报告等。

详细信息，请参阅第 134 页的 “使用 VuGen 的 Web Vuser 脚本调试功能”。

### 回放 Vuser 脚本

1 选择 “Vuser” > “运行”。

将在 VuGen 主窗口的底部打开 “输出”窗口 （如果已经打开，则清除该窗口），VuGen 开始执行 Vuser 脚本。

VuGen 将从脚本的第一行开始运行 Vuser 脚本。“执行日志”中显示的消息用于描述 Vuser 运行时执行的操作。该信息可显示在场景中执行脚本时，该脚本是如何运行的。

---

**注意**：对于支持树视图的协议 （如 “视图”菜单所示），在树视图中运行 Vuser 脚本时， VuGen 将从 Vuser 脚本中的第一个图标开始运行该脚本。

---

2 要在脚本运行期间或之后隐藏 “输出”窗口，请选择 “视图” > “输出窗口”。VuGen 将关闭 “输出”窗口，并删除 “输出”窗口的 “视图”菜单旁边的复选标记。

3 要中断正在运行的 Vuser 脚本，请选择 “Vuser” > “暂停”以便临时暂停脚本的运行，或者选择 “Vuser” > “停止”以便结束脚本的运行。

## 输出窗口：执行日志

“输出”窗口的 “执行日志”显示的消息用于描述 Vuser 运行时执行的操作。该信息可说明在场景中执行脚本时，该脚本的运行方式。

脚本执行完成后，可以检查 “执行日志”中的消息，以查看脚本在运行时是否发生错误。

“执行日志”中使用了不同颜色的文本。

- 黑色：标准输出消息
- 红色：标准错误消息
- 绿色：用引号括起来的文字字符串 （例如 URL）
- 蓝色：事务信息 （开始、结束、状态和持续时间）

如果双击以操作名开始的行，光标将会跳到生成的脚本中的相应步骤上。

有关关闭和打开 “输出”窗口的详细信息，请参阅第 132 页的 “回放 Vuser 脚本”。

下例显示 Web Vuser 脚本运行时的 “执行日志”消息。

*“执行日志”消息*

# 使用 VuGen 的调试功能

VuGen 包含两个选项来帮助调试 Vuser 脚本：“分步运行”命令和断点。这些选项不适用于 VBscript 和 VB 应用程序类型的 Vuser。

VuGen 还包含其他一些功能来帮助调试 Web Vuser 脚本。详细信息，请参阅第 134 页的 “使用 VuGen 的 Web Vuser 脚本调试功能”。

## “分步运行”命令

“分步运行”命令在运行脚本时一次运行一行。通过该命令，可以依次查看脚本每一行的执行情况。

**要分步运行脚本，请执行下列操作：**

1 选择 “Vuser” > “分步运行”，或者单击 “步骤”按钮。

VuGen 将执行脚本的第一行。

2 继续单击 “步骤”按钮来执行该脚本，直到脚本运行完成为止。

### 断点

通过断点可以使脚本在特定位置暂停执行。它可用于检查在执行期间的预定时间点处，该脚本对应用程序的影响。

**要设置断点，请执行下列操作：**

1 将光标置于脚本中要停止执行的行上。

2 单击 “断点”按钮。将在脚本的左边距显示 “断点”符号 ( )。

3 要删除断点，请将光标置于含有断点符号的行上，然后单击 “断点”按钮。

**要运行含有断点的脚本，请执行下列操作：**

1 照常运行脚本。

到达断点时， VuGen 将暂停脚本的执行。可以检查脚本运行到断点时的效果，并进行必要的更改，然后从断点处重新启动脚本。

2 要继续执行，请选择 “Vuser” > “运行”。

重新启动后，脚本将继续执行，直到遇到下一个断点或脚本完成。

## 使用 VuGen 的 Web Vuser 脚本调试功能

VuGen 还提供了两种工具来帮助您调试 Web Vuser 脚本：运行时查看器 （联机浏览器）和 “结果概要”报告。

➤ 可以指示 VuGen 在运行 Web Vuser 脚本时显示运行时查看器。运行时查看器是 Mercury Interactive 为了与 VuGen 配合使用而专门开发的，它与录制 Vuser 脚本时使用的浏览器无关。该运行时查看器可显示由 Vuser 访问的每个网页。这对调试 Web Vuser 脚本非常有用，因为通过它可以检查 Vuser 访问的网页是否正确。有关运行时查看器的详细信息，请参阅第 45 章 “有关 Web Vuser 的 Power User 提示”。

---

**注意**：要显示运行时查看器，必须安装 Microsoft Internet Explorer 4.0 或更高版本。

---

➤ 可以指定 Web Vuser 是否在脚本执行期间生成 “结果概要” 报告。“结果概要” 报告概述 Web Vuser 脚本中的每一步是成功还是失败，并可以通过它查看每一步返回的网页。有关使用 “结果概要” 报告的详细信息，请参阅第 44 章 “使用报告调试 Vuser 脚本”。

---

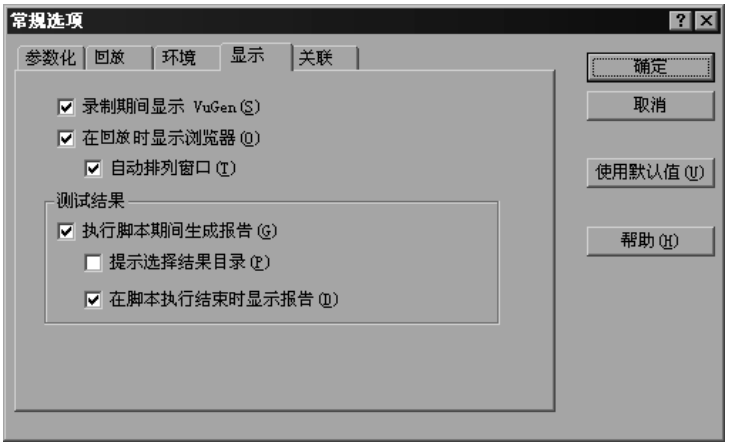
**注意**：Vuser 生成 “结果概要” 报告时，事务时间可能会增加。

Vuser 可以仅在从 VuGen 运行时才生成 “结果概要” 报告。使用 Controller 运行 Web Vuser 脚本时， Vuser 不能生成报告。

---

**要启用 Web Vuser 脚本调试功能，请执行下列操作：**

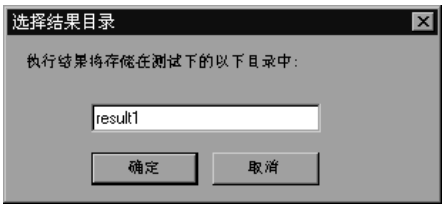
**1** 从 VuGen 菜单中选择 “工具” > “常规选项”。将打开 “常规选项” 对话框。选择 “显示” 选项卡。




**2** 选中 “录制期间显示 VuGen” 或 “在回放期间显示浏览器” 复选框，以便在录制期间查看 VuGen 或启用运行时查看器。选中 “自动排列窗口” 复选框，以便在脚本执行完成时使运行时查看器最小化。

3 选中 “测试结果” 部分中的 “执行脚本期间生成报告” 复选框，以便指示 Vuser 生成 “结果概要” 报告。

4 选中 “提示选择结果目录” 复选框，以便在执行 Vuser 脚本之前显示 “选择结果目录” 对话框。




键入要存储执行结果的文件夹名，或者接受默认名，然后单击 “确定”。

如果未选中 “提示选择结果目录” 复选框， VuGen 会自动将该文件夹命名为 *result1*。以后的脚本执行过程将自动覆盖以前的文件，除非另外指定了结果文件。注意，结果总是存储在脚本文件夹的子文件夹中。

5 选中 “在脚本执行结束时显示报告” 复选框，以便在脚本执行结束时自动显示 “结果概要” 报告。如果不选择该选项，则可以在脚本执行之后通过选择 “视图” > “可视日志” 来打开 “结果概要” 报告。

6 要使用快照关联机制，请选中 “回放期间保存关联信息” 复选框。可以将每张快照与最初录制的脚本相比较。

7 单击 “确定” 接受设置并关闭 “常规选项” 对话框。

## 使用 VuGen 窗口

创建 Vuser 脚本时，可能需要查看若干个脚本和窗口。可以使用下列 VuGen 功能：

➤ 显示 / 隐藏输出窗口

选择 “视图” > “输出窗口” 可显示和隐藏 VuGen 脚本编辑器下方的 “输出” 窗口。

➤ 显示网格

选择 “视图” > “数据网格” 可显示或隐藏包含结果数据的网格。

➤ 关闭所有窗口

选择 “窗口” > “关闭全部” 可关闭所有打开的窗口。

## 从命令提示符运行 Vuser 脚本

可以从命令提示符或 Windows 的 “运行” 对话框测试 Vuser 脚本，此时不需要使用 Controller 或 VuGen 用户界面。

**要从 DOS 命令行或 “运行” 对话框运行脚本，请执行下列操作：**

1 选择 “开始” > “程序” > “命令提示符”，打开 “命令提示符” 窗口，或者选择 “开始” > “运行”，打开 “运行” 对话框。

2 键入以下命令，然后按 Enter 键：

*loadRunner VuGen 路径* /bin/mdrv.exe -usr *script_name* -vugen_win 0

*script_name* 是 *.usr* 脚本文件的完整路径，例如， *c:\temp\mytest\mytest.usr*。

mdrv 程序将运行单个脚本实例，而不使用用户界面。检查输出文件中的运行时信息。

## 从 UNIX 命令行运行 Vuser 脚本

使用 VuGen 开发基于 UNIX 的 Vuser 时，必须检查录制的脚本是否运行在 UNIX 平台上。要确保脚本正确运行，请执行下列步骤：

1 **从 VuGen 对录制的脚本进行测试。**

从 VuGen 运行录制的脚本，以确保该脚本在基于 Windows 的系统上正确运行。

2 **将 Vuser 脚本文件复制到 UNIX 驱动器。**

将文件传输到本地 UNIX 驱动器。

---

**注意：**如果尚未执行上述操作，请使用 *vu_verify* 来检查 UNIX 计算机上的 Vuser 设置。有关 UNIX Vuser 设置的详细信息，请参阅 《LoadRunner 安装指南》。

---

3 **从 UNIX 命令行测试脚本。**

使用 *run_db_vuser* shell 脚本，从 Vuser 脚本目录中以独立模式运行要测试的脚本：

```
run_db_vuser.sh script_name.usr
```

### 命令行选项：run_db_vuser Shell 脚本

*run_db_vuser* shell 脚本具有下列命令行选项：

**-help**
显示可用的选项。（该选项的前面必须有两条短线。）

**-cpp_only**
仅在脚本上运行 cpp （预处理）。

**-cci_only**
仅在脚本上运行 cci（预编译）以创建扩展名为 .ci 的文件。只有在成功运行 cpp 后才能运行 cci。

**-driver** *driver_path*
使用特定的驱动程序。每个数据库都在 /bin 目录下有自己的驱动程序。例如，位于 /bin 目录中的 CtLib 驱动程序是 *mdrv*。通过该选项，可以指定外部驱动程序。

**-exec_only**
执行 Vuser .ci 文件。只有存在有效的 .ci 文件，该选项才可用。

**-ci** *ci_file_name*
执行特定的 .ci 文件。

**-out** *output_path*
将结果置于特定目录中。

默认情况下， *run_db_vuser.sh* 将在详细模式下运行 **cpp**、 **cci** 和 **exec**。它使用 *LoadRunner 安装目录* /bin 中的驱动程序，并将结果保存在 Vuser 脚本目录下的输出文件中。始终必须指定 *.usr* 文件。如果当前所在目录不是脚本目录，请指定 *.usr* 文件的完整路径。

例如，以下命令行将执行 Vuser 脚本 *test1*，并将输出文件放在 *results1* 目录中。该结果目录必须是现有目录，它不会自动创建：

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

# 将 Vuser 脚本集成到场景中

成功地在独立模式下运行脚本以确认其可用之后，可以将该脚本集成到场景中。场景包含下列信息：

- 要运行脚本的 Vuser
- 要执行脚本的负载生成器

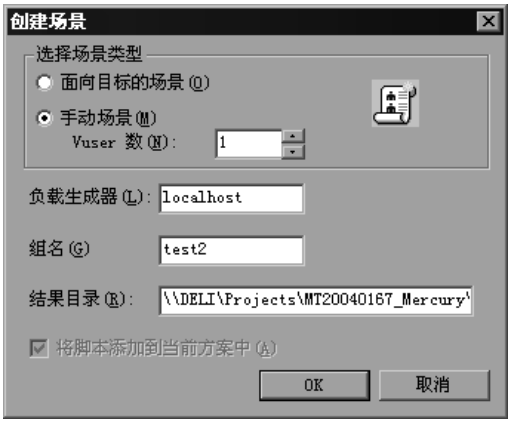
## 使用 Controller

通常是从 LoadRunner Controller 创建场景。也可以从 VuGen 创建一个使用当前 Vuser 脚本的简单场景。详细信息，请参阅 《LoadRunner Controller 用户指南 (Windows)》。

## 使用 VuGen

**要从 VuGen 创建场景，请执行下列操作：**

**1** 选择 “工具” > “创建 Controller 场景”。将打开 “创建场景”对话框。




**2** 选择面向目标的场景或手动场景。

在面向目标的场景中， LoadRunner 将根据指定的目标自动创建场景，而在手动场景中，则需要指定要运行的 Vuser 的数量。

**3** 对于手动场景，输入要执行该脚本的 Vuser 的数量。

**4** 在 “负载生成器”框中，输入要运行 Vuser 的计算机的名称。

5 对于手动场景，具有共同特征的用户将组织成组。在 “组名” 框中为 Vuser 指定一个新组名。

6 对于面向目标的场景，指定 “脚本名”。

7 在 “结果目录” 框中输入要保存结果的位置。

8 如果当前已在 Controller 中打开了一个场景，并希望在该场景中添加脚本，请选中 “将脚本添加到当前场景中” 复选框。如果清除该复选框， LoadRunner 将打开一个含有指定数量的 Vuser 的新场景。

9 单击 “确定”。 VuGen 将在 Vuser 视图中打开 Controller。

10 如果对 Controller 进行配置，使其在共享网络驱动器上保存脚本，则可能需要执行路径转换。

详细信息，请参阅 《LoadRunner Controller 用户指南 (Windows)》。

# 12

# 使用 TestDirector 管理脚本

通过将 VuGen 与 TestDirector 进行集成，可以使用 TestDirector 来管理 Vuser 脚本。

本章描述：

- 与 TestDirector 连接以及与其断开连接
- 打开 TestDirector 项目中的脚本
- 将脚本保存到 TestDirector 项目

## 关于使用 TestDirector 管理脚本

VuGen 可与 TestDirector 一起使用，TestDirector 是 Mercury Interactive 的基于 Web 的测试管理工具。 TestDirector 提供了存储和检索 Vuser 脚本、场景以及收集结果的有效方法。您可以在 TestDirector 项目中存储脚本，并将它们组织成唯一的组。

为了使 VuGen 可以访问 TestDirector 项目，必须将其连接到安装有 TestDirector 的 Web 服务器上。您可以连接本地或远程 Web 服务器。

有关使用 TestDirector 的详细信息，请参阅 《TestDirector 用户指南》。

## 与 TestDirector 连接以及与其断开连接

如果同时使用 VuGen 和 TestDirector，则 VuGen 可以与 TestDirector 项目通信。在测试过程中，您可以随时将 VuGen 与 TestDirector 项目连接或断开连接。

## 将 VuGen 连接到 TestDirector

连接过程分两个阶段。首先，将 VuGen 连接到本地或远程 TestDirector Web 服务器。该服务器用于处理 VuGen 与 TestDirector 项目之间的连接。

然后，选择希望 VuGen 访问的项目。该项目存储了您要测试的应用程序的脚本。注意， TestDirector 项目有密码保护，因此您必须提供用户名和密码。

**要将 VuGen 连接到 TestDirector，请执行下列操作：**

**1** 在 VuGen 中，选择 “工具” > “TestDirector 连接”。将打开 “TestDirector 连接”对话框。

**2** 在 “服务器”框中，键入安装有 TestDirector 的 Web 服务器的 URL 地址。

---

**注意：**您可以选择可通过局域网 (LAN) 或广域网 (WAN) 访问的 Web 服务器。

---

**3** 单击 “连接”。建立与服务器的连接后，服务器名将以只读格式显示在 “服务器”框中。

**4** 从 “项目连接”部分的 “项目”框中，选择 TestDirector 项目。

**5** 在 “用户名”框中键入用户名。

6 在 “密码”框中键入密码。

7 单击 “连接”，将 VuGen 连接到选定的项目。

建立与选定项目的连接后，项目名将以只读格式显示在 “项目”框中。

8 要在启动时自动重新连接到 TestDirector 服务器和选定的项目，请选中 “启动时重新连接”复选框。

9 如果选中了 “启动时重新连接”复选框，可以保存指定的密码以便在启动时重新连接。选中 “保存密码以便在启动时重新连接”复选框。

如果没有保存密码，当 VuGen 在启动时连接到 TestDirector 时，系统会提示您输入密码。

10 单击 “关闭”以关闭 “TestDirector 连接”对话框。

状态栏将指明 VuGen 当前已连接到 TestDirector 项目。

### 断开 VuGen 与 TestDirector 的连接

可以断开 VuGen 与选定 TestDirector 项目和 Web 服务器的连接。

**要断开 VuGen 与 TestDirector 的连接，请执行下列操作：**

1 在 VuGen 中，选择 “工具” > “TestDirector 连接”。将打开 “TestDirector 连接”对话框。

2 要断开 VuGen 与选定项目的连接，请在 “项目连接”部分中单击 “断开连接”。

3 要断开 VuGen 与选定服务器的连接，请在 “服务器连接”部分中单击 “断开连接”。

4 单击 “关闭”以关闭 “TestDirector 连接”对话框。

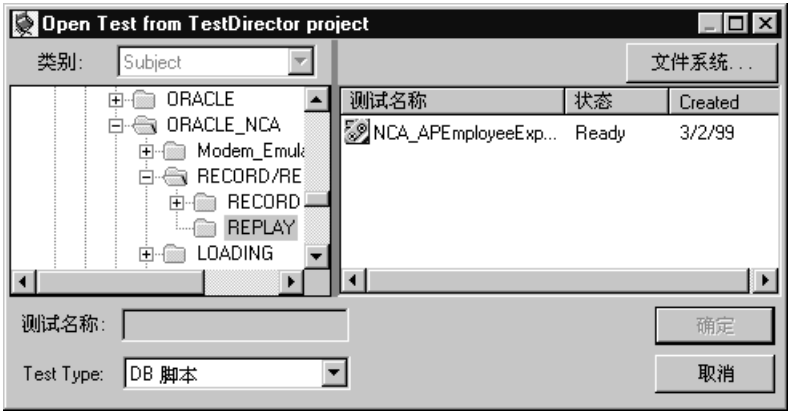
## 打开 TestDirector 项目中的脚本

当 VuGen 连接到 TestDirector 项目时，您可以从 TestDirector 中打开脚本。请根据测试在测试计划树中的位置对其进行查找，而不是按照其在文件系统中的实际位置进行查找。

要从 TestDirector 项目打开脚本，请执行下列操作：

1 连接到 TestDirector 服务器 （请参阅第 142 页的 “将 VuGen 连接到 TestDirector”）。

2 在 VuGen 中，选择 “文件” > “打开” 或单击 “打开文件” 按钮。将打开 “从 TestDirector 项目打开测试” 对话框，并显示测试计划树。




要从文件系统直接打开脚本，请单击 “文件系统” 按钮。将打开 “打开测试” 对话框。（在 “打开测试” 对话框中，通过单击 “TestDirector” 按钮可以返回 “从 TestDirector 项目打开测试” 对话框）。

3 在测试计划树中单击相关主题。要展开该树并查看子级别，请双击关闭的文件夹。要折叠该树，请双击打开的文件夹。

注意，选择某个主题时，属于该主题的脚本将出现在 “测试名称” 列表中。

4 从 “测试名称” 列表中选择脚本。该脚本将出现在只读的 “测试名称” 框中。

5 单击 “确定” 打开脚本。 VuGen 将加载该脚本。脚本的名称将出现在 VuGen 的标题栏中。“设计” 选项卡将显示测试计划树中的所有脚本。

---

**注意：**您还可以从 “文件” 菜单中最近使用过的文件列表中打开脚本。如果您选择了位于 TestDirector 项目中的脚本，但 VuGen 当前并未连接到该项目，则将打开 “TestDirector 连接” 对话框。请输入您的用户名和密码以登录到该项目，然后单击 “确定”。

---

# 将脚本保存到 TestDirector 项目

当 VuGen 连接到 TestDirector 项目时，可以在 VuGen 中新建脚本并将其直接保存到您的项目中。要保存脚本，请为其赋予一个描述性名称，并将其与测试计划树中的相关主题关联起来。这有助于您跟踪为每个主题创建的脚本，以及快速查看测试计划的进度和创建进度。

**要将脚本保存到 TestDirector 项目，请执行下列操作：**

1 连接到 TestDirector 服务器 （请参阅第 142 页的 “将 VuGen 连接到 TestDirector”）。

2 在 VuGen 中，选择 “文件” > “另存为”。将打开 “将测试保存到 TestDirector 项目”对话框，并显示测试计划树。

要将脚本直接保存到文件系统中，请单击 “文件系统”按钮。将打开 “保存测试”对话框。（在 “保存测试”对话框中，通过单击 “TestDirector”按钮可以返回 “将测试保存到 TestDirector 项目”对话框）。

3 在测试计划树中选择相关主题。要展开该树并查看子级别，请双击关闭的文件夹。要折叠子级别，请双击打开的文件夹。

4 在 “测试名称”框中输入脚本的名称。使用描述性名称有助于方便地标识脚本。

5 单击 “确定”保存脚本并关闭对话框。

下次启动 TestDirector 时，将在 TestDirector 的测试计划树中显示新脚本。

# 第 III 部分

# 使用 Java 语言协议

使用引用 RMI-Java、 CORBA-Java、 EJB 和 Jacada 类型的 Java 语言协议。对于每种提及的协议，请参阅相应的部分。本部分包含适用于所有 Java Vuser 类型的信息。

# 13

# 录制 Java 语言 Vuser 脚本

通过 VuGen，可以使用 CORBA、RMI、EJB 或 Jacada 等协议来录制用 Java 编写的应用程序或小程序。也可以使用 VuGen 的导航工具向脚本中添加任意方法。

本章描述：

- 录制入门
- 了解 Java 语言 Vuser 脚本
- 作为程序包的一部分运行脚本
- 查看 Java 方法
- 手动插入 Java 方法
- 配置脚本生成设置

***下列信息适用于 CORBA-Java、RMI-Java、EJB 和 Jacada Vuser 脚本。***

## 关于录制 Java 语言 Vuser 脚本

使用 VuGen 可以录制 Java 应用程序或小程序。VuGen 将创建一个完全使用 Java 编写的脚本，该脚本通过 LoadRunner 特有的 Java 函数进行了增强。录制完后，可以使用 JDK 库或自定义类通过标准 Java 代码来增强或修改该脚本。

准备好脚本后，可以在 VuGen 中以独立模式运行它。 Sun 的标准 Java 编译器 *javac.exe* 将检查脚本中是否有错误，并对其进行编译。确认脚本可正常运行后，可将该脚本合并到 LoadRunner 场景中。

在通过录制和手动增强创建脚本时，与 Java Vuser 脚本相关的所有规则和限制都适用。此外，该脚本中使用的所有具体的类都必须位于执行 Vuser 的计算机上，并在 *classpath* 环境变量中加以指定。有关函数语法和系统配置的重要信息，请参阅第 24 章 “编写 Java 脚本”。

注意，录制过程中从 VuGen 加载小程序或应用程序所需要的时间要比单独加载它们（独立于 LoadRunner）长几秒钟。

VuGen 提供了一种工具，通过该工具，可以将创建的 Web Vuser 脚本转换为 Java。详细信息，请参阅第 374 页的“将 Web Vuser 脚本转换为 Java”。

## 录制入门

以下过程概括了如何录制 Java 语言 Vuser 脚本。

**1 确保录制计算机配置正确。**

开始录制之前，确保计算机针对 Java 配置正确。详细信息，请参阅第 24 章“编写 Java 脚本”和自述文件。

**2 创建新的 Vuser 脚本。**

选择一种协议类型（分布式组件、 EJB 或中间件），然后选择所需的 Vuser 类型。

**3 为该脚本设置录制参数和选项。**

为小程序或应用程序指定参数，例如工作目录和路径。还可以设置 JVM、序列化、关联、录制器和调试录制选项。详细信息，请参阅第 14 章“设置 Java 录制选项”。

**4 录制典型的用户操作。**

开始录制脚本。在小程序或应用程序中执行典型的操作。 VuGen 将录制您所执行的操作并生成 Vuser 脚本。

**5 增强 Vuser 脚本。**

添加 LoadRunner 特有的函数以增强 Vuser 脚本。详细信息，请参阅第 24 章“编写 Java 脚本”。可以使用内置的 Java 函数导航器。详细信息，请参阅第 153 页的“查看 Java 方法”。

**6 参数化 Vuser 脚本。**

将录制的常量替换为参数。可以参数化整个字符串，也可以参数化字符串的一部分。注意，可以为具有多个变量的函数定义多个参数。详细信息，请参阅第 7 章“定义参数”。

**7 为脚本配置运行时设置。**

为 Vuser 脚本配置运行时设置。运行时设置用于定义脚本执行的运行时特征。有关 Java 特定的运行时设置，请参阅第 16 章 “配置 Java 运行时设置”。

**8 保存和运行 Vuser 脚本。**

从 VuGen 运行该脚本，并查看执行日志中的运行时信息。详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

有关录制过程的详细信息，请参阅对应于各个 Vuser 类型的章节内容。

## 了解 Java 语言 Vuser 脚本

录制会话时，VuGen 将记录对服务器的所有调用，并生成带有 LoadRunner 增强功能的脚本。这些函数描述了应用程序或小程序中的所有操作。该脚本还包含进行正确播放所需的补充代码（例如属性设置）以及命名服务初始化 (JNDI)。

录制的脚本由下列三部分组成：

- 导入
- 代码
- 变量

“导入”部分位于脚本的开头。这一部分包含对编译该脚本所需的所有程序包的引用。“代码”部分包含 Actions 类以及在 *init*、*actions* 和 *end* 方法中录制的代码。“变量”部分位于 *end* 方法之后，包含代码中使用的变量的所有类型声明。

录制完后，可以修改脚本中的函数，也可以添加更多的 Java 或 LoadRunner 函数来增强该脚本的功能。注意，如果希望将 Java Vuser 作为线程运行，则添加到脚本中的 Java 代码必须是线程安全的。有关函数语法的详细信息，请参阅《Online Function Reference》（“帮助” > “函数参考”）。此外，还可以对该脚本进行修改，使其作为另一个程序包的一部分运行。详细信息，请参阅第 302 页的 “将脚本作为包的一部分编译和运行”。

## 作为程序包的一部分运行脚本

本节不适用于 Jacada 类型的脚本。

创建或录制 Java 脚本时，可能需要使用某些类中的方法，而在这些类中，方法或类是受到保护的。尝试编译这种脚本时，会显示一条编译错误，指明方法不可访问。

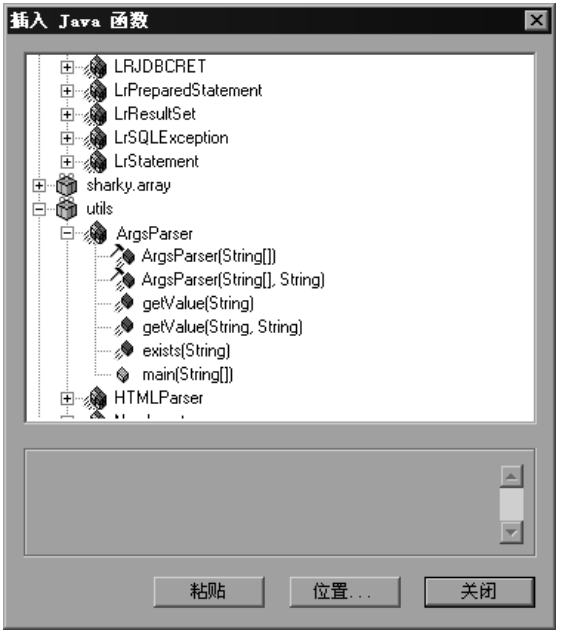
要使用受保护的方法，可向包含要使用的方法的程序包中添加 Vuser。在脚本的开头，添加以下行：

```
package a.b.c;
```

其中 a.b.c 表示目录层次结构。VuGen 将在用户目录中创建 a/b/c 目录层次结构，并在该目录层次结构中编译 *Actions.java* 文件，从而使其成为程序包的一部分。注意， **package** 语句不是录制的，需要手动插入。

# 查看 Java 方法

VuGen 提供了一个导航器，通过该导航器可以查看应用程序程序包中的所有 Java 类和方法。




要向脚本中插入类或方法，请选择该类或方法，然后将其粘贴到脚本中。有关分步说明，请参阅第 154 页的 “手动插入 Java 方法”。

对话框的下半部分显示 Java 对象及其原型、返回值和路径的描述。在以下示例中，描述部分说明了 *deserialize* 方法是一个公共静态方法，该方法接收两个参数：一个字符串和一个整数。该方法返回 java.lang.object 并引发异常。

```
public static synchronized java.lang.Object deserialize (java.lang.String,
int) throws Exception
```

下表描述了表示各种 Java 对象的图标：

| 图标 | 项目 | 示例 |
|---|---|---|
| | 程序包 | java.util |
| | 类 | public class Hashtable extends java.util.Dictionary implements java.lang.Cloneable, java.io.Serializable |
| | 接口类（灰色图标） | public interface Enumeration |
| | 方法 | public synchronized java.util.Enumeration keys () |
| | 静态方法（黄色图标） | public static synchronized java.util.TimeZone getTimeZone |
| | 构造函数方法 | public void Hashtable () |

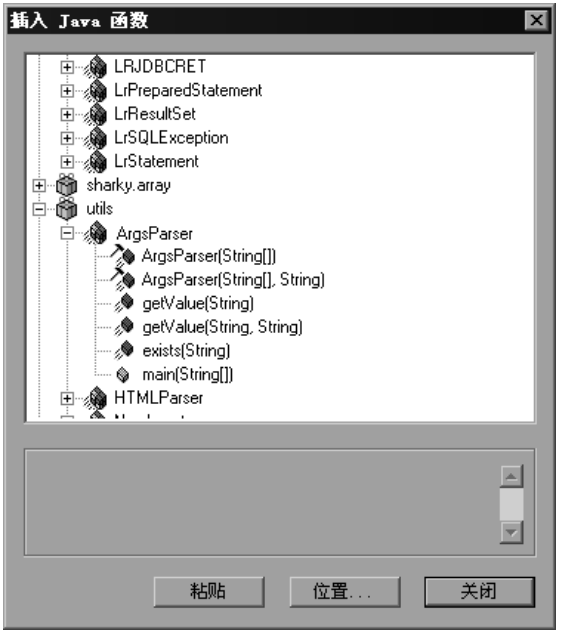
# 手动插入 Java 方法

使用 Java 函数导航器可以查看各种 Java 函数并将这些函数添加到脚本中。以下部分适用于 EJB 测试、 RMI-Java 和 CORBA-Java Vuser。可以通过修改配置文件来自定义函数生成设置。详细信息，请参阅第 157 页的 “配置脚本生成设置”。

**要插入 Java 函数，请执行下列操作：**

1 在脚本中的所需插入点处单击。粘贴函数时， VuGen 即会将该函数放置在光标所在位置上。

**2** 选择 “插入” > “插入 Java 函数”。将打开 “插入 Java 函数”对话框。



**3** 单击“位置”。将打开“位置”对话框。默认情况下， VuGen 将列出 CLASSPATH 环境变量中定义的路径。

**4** 单击“浏览”向列表中添加其他路径或存档文件。要添加路径，请选择“浏览”> “文件夹”。要添加存档文件（*jar* 或 *zip*），请选择“浏览”> “文件”。选择文件夹或文件后， VuGen 即会将其插入到“添加位置”框中。

**5** 单击“添加”将项目添加到列表中。

**6** 针对要添加的每个路径或存档文件，重复步骤 4 和 5。

**7** 选中或清除列表中每个项目左侧的复选框。如果某个项目已选中，则其成员将在 Java 类导航器中列出。

**8** 单击“确定”关闭“位置”对话框并查看可用的程序包。

**9** 单击导航器中每个项目左侧的加号或减号可以展开或折叠树。

**10** 选择对象并单击“粘贴”。 VuGen 即会将该对象放置在脚本中光标所在的位置上。要将类中的所有方法粘贴到脚本中，请选择该类，然后单击“粘贴”。

**11** 针对所需的所有方法或类，重复上述步骤。

**12** 修改方法的参数。如果将脚本生成设置 **DefaultValues** 设置为 *true*，则可以使用 VuGen 插入的默认值。如果将 **DefaultValues** 设置为 *false*，则必须为要插入到脚本中的所有方法添加参数。

此外，还需要修改任何返回值。例如，如果脚本生成以下语句 "(String)=LavaVersion.getVersionId();"，则请将 *(String)* 替换为字符串类型的变量。

**13** 向脚本中添加任何必要的语句 （例如导入语句或 LoadRunner Java 函数 （如第 24 章 "编写 Java 脚本" 中所述））。

**14** 保存该脚本并在 VuGen 中运行它。

## 配置脚本生成设置

可以在下列区域中自定义导航器如何向脚本中添加方法：

- 类名路径
- 自动事务
- 默认参数值
- 类粘贴

要查看配置设置，请打开 LoadRunner 的 dat 目录中的 *jquery.ini* 文件。

```
[Display]
FullClassName=False

[Insert]
AutoTransaction=False
DefaultValues=True
CleanClassPaste=False
```

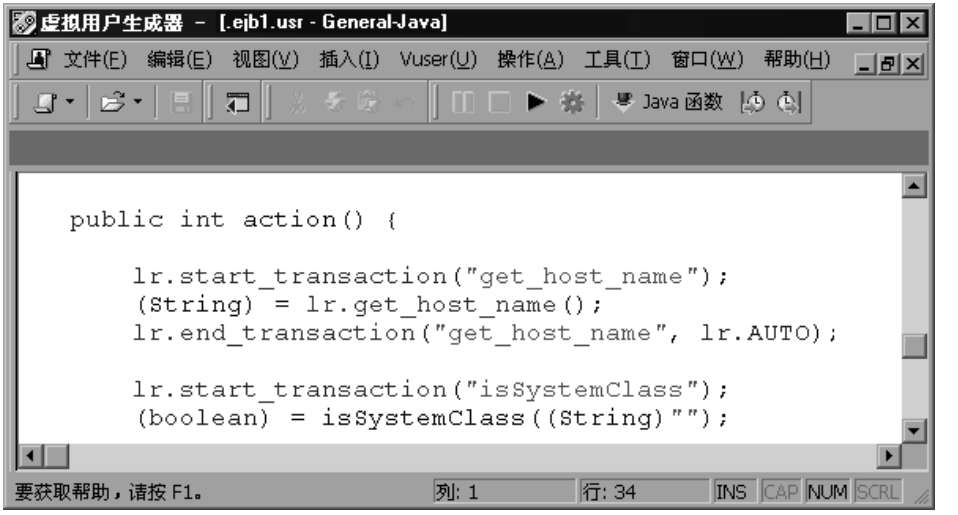
## 类名路径

**FullClassName** 选项在 Java 函数导航器中显示完整的程序包名和类名。此选项不影响向脚本中添加函数的方式， 它只影响在导航器中显示类的方式。 默认情况下，该选项设置为 false。如果程序包中有许多类，并且无法同时查看程序包名和类名，则应该启用该选项。

| 已启用 FullClassName | 已禁用 FullClassName |
| --- | --- |
| | |

## 自动事务

**AutoTransaction** 设置为所有方法创建 LoadRunner 事务。启用该选项后，VuGen 会自动将所有 Java 方法括在 **lr.start_transaction** 和 **lr.end_transaction** 函数之间。这样可以单独跟踪每个方法的性能。默认情况下，该选项是禁用的。



### 默认参数值

**DefaultValues** 设置包含粘贴到脚本中的所有方法的默认值。该选项在默认情况下是启用的，并为所有对象插入一个空值。如果禁用该选项，则必须手动为脚本中的所有函数插入参数值。下表显示了 **DefaultValues** 标志启用和禁用时的状况。

| 已启用 DefaultValues | 已禁用 DefaultValues |
|---|---|
| lr.message((String)"");<br>lr.think_time((int)0);<br>lr.enable_redirection((boolean)false);<br>lr.save_data((byte[])null, (String)""); | lr.message((String));<br>lr.think_time((int));<br>lr.enable_redirection((boolean));<br>lr.save_data((byte[]), (String)); |

### 类粘贴

可以通过 **CleanClassPaste** 设置粘贴类，以便使用构造函数返回的实例，并使用默认值作为参数重新进行编译，而不需要导入语句。使用该选项，很可能无需进一步修改即可运行脚本。如果禁用该选项 （默认），则可能需要手动定义参数，并在其中包含导入语句。注意，仅当将完整的类粘贴到脚本中时 （而不是仅粘贴单个方法），该设置才会有效。

以下代码段显示了在启用 **CleanClassPaste** 选项的情况下粘贴到脚本中的 *toString* 方法。

```
_class.toString();
    // Returns:java.lang.String
```

下面显示了在禁用 **CleanClassPaste** 选项的情况下粘贴同一方法时的情形：

```
(String) = toString();
```

下一个代码段显示了在启用 **CleanClassPaste** 选项的情况下粘贴在脚本中的 *NumInserter* 构造函数方法。

```
utils.NumInserter _numinserter = new utils.NumInserter
            ((java.lang.String)"", (java.lang.String)"", (java.lang.String)""…);
// Returns:void
```

下面显示了在禁用 **CleanClassPaste** 选项的情况下粘贴同一方法时的情形：

```
new utils.NumInserter((String)"", (String)"", (String)"",...);
```

# 14

# 设置 Java 录制选项

通过 VuGen，可以控制录制 CORBA、 RMI 或 EJB 应用程序的方式。可以使用默认的录制选项，也可以根据具体需要自定义这些选项。

本章描述：

- Java 虚拟机 (JVM) 录制选项
- 设置类路径录制选项
- 录制器选项
- 序列化选项
- 关联选项
- 调试选项

***下列信息适用于 CORBA-Java、 RMI-Java 和 EJB Vuser 脚本。***

## 关于设置 Java 录制选项

使用 VuGen 可以录制 CORBA （公共对象请求代理体系结构）或 RMI （远程方法调用） Java 应用程序或小程序。有关录制 EJB 测试的信息，请参阅第 46 章 “执行 EJB 测试”。

录制前，可以通过 VuGen 为 Java 虚拟机 (JVM) 和代码生成阶段设置录制选项。设置录制选项并不是强制的；如果不设置这些选项， VuGen 将使用默认值。

本章描述的选项已经在前面通过修改 *mercury.properties* 文件进行了介绍。

可以在下列区域中设置录制选项：

- Java 虚拟机 (JVM) 录制选项
- 设置类路径录制选项
- 录制器选项
- 序列化选项
- 关联选项
- 调试选项

## Java 虚拟机 (JVM) 录制选项

**Java VM** 选项指示录制 Java 应用程序时使用的其他参数。

录制 Vuser 时， VuGen 会自动将 *Xbootclasspath* 变量设置为默认参数。如果使用该对话框将 *Xbootclasspath* 设置为其他参数，则它将使用所设置的命令参数，而不是默认参数。

还可以指示 VuGen 在 *Xbootclasspath* 之前添加类路径 （预设该字符串），以创建单个类路径字符串。

默认情况下， VuGen 会在录制过程中使用传统的虚拟机。还可以指示 VuGen 使用其他虚拟机 (Sun Java Hotspot VM)。

**要设置 Java 虚拟机录制选项，请执行下列操作：**

1 在 “开始录制”对话框中单击 “选项”。在 “录制选项”树中选择 “Java 环境设置:Java 虚拟机”节点。

2 在 “其他 VM 参数”框中，列出 Java 命令行参数。这些参数可以是任意 Java VM 参数。常见参数有调试标志 (*-verbose*) 或内存设置 （*-ms* 和 *-mx*）。有关 Java VM 标志的详细信息，请参阅 JVM 文档。此外，还可以通过 *-D* 标志的形式将属性传递给 Java 应用程序。

VuGen 会自动将 *-Xbootclasspath* 变量 （适用于 JDK 1.2 和更高版本）设置为默认参数。如果将 *-Xbootclasspath* 参数指定为其他参数， VuGen 将使用所设置的参数，而不是默认参数。

3 要在回放时使用相同的 “其他虚拟机参数”，请选中 “回放时使用指定的其他虚拟机参数”复选框。

4 要使用传统的 VM，请选中 “使用传统 Java 虚拟机”复选框 （默认）。要使用其他虚拟机 （Sun Java HotSpot），请清除该复选框。

5 要在 *Xbootclasspath* 之前添加类路径 （预设该字符串），请选中 “将 CLASSPATH 预设为 -Xbootclasspath 参数”复选框。

6 单击 “确定”关闭对话框并开始录制。

## 设置类路径录制选项

通过 “类路径”部分可以指定在系统的 classpath 环境变量中未包括的其他类的位置。要运行 Java 应用程序并确保正常录制，可能需要用到这些类。

可以在计算机或网络中浏览所需的类，并在特定的测试中禁用这些类。还可以通过更改类路径项的顺序对类路径项进行处理。

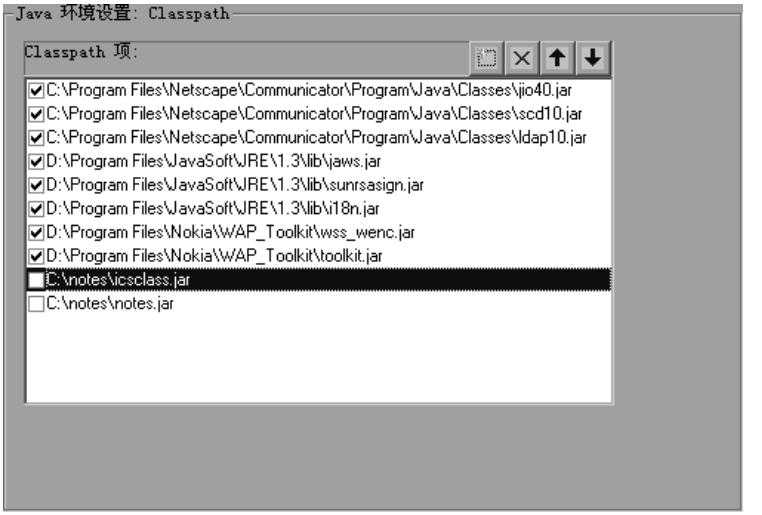
**要设置类路径录制选项，请执行下列操作：**

1 在 “开始录制”对话框中单击 “选项”。在 “录制选项”树中选择 “Java 环境设置 : 类路径”节点。

2 向列表中添加类路径：

单击 “添加类路径”按钮。 VuGen 会在类路径列表中添加一个新行。

键入该类的 *jar*、 *zip* 或其他存档文件的路径和名称。或者单击字段右侧的 “浏览” 按钮，查找所需的文件。 VuGen 会向类路径列表中添加这一新位置，并将其标记为启用状态。



**3** 要永久删除某一项，请选择该项并单击 “删除” 按钮。

**4** 要在特定的测试中禁用某个类路径项，请清除该项左侧的复选框。

**5** 要在列表中下移某一项，请选择该项并单击向下箭头。

**6** 要在列表中上移某个类路径项，请选择该项并单击向上箭头。

**7** 单击 “确定” 关闭对话框并开始录制。

## 录制器选项

“录制器” 选项为 VuGen 提供了生成 Vuser 脚本的指南。可以在下列区域中设置选项：

- 常规选项
- 录制日志

- 样式选项
- 字节格式选项

## 常规选项

**录制返回值**：在脚本中生成一条注释，以指示每次调用的返回值。（默认情况下是禁用的）

**录制进度消息**：在每次调用之前录制一个 lr.log_message 函数，使您可以了解回放的进度。（默认情况下是禁用的）

**录制思考时间**：在脚本中录制思考时间并包含思考时间函数 lr.think_time。（默认情况下是启用的）

**录制异常处理**：出现异常时，调用将被置于 “try-catch” 块中。（默认情况下是启用的）

**插入功能检查**：插入验证代码，将回放时收到的返回值与录制期间应生成的返回值进行比较。该选项只适用于基元返回值。（默认情况下是禁用的）

**使用 LR- API**：在脚本中包含 LR API 函数。如果要在 VuGen 之外使用该脚本，请禁用该选项以删除诸如思考时间之类的所有 LR API 函数以及其他常量。（默认情况下是启用的）

**输出重定向：**将 Java 应用程序的 *Stdout* 和 *Stderr* 输出重定向到文件。（默认情况下是禁用的）

**扩展名列表：**支持的所有扩展名的列表。每个扩展名都有其各自的挂钩文件。要指定其他扩展名，请将其添加到默认扩展列表中。如果向列表中添加扩展名，请确保其挂钩文件位于 Vuser 脚本中。默认扩展名为 JNDI、JMS 和 EJB。

**使用 _JAVA_OPTION 标志：**强制 JVM 1.2 及更高版本使用包含所需 JVM 参数的 _JAVA_OPTION 环境变量。（默认情况下是禁用的）

### 录制日志

**生成录制日志：**生成录制日志，并显示在 “输出”窗口的 “录制”选项卡中。如果禁用该选项，性能会有所提高，但在录制期间不会向 “输出”窗口发送任何信息。（默认情况下是启用的）

**生成变量信息：**将变量的内部值写入录制日志中。如果启用该选项，性能会有所降低。（默认情况下是禁用的）

**详细级别：**在录制数组类型参数或返回值时，要在日志中显示的数组元素的数目。默认级别为 5。

### 样式选项

**使用块语义：**通过用花括号将调用括起来，将每个调用放在一个单独的范围中。如果该选项被禁用，则会用花括号将整个 Action 方法 （而不是每个调用）括起来。（默认情况下是禁用的）

**带下划线的变量名：**在脚本中生成的所有变量的前面加上下划线前缀。要防止与同名程序包相冲突，这一点是必需的。（默认情况下是启用的）

**最大行长度：**所录制行的最大长度。如果录制的行超过该值，则将被截断。VuGen 采用智能截断方式，以防止破坏代码的一致性，例如引号或函数参数。默认值为 1000 个字符。最大长度为 30000 个字符。

**最大 Action 长度：**Action 方法的最大长度。默认值为 3000 个字符。如果 Action 方法超过该值， VuGen 会将其分解为较小的 Action 方法。

**注释掉包含以下字符串的行：**注释掉脚本中包含指定字符串之一的所有行。要指定多个字符串，请用逗号分隔各项。默认情况下，包含标有 < 未定义 > 字样的字符串的所有行都会被注释掉。

**删除包含以下字符串的行**：从脚本中删除包含指定字符串之一的所有行。要指定多个字符串，请用逗号分隔各项。在针对特定测试目标自定义脚本时，该功能会很有用。

### 字节格式选项

**字节作为字符**：通过必要的转换显示可读字符，而不使用字节或十六进制格式。（默认情况下是启用的）

**隐式转换**：指示 VuGen 自动对所有调用进行转换。如果启用该选项，则转换将不被添加到录制的调用中，而是由编译器对转换进行隐式处理。如果禁用该选项， VuGen 会将转换添加到调用中，从而使脚本变长。（默认情况下为 false）

**不可读字符串作为字节**：将包含不可读字符的字符串表示为字节数组。该选项适用于作为参数传递给调用的字符串。（默认情况下为 true）

**字节数组格式**：脚本中字节数组的格式：常规、展开的序列化对象或未展开的序列化对象。在录制很长的字节数组时，请使用其中的某个序列化对象选项。默认值为 “常规”。

**忽略系统属性**：录制 EJB 属性时筛选出指定的系统属性。

**要设置 Java 录制器选项，请执行下列操作：**

1. 在 “开始录制”对话框中单击 “选项”，并选择 “录制属性 : 录制器选项”节点。
2. 设置所需的选项。对于带有复选框的选项，选中或清除该选项旁边的复选框。对于要求输入数字或字符串的选项，键入所需的值。
3. 要将所有选项设置为默认值，请单击 “使用默认值”。
4. 单击 “确定”关闭对话框并开始录制。

# 序列化选项

通过 “序列化”选项，可以控制元素的序列化方式。有关序列化的概述，请参阅第 180 页的 “使用序列化机制”。可使用下列选项：

**展开序列化对象**：以 ASCII 格式展开序列化对象。通过该选项，可以查看对象的 ASCII 值，以便进行参数化。（默认情况下是启用的）

**限制对象大小 （字节）**：将可序列化的对象限制为指定的值。大小超过该值的对象将不在脚本中提供 ASCII 表示形式。默认值为 3072。

**序列化分隔符**：指示在对象的 ASCII 表示形式中用于分隔元素的分隔符。VuGen 将只参数化这些分隔符中的字符串。默认分隔符为 “#”。

**展开数组**：以 ASCII 格式展开序列化对象中的数组元素。如果禁用该选项，而对象包含数组，则对象将不被展开。默认情况下，该选项是启用的，即所有反序列化的对象将被完全展开。

**限制数组项数**：指示录制器不打开超过指定元素数的数组。默认值为 200。

**启用存根序列化**：序列化不关联的存根对象，否则这些对象将为 < 未定义 >。在新的服务器上下文中回放该代码时，可能要求重新录制。（默认情况下是禁用的）

### 调试选项

**序列化调试消息：**提供序列化机制中的调试输出。（默认情况下是禁用的）

**显示序列化异常：**在日志中显示所有序列化异常。（默认情况下是禁用的）

**要设置序列化选项，请执行下列操作：**

1. 在 “开始录制”对话框中单击 “选项”，并选择 “录制属性 : 序列化选项”节点。
2. 设置所需的选项。
3. 要将所有选项设置为默认值，请单击 “使用默认值”。
4. 单击 “确定”关闭对话框并开始录制。

## 关联选项

通过 “关联”选项，可以指示 VuGen 是否应执行自动关联，并控制其深度。有关关联的信息，请参阅第 15 章 “关联 Java 脚本”。可使用下列选项：

**关联字符串：**关联所有要求关联的字符串。如果该选项是禁用的， VuGen 会在脚本中用引号将这些字符串括起来。（默认情况下是禁用的）

**关联字符串数组：**关联字符串数组中的文本 （默认情况下是启用的）

**高级关联**：启用 CORBA 容器构造和数组中的深度关联。（默认情况下是启用的）

**关联级别**：指示深度关联的级别，即要扫描的内部容器的数量。（默认情况下为 15）

**关联集合类型**：关联 JDK 1.2 及更高版本的 Collection 类中的对象。（默认情况下是禁用的）

**关联调试级别**：将与关联相关的调试信息发送到日志中。可以指定 0 到 5 之间的值。（默认情况下为 0，表示无关联调试信息）

**要设置关联选项，请执行下列操作：**

1. 在 “开始录制”对话框中单击 “选项”，并选择 “录制属性 : 关联选项”节点。
2. 启用所需的选项，或者如果选项要求输入值，则输入所需的值。
3. 要将所有选项设置为默认值，请单击 “使用默认值”。
4. 单击 “确定”关闭对话框并开始录制。

## 调试选项

通过 “调试”选项，可以确定录制期间生成的调试信息的级别。可使用下列选项：

## 常规选项

**启用常规调试选项**：启用常规调试选项：类转储、挂钩调试级别、堆栈跟踪和跟踪支持。如果启用该选项，即使第一个“堆栈跟踪”选项被禁用， VuGen 仍会执行堆栈跟踪。将堆栈跟踪与类转储结合使用，可以确定应用程序中挂钩部分的上下文。跟踪有助于确定放置其他挂钩的位置。（默认情况下是禁用的）

**堆栈跟踪**：在堆栈跟踪中记录所有调用。该设置为每个录制的函数提供了 Java 堆栈跟踪。将该选项与“类转储”结合使用，可以确定应用程序中挂钩部分的上下文。该跟踪有助于解决参数不关联的情况，并可确定放置其他挂钩的位置。注意，启用该选项后，应用程序会变慢 （默认情况下是禁用的）

**堆栈跟踪限制**：堆栈中存储的最大调用数。启用堆栈跟踪后，如果调用数超过指定的值，堆栈跟踪将被截断。默认值为 20 个调用。

**跟踪支持**：跟踪所有主要支持调用，并将其写入 Vuser 目录的 *trace.log* 文件中。（默认情况下是禁用的）

**显示进度窗口**：启用 Mercury 产品的进度窗口。（默认情况下是启用的）

**调试类加载程序**：为非系统的类加载程序支持提供调试输出。（默认情况下是禁用的）

**同步线程**：对于多线程应用程序，指示 VuGen 在不同的线程之间进行同步。（默认情况下是禁用的）

**摘要计算**：生成录制的所有对象的摘要。（默认情况下是禁用的）

**从摘要中排除**：在摘要计算中不包括的对象的列表。

**调试变量**：将 < 未定义 > 变量转换为其类型。此外，变量部分中同时也是接口的每个变量会有一条注释，指示其原始类型。（默认情况下是禁用的）

**指定挂钩**：在调用脚本前插入一个字符串，指示引起该调用的挂钩。这对于捕获冗余录制非常有用。（默认情况下是禁用的）

**指定线程**：在调用脚本前插入一个字符串，指示运行该调用的线程。这对于标识多线程应用程序非常有用。（默认情况下是禁用的）

## 挂钩选项

**挂钩调试级别**：录制器中挂钩调试输出的级别。级别 0 表示不发出调试输出。

**忽略类**：要忽略的类的列表。包含指定字符串的所有类将从挂钩机制中被排除。

**打印输出重定向**：确定要将挂钩机制中的打印输出重定向到何处。选项有控制台、单独的文件或调试文件。默认值为控制台。

**将方法设为公共**：将挂钩的方法设为公共。（默认情况下是禁用的）

**将类设为公共**：将挂钩的类设为公共。（默认情况下是启用的）

**日志类挂钩**：创建一个日志文件，该文件包含挂钩前和挂钩后所有类的字符串表示形式。该选项只应在集中调试中使用，因为它会显著降低性能。（默认情况下是禁用的）

**日志特定类挂钩**：应生成挂钩日志文件的类的列表。如果未指定类，则会记录所有类。

**类转储选项**

**类转储**：将加载的类转储到 Vuser 目录中。（默认情况下是禁用的）

**要转储的类**：挂钩后要转储的类的列表。包含指定字符串之一的所有类都将被转储。如果未指定类，则会转储所有类。

**转储后缀**：追加到转储类名中的后缀。默认后缀为 _DUMP。

**类转储目录**：用于转储类的目录。

**标记类转储**：将所有类转储到单个目录中，并在每个类的前面标记完整的程序包。如果该选项被禁用，则会创建目录层次结构。（默认情况下是禁用的）

**要设置调试选项，请执行下列操作：**

1. 在 “开始录制”对话框中单击 “选项”，并选择 “录制属性 : 调试选项”节点。
2. 启用所需的选项，或者如果选项要求输入值，则输入所需的值。
3. 要将所有选项设置为默认值，请单击 “使用默认值”。
4. 单击 “确定”关闭对话框并开始录制。

# CORBA 选项

下面是 Corba-Java 协议特有的选项。通过这些选项，可以设置 Corba 特定的录制属性以及若干个回调选项。可使用下列选项：

**录制属性**：指示 VuGen 录制与协议相关的系统和自定义属性。默认情况下，该选项是启用的。

**显示 IDL 构造**：显示在作为参数传递给 CORBA 调用时使用的 IDL 构造。默认情况下，该选项是启用的。

**只录制 DLL**：指示 VuGen 只在 DLL 级别上进行录制。默认情况下，该选项是禁用的。

**解析 CORBA 对象**：如果关联在解析 CORBA 对象时失败，则使用其二进制数据重新创建该对象。默认情况下，该选项是禁用的。

### 回调选项

**录制回调连接**：指示 VuGen 为每个回调对象与 ORB 的连接生成一条连接语句。默认情况下，该选项是禁用的。

**调试回调**：允许在回调中生成调试信息。默认情况下，该选项是禁用的。

**要设置 Corba 选项，请执行下列操作：**

1. 在“开始录制”对话框中单击“选项”，并选择“录制属性 :Corba 选项”节点。
2. 启用或禁用所需的选项。
3. 要将所有选项设置为默认值，请单击“使用默认值”。
4. 单击“确定”关闭对话框并开始录制。

# 15

# 关联 Java 脚本

通过 VuGen 的关联功能，可以使用一个语句的结果作为另一个语句的输入来将 Java Vuser 函数链接在一起。

本章描述：

- 标准关联
- 高级关联
- 字符串关联
- 使用序列化机制

***下列信息仅适用于 Corba-Java 和 RMI-Java Vuser 脚本。***

# 关于关联 Java 脚本

包含 Java 代码的 Vuser 脚本通常包含动态数据。录制 Corba 或 RMI Vuser 脚本时，该动态数据会录制到脚本中，但在回放过程中不能重新使用它。如果在运行 Vuser 时遇到错误，请检查脚本中出现错误的位置。多数情况下，可通过关联功能将一个语句的结果作为另一个语句的输入来解决问题。

VuGen 的 Corba 录制器将尝试自动关联所生成的脚本中的语句。它只对 Java 对象执行关联。在录制过程中遇到 Java 基元（字节型、字符型、布尔型、整型、浮点型、双精度型、短整型和长整型）时，出现在脚本中的参数值不会与变量相关。 VuGen 会自动关联所有对象、对象数组及基元数组。注意， Java 数组和字符串也被视为对象。

VuGen 采用多级别关联：标准、增强、字符串。可以通过 “录制”选项来启用或禁用关联。如果前面所述的方法都不适用，则可以使用另一种 “序列化”方法来处理脚本。详细信息，请参阅第 180 页的 “使用序列化机制”。

# 标准关联

标准关联是指在录制过程中对简单对象 （不包括对象数组、矢量和容器结构）执行的自动关联。

如果录制的应用程序调用将返回对象的方法， VuGen 的关联机制就会录制这些对象。运行脚本时， VuGen 会将生成的对象与录制的对象进行对比。如果对象匹配，则使用同一对象。下例显示了两个 Corba 对象：*my_bank* 和 *my_account*。第一个对象 *my_bank* 将被调用；第二个对象 *my_account* 将被关联，并作为代码段最后一行中的参数进行传递：

```
public class Actions {

 // 公共函数：init
    public int init() throws Throwable {

     Bank my_bank = bankHelper.bind("bank", "shunra");
     Account my_account = accountHelper.bind("account","shunra");

     my_bank.remove_account(my_account);
    }
:
}
```

## 高级关联

高级关联或*深度*关联是指在录制过程中对复杂对象 （如对象数组和 Corba 容器结构）执行的自动关联。

深度关联机制将 Corba 结构 （结构、联合、序列、数组、持有者、“任意”等）作为容器进行处理。这样可以引用容器内部的成员、其他对象或不同的容器。每当调用对象或作为参数传递对象时，还会将该对象与容器内部的成员进行比较。

在下例中， VuGen 通过引用数组元素来执行深度关联。 *remove_account* 对象收到一个帐户对象参数。录制时，关联机制将搜索返回的数组 *my_accounts* 并确定其第六个元素应作为参数传递。

```
public class Actions {

  // 公共函数：init
  public int init() throws Throwable {

    my_banks[] = bankHelper.bind("banks", "shunra");
    my_accounts[] = accountHelper.bind("accounts","shunra");

    my_banks[2].remove_account(my_accounts[6]);
  }
:
}
```

以下代码段进一步说明了增强的关联功能。该脚本调用接收到 *address* 类型参数的 *send_letter* 对象。关联机制将在 *my_accounts* 数组的第六个元素中检索内部成员 *address*。

```
public class Actions {

  // 公共函数：init
  public int init() throws Throwable {

     my_banks = bankHelper.bind("bank", "shunra");
     my_accounts = accountHelper.bind("account", "shunra");

    my_banks[2].send_letter(my_accounts[6].address);
   }
:
}
```

## 字符串关联

字符串关联是指将录制的值表示为实际的字符串或变量。如果禁用字符串关联（默认设置），则实际录制的字符串值将在脚本中显式指定。如果启用字符串关联，它将为每个字符串创建一个变量，从而可以在脚本中稍后的某个位置上使用它。

以下代码段启用了字符串关联：从 *get_id* 方法返回的值存储在字符串型变量中，以便在脚本中稍后的某个位置上使用。

```
public class Actions {

  // 公共函数：init
  public int init() throws Throwable {

    my_bank = bankHelper.bind("bank", "shunra");
    my_account1 = accountHelper.bind("account1", "shunra");
    my_account2 = accountHelper.bind("account2", "shunra");

     string = my_account1.get_id();
     string2 = my_account2.get_id();
    my_bank.transfer_money(string, string2);
  }
:
}
```

可以从录制选项的 “关联”选项卡中设置关联方法。

**关联字符串**：录制过程中关联脚本中的字符串。如果禁用该选项，实际录制的值将在脚本中用引号括起来。如果禁用该选项，则会忽略其他所有关联选项。（默认情况下是禁用的）

**关联字符串数组**：录制过程中关联字符串数组中的字符串。如果禁用该选项，则不会关联数组中的字符串，实际的值会直接放在脚本中。（默认情况下是启用的）

**高级关联**：对复杂对象 （如数组及 Corba 容器结构和数组）启用关联。此类关联也称为*深度*关联。（默认情况下是启用的）

**关联级别**：确定深度关联的级别，即要搜索的内部容器的数量。

**关联集合类型**：关联 JDK 1.2 或更高版本的 Collection 类中包含的对象。（默认情况下是禁用的）

## 使用序列化机制

在 RMI （有时是 Corba）中，客户端 AUT 会使用 **java.io.serializable** 接口新建 Java 对象实例。该实例将作为参数进行传递，以便由服务器来调用。以下代码段创建实例 *p* 并将其作为参数进行传递。

```
// AUT 代码：
java.awt.Point p = new java.awt.Point(3,7);
map.set_point(p);
:
```

由于对象不是从先前的调用中返回，因此自动关联机制在此处无效。此时，VuGen 将激活序列化机制并将传递的对象作为参数存储。它会将该信息保存在用户目录下的二进制数据文件中。其他参数则保存为新的二进制数据文件，并依次进行编号。 VuGen 生成以下代码：

```
public class Actions {

 // 公共函数：init
    public int init() throws Throwable {
        java.awt.Point p = (java.awt.Point)lr.deserialize(0, false);
        map.set_point(p);
    }
:
}
```

传递给 **lr.deserialize** 的整数是 Vuser 目录中的二进制数据文件的编号。

要将录制的值参数化，请使用公共的 **setLocation** 方法 （有关信息，请参阅

JDK 函数参考）。以下示例使用 **setLocation** 方法设置对象 *p* 的值。

```
public class Actions {

 // 公共函数：init
    public int init() throws Throwable {
       java.awt.Point p = (java.awt.Point)lr.deserialize(0, false);
       p.setLocation(2,9);
       map.set_point(p);
    }
:
:
}
```

在某些情况下， **setLocation** 的公共方法不适用。为此，可以使用包含 get 或 set 访问器方法的类的 API。如果使用的是不带 get/set 方法或不使用专用方法的 AUT 类，或者如果您不熟悉这些类的 API，则可以使用 VuGen 的内置序列化机制。通过该机制，可以利用 ASCII 形式展开对象并手动参数化该脚本。可以在“录制选项”对话框中启用该机制。（请参阅第 14 章 “设置 Java 录制选项”）。

VuGen 将生成 **lr.deserialize** 方法，该方法用于反序列化数据或将复杂数据结构显示为串联字符串。拆分为各个组成部分的结构比较容易进行参数化。**lr.deserialize** 方法接收两个参数：一个字符串和一个整数。字符串是要在回放时替换的参数值。整数是要加载的二进制文件的编号。

如果通过清除 “展开序列化对象”复选框而选择不展开脚本中的对象，则可以通过向 **lr.deserialize** 方法传递参数来控制序列化机制。第一个参数是一个整数，指明要加载的二进制文件的编号。第二个整数是布尔值：

| | |
|---|---|
| *true* | 使用 VuGen 的序列化机制 |
| *false* | 使用标准的 Java 序列化机制 |

以下代码段显示了生成的脚本，该脚本已启用序列化机制。

```
public class Actions {

 // 公共函数：init
    public int init() throws Throwable {
        _string = "java.awt.Point __CURRENT_OBJECT = {" +
                  "int x = "#5#" +
                  "int y = "#8#" +
        "}";
        java.awt.Point p = (java.awt.Point)lr.deserialize(_string,0);
        map.set_point(p);
    }
:
}
```

字符串值用分隔符分隔。默认分隔符是 “#”。可以在录制选项的 “序列化” 选项卡中更改分隔符。使用分隔符可加快回放过程中分析字符串的速度。

修改字符串时，必须遵循下列规则：

- 不能更改各行的顺序。分析器依次读取各个值 （而不是成员名称）。
- 只能修改两个分隔符之间的值。
- 不能修改对象引用。指定对象引用的目的只是保持内部的一致性。
- “_NULL_” 可以显示为一个值，用于表示 Java 的 null 常数。只能使用字符串类型值来替换它。
- 可以在脚本中的任意位置反序列化对象。例如，可以反序列化 *init* 方法中的所有对象，并使用 *action* 方法中的值。
- 保持对象的内部一致性。例如，如果矢量成员是 *element count*，并且添加了一个元素，则必须修改该元素计数。

在以下代码段中，矢量包含两个元素。

```
public class Actions {

 // 公共函数：init
    public int init() throws Throwable {
       _string = "java.util.Vector CURRENTOBJECT = {" +
                 "int capacityIncrement = "#0#" +
                 "int elementCount = #2#" +
                 "java/lang/Object elementData[] = {" +
                       "elementData[0] = #First Element#" +
                       "elementData[1] = #Second Element#" +
                       "elementData[2] = _NULL_" +
                        ....
                       "elementData[9] = _NULL_" +

                 "}" +
       "}";
       _vector = (java.util.Vector)lr.deserialize(_string,0);
       map.set_vector(_vector);
    }
:
}
```

以下示例更改了一个矢量元素，有个 “_NULL_” 值更改为了 “Third element”。添加新元素后，“elementCount” 成员被修改为 “3”。

```
public class Actions {

 // 公共函数：init
    public int init() throws Throwable {
        _string = "java.util.Vector CURRENTOBJECT = {" +
                  "int capacityIncrement = "#0#" +
                  "int elementCount = #3#" +
                  "java/lang/Object elementData[] = {" +
                       "elementData[0] = #First Element#" +
                       "elementData[1] = #Second Element#" +
                       "elementData[2] = #Third Element#" +
                        ....
                       "elementData[9] = _NULL_" +
                  "}" +
        "}";
        _vector = (java.util.Vector)lr.deserialize(_string,0);
        map.set_vector(_vector);
    }
:
}
```

由于序列化机制比较复杂 （即需要打开对象的 ASCII 表示形式），因此录制时打开大型对象可能会增加生成脚本所需的时间。要缩短该时间，可以指定提高序列化机制性能的标志。

在向脚本中添加 **lr.deserialize** 时，建议将其添加到 *init* 方法 （而不是 *action* 方法）中。由于 VuGen 只反序列化字符串一次，因此这会提高性能。如果它位于 *action* 方法中，则 VuGen 将在每次循环时反序列化字符串。

以下列表显示了可在录制选项的“序列化”选项卡中设置的选项。

- 序列化分隔符
- 展开序列化对象
- 展开数组
- 限制数组项数
- 忽略序列化对象

有关录制选项的完整信息，请参阅第 14 章“设置 Java 录制选项”。

# 16

# 配置 Java 运行时设置

录制 Java Vuser 脚本之后，需要配置 Java 虚拟机的运行时设置。

本章描述：

- 指定 JVM 运行时设置
- 设置运行时类路径选项

***下列信息适用于 Java、EJB 测试、Corba-Java 以及 RMI-Java 类型 Vuser。***

## 关于 Java 运行时设置

开发 Java Vuser 脚本之后，需要设置 Java VM （虚拟机）的运行时设置。通过这些设置可以设置其他路径和参数，以及确定运行模式。

通过 “运行时设置”对话框中的 “Java 虚拟机”选项来设置与 Java 相关的运行时设置。

要显示 “运行时设置”对话框，请单击 VuGen 工具栏上的 “运行时设置”按钮。还可以从 LoadRunner Controller 中修改运行时设置。在 “Controller”窗口中，选定要修改其设置的脚本，然后单击 “运行时设置”按钮。

本章只讨论适用于 Java 类型 Vuser （Java、 EJB 测试、 Corba-Java 和 RMI-Java）的运行时设置。有关适用于所有 Vuser 的运行时设置的信息，请参阅第 9 章 “配置运行时设置”。

# 指定 JVM 运行时设置

在 Java 虚拟机部分中，可以提供有关 Java 虚拟机设置的信息。可以使用下列设置：

**其他虚拟机参数**：输入由虚拟机使用的任何可选参数。

**使用 -Xbootclasspath 虚拟机参数**：运行 Vuser 脚本时， VuGen 将自动设置 *Xbootclasspath* 变量。除了在 *Xbootclasspath* 中定义的参数以外，还可以使用该对话框指定其他参数。如果指定了其他虚拟机参数以便进行录制，则可以指示 VuGen 保存这些参数，然后在回放期间使用这些参数。

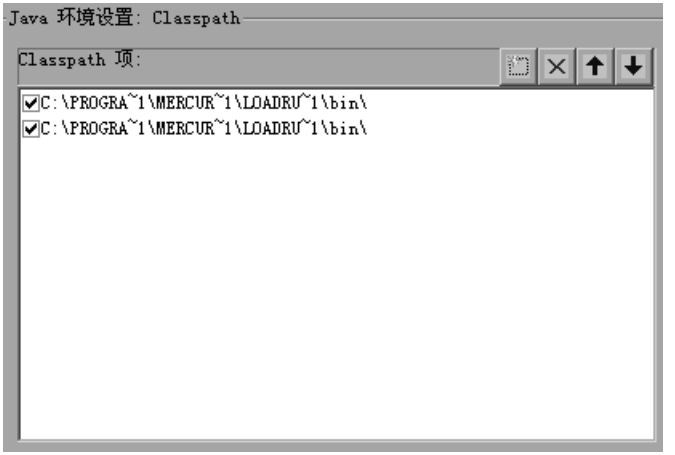
**要设置 Java VM 运行时设置，请执行下列操作：**

1. 选择 “Vuser” > “运行时设置”，然后在 “运行时设置”树中选择 “Java 环境设置 : Java 虚拟机”节点。
2. 在 “其他虚拟机参数”框中，输入由负载生成器计算机使用的任何可选参数。
3. 要用 *-Xbootclasspath/p* 选项回放，请选择 “使用 -Xbootclasspath 虚拟机参数”选项。
4. 单击 “确定”。

# 设置运行时类路径选项

通过 “类路径”部分可以指定未包含在系统类路径环境变量中的其他类的位置。可能需要这些类来运行 Java 应用程序并确保正确回放。

可以在计算机或网络上浏览所需的类，然后对特定的测试禁用这些类。还可以通过更改其顺序来操纵类路径项。




**要设置类路径运行时设置，请执行下列操作：**

1. 打开 “运行时”设置 (F4)。在 “运行时”设置树中选择 “Java 环境设置 : 类路径”节点。

2. 向列表中添加类路径：

   单击 “添加类路径”按钮。 VuGen 将向类路径列表中添加新行。

   键入类的 *jar*、 *zip* 或其他存档文件的路径和名称。还可以单击字段右侧的 “浏览”按钮，以查找所需的文件。 VuGen 将向类路径列表中添加新位置，并将其状态设置为已启用。

3. 要永久删除某个类路径项，请选择该项，然后单击 “删除”按钮。
4. 要对特定测试禁用某个类路径项，请清除该项左侧的复选框。

5. 要将列表中的某个类路径项下移，请选择该项，然后单击向下箭头。

6. 要将列表中的某个类路径项上移，请选择该项，然后单击向上箭头。
7. 单击 “确定”关闭该对话框。

# 第 IV 部分

## 应用程序部署解决方案协议

# 17

# 开发 Citrix Vuser 脚本

通过 VuGen，可以录制使用 Citrix ICA 协议与服务器进行通信的 Citrix 客户端的操作。得到的脚本称为 Citrix Vuser 脚本。

本章描述：

- Citrix Vuser 脚本入门
- 设置 Citrix 录制选项
- 同步回放
- 使用 Citrix 函数
- 查看 Citrix Vuser 脚本
- 设置 Citrix 显示设置
- 设置 Citrix 运行时设置
- 了解 ICA 文件
- 断开与 Citrix 服务器的连接
- Citrix Vuser 脚本使用提示

***下列信息适用于 Citrix ICA 协议。***

## 关于录制 Citrix Vuser 脚本

Citrix Vuser 脚本模拟 Citrix 客户端与服务器间的 Citrix ICA 协议通信。VuGen 录制通信期间的所有活动，并创建 Vuser 脚本。

对远程服务器执行操作时， VuGen 将生成描述这些操作的函数。每个函数都以 **ctrx** 前缀开头。这些函数模拟鼠标和键盘的移动。此外，使用 **ctrx** 函数还可以通过等待打开特定的窗口同步对操作的回放。

VuGen 还可以录制 Citrix NFUSE 会话。 NFuse 会话使用浏览器，而不是客户端。要录制 NFUSE 会话，必须为 Citrix 和 Web Vuser 执行多协议录制。（请参阅第 3 章 “使用 VuGen 进行录制”）。在多协议模式中， VuGen 将在录制会话期间生成来自两种协议的函数。

下例中， **ctrx_mouse_click** 模拟鼠标左键单击。

```
ctrx_mouse_click(44, 318, LEFT_BUTTON, 0);
```

有关语法和参数的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

在 VuGen 的主窗口中，可以查看和编辑录制的脚本。会话期间录制的 API 调用将显示在窗口中，以便对操作进行跟踪。

## Citrix Vuser 脚本入门

本部分概述使用 VuGen 开发 Citrix ICA Vuser 脚本的过程。另请参考第 213 页的 “Citrix Vuser 脚本使用提示”。

**要开发 Citrix ICA 脚本，请执行下列操作：**

**1 使用 VuGen 录制操作。**

调用 VuGen 并新建 Vuser 脚本。注意，录制 Citrix ICA 客户端时，可以使用单协议 Vuser 脚本 Citrix ICA。而录制 NFUSE 会话时，则必须创建多协议 Vuser 脚本 Citrix ICA 和 HTTP，这样才能录制这两种协议。

录制 Citrix ICA 客户端会话时，必须在 “录制程序”框中指定 Mercury Interactive 版本的客户端作为要录制的应用程序。此文件名为 *runDlg.exe*，位于 LoadRunner 安装目录中的 *bin* 目录下。

有关录制的详细信息，请参阅第 3 章 “使用 VuGen 进行录制”。

**2 增强 Vuser 脚本。**

通过将事务、集合点和控制流结构插入到脚本中，可以增强 Vuser 脚本。

详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

**3 定义参数（可选）。**

为录制到 Vuser 脚本中的固定值定义参数。通过使用参数替换固定值，可以用不同的值多次重复同一业务流程。

详细信息，请参阅第 7 章 “定义参数”。

**4 配置 Citrix 显示选项。**

配置回放 Citrix Vuser 的显示选项。使用这些选项可以显示回放期间的 Citrix 客户端，并可以在发生错误时打开快照。详细信息，请参阅第 206 页的 “设置 Citrix 显示设置”。

**5 配置运行时设置。**

运行时设置控制脚本执行期间 Vuser 的行为。这些设置包括步、日志记录、思考时间和连接信息。

有关常规运行时设置的信息，请参阅第 9 章 “配置运行时设置”。有关 Citrix 特定运行时设置的详细信息，请参阅第 207 页的 “设置 Citrix 运行时设置”。

**6 在 VuGen 中保存并运行 Vuser 脚本。**

在 VuGen 中保存并运行 Vuser 脚本以验证该脚本是否可以正常运行。录制时，VuGen 会创建一系列配置、数据和源代码文件。这些文件包含 Vuser 运行时信息和设置信息。 VuGen 将这些文件连同脚本一起保存。

有关作为单独测试而运行 Vuser 脚本的详细信息，请参阅第 213 页的 “Citrix Vuser 脚本使用提示”和第 11 章 “在独立模式下运行 Vuser 脚本”。

创建 Citrix Vuser 脚本后，请将其集成到场景中。详细信息，请参阅《LoadRunner Controller 用户指南》。

## 了解 Citrix 录制选项

可以在下列区域中设置 Citrix 录制选项。

- 登录
- 配置
- 录制器

## 登录

在 “Citrix：登录”录制选项中，可以为录制会话设置连接信息和登录信息。必须提供服务器的名称，否则连接将会失败。如果没有提供登录信息，在客户端查找指定服务器时，系统将提示输入该信息。

可以直接提供登录信息，也可以指示 VuGen 使用存储在 *ica* 文件中的原有配置。

必须提供服务器的名称，否则连接将会失败。提供的登录信息将被录制到 **ctrx_connect_server** 函数中：

```
ctrx_connect_server(“steel”, “test”,“test”, “testlab”);
```

如果没有提供登录信息，在客户端查找指定服务器时，系统将提示输入该信息。

### 使用连接信息登录

“连接”部分 – 输入连接信息：

- Citrix 服务器名
- Citrix 服务器可识别的发行的应用程序名。如果没有指定发行的应用程序，VuGen 将使用服务器的桌面。

➤ 首选的“网络协议”：TCP/IP 或 TCP/IP+HTTP。

“登录信息”部分 – 输入登录信息：

➤ Citrix 服务器的登录名

➤ Citrix 服务器的密码

➤ Citrix 服务器的域

### 使用 ICA 文件登录

如果已有包含所有相关配置信息的 *.ica* 文件，请选择“使用 ICA 文件登录”。在以下行中，指定 *.ica* 文件的完整路径。

有关 ICA 文件格式的信息，请参阅第 210 页的“了解 ICA 文件”。跳到步骤 6。

## 配置

在“Citrix：配置”录制选项中，可以设置录制会话期间 Citrix 客户端的窗口属性和加密设置。

➤ **加密级别**：ICA 连接的加密级别：“基本”、“128 位 ( 仅用于登录 )”、“40 位”、“56 位”、“128 位”或“使用服务器默认值”以使用计算机的默认配置。

➤ **窗口大小**：客户端窗口的大小：“640 x 480”、“800 x 600”（默认）或“1024 x 768”。

### 录制器

使用 “Citrix：录制器”录制选项可以指定在已更改标题的位置如何生成窗口名。

有下列选项：

- 选择 “使用显示的标题作为新窗口名”，将使用当前显示的窗口标题作为窗口名。
- 选择 “为新名称使用通用前缀”，将使用标题开头处的公共字符串作为窗口名。
- 选择 “为新名称使用通用后缀”，将使用标题结尾处的公共字符串作为窗口名。

## 设置 Citrix 录制选项

录制之前，应首先设置所需的录制选项。

---

**注意**：请确保 Controller、负载生成器计算机和屏幕上的 “配置”设置相同。否则， VuGen 可能在回放期间找不到正确的位图。

---

**要设置 Citrix 录制选项，请执行下列操作：**

1. 打开 “录制选项”对话框。选择 “工具” > “录制选项”，或者单击 “开始录制”对话框中的 “选项”按钮。键盘快捷键为 CTRL+F7。
2. 选择 “Citrix：登录”节点。
   - 如果已有包含所有相关配置信息的 *.ica* 文件，请选择 “使用 ICA 文件登录”。在下一行中，指定 ICA 文件的完整路径，或者单击 “浏览”按钮在本地磁盘或网络中查找该文件。
   - 如果没有 ICA 文件，请选择 “使用连接信息登录”。该选项为默认设置。在 “连接”部分中，输入连接信息。在 “登录信息”部分中，输入登录信息。
3. 选择 “Citrix：配置”节点。选择加密级别和窗口大小。

4 选择 “Citrix：录制器”节点。指定如何为在录制会话期间更改了标题的窗口生成窗口名。

5 录制 NFUSE 会话时，将 Web 录制模式设置为基于 URL。选择 “Internet 协议：录制”录制选项，并选择 “基于 URL 的脚本”。

6 单击 “确定”接受设置并关闭该对话框。

## 同步回放

回放期间， VuGen 会自动生成对操作进行同步的函数。此外，还可以添加手动同步函数。

### 自动同步

录制期间， VuGen 会自动生成一些函数，这些函数有助于同步 Vuser 的脚本回放：**ctrx_sync_on_window**。

**ctrx_sync_on_window** 函数指示 Vuser 等待特定事件之后再继续回放。可用的事件为 *Create* 或 *Active*。 Create 事件等到窗口被创建时发生。 Active 事件等到窗口被创建并激活 （置为焦点）时发生。对于从未完全激活的非窗口对象 （如菜单）， VuGen 通常会生成带有 *Create* 事件的函数。对于标准窗口对象， VuGen 会生成带有 *Active* 事件的函数。

对于由 **ctrx_sync_on_window** 函数录制的所有窗口， VuGen 将保存屏幕快照，并将其存储在脚本的 data/bitmap 目录中。该函数的第七个参数为位图文件的名称。

```
ctrx_sync_on_window("ICA Administrator Toolbar", ACTIVATE, 768, 0, 33,
                    573, "129c54d64c9679ea1a0cb8bb02c2a981", CTRX_LAST);
```

### 手动同步

还可以通过 VuGen 用户界面或插入自定义同步函数来添加手动同步。

手动同步的第一种类型是通过 VuGen 的用户界面。录制期间， VuGen 会在浮动录制工具栏中提供一个标记工具。使用该工具，可以在恢复回放之前需要置为焦点的客户端窗口中标记位图区域。该功能常用于浏览器窗口中的特定图像。录制会话期间，可以使用 VuGen 的标记工具 （位于浮动工具栏中）将区域标记为位图。 VuGen 将生成 **ctrx_sync_on_bitmap** 函数。

```
ctrx_sync_on_bitmap(93, 227, 78, 52,
                    "66de3122a58baade89e63698d1c0d5dfa");
```

回放期间，该函数会在指定位置查找位图。要标记同步部分，请单击标记按钮。通过将鼠标从窗口中某部分的左上方拖动到右下方，将该部分标记为要等待的部分。 VuGen 将生成 **ctrx_sync_on_bitmap** 函数，所选坐标为该函数的参数。

要实现其他类型的同步，必须将下列函数之一手动输入到脚本中。

- **ctrx_sync_on_bitmap_change:** 指定某个区域 （位置和大小）， VuGen 等待该区域中的更改。若不知道该区域中所显示的数据或图像，但确实需要等待该区域更改时，该函数非常有用。为帮助您为该位图区域派生出正确的坐标，可以使用标记工具 （请参阅上述内容）录制 **ctrx_sync_on_bitmap** 函数，手动修改函数名并删除第五个参数。

  各函数语法如下：

  **ctrx_sync_on_bitmap** (x_start, y_start, width, height, hash);
  **ctrx_sync_on_bitmap_change** (x_start, y_start, width, height,
  [initial_wait_time,] [timeout,] CTRX_LAST};

可以向 **ctrx_sync_on_bitmap_change** 函数中添加可选参数：初始等待时间值 （开始检查更改的时间）和超时 （失败之前等待更改发生的时间长度，以秒为单位）。

```
/* 录制的函数 */
ctrx_sync_on_bitmap(93, 227, 78, 52,
                    “66de3122a58baade89e63698d1c0d5dfa“);
```

```
/* 修改后的函数，初始等待时间为 300，超时为 400 */
ctrx_sync_on_bitmap_change(93, 227, 78, 52, 300, 400, CTRX_LAST);
```

录制期间，**ctrx_sync_on_bitmap** 函数的生成位图将保存在脚本的 *data/bitmap* 目录下。位图名的格式为 *hash_value*.bmp。如果回放期间同步失败，则生成的位图将写入脚本的输出目录中，如果是在场景中运行脚本，则将写入输出文件所写入的位置。可以检查新建位图以确定同步失败的原因。

- **ctrx_set_waiting_time:** 设置其他 Citrix 同步函数的等待时间。该设置将应用于脚本中后面的所有函数。例如，如果 **ctrx_set_window** 函数超时，您可以将默认超时从 60 秒增加至 180 秒。

```
ctrx_set_waiting_time(180);
```

- **ctrx_win_exist:** 检查 Citrix 客户端中窗口是否可见。例如，可以使用该函数检查浏览器是否已启动。第二个参数指示使窗口变为可见时函数需等待的时间。如果函数尚未启动，请双击其图标。

```
if (!ctrx_win_exist(“Welcome to MSN.com- Microsoft Internet
Explorer“,6))
  ctrx_mouse_double_click(34, 325, LEFT_BUTTON, 0)
```

有关这些函数的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

# 使用 Citrix 函数

在 Citrix 录制会话期间， VuGen 将生成模拟客户端与远程服务器间通信的函数。生成的函数前缀为 **ctrx**。录制会话后，还可以将任一函数手动编辑到 Vuser 脚本中。有关 **ctrx** 函数的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

注意，对于指定了窗口名的函数，可以使用通配符，即星号 (*)。可以将通配符放在字符串的开头、中间或末尾，但每个窗口名仅能使用一个通配符。

## Citrix 函数

| | |
|---|---|
| **ctrx_connect_server** | 使用 Citrix 客户端连接远程服务器 |
| **ctrx_disconnect_server** | 关闭与服务器的连接 |
| **ctrx_get_bitmap_value** | 获得指定位图的哈希值 |
| **ctrx_get_window_name** | 获得处于焦点的窗口的名称 |
| **ctrx_get_window_position** | 获得指定窗口或处于焦点的窗口的位置 |
| **ctrx_key** | 模拟按下非字母数字键 |
| **ctrx_key_down** | 模拟按下键盘键 |
| **ctrx_key_up** | 模拟释放键盘键 |
| **ctrx_mouse_click** | 模拟鼠标单击 |
| **ctrx_mouse_double_click** | 模拟鼠标双击 |
| **ctrx_mouse_down** | 模拟按下鼠标按钮 |
| **ctrx_mouse_move** | 模拟鼠标移动 |
| **ctrx_mouse_up** | 模拟释放鼠标按钮 |
| **ctrx_nfuse_connect** | 通过 NFUSE 门户连接到 Citrix 服务器 |
| **ctrx_set_connect_opt** | 设置连接选项 |
| **ctrx_set_exception** | 指定异常处理 |
| **ctrx_set_waiting_time** | 设置所有后续计时函数的等待时间 |
| **ctrx_set_window** | 等待指定的窗口打开 |

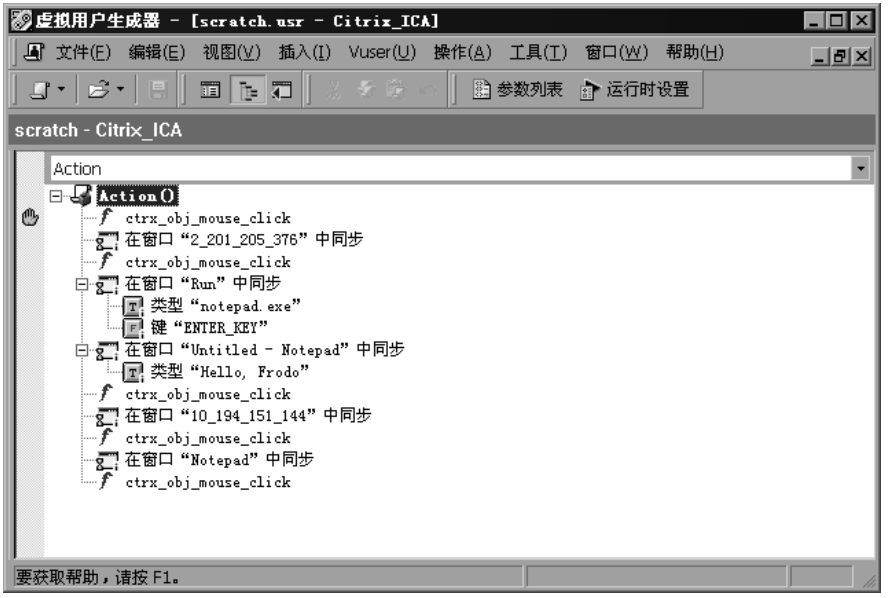
| | |
|---|---|
| **ctrx_set_window_ex** | 等待特定的一段时间 （以秒为单位）直到指定的窗口打开 |
| **ctrx_sync_on_bitmap** | 等待坐标指定的位图显示 |
| **ctrx_sync_on_bitmap_change** | 等待坐标指定的区域发生改变 |
| **ctrx_sync_on_window** | 等待窗口被创建或变为活动状态 |
| **ctrx_type** | 模拟键入字母数字键 |
| **ctrx_unset_window** | 等待指定的窗口关闭 |
| **ctrx_wait_for_event** | 等待指定的事件发生 |
| **ctrx_win_exist** | 检查指定的窗口是否存在 |

# 查看 Citrix Vuser 脚本

在 VuGen 脚本编辑器中，可以查看 *vuser_init*、 *Actions* 和 *vuser_end* 部分的内容。要显示某个操作，请在左窗格中选择该操作的名称。 VuGen 将在右窗格中显示其内容。在 VuGen 中查看和编辑 Citrix Vuser 脚本时，可以选择显示基于图标的树视图或基于文本的脚本视图。

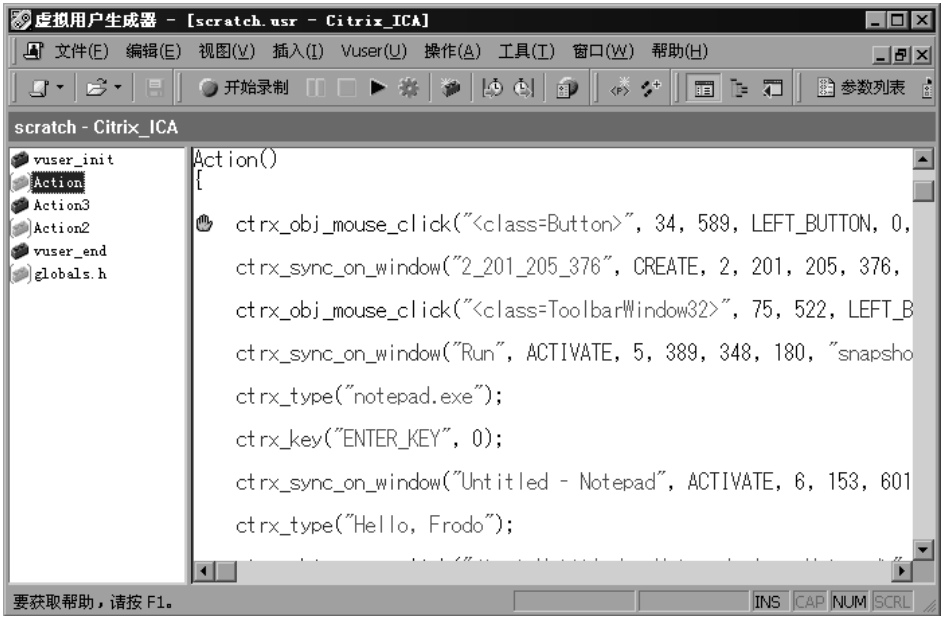
**要显示树视图，请执行下列操作：**

从 VuGen 主菜单中选择 “视图” > “树视图”，或单击 “桉树查看脚本” 图标。将在基于图标的树视图中显示 Vuser 脚本的 Actions 部分。如果已经显示了树视图，该菜单项将被禁用。

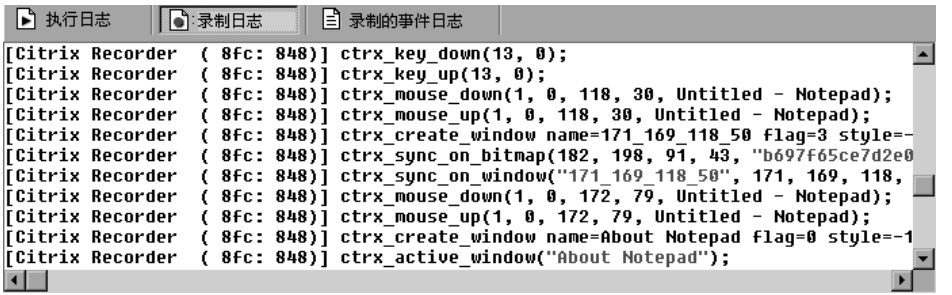


要显示脚本视图，请执行下列操作：

从 VuGen 主菜单中，选择 “视图” > “脚本视图”，或单击 “按文本查看脚本” 图标。将在基于文本的脚本视图中显示 Vuser 脚本。如果已经显示了脚本视图，该菜单项将被禁用。




要了解有关录制会话的详细信息，可以查看录制期间生成的操作日志。选择 “视图” > “输出窗口”，然后选择 “录制日志” 选项卡。VuGen 将会显示它执行的所有操作的详细日志。



# 设置 Citrix 显示设置

运行 Citrix Vuser 脚本之前，可以首先设置回放期间将使用到的一些显示选项。尽管这些选项会增加服务器的负载，但它们对于调试和分析会话非常有用。

在 “常规选项”对话框中可以设置 Citrix 显示选项。

**要设置 Citrix 显示选项，请执行下列操作：**

1 打开 “常规选项”对话框。选择 “工具” > “常规选项”。

2 选择 “Citrix 显示”选项卡。

3 选择 “回放期间显示客户端”，在回放 Vuser 脚本时显示 Citrix 客户端。

4 选择 “出现错误时显示快照”，当出现错误时显示屏幕的快照。

5 单击 “确定”。

# 设置 Citrix 运行时设置

创建 Citrix Vuser 脚本后，可以设置运行时设置。使用运行时设置可以控制脚本运行时 Vuser 的行为。“配置”节点中的 Citrix 运行时设置应与 Citrix 客户端的属性相对应。这些设置将影响服务器负载。要查看连接属性，请在 “Citrix 程序邻居”中选择表示 ICA 连接的图标，并在右键单击菜单中选择 “属性”。选择“默认选项”选项卡。

---

**注意**：Citrix Vuser 不支持调制解调器速度模拟和 IP 欺骗。

---

在 “运行时设置”对话框中，可以设置下列区域的 Citrix 运行时设置：

- Citrix 配置运行时设置
- Citrix 超时运行时设置

## Citrix 配置运行时设置

该配置设置与屏幕延迟时间、数据压缩、磁盘缓存和鼠标动作队列有关。

**要设置 “配置运行时设置”，请执行下列操作：**

1 打开 “运行时设置”对话框。单击 VuGen 工具栏上的 “运行时设置”按钮，或者选择 “Vuser” > “运行时设置”。

2 选择 “Citrix：配置”节点。指定 “常规”属性：

➤ **SpeedScreen 延迟缩短**：当网络速度较慢时，该机制用于增强用户交互。根据网络速度，可以“打开”或“关闭”该机制。选择“自动”选项，会根据当前的网络速度打开或关闭该机制。如果不知道网络速度如何，请将该选项设置为“使用服务器默认值”以使用计算机的默认值。

3 设置“使用数据压缩”选项。该选项指示 Vuser 压缩传输的数据。要启用该选项，请选中其左侧的复选框；要禁用该选项，请清除该复选框。如果带宽有限，则应启用数据压缩。（默认情况下启用）

4 设置“使用位图磁盘缓存”选项。该选项指示 Vuser 使用本地缓存存储位图和常用的图形对象。要启用该选项，请选中其左侧的复选框；要禁用该选项，请清除该复选框。如果带宽有限，则应启用该选项。（默认情况下禁用）

5 设置“鼠标动作和击键队列”选项。该选项指示 Vuser 创建鼠标动作和击键的队列，并将其作为数据包以较慢频率发送到服务器。其目的是减慢连接速度，从而减少网络流量。启用该选项可以减慢会话对键盘和鼠标动作的响应速度。要启用该选项，请选中其左侧的复选框；要禁用该选项，请清除该复选框。（默认情况下禁用）

## Citrix 超时运行时设置

该超时设置与连接时间和等待时间有关。

**要设置 “超时运行时设置”，请执行下列操作：**

1 打开 “运行时设置”对话框。单击 VuGen 工具栏上的 “运行时设置”按钮，或者选择 “Vuser” > “运行时设置”。

2 选择 “Citrix：超时”节点。

- **连接时间：**退出之前在已建立连接处空闲等待的时间 （以秒为单位）。默认值为 180 秒。
- **等待时间：**退出之前在同步点空闲等待的时间 （以秒为单位）。默认值为 60 秒。

3 单击 “确定”接受设置并关闭该对话框。

# 了解 ICA 文件

Citrix ICA 客户端文件是一些文本文件，其中包含通过 Citrix 客户端访问的应用程序的配置信息。这些文件均带有 *ica* 扩展名，且必须符合下列格式：

```
[WFClient]
Version=
TcpBrowserAddress=

[ApplicationServers]
AppName1=

[AppName1]
Address=
InitialProgram=#
ClientAudio=
AudioBandwidthLimit=
Compress=
DesiredHRES=
DesiredVRES=
DesiredColor=
TransportDriver=
WinStationDriver=

Username=
Domain=
ClearPassword=
```

---

**注意**：使用 “录制选项”加载 ICA 文件时， VuGen 将文件与脚本一同保存，因而不必再将 ICA 文件复制到每个注射器计算机上。

---

下例显示了通过 Citrix 客户端使用远程计算机上的 MicroSoft Word 的 ICA 示例文件：

```
[WFClient]
Version=2
TcpBrowserAddress=235.119.93.56

[ApplicationServers]
Word=

[Word]
Address=Word
InitialProgram=#Word
ClientAudio=On
AudioBandwidthLimit=2
Compress=On
DesiredHRES=800
DesiredVRES=600
DesiredColor=2
TransportDriver=TCP/IP
WinStationDriver=ICA 3.0

Username=test
Domain=user_lab
ClearPassword=test
```

## 断开与 Citrix 服务器的连接

Citrix 客户端关闭连接后，默认情况下，服务器将保存会话，以便下次同一客户端打开新建连接时使用。因此，同一客户端的新建连接将使用与上次断开连接时相同的工作区。通常，每次新的测试运行需要一个干净的工作区，因此这不是运行测试的理想方法。

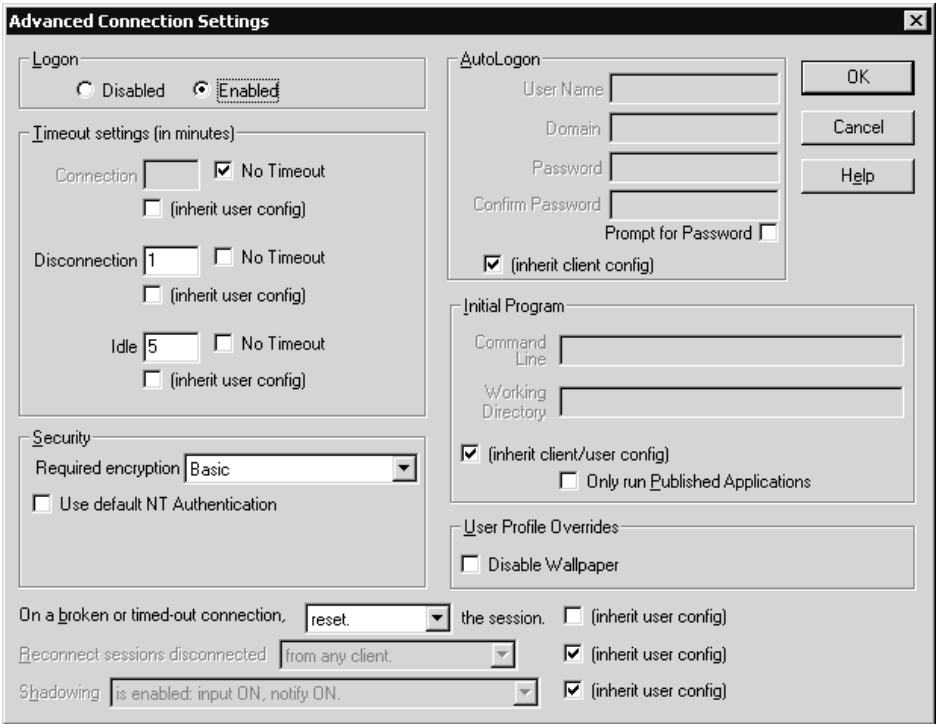
解决方案是将 Citrix 服务器配置为不保存上次会话，在每次客户端断开连接或超时时完全断开与客户端的连接。

**要强制断开 Citrix 服务器与 XP 服务器的连接，请执行下列操作：**

**1** 打开 “Citrix 连接配置”对话框。选择 “开始” > “程序” > “Citrix” > “MetaFrame XP” > “Citrix 连接配置”。

**2** 双击 ica-tcp 连接名。将显示 “编辑连接”对话框。

**3** 单击 “高级”按钮。将显示 “高级连接设置”对话框。




**4** 在该对话框的底部，清除 “连接断开或超时时”列表框旁的 “继承用户配置”复选框。将该列表框中的选项更改为 “重置”。

# Citrix Vuser 脚本使用提示

### 录制提示

- 录制会话时，请确保执行完整的业务流程，以连接开始，以清理结束。在可以从头开始整个流程的一点结束会话。不要将任何客户端或应用程序窗口保留为打开状态。
- 建议在录制会话期间不要移动窗口或调整窗口的大小。
- Vugen 使用桌面的颜色设置。
- 为保证位图同步成功，请确保设置匹配的分辨率。在录制计算机上，请检查“ICA 客户端”、“录制选项”以及 “运行时设置”设置。在注射器计算机上，检查 ICA 客户端的设置，并确保所有注射器计算机与录制计算机中的设置相一致。
- 支持的分辨率 （窗口大小）为 640 x 480、800 x 600 和 1024 x 768。建议录制计算机的显示器设置为 1024 x 768，以便 Citrix 窗口（默认大小为 800 x 600）能够恰当显示。
- 录制期间等待事件（如加载 HTML 网页）时，建议使用 **ctrx_sync_on_bitmap** 函数添加手动同步点。详细信息，请参阅第 199 页的 “同步回放”。
- 配置 Citrix 服务器，以完全关闭会话。请参阅第 211 页的 “断开与 Citrix 服务器的连接”。
- 将连接进程录制到 *vuser_init* 部分中，将关闭进程录制到 *vuser_end* 部分中。这样可以避免对连接进程执行循环。
- 如果未能成功录制 Citrix 会话中的任何操作，请验证计算机中是否仅安装了一个 Citrix 客户端 - Mercury Interactive 版本的客户端。要验证是否仅安装了一个客户端，请从 “控制面板”中打开 “添加 / 删除程序”对话框，确认只有一个 Citrix ICA 客户端项。
- 打开展开的菜单选项时，明确单击每一选项，不要依赖菜单自动展开功能。例如，“开始” > “程序” > “Microsoft Word”，确保要单击 “程序”菜单项。
- 在 Citrix 客户端发出提示时，禁用客户端更新。

### 回放提示

- 为防止连接时多个 Vuser 造成过载，请设置 4 至 10 个 Vuser 初始化配额 （根据服务器的性能）或使用 “计划程序”应用加压初始化。
- 要获得最佳结果，请不要禁用运行时设置中的思考时间。思考时间在 **ctrx_sync_on_window** 和 **ctrx_sync_on_bitmap** 函数之前尤为重要，因为这两个函数需要一定时间以达到稳定。

- 如果要在另一台计算机上回放脚本，请确保录制计算机和回放计算机的下列各项相互一致：“窗口大小（分辨率）”、“窗口颜色”、“系统字体”以及 Citrix 客户端的其他“默认选项”设置。这些设置会影响位图的哈希值，不一致的设置可能会导致回放失败。要查看 Citrix 客户端设置，请从 Citrix 程序组中选择一项，并选择“应用程序集设置”，或在 ICA 连接图标的右键单击菜单中选择“自定义连接设置”。单击“默认选项”选项卡。
- 由于图形资源有限，运行 Citrix Vuser 的计算机可能只能运行有限数量的 Vuser。要增加每台计算机的 Vuser 数目，请在该计算机上打开一个终端服务器会话。作为一台新的注射器计算机关联到该终端服务器。要在 Controller 中标识该虚拟注射器计算机，请使用下列格式：计算机名 :1、计算机名 :2、…使用计算机名或其 IP 地址。注意，默认情况下，终端服务器上的会话使用 256 色设置。要使用终端会话进行负载测试，请确保录制计算机同样采用 256 色设置。

**调试提示**

- 向 VuGen 中的脚本添加断点有助于确定有问题的代码行。
- 如果回放失败，可能需要在脚本中插入同步函数，以便有更长的时间使所需的窗口获得焦点。虽然也可以使用 **lr_think_time** 函数手动添加延迟，但建议使用第 199 页的“同步回放”中讨论的同步函数之一。
- 可以在扩展日志中查看其他回放信息。可以在“运行时设置”（F4 快捷键）的“日志”选项卡中启用扩展日志。可以在“执行日志”选项卡或脚本目录中的 *output.txt* 文件中查看这些信息。
- 发生错误时， VuGen 将在脚本的 *output* 目录中保存屏幕快照。可以查看位图以确定发生错误的原因。
- 录制期间， **ctrx_sync_on_bitmap** 函数生成的位图将保存在脚本的 *data* 目录下。位图名的格式为 *hash_value*.bmp。如果回放期间同步失败，则生成的位图将被写入脚本的输出目录下，或者，如果是在场景中运行脚本，将写入输出文件被写入的任何位置。可以检查新建位图以确定同步失败的原因。
- 要显示场景运行期间的所有 Vuser，请在 Vuser 命令行框中输入：-lr_citrix_vuser_view。在 Controller 中，打开“Vuser 详细信息”对话框，单击“更多”展开该对话框。注意，这将影响测试的可伸缩性，因此只用于检查有问题的 Vuser 行为。
- 要检查服务器的版本，请确保安装了 MetaFrame XP 或 1.8 服务器。在服务器控制台的工具栏中选择“Citrix 连接配置”，然后选择“帮助”>“关于”。

# 第 V 部分

## 客户端服务器协议

# 18

# 开发数据库 Vuser 脚本

您可以使用 VuGen 录制数据库客户端应用程序与服务器之间的通信。得到的脚本称为数据库 Vuser 脚本。

本章描述：

- 介绍数据库 Vuser
- 了解数据库 Vuser 技术
- 数据库 Vuser 脚本入门
- 设置数据库录制选项
- 使用 LRD 函数
- 了解数据库 Vuser 脚本
- 评估错误代码
- 处理错误

***下列信息仅适用于客户端服务器数据库（CtLib、DbLib、Informix、MS SQL Server、 Oracle、 ODBC 和 DB2-CLI）和 ERP Siebel Vuser 脚本。***

## 关于录制数据库 Vuser 脚本

录制与服务器通信的数据库应用程序时， VuGen 将生成数据库 Vuser 脚本。VuGen 支持下列数据库类型：CtLib、 DbLib、 Informix、 Oracle、 ODBC 和 DB2-CLI。得到的脚本包含描述数据库活动的 LRD 函数。每个 LRD 函数都具有 **lrd** 前缀，表示一个或多个数据库函数。例如， **lrd_fetch** 函数表示提取操作。

运行录制的会话时， Vuser 脚本将直接与数据库服务器通信，执行与原用户相同的操作。您可以设置 Vuser 行为 （运行时设置）来指明重复操作次数和重复操作之间的时间间隔。详细信息，请参阅第 9 章 “配置运行时设置”。

使用 VuGen，您可以将脚本参数化，从而将录制的常量替换为参数。详细信息，请参阅第 7 章 “定义参数”。

此外，您还可以对脚本中的查询和数据库语句进行关联，从而将查询结果互相链接起来。详细信息，请参阅第 8 章 “关联语句”。

有关疑难解答信息和脚本提示，请参阅第 69 章 “VuGen 调试提示”。

## 介绍数据库 Vuser

假设您拥有的客户信息数据库可以被遍布全国的客户服务人员访问。您可以使用数据库 Vuser 来模拟数据库服务器处理众多信息请求的情况。数据库 Vuser 能够：

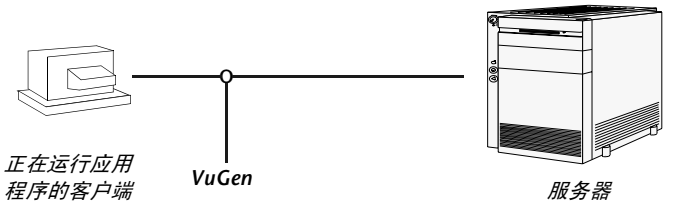
- 连接到服务器
- 提交 SQL 查询
- 检索并处理信息
- 断开与服务器的连接

您可以将数以百计的数据库 Vuser 分配到可用的负载生成器中，每个 Vuser 都使用服务器 API 访问数据库。这样，您就能够度量服务器在用户负载较重时的性能。

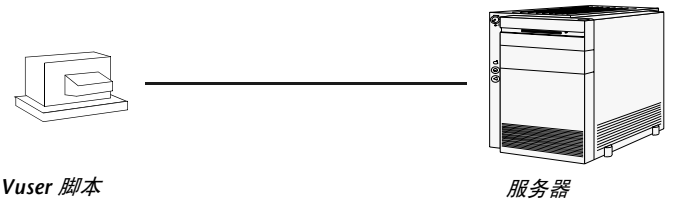
包含对服务器 API 的调用的程序称为数据库 Vuser 脚本。它模拟客户端应用程序及其执行的所有操作。使用 Controller，可以将脚本分配给多个 Vuser。 Vuser 将执行该脚本并在客户端 / 服务器系统上模拟用户负载。 LoadRunner 将生成性能数据，您可以以报告和图等格式分析该数据。

# 了解数据库 Vuser 技术

VuGen 通过录制数据库客户端和服务器之间的所有活动来创建数据库 Vuser 脚本。 VuGen 将监视数据库的客户端，并跟踪数据库服务器上接收和发送的所有请求。




像其他所有使用 VuGen 创建的 Vuser 那样，数据库 Vuser 不依靠客户端软件与服务器通信。相反，每个数据库 Vuser 都执行特定脚本，该脚本直接调用服务器 API 函数。




您可以使用 VuGen 在 Windows 环境中创建数据库 Vuser 脚本。创建了脚本后，即可将其分配给 Windows 和 UNIX 环境下的 Vuser。有关录制脚本的详细信息，请参阅第 3 章 “使用 VuGen 进行录制”。

仅使用 UNIX 环境的用户可以通过将 LoadRunner 模板用作脚本基础，以编程的方式创建数据库 Vuser 脚本。有关在 UNIX 上编写数据库 Vuser 脚本的信息，请参阅附录 D “在 UNIX 平台上编写脚本”。

# 数据库 Vuser 脚本入门

本部分概述使用 VuGen 开发数据库 Vuser 脚本的过程。

**要开发数据库 Vuser 脚本，请执行下列操作：**

**1 使用 VuGen 录制基本脚本。**

调用 VuGen 并新建一个 Vuser 脚本。指定 Vuser 的类型 （客户端服务器或 ERP 协议类型）。选择要录制的应用程序，并设置录制选项。录制应用程序的典型操作。

详细信息，请参阅第 3 章 “使用 VuGen 进行录制”。

**2 增强脚本。**

通过将事务、集合点和控制流结构插入脚本以增强 Vuser 脚本。

详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

**3 定义参数 （可选）。**

为脚本中录制的固定值定义参数。通过用参数替换固定值，您可以用不同的值多次重复相同查询。

详细信息，请参阅第 7 章 “定义参数”。

**4 关联查询 （可选）。**

关联数据库语句后可以在后续查询中使用以前查询的结果。当使用具有用户约束的数据库时，此功能非常有用。

详细信息，请参阅第 8 章 “关联语句”。

**5 配置运行时设置。**

运行时设置在脚本执行过程中控制 Vuser 脚本行为。这些设置包括循环、日志和计时信息。

详细信息，请参阅第 9 章 “配置运行时设置”。

**6 通过 VuGen 运行脚本。**

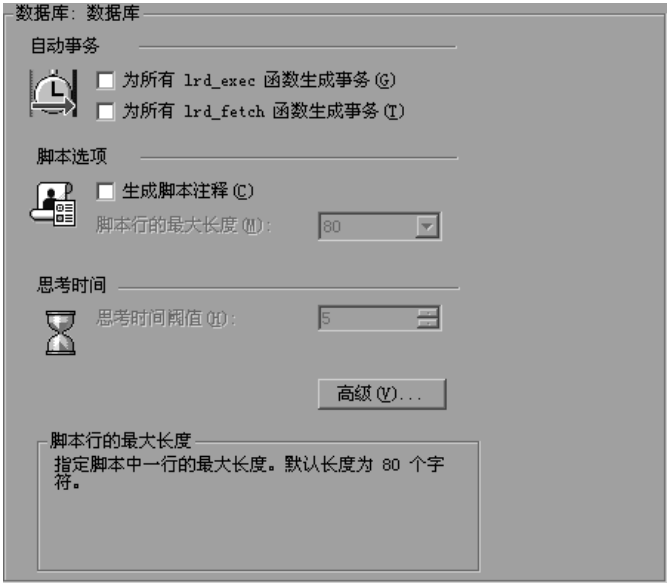
在 VuGen 中保存并运行脚本，以验证其是否能够正确运行。

详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

创建数据库 Vuser 脚本后，在 Windows 或 UNIX 平台上都可以将其集成到场景中。有关将 Vuser 脚本集成到场景中的详细信息，请参阅 《LoadRunner Controller 用户指南》。

# 设置数据库录制选项

在录制数据库会话之前，应该设置录制选项。您可以为自动函数的生成、脚本选项和思考时间设置基本录制选项：




**自动事务**：您可以指示 VuGen 把每个 **lrd_exec** 和 **lrd_fetch** 函数标记为事务。启用这些选项后，VuGen 将把 **lr_start_transaction** 和 **lr_end_transaction** 插入到每个 **lrd_exec** 和 **lrd_fetch** 函数周围。默认情况下，自动事务将禁用。

**脚本选项**：您可以指示 VuGen 在录制的脚本中生成注释，描述 **lrd_stmt** 选项值。此外，还可以指定脚本中行的最大长度。默认长度为 80 个字符。

**思考时间**：VuGen 自动录制操作者的思考时间。您可以设置阈值级别，录制的思考时间如果低于此阈值，则将忽略。如果录制的思考时间超出了阈值级别，VuGen 将在 LRD 函数之前放置 **lr_think_time** 语句。如果录制的思考时间低于阈值级别，将不会生成 **lr_think_time** 语句。默认值为五秒。

要设置数据库录制选项，请执行下列操作：

1 选择 “工具” > “录制选项”。将打开 “录制选项”对话框。

2 选择 “为所有 lrd_exec 函数生成事务”，为 **lrd_exec** 语句启用自动事务。

选择 “为所有 lrd_fetch 函数生成事务”，为 **lrd_fetch** 语句启用自动事务。

3 选择 “生成脚本注释”，指示 VuGen 将描述性注释插入到脚本中。

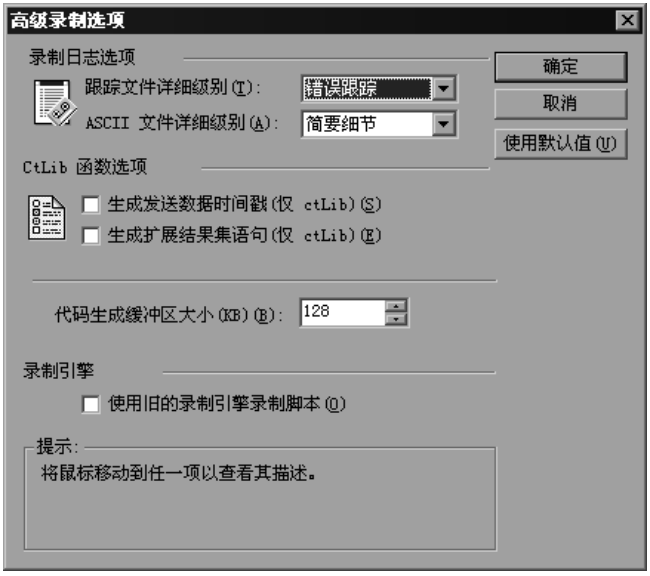
4 要在 VuGen 编辑器中更改行的最大长度，请在 “脚本行最大长度”框中指定所需的值。

5 要更改思考时间的阈值 （默认为五秒），请在 “思考时间阈值”框中指定所需的值。

您还可以设置有关跟踪级别、Ctlib 函数生成和代码生成缓冲区的高级录制选项。

## 数据库高级录制选项

除了基本录制选项，您还可以为日志文件详细信息、 CtLib 特定函数、缓冲区大小和录制引擎配置高级选项：



**录制日志选项**：您可以为跟踪和 ASCII 日志文件设置详细级别。跟踪文件可用的级别有 “关闭”、“错误跟踪”、“简单跟踪”和 “完全跟踪”。错误跟踪仅记录错误消息。简单跟踪记录错误并列出录制过程中生成的函数。完全跟踪将记录所有消息、通知和警告。

您还可以指示 VuGen 生成录制会话的 ASCII 类型日志。可用的级别有 “关闭”、“简单明细”和 “完全明细”。简单明细记录所有函数，而完全明细则记录所有生成的函数和 ASCII 代码消息。

**CtLib 函数选项**：您可以指示 VuGen 生成发送数据时间戳和扩展结果集语句。

- **时间戳**：默认情况下， VuGen 为 *mpszReqSpec* 参数生成带有 “TotalLen”和 “Log”关键字的 lrd_send_data 语句。通过 “高级录制选项”对话框可以指示 VuGen 生成 “TimeStamp”关键字。如果要对现有脚本更改此设置，必须通过选择 “工具” > “重新生成”来重新生成 Vuser 脚本。建议不要默认地生成 “Timestamp”关键字。录制过程中生成的时间戳不同于回放过程中所生成的，并且脚本执行将失败。应该仅在运行脚本的尝试失败（**lrd_send_data** 失败后出现 **lrd_result_set**）后使用此选项。现在，生成的时间戳可与先前的 **lrd_send_data** 生成的时间戳关联。
- **扩展结果集**：默认情况下， VuGen 会在准备结果集时生成 **lrd_result_set** 函数。此设置将指示 VuGen 生成 **lrd_result_set** 函数 **lrd_result_set_ext** 的扩展形式。除了准备结果集，此函数还从 *ct_results* 发出返回代码和类型。

**代码生成缓冲区大小**：以 KB 为单位指定代码生成缓冲区的最大大小。默认值为 128 KB。对于长数据库会话，可以指定较大的大小。

**录制引擎**：6.0 版及更高版本的 VuGen 使用新的录制引擎。您可以指示 VuGen 使用较旧的录制引擎来录制脚本，以与前几个版本的 VuGen 兼容。

**要设置高级录制选项，请执行下列操作：**

1. 在 “录制选项”对话框的 “数据库”节点中单击 “高级”按钮。将打开 “高级录制选项”对话框。
2. 选择 “跟踪文件详细级别”。要禁用跟踪文件，请选择 “关闭”。
3. 要生成 ASCII 日志文件，请在 “ASCII 文件详细级别”框中选择所需的详细级别。
4. 对于 CtLib：要指示 VuGen 为 lrd_send_data 函数生成 “TimeStamp”关键字，请选择 “生成发送数据时间戳”选项。
5. 对于 CtLib：要指示 VuGen 生成 **lrd_result_set_ext** 而非 **lrd_result_set**，请选择 “生成扩展结果集语句”选项。

6 要修改默认值为 128 KB 的代码生成缓冲区的大小，请在 “代码生成缓冲区大小”框中输入所需的值。

7 要使用旧的 VuGen 录制引擎，以实现向后兼容性，请选择 “使用旧录制引擎录制脚本”选项。

8 单击 “确定”以保存设置并关闭 “高级录制选项”对话框。

## 使用 LRD 函数

为模拟数据库客户端和服务器之间的通信而开发的函数称为 LRD Vuser 函数。每个 LRD Vuser 函数都有 lrd 前缀。在数据库会话 （CtLib、 DbLib、 Informix、 Oracle （第 2 层）和 ODBC）过程中， VuGen 会自动录制本部分中列出的大多数 LRD 函数。也可以手动将任一函数编写到脚本中。有关 LRD 函数的语法和示例，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

### 访问管理函数

| | |
|---|---|
| **lrd_alloc_connection** | 分配连接结构 |
| **lrd_close_all_cursors** | 关闭所有打开的光标 |
| **lrd_close_connection** | 断开与数据库的连接 （注销） |
| **lrd_close_context** | 关闭上下文 |
| **lrd_close_cursor** | 关闭数据库光标 |
| **lrd_ctlib_cursor** | 指定 CtLib 光标命令 |
| **lrd_commit** | 提交当前事务 |
| **lrd_db_option** | 为当前数据库设置选项 |
| **lrd_free_connection** | 释放连接结构 |
| **lrd_rollback** | 回滚当前事务 |
| **lrd_open_connection** | 连接到 （登录）数据库 |
| **lrd_open_context** | 打开上下文 |
| **lrd_open_cursor** | 打开数据库光标 |

### LRD 环境函数

| | |
|---|---|
| **lrd_msg** | 发出输出消息 |
| **lrd_option** | 设置 LRD 选项 |
| **lrd_end** | 关闭 lrd 环境 |
| **lrd_init** | 初始化 lrd 环境 |

### 检索处理函数

| | |
|---|---|
| **lrd_col_data** | 设置表明数据位置的指针 |
| **lrd_fetch** | 在结果集中提取下一行 |
| **lrd_fetchx** | 使用扩展提取在结果集中提取下一行（仅适用于 ODBC） |
| **lrd_result_set** | 返回结果集 （仅适用于 CtLib） |
| **lrd_result_set_ext** | 返回 CtLib 结果代码和结果类型 （扩展） |
| **lrd_fetch_adv** | 使用扩展提取从结果集当中提取多行。（仅适用于 ODBC） |
| **lrd_reset_rows** | 为更新操作准备提取行。（仅适用于 ODBC） |
| lrd_row_count | 返回受 UPDATE、DELETE 和 INSERT 语句影响的行数。（ODBC、 DB2） |

### 语句处理函数

| | |
|---|---|
| lrd_bind_col | 将主机变量绑定到输出列 |
| lrd_bind_cols | 将主机变量数组绑定到列 |
| lrd_bind_cursor | 将光标绑定到占位符 |
| lrd_bind_placeholder | 将主机变量或数组绑定到占位符 |
| **lrd_cancel** | 取消上一个语句 |
| lrd_data_info | 获得 I/O 信息。（仅适用于 CtLib） |
| lrd_dynamic | 指定要处理的动态 SQL 语句（仅适用于 CtLib） |

| | |
|---|---|
| lrd_exec | 执行指定的上一条 SQL 语句 |
| lrd_send_data | 将数据发送到服务器 |
| **lrd_stmt** | 指定要处理的 SQL 语句 |

### 语句关联函数

| | |
|---|---|
| **lrd_save_col** | 将表单元格的值保存到参数 |
| **lrd_save_value** | 将占位符描述符值保存到参数 |
| **lrd_save_ret_param** | 将返回参数的值保存到参数 （仅适用于 CtLib） |
| **lrd_save_last_rowid** | 将上一个行 ID 保存到参数 (Oracle) |

### 变量处理函数

| | |
|---|---|
| **lrd_assign** | 将以 NULL 结尾的字符串分配到变量 |
| **lrd_assign_ext** | 将存储区域分配到变量 |
| **lrd_assign_literal** | 将包含 NULL 字符的文字字符串分配到变量 |
| **lrd_assign_bind** | 将以 NULL 结尾的字符串分配到变量并将其绑定到占位符 |
| **lrd_assign_bind_ext** | 将存储区域值分配到变量并将其绑定到占位符。 |
| **lrd_assign_bind_literal** | 将包含 NULL 字符的文字字符串分配到变量并将其绑定到占位符 |
| **lrd_to_printable** | 将变量转换为可打印字符串 |

### Siebel 函数

| | |
|---|---|
| lrd_siebel_incr | 按指定的值增加字符串 |
| lrd_siebel_str2num | 将 base 36 字符串转换为 base 10 数字 |
| **SiebelPostSave_x** | 保存 Siebel 参数的未来值 |
| **SiebelPreSave_x** | 指明关联所必需的参数 |

## Oracle 8 函数

VuGen 提供对 Oracle 8.x 的部分支持。在以前的 Oracle 版本中录制的所有数据库操作均会录制下来。在许多实例中，录制的函数特定于 Oracle 8.x。例如，对于提取操作， VuGen 将录制 lrd_ora8_fetch，而不是 lrd_fetch。

| | |
|---|---|
| **lrd_attr_set** | 为 LRDDBI 句柄设置属性 |
| **lrd_attr_set_from_handle** | 使用 LRDDBI 句柄指针设置属性 |
| **lrd_attr_set_literal** | 使用文字字符串设置 LRDDBI 句柄属性 |
| **lrd_env_init** | 分配并初始化 LRDDBI 句柄 |
| **lrd_handle_alloc** | 显式分配并初始化 LRDDBI 句柄 |
| **lrd_handle_free** | 显式释放 LRDDBI 句柄 |
| **lrd_initialize_db** | 初始化数据库进程环境 |
| **lrd_logoff** | 终止简单数据库会话 |
| **lrd_logon** | 开始简单数据库会话 |
| **lrd_logon_ext** | 开始简单数据库会话 （扩展） |
| **lrd_ora8_attr_set** | 为 LRDDBI 句柄设置属性 – 速记 |
| **lrd_ora8_attr_set_from_handle** | 使用 LRDDBI 句柄指针设置属性 |
| **lrd_ora8_attr_set_literal** | 使用文字字符串设置 LRDDBI 句柄属性 – 速记 |
| **lrd_ora8_bind_col** | 将主机变量绑定到输出列 |
| **lrd_ora8_bind_placeholder** | 将主机变量绑定到占位符 |
| **lrd_ora8_commit** | 提交 Oracle 8.x 客户端的当前事务 |
| **lrd_ora8_exec** | 在 Oracle 8.x 中执行 SQL 语句 |
| **lrd_ora8_fetch** | 在结果集中提取下一行 |
| **lrd_ora8_handle_alloc** | 显式分配并初始化 LRDDBI 句柄 – 速记 |
| **lrd_ora8_rollback** | 回滚 Oracle 8.x 客户端的当前事务 |
| **lrd_ora8_save_col** | 将表单元格的值保存到参数 |
| **lrd_ora8_stmt** | 准备以 NULL 结尾的 SQL 语句以供执行 |

| | |
|---|---|
| **lrd_ora8_stmt_ext** | 准备带有 NULL 字符的 SQL 语句以供执行 |
| **lrd_ora8_stmt_literal** | 准备文字 SQL 语句字符串以供执行 |
| **lrd_server_attach** | 为数据库操作而创建对数据源的访问路径 |
| **lrd_server_detach** | 为数据库操作而删除对数据源的访问路径 |
| **lrd_session_begin** | 为服务器创建并开始用户会话 |
| **lrd_session_end** | 为服务器终止用户会话 |

## 了解数据库 Vuser 脚本

录制数据库会话之后，您可以在 VuGen 的内置编辑器中查看录制的代码。您可以在脚本中滚动，查看应用程序生成的 SQL 语句，并检查服务器返回的数据。VuGen 窗口提供下列关于录制的数据库会话的信息：

- 函数的录制顺序
- 用于显示数据库查询所返回的数据的网格
- 查询中提取的行数

### 函数顺序

在 VuGen 窗口中查看 Vuser 脚本时，您可看到 VuGen 录制活动的顺序。例如，以下函数顺序即为在典型的 Oracle 数据库会话过程中录制下来的：

| | |
|---|---|
| **lrd_init** | 初始化环境 |
| **lrd_open_connection** | 连接到数据库服务器 |
| **lrd_open_cursor** | 打开数据库光标 |
| **lrd_stmt** | 将 SQL 语句与光标关联 |
| **lrd_bind_col** | 将主机变量绑定到列 |
| **lrd_exec** | 执行 SQL 语句 |
| **lrd_fetch** | 在结果集中提取下一条记录 |
| **lr_commit** | 提交数据库事务 |
| **lr_close_cursor** | 关闭光标 |

| | |
|---|---|
| **lrd_close_connection** | 断开与数据库服务器的连接 |
| **lrd_end** | 清理环境 |

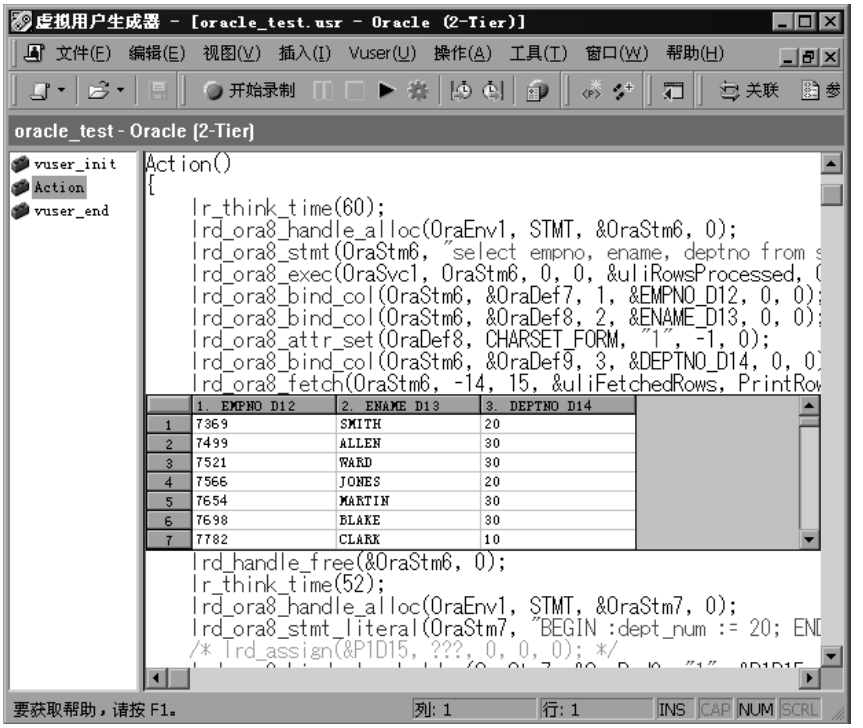
在以下脚本中， VuGen 录制了打开 Oracle 服务器连接的操作者的操作，然后执行对本地设置进行请求的查询。

```
lrd_init(&InitInfo, DBTypeVersion);
lrd_open_connection(&Con1, LRD_DBTYPE_ORACLE, "s1", "tiger", "hp1",
"", 0, 0, 0);
lrd_open_cursor(&Csr1, Con1, 0);
lrd_stmt(Csr1, "select parameter, value     from v$nls_parameters "
      "    where (upper(parameter) in ('NLS_SORT','NLS_CURRENCY',"
      "'NLS_ISO_CURRENCY', 'NLS_DATE_LANGUAGE',"
      "'NLS_TERRITORY'))", -1, 0 /*Non deferred*/, 1 /*Dflt Ora Ver*/, 0);
lrd_bind_col(Csr1, 1, &D1, 0, 0);
lrd_bind_col(Csr1, 2, &D2, 0, 0);
lrd_exec(Csr1, 0, 0, 0, 0, 0);
lrd_fetch(Csr1, 7, 7, 0, PrintRow2, 0);
…
lrd_close_cursor(&Csr1, 0);
lrd_commit(0, Con1, 0);
lrd_close_connection(&Con1, 0, 0);
lrd_end(0);
```

### 网格

在录制会话过程中，数据库查询所返回的数据将显示在网格中。通过查看网格，您可以确定应用程序生成 SQL 语句的方式以及客户端 / 服务器系统的效率。

在下例中， VuGen 显示对雇员数据库执行的查询的网格。查询使用标题 “工程师”来为所有雇员检索姓名、 ID 和职务。




网格的列宽可调整。您可以使用滚动条向上滚动 100 行。

要显示或隐藏网格，请选择 “视图” > “数据网格”。

## 行信息

VuGen 为每个 SQL 查询都生成 **lrd_fetch** 函数。

```
lrd_fetch(Csr1, -4, 1, 0, PrintRow7, 0);
```

该函数的第二个参数表明提取的行数。该数字可以为正也可以为负。

### 正行值

正值显示录制过程中提取的行值，并表明并未提取所有的行。（例如，当操作者在查询完成之前将其取消时。）

在下例中，在数据库查询过程中检索了四行，但并未提取所有的数据。

```
lrd_fetch(Csr1, 4, 1, 0, PrintRow7, 0);
```

在执行过程中，脚本总是检索正值所表示的行数 （假设存在这些行）。

### 负行值

负行值表示在录制过程中提取了所有可用行。负行数的绝对值就是提取的行数。

在下例中，检索了结果集的全部四行：

```
lrd_fetch(Csr1, -4, 1, 0, PrintRow7, 0);
```

当执行包含负行值的 lrd_fetch 语句时，它会在运行时检索表中的所有可用行，而不必一定是在录制时的行数。在上例中，在录制会话过程中检索了表中的全部四行。但是，如果在脚本执行过程中有更多行可用，则它们全都将被检索。

有关 **lrd_fetch** 的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

## 评估错误代码

Vuser 在执行 LRD 函数时，该函数总是会生成返回代码。返回代码 0 表示该函数成功。例如，返回代码 0 表示结果集中有另外一行可用。如果发生错误，返回代码将指示错误类型。例如，返回代码 2014 表示在初始化过程中出现错误。

返回代码有四种类型，均由数值范围表示：

| 返回代码的类型 | 范围 |
|---|---|
| 信息性 | 0 到 999 |
| 警告 | 1000 到 1999 |

| 返回代码的类型 | 范围 |
|---|---|
| 错误 | 2000 到 2999 |
| 内部错误 | 5000 到 5999 |

有关返回代码的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

您可以评估 LRD 函数的返回代码，以确定该函数是否成功。以下脚本段评估 **lrd_fetch** 函数的返回代码：

```
static int rc;
rc=lrd_fetch(Csr15, -13, 0, 0, PrintRow4, 0);
if (rc==0)
      lr_output_message("The function succeeded");
else
      lr_output_message("The function returned an error code:%d",rc);
```

## 处理错误

您可以在运行数据库 Vuser 脚本时控制数据库 Vuser 处理错误的方式。默认情况下，如果脚本执行过程中发生错误，该脚本的执行将终止。要更改默认行为，可以指示 Vuser 在发生错误时仍继续。您可以应用下面的行为：

- 全局 – 应用于整个脚本或脚本中的一段
- 本地 – 仅应用于特定函数

### 全局修改错误处理

您可以发出 LRD_ON_ERROR_CONTINUE 或 LRD_ON_ERROR_EXIT 语句来更改 Vuser 处理错误的方式。默认情况下，当遇到与数据库、参数等相关的任何错误类型时， Vuser 将中止脚本执行。要更改此默认行为，请将下行插入到脚本中：

```
LRD_ON_ERROR_CONTINUE;
```

从此时起， Vuser 将继续执行脚本，甚至发生错误时也不例外。

您还可以指定仅当错误发生在脚本的某一段内时， Vuser 才继续执行脚本。例如，以下代码指示 Vuser 继续脚本执行，即使在 **lrd_stmt** 或 **lrd_exec** 函数中发生了错误：

```
LRD_ON_ERROR_CONTINUE;
lrd_stmt(Csr1, "select..."...);
lrd_exec(...);
LRD_ON_ERROR_EXIT;
```

请谨慎使用 LRD_ON_ERROR_CONTINUE 语句，因为可能会忽略重大和严重的错误。

### 本地修改错误处理

您可以通过修改严重级别来为特定函数设置错误处理。执行数据库操作的函数（例如 **lrd_stmt** 和 **lrd_exec**）可以使用严重级别。严重级别由函数的最后一个参数 *miDBErrorSeverity* 表示。该参数指示 Vuser 是否在数据库发生错误时 （错误代码 2009）仍继续执行脚本。默认值 0 表示当发生错误时， Vuser 应该中止脚本。

例如，如果以下数据库语句失败 （例如，表不存在），则脚本的执行将终止。

```
lrd_stmt(Csr1, "insert into EMP values ('Smith',301)\n", -1, 1 /*Deferred*/,
        1 /*Dflt Ora Ver*/, 0);
```

要指示 Vuser 即使在发生该函数的数据库操作错误时仍然继续执行脚本，请将语句的严重度参数从 0 更改为 1。

```
lrd_stmt(Csr1, "insert into EMP values ('Smith',301)\n", -1, 1 /*Deferred*/,
        1 /*Dflt Ora Ver*/, 1);
```

当严重度设置为 1 且发生数据库错误时，将发出警告。注意，为特定函数设置的严重级别仅适用于该函数。

### CtLib 结果集错误

在 CtLib 录制中，应用程序在执行语句后将检索所有可用的结果集。如果返回的结果集包含可提取数据，则应用程序将对该数据执行绑定和提取操作，如下例所示：

```
lrd_stmt(Csr15, "select * from all_types", -1, 148, -99999, 0);
lrd_exec(Csr15, 0, 0, 0, 0, 0);
lrd_result_set(Csr15, 1 /*Succeed*/, 4040 /*Row*/, 0);
lrd_bind_col(Csr15, 1, &tinyint_D41, 0, 0);
…
lrd_fetch(Csr15, -9, 0, 0, PrintRow3, 0);
```

如果结果集不包含可提取数据，绑定和提取操作将无法执行。

当参数化脚本时，结果数据有可能变得不可提取 （取决于参数）。因此，如果新数据不可提取，那么为特定语句录制绑定和提取操作的 CtLib 会话将无法运行。如果试图执行 **lrd_bind_col** 或 **lrd_fetch** 操作，将会发生 LRDRET_E_NO_FETCHABLE_DATA 错误 （错误代码 2064）， Vuser 将终止脚本的执行。

通过指示 Vuser 在发生此类错误时仍然继续执行脚本，您可以覆盖错误。将以下行插入脚本中：

```
LRD_ON_FETCHABLE_SET_ERR_CONT;
```

要返回到终止脚本执行的默认模式下，请将以下行键入脚本中：

```
LRD_ON_FETCHABLE_SET_ERR_EXIT;
```

请谨慎使用此语句，因为可能会忽略重大和严重的错误。

# 19

# 关联数据库 Vuser 脚本

录制数据库会话后，可能需要将脚本中的一个或多个查询关联起来，以便在数据库会话稍后部分使用前面检索到的值。

本章描述：

- 在脚本中扫描关联
- 关联已知值
- 数据库关联函数

***下列信息仅适用于数据库（CtLib、DbLib、Informix、Oracle、ODBC 和 DB2-CLI）Vuser 脚本。***

## 关于关联数据库 Vuser 脚本

如果在运行脚本时遇到错误，请检查该脚本中发生错误的位置。多数情况下，可以通过关联查询并使用一个语句的结果作为另一个语句的输入来解决问题。关联数据库 Vuser 脚本的两个主要原因是：

- 重复值非法

假设脚本在数据库中创建了一个新职员记录，并为每个职员分配了唯一的 ID。数据库要求每个记录必须唯一，不能重复。尝试回放该脚本时，操作会失败，因为在录制会话过程中已经创建了该职员 ID，它不能重复。

要解决该问题，可以使用关联来捕获分配给新职员的 ID，并在数据库会话其余部分中使用它。此外，还可以使用参数化来检索每个职员的唯一数据。（请参阅第 7 章 “定义参数”）。

➤ 简化或优化代码

如果连续执行一系列相互依赖的查询，则代码可能会很长。要减小代码的大小，可以嵌套查询，但那样做会丧失精确性，并且代码将变得很复杂且难于理解。通过关联语句，可以将查询链接在一起而无需采用嵌套。

## 在脚本中扫描关联

VuGen 提供了一个关联实用程序，用于帮助您修复脚本以确保成功回放。该实用程序执行下列步骤：

➤ 扫描可能的关联

➤ 插入适当的关联函数以便将结果保存在参数中

➤ 将语句值替换为参数

可以在整个脚本中或在脚本的特定位置上执行自动关联。

本节描述如何确定需要关联的语句。如果已经知道要关联的值，请转到下一节以了解如何关联特定的值。

**要扫描并使用自动关联来关联脚本，请执行下列操作：**

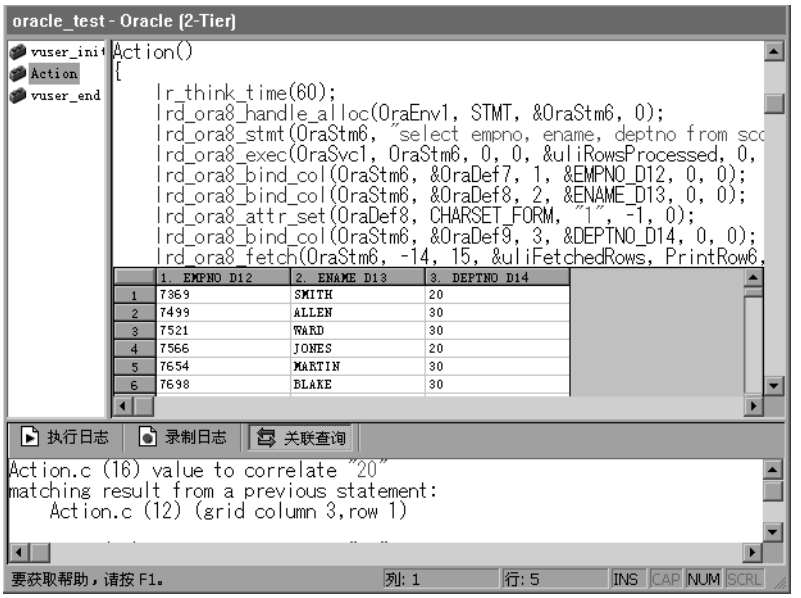
**1** 打开 “输出” 窗口。

选择 “视图” > “输出” 以便在窗口底部显示输出选项卡。检查 “执行日志” 文件夹中是否存在错误。这些错误通常可以通过关联来更正。

**2** 选择 “Vuser” > “扫描关联”。

VuGen 将扫描整个脚本，并在 “关联查询” 选项卡中列出所有可能要关联的值。

在下例的 **lrd_ora8_stmt_literal** 语句中， VuGen 检测到一个要关联的值。




**3** 在 “关联查询”选项卡列出的结果中，双击要关联的值。该结果位于文本消息的第三行，它显示 “grid column x, row x”。 VuGen 会将光标发送到该值在脚本中的网格位置。

**4** 在网格中，选择要关联的值，然后选择 “Vuser” > “创建关联”。 VuGen 将提示您输入结果值的参数名称。

**5** 指定一个名称或接受默认名称。单击 “确定”继续操作。 VuGen 会插入适当的关联语句（**lrd_save_value**、 **lrd_save_col** 或 **lrd_save_ret_param**），以便将结果保存在参数中。

6 单击 “是”确认关联。

将显示一个消息，询问您是否要在脚本中搜索出现该值的所有位置。

7 单击 “否”只替换选定语句中的值。

8 要搜索出现该值的其他位置，请单击 “是”。将打开 “搜索和替换”对话框。

9 确认所有替换 （包括原始语句）。关闭 “搜索和替换”对话框。

VuGen 将使用参数引用替换该语句值。注意，如果选择取消关联， VuGen 还会清除在上一步中创建的语句。

## 关联已知值

如果知道需要关联的值，请执行以下过程。

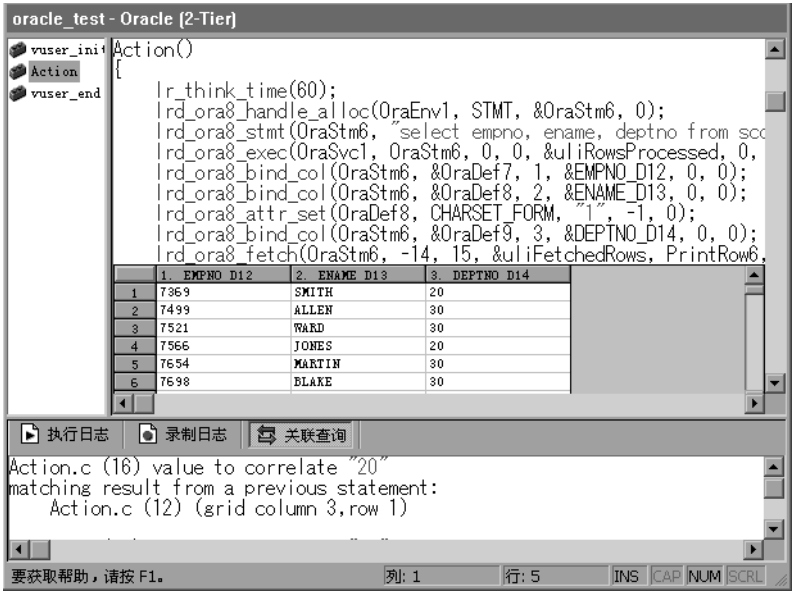
**要关联特定值，请执行下列步骤：**

1 找到要关联的值，并选择该值 （不带引号）。

2 选择 “Vuser” > “扫描关联 （光标处）”。

VuGen 将扫描该值并列出脚本中与该值匹配的所有结果。“关联查询”选项卡中将列出关联值。

在以下示例中， VuGen 找到几个与 “20”关联的匹配结果值。




**3** 在 “关联查询”选项卡中，双击要关联的结果。该结果位于消息的第三行，它显示“grid column x, row x”。VuGen 会将光标发送到该值在脚本中的网格位置。

**4** 在网格中，选择要关联的值，然后选择 “Vuser” > “创建关联”。 VuGen 将提示您输入参数名称。

**5** 指定一个名称或接受默认名称。单击 “确定”继续操作。 VuGen 会插入适当的关联语句（**lrd_save_value**、 **lrd_save_col** 或 **lrd_save_ret_param**），以便将结果保存在参数中。

6 单击 “是”确认关联。

将显示一个消息，询问您是否要在脚本中搜索出现该值的所有位置。

7 单击 “否”只替换选定语句中的值。

8 要搜索出现该值的其他位置，单击 “是”。将打开 “搜索和替换”对话框。

9 确认所有替换 （包括原始语句）。关闭 “搜索和替换”对话框。

VuGen 将使用参数引用替换该语句值。注意，如果选择撤消关联， VuGen 将清除在上一步中创建的语句。

---

**注意**：如果要关联 **lrd_stmt** 函数中的值，则不支持下列数据类型：日期、时间和二进制 （RAW、 VARRAW）。

---

## 数据库关联函数

使用数据库 Vuser 脚本（DbLib、CtLib、Oracle、Informix 等）时，可以使用 VuGen 的自动关联功能在脚本中插入适当的函数。关联函数包括：

- **lrd_save_col** 将网格中的查询结果保存在参数中。该函数在提取数据之前放置。它会将随后的 **lrd_fetch** 检索到的值分配给指定的参数。
- **lrd_save_value** 将占位符描述符的当前值保存在参数中。该函数与设置输出占位符的数据库函数 （如 Oracle 中的某些存储过程）配合使用。
- **lrd_save_ret_param** 将存储过程的返回值保存在参数中。该函数主要与存储在 DbLib 中的、生成返回值的数据库过程配合使用。

---

**注意**：如果保存的值无效或为 NULL （不返回行），则 VuGen 将不应用关联。

---

有关这些函数及其参数的详细信息，请参阅《LoadRunner Function Reference》。

# 20

# 开发 DNS Vuser 脚本

使用 VuGen 可以通过直接访问 DNS 服务器模拟网络活动。

本章描述：

➤ 使用 DNS 函数

***以下信息仅适用于 DNS 虚拟用户脚本。***

## 关于开发 DNS Vuser 脚本

DNS 协议是一种低级协议，通过它可以模拟使用 DNS 服务器的用户的操作。

DNS 协议模拟用户访问域名服务器，以将主机名解析为 IP 地址。此协议仅支持回放 – 需要手动向脚本中添加函数。

要为 DNS 协议创建脚本，请选择 “文件” > “新建”，打开 “新建虚拟用户” 对话框。在 “客户端 / 服务器” 类别中选择 “域名解析 (DNS)” 协议类型。由于 DNS 不支持录制，请用适当的 DNS、 LoadRunner 和 C 函数来编写脚本。有关这些函数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

创建 Vuser 脚本后，在 Windows 或 UNIX 平台上都可将其集成到场景中。有关将 Vuser 脚本集成到场景的详细信息，请参阅 《LoadRunner Controller 用户指南》。

# 使用 DNS 函数

DNS Vuser 脚本函数用于录制发送到或来自域名解析 (DNS) 服务器的查询。每个 DNS 函数都以 **dns** 前缀开头。有关这些函数的详细语法信息，请参阅《LoadRunner Function Reference》（“帮助” > “函数参考”）。

| 函数名 | 描述 |
| --- | --- |
| **ms_dns_query** | 解析主机的 IP 地址 |
| **ms_dns_nextresult** | 前进到 **ms_dns_query** 所返回列表中的下一个 IP 地址 |

下例中会向 DNS 服务器提交一个查询，并将结果打印到日志文件。

```
Actions()
{
int    rescnt = 0;
      char results = NULL;
      results = (char *) ms_dns_query("transaction",
                                      "URL=dns://<DnsServer>",
                                      "QueryHost=< 主机名 >",
                                      LAST);

      // 列出主机名的所有 IP 地址……
      while (*results) {
           rescnt++;
           lr_log_message(lr_eval_string("(%d) IP of< 主机名 > is %s"),
                     rescnt, results);
           results = (char *) ms_dns_nextresult(results);
      }
      return 1;
}
```

# 21

# 开发 WinSock Vuser 脚本

对于采用 Windows 套接字协议进行通信的客户端应用程序和服务器，可以使用 VuGen 录制它们之间的通信。生成的脚本称为 Windows 套接字 Vuser 脚本。

本章描述：

- Windows 套接字 Vuser 脚本入门
- 设置 WinSock 录制选项
- 使用 LRS 函数
- 在树视图和脚本视图之间切换

***下列信息适用于 Windows 套接字级别上录制的所有协议，包括 Web/Winsock 双协议。***

## 关于录制 Windows 套接字 Vuser 脚本

Windows 套接字协议非常适合于分析应用程序的低级代码。例如，要检查网络，可以使用 Windows 套接字 (WinSock) 脚本来查看缓冲区发送和接收到的实际数据。 WinSock 类型还可以用于录制其他低级通信会话。此外，您可以录制和回放其他任何 Vuser 类型都不支持的应用程序。

录制使用 Windows 套接字协议的应用程序时，VuGen 将生成描述所录制的操作的函数。每个函数都以 **lrs** 前缀开头。 LRS 函数与套接字、数据缓冲区和 Windows 套接字环境相关。使用 VuGen，您可以录制应用程序对 Winsock.dll 或 Wsock32.dll 的 API 调用。例如，可以通过录制 *telnet* 应用程序的操作来创建脚本。

在下例中， **lrs_send** 将数据发送到指定的套接字：

```
lrs_send("socket22", "buf44", LrsLastArg);
```

在 VuGen 的主窗口中，可以查看并编辑录制的脚本。会话期间录制的 Windows 套接字 API 调用将显示在窗口中，可以用于跟踪网络活动。

VuGen 可以以两种方式显示 WinSock 脚本：

- 作为基于图标的脚本表示形式。这是默认的视图，称为树视图。
- 作为基于文本的脚本表示形式，它显示 Windows 套接字 API 调用。这称为脚本视图。

可以使用 VuGen 来查看和编辑脚本的树视图和脚本视图。您可以方便地在两种视图之间切换。详细信息，请参阅第 252 页的 “在树视图和脚本视图之间切换”。

## Windows 套接字 Vuser 脚本入门

本节概述使用 VuGen 开发 Windows 套接字 Vuser 脚本的过程。

**要开发 Windows 套接字脚本，请执行下列操作：**

**1 使用 VuGen 录制操作**

调用 VuGen 并新建 Vuser 脚本，将类型指定为 Windows 套接字。选择要录制的应用程序，并设置录制选项。录制应用程序的典型操作。

详细信息，请参阅第 3 章 “使用 VuGen 进行录制”。

**2 增强 Vuser 脚本**

通过将事务、集合点和控制流结构插入到脚本中，可以增强 Vuser 脚本。

详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

**3 定义参数 （可选）**

为录制到 Vuser 脚本中的固定值定义参数。通过将这些固定值替换为参数，可以使用不同的值多次重复相同的业务流程。

详细信息，请参阅第 7 章 “定义参数”。

**4 关联语句（可选）**

通过关联语句，可以在后面的业务流程中使用前一个流程的结果。

详细信息，请参阅第 8 章 “关联语句”。

**5 配置运行时设置**

运行时设置控制脚本执行过程中 Vuser 的行为。这些设置包括循环、日志和计时信息。

详细信息，请参阅第 9 章 “配置运行时设置”和第 10 章 “配置网络运行时设置”。

**6 从 VuGen 中运行该 Vuser 脚本**

在 VuGen 中保存并运行 Vuser 脚本，验证该脚本是否可以正常运行。

详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

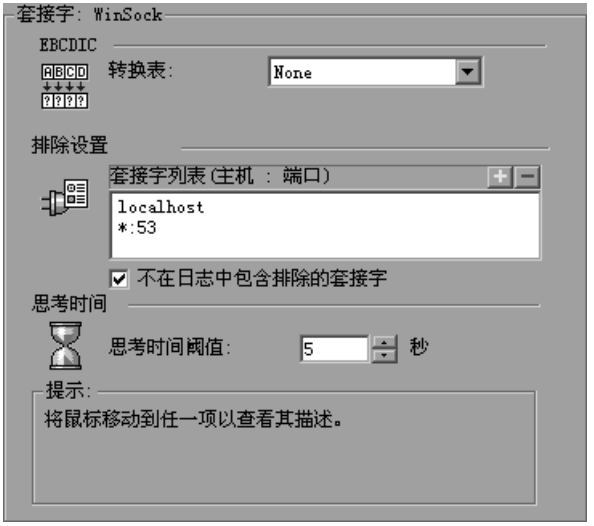
创建 Windows 套接字 Vuser 脚本后，在 Windows 或 UNIX 平台上都可以将其集成到场景中。有关将 Vuser 脚本集成到场景的详细信息，请参阅《LoadRunner Controller 用户指南》。

# 设置 WinSock 录制选项

下列录制选项可用于 WinSock Vuser：

- 配置转换表
- 排除套接字
- 设置思考时间阈值

要打开 “录制选项”对话框，请选择 “工具” >“录制选项”，或在 “开始录制”对话框中单击 “选项”按钮。 VuGen 将显示 WinSock 选项。




## 配置转换表

要以 EBCDIC 格式显示数据，请在录制选项中指定转换表。

通过转换表可以指定录制会话的格式。这适用于在大型机或 AS/400 服务器上运行的用户。服务器和客户端计算机都从系统上所安装的转换表中确定数据的格式。可以从列表框中选择转换选项。

列表框项的前四位数字表示服务器格式。后四位表示客户端格式。上例中，选定的转换表为 *002501b5*。服务器格式为 *0025*，客户端格式为 *01b5*，该格式指示从服务器传输到客户端。在从客户端到服务器的传输中，应该选择与该格式相反的项 – *01b50025*，指明客户端的 *01b5* 格式需要转换成服务器的 *0025* 格式。

转换表位于 LoadRunner 安装目录下的 *ebcdic* 目录中。如果系统使用另外的转换表，请将其复制到 *ebcdic* 目录下。

---

**注意**：如果数据是 ASCII 格式，则不需要转换。此时您必须选择默认的 “无” 选项。如果确实选择了转换表，则 VuGen 将转换该 ASCII 数据。

---

使用 Solaris 计算机时，必须设置下列环境变量：在运行 Vuser 脚本的所有计算机上。

```
setenv LRSDRV_SERVER_FORMAT 0025
setenv LRSDRV_CLIENT_FORMAT 04e4
```

### 排除套接字

VuGen 支持 “排除套接字” 功能，您可以从录制会话中排除特定的套接字。要从脚本中排除某个套接字的所有操作，请在 “排除套接字” 列表中指定该套接字的地址。要向列表中添加套接字，请单击该框右上角的加号，然后以下列格式之一输入套接字地址：

| 值 | 含义 |
|---|---|
| 主机 : 端口 | 仅排除指定主机上的指定端口 |
| 主机 | 排除指定主机上的所有端口 |
| : 端口 | 排除本地主机上的指定端口号 |
| *: 端口 | 排除所有主机上的指定端口号 |

可以通过将多个主机和端口添加到列表中来排除。要从排除列表中删除套接字，请选择该套接字地址，然后单击该框右上角的减号。建议您排除不影响测试期间服务器负载的主机和端口，例如在默认情况下即会排除本地主机和 DNS 端口 (53)。

默认情况下， VuGen 不记录在 “已排除套接字列表”中排除的套接字的操作。要指示 VuGen 记录已排除的套接字的操作，请清除 “日志中不包括已排除的套接字”复选框。如果对已排除的套接字启用了日志记录，则在日志文件中，这些套接字的操作之前会加上 “Exclude”字样。

```
Exclude :/* recv():15 bytes were received from socket 116 using flags 0 */
```

### 设置思考时间阈值

录制期间， VuGen 会自动插入操作者的思考时间。您可以设置阈值级别，录制的思考时间如果低于此阈值，则被忽略。如果录制的思考时间超出了阈值级别，VuGen 将在 LRS 函数之前放置 **lr_think_time** 语句。如果录制的思考时间低于阈值级别，则不会生成 **lr_think_time** 语句。

要设置思考时间阈值，请在 “思考时间阈值”框中输入所需的值 （以秒为单位）。默认值为五秒。

## 使用 LRS 函数

为模拟客户端和服务器之间使用 Windows 套接字协议进行通信的情况而开发的函数称为 LRS Vuser 函数。每个 LRS Vuser 函数都有 **lrs** 前缀。 Windows 套接字会话期间， VuGen 将自动录制本节中列出的大多数 LRS 函数。您还可以在 Vuser 脚本中手动编写任何函数。有关 LRS 函数的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

### 套接字函数

| | | |
|---|---|---|
| | **lrs_accept_connection** | 接受侦听套接字连接 |
| | **lrs_close_socket** | 关闭打开的套接字 |
| | **lrs_create_socket** | 初始化套接字 |
| | **lrs_disable_socket** | 禁用套接字操作 |
| | **lrs_exclude_socket** | 回放期间排除套接字 |

| | |
|---|---|
| **lrs_get_socket_attrib** | 获取套接字属性 |
| **lrs_get_socket_handler** | 获取指定套接字的套接字处理程序 |
| **lrs_length_receive** | 接收来自指定长度的缓冲区的数据 |
| **lrs_receive** | 接收来自套接字的数据 |
| **lrs_receive_ex** | 接收来自数据报或流套接字的数据（具有特定长度） |
| **lrs_send** | 将数据发送到数据报上或流套接字中 |
| **lrs_set_receive_option** | 设置套接字接收选项 |
| **lrs_set_socket_handler** | 设置特定套接字的套接字处理程序 |
| **lrs_set_socket_options** | 设置套接字选项 |

## 缓冲区函数

| | |
|---|---|
| **lrs_free_buffer** | 释放分配给缓冲区的内存 |
| **lrs_get_buffer_by_name** | 从数据文件中获取缓冲区及其大小 |
| **lrs_get_last_received_buffer** | 获取套接字上接收到的最后的缓冲区及其大小 |
| **lrs_get_last_received_buffer_size** | 获取套接字上接收到的最后一个缓冲区的大小 |
| **lrs_get_received_buffer** | 获取最后接收到的缓冲区或其一部分 |
| **lrs_get_static_buffer** | 获取静态缓冲区或其一部分 |
| **lrs_get_user_buffer** | 获取套接字的用户数据的内容 |

| | |
|---|---|
| **lrs_get_user_buffer_size** | 获取套接字的用户数据的大小 |
| **lrs_set_send_buffer** | 指定要在套接字上发送的缓冲区 |

### 环境函数

| | |
|---|---|
| **lrs_cleanup** | 终止 Windows 套接字 DLL 的使用 |
| **lrs_startup** | 初始化 Windows 套接字 DLL |

### 关联语句函数

| | |
|---|---|
| **lrs_save_param** | 将静态或接收到的缓冲区 （或缓冲区部分）保存到参数中 |
| **lrs_save_param_ex** | 将用户、静态或接收到的缓冲区（或缓冲区部分）保存到参数中 |
| **lrs_save_searched_string** | 在静态或接收到的缓冲区中搜索出现的字符串，将出现字符串的缓冲区部分保存到参数中 |

### 转换函数

| | |
|---|---|
| **lrs_ascii_to_ebcdic** | 将缓冲区数据从 ASCII 格式转换成 EBCDIC 格式 |
| **lrs_decimal_to_hex_string** | 将十进制整数转换为十六进制字符串 |
| **lrs_ebcdic_to_ascii** | 将缓冲区数据从 EBCDIC 格式转换成 ASCII 格式 |
| **lrs_hex_string_to_int** | 将十六进制字符串转换为整数 |

### 超时函数

**lrs_set_accept_timeout** 为接受套接字设置超时

**lrs_set_connect_timeout** 为连接到套接字设置超时

**lrs_set_recv_timeout** 为接收套接字上的初始预期数据设置超时

**lrs_set_recv_timeout2** 为建立连接后接收套接字上的预期数据设置超时

**lrs_set_send_timeout** 为发送套接字数据设置超时

录制会话之后，通过 VuGen 的内置编辑器可以查看录制的代码。您可以在脚本中滚动，查看应用程序生成的函数，并检查传输的数据。在主窗口中查看脚本时，可以看到 VuGen 录制活动的顺序。在典型的会话期间，将录制下列函数顺序：

| | |
|---|---|
| **lrs_startup** | 初始化 WinSock DLL |
| **lrs_create_socket** | 初始化套接字 |
| **lrs_send** | 在数据报上或者向流套接字发送数据 |
| **lrs_receive** | 接收来自数据报或流套接字的数据 |
| **lrs_disable_socket** | 禁用套接字操作 |
| **lrs_close_socket** | 关闭打开的套接字 |
| **lrs_cleanup** | 终止 WinSock DLL 的使用 |

VuGen 在 Windows 上使用 Windows 套接字协议支持应用程序的录制和回放；而在 UNIX 平台上仅支持回放。

# 在树视图和脚本视图之间切换

在 VuGen 中查看和编辑 Windows 套接字 Vuser 脚本时，可以选择是以基于图标的树视图来查看脚本，还是以基于文本的脚本视图来查看脚本。

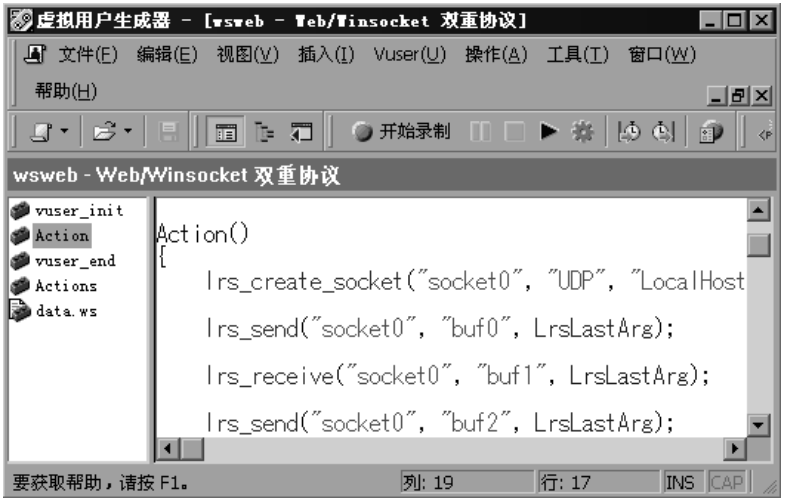
**要显示 WinSocket Vuser 脚本的树视图，请执行下列操作：**

在 VuGen 主菜单中选择 “视图” > “树视图”，或者单击 “按树查看脚本” 图标。

将在树视图中显示 Vuser 脚本的 “Actions” 部分。如果您已经位于此视图中，则该菜单项将被禁用。

**要显示脚本视图，请执行下列操作：**

在 VuGen 主菜单中选择 “视图” > “脚本视图”，或者单击 “按文本查看脚本” 图标。 Vuser 脚本即会显示在基于文本的脚本视图中。如果已经位于脚本视图中，则该菜单项将被禁用。




创建脚本之后，可以以快照或原始数据文件的形式查看数据。详细信息，请参阅第 22 章 “使用 Window 套接字数据”。

# 22

# 使用 Window 套接字数据

在 Windows 套接字协议中录制会话之后，您可以查看并操纵数据。

本章描述：

- 查看快照窗口中的数据
- 在数据中导航
- 修改缓冲区数据
- 修改缓冲区名称
- 了解数据文件格式
- 查看十六进制格式的缓冲区数据
- 设置显示格式
- 调试提示
- 手动关联 WinSock 脚本

***下列信息适用于所有在 Windows 套接字级别上录制的协议。***

## 关于使用 Windows 套接字数据

使用 VuGen 录制应用程序之后，您就拥有多个包含数据的数据缓冲区。

在树视图中查看 WinSocket 脚本时， VuGen 提供快照窗口，以便您在数据缓冲区内进行导航以及修改数据。

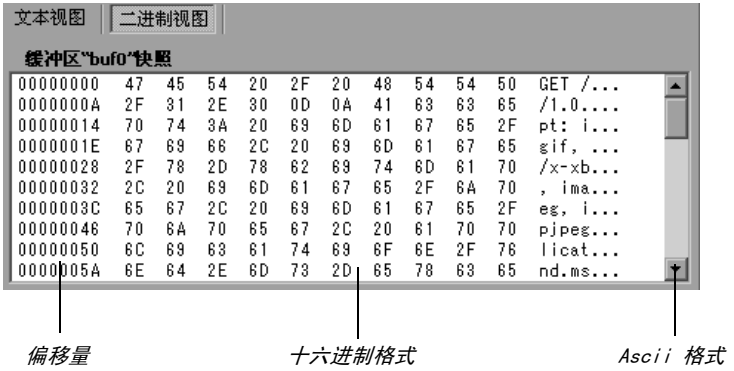
使用脚本视图时，可以在 *data.ws* 文件中查看原始数据。详细信息，请参阅第 267 页的 “在脚本视图中查看 Windows 套接字数据”。

# 查看快照窗口中的数据

在树视图中查看 Windows 套接字脚本时，VuGen 提供缓冲区快照窗口，在可编辑的窗口中显示数据。您可以在文本视图或二进制视图中查看快照。

文本视图显示缓冲区快照，以文本形式表示其内容。

二进制视图显示以十六进制表示的数据。左列显示该行中第一个字符的偏移量。中间的列显示数据的十六进制值。右列以 ASCII 格式显示数据。

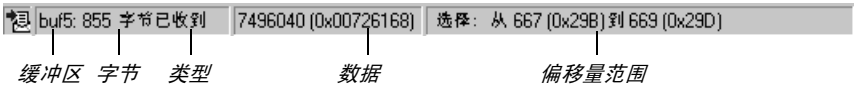



缓冲区快照下面的状态栏提供有关数据和缓冲区的下列信息：

- **缓冲区编号**：选定缓冲区的缓冲区编号。
- **总字节数**：缓冲区中的总字节数。
- **缓冲区类型**：缓冲区的类型 – 接收或发送。

➤ **数据**：以十进制和十六进制格式以及“小端”顺序（与在缓冲区中显示的方式相反）表示的光标处数据的值。

➤ **偏移量**：所选内容（或文本视图中的光标）距缓冲区开始处的偏移量。如果选择多个字节，则它可以指明所选内容的范围。




状态栏还可指明是否修改了原始数据。

## 在数据中导航

在树视图中， VuGen 提供了几个用于在数据中导航的工具，以便标识和分析特定的值。

➤ 缓冲区导航器

➤ 转至偏移量

➤ 书签

## 缓冲区导航器

默认情况下， VuGen 在左窗格中显示所有的步骤和缓冲区。缓冲区导航器是一个浮动窗口，通过它仅可以显示接收和发送缓冲区步骤（**lrs_send**、 **lrs_receive**、**lrs_receive_ex** 和 **lrs_length_receive**）。此外，您还可以应用筛选器，以查看发送或接收缓冲区。

在导航器中选择缓冲区时，其内容将显示在缓冲区快照窗口中。

如果在录制之后更改缓冲区的名称，在步骤上单击时，其内容将不会显示在快照窗口中。要查看已重命名的缓冲区的数据，请使用缓冲区导航器，并选择新缓冲区的名称。 VuGen 将发出警告消息，指明选定的缓冲区将禁止创建参数。

要打开缓冲区导航器，请选择 “视图” > “缓冲区导航器”。要关闭导航器，请单击导航器对话框右上角的 X。

注意，通过在左窗格树视图中的缓冲区步骤上单击，还可以在缓冲区之间导航。缓冲区导航器的优势在于，它是具有筛选功能的浮动窗口。

## 转至偏移量

通过指定偏移量，可以在数据缓冲区中移动。您可以指示数据的绝对位置，也可以指示与缓冲区中光标当前位置相对的位置。通过此对话框，还可以指定开始和结束偏移量选择数据的范围。

**要转至偏移量，请执行下列操作：**

1. 在快照窗口中进行单击。然后从右键单击菜单中选择 “转至偏移量”。将打开 “转至偏移量”对话框。

2. 要转至缓冲区中特定的偏移量 （绝对偏移量），请单击 “转至偏移量”，然后指定偏移量值。
3. 要跳至与光标相对的位置，请单击 “前进”，然后指定想要前进的字节数。要前进，请输入正值。要在缓冲区内后退，请使用负值。
4. 要选择缓冲区中数据的范围，请单击 “选择范围：从”，指定开始和结束偏移量。

## 书签

通过 VuGen，可以将缓冲区中的位置标记为书签。可以为每个书签赋予一个描述性名称，单击该名称可以直接跳至该书签的位置。书签列在缓冲区快照下方“输出”窗口的 “书签”选项卡中。

文本和二进制视图中都可以使用书签。您可以在文本视图中查找需要的数据，并将该位置保存为书签，然后在二进制视图中即可直接跳至该书签。

书签可以标记单个字节或多个字节。在列表中单击书签时，它在缓冲区快照窗口中将作为选定内容显示。数据最初在文本视图中以蓝色突出显示，而在二进制视图中，书签块则以红色标记。将光标置于缓冲区中的书签上时，将出现一个弹出文本框，指明该书签的名称。

您可以创建永久书签和简单书签。永久书签总是在缓冲区的二进制视图中标记（以蓝色框括起来）。该书签总是保持蓝色选中状态，即使指向缓冲区中的另一个位置也不例外。光标位置以红色标记。但是简单书签不会永久标记。当跳至简单书签时，书签以红色标记，但是如果在缓冲区内移动光标，则该书签不会再选定。默认情况下，书签都是永久的。

使用书签：

1. 要创建书签，请在缓冲区快照 （文本或二进制视图）中选择一个或多个字节，并从右键单击菜单中选择 “新建书签”。
2. 要查看书签列表，请选择 “视图” > “输出窗口”，然后选择 “书签”选项卡。
3. 要为书签分配名称，请在书签列表中单击该书签，然后编辑标题。
4. 要更改书签的位置，请在 “书签”选项卡中选择该书签，然后在缓冲区快照中选择新数据。在 “书签”选项卡中，单击 “修改”。
5. 要将书签形式从永久更改为简单 （永久意味着始终被标记，即使将光标移至新位置），请选中该书签，右键单击，然后清除 “永久书签”旁边的复选框。

6 要在列表中仅显示永久书签，请在 “书签” 选项卡中选中 “仅显示永久标签” 复选框。

7 要查看特定缓冲区中的书签，请在需要的缓冲区中选择书签，然后在 “筛选器” 框中选择 “仅选定的缓冲区”。

8 要删除书签，请在 “书签” 选项卡中选择该书签，然后单击 “删除”。

## 修改缓冲区数据

在树视图中， VuGen 提供几个工具，使用这些工具，可以通过删除、更改或向现有数据中添加数据对数据进行修改。

- 插入数据
- 编辑数据
- 参数化数据

### 插入数据

数据缓冲区中可以插入数值。您可以插入单字节、双字节或 4 字节值。

**要向数据缓冲区中插入数字，请执行下列操作：**

1 在缓冲区中单击任意位置。

2 打开右键单击菜单，选择 “高级” > “插入数” > “指定 ...”

3 输入想要插入到 “值” 框中的 ASCII 值。

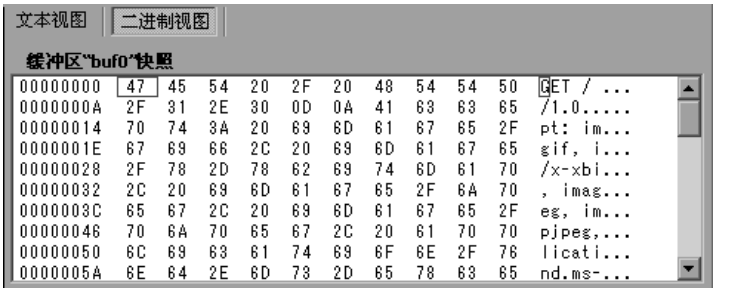
4 选择要插入的数据大小：从 “大小” 框中选择单字节、 2 字节或 3 字节。

5 单击 “确定” 结束。 VuGen 将以十六进制的形式将数据插入缓冲区。

## 编辑数据

可以对缓冲区数据执行下列所有标准编辑操作：复制、粘贴、剪切、删除和撤消。在二进制视图中可以指定要插入的实际数据。通过 VuGen 可以指定数据的格式：单字节、 2 字节或 4 字节以及十六进制或十进制值。您可以复制二进制数据，并将其作为数字插入缓冲区中。在二进制视图的右列，可以查看十进制或十六进制数字。

在下例中，选定了单词 “OK”。




如果在下一行数据的开始处执行简单的复制 (CTRL+C) 和粘贴 (CTRL+V) 操作，将插入实际的文本。

```
00000014 4F 4B 76 65 72 3A 20 4D 69 63 OKver: Mic
```

如果选择 “作为数字进行高级复制” > “十进制”，然后粘贴数据，则将插入选定字符的 ASCII 码的十进制值：

```
00000014 31 39 32 37 39 76 65 72 3A 20 19279ver:
```

如果选择 “作为数字进行高级复制” > “十六进制”，然后粘贴数据，则将插入选定字符的 ASCII 码的十六进制值：

```
00000014 30 78 34 42 34 46 76 65 72 3A 0x4B4Fver:
```

“撤消缓冲区”操作仍将保留对缓冲区所作的所有修改。此信息将随文件保存，如果关闭该文件，此信息仍然可用。如果要防止其他人撤消您的更改，可以清空 “撤消”缓冲区。要清空 “撤消”缓冲区，请在右键单击菜单中选择 “高级” > “清空撤消缓冲区”。

在二进制视图中编辑缓冲区数据：

1 要复制缓冲区数据，请执行下列操作：

- 对于字符，选择一个或多个字节，然后按 CTRL+C。
- 对于十进制数字，在右键单击菜单中选择 “高级” > “作为数字进行复制” > “十进制”。
- 对于十六进制数字，在右键单击菜单中，选择 “高级” > “作为数字进行复制” > “十六进制”。

2 要粘贴数据，请执行下列操作：

- 对于单字节 （假设剪贴板中的数据大小为单字节），在缓冲区中单击需要的位置，然后按 CTRL+V。
- 在短格式 （2 字节）中，从右键单击菜单中选择 “高级” > “插入数字” > “粘贴短格式 （2 字节）”。
- 在长格式 （4 字节）中，从右键单击菜单中选择 “高级” > “插入数字” > “粘贴长格式 （4 字节）”。

3 要删除数据，请在其中一个视图中选择该数据，然后从右键单击菜单中选择 “删除”。

## 参数化数据

在树视图中，通过 VuGen 可以直接从缓冲区快照视图中将数据参数化。您可以指定要参数化的范围，并指定边界。如果不为已参数化的字符串指定边界，VuGen 将向脚本中插入 **lrs_save_param** 函数。如果指定边界， VuGen 将向脚本中插入 **lrs_save_searched_string** 函数，因为通过此函数可以指定边界参数。注意， **lrs_save_param** 和 **lrs_save_searched_string** 函数会关联数据。这意味着接收到的数据将被存储，以供测试过程中稍后的时间点处使用。因为这种关联会存储接收到的数据，所以它仅适用于接收缓冲区而不适用于发送缓冲区。推荐的过程是，在要参数化的接收缓冲区中选择动态数据的字符串。在其后的发送缓冲区中使用相同的参数。

这种关联不应该与简单参数化相混淆。简单参数化 （“插入” > “新建参数”）仅适用于发送缓冲区中的数据。您可以设置一个参数，然后为其分配几个值。VuGen 在每次测试运行或循环中都使用不同的值。详细信息，请参阅第 7 章 “定义参数”。下面几部分仅讨论接收缓冲区中数据的关联。

创建参数之后， VuGen 会列出将使用参数替换字符串的所有位置。 VuGen 还提供有关创建该参数的信息，即创建参数的缓冲区以及其在该缓冲区中的偏移量。快照视图下方输出窗口的 “参数”选项卡中将列出出现的所有参数。

通过 VuGen 可以操纵参数：

**筛选**：可以按参数名筛选参数替换。

**转至源**：选择替换，再单击 “转至源”，跳至缓冲区中被替换参数的准确位置。

**删除**：可以删除任何参数。删除参数时， VuGen 用数据的原始值替换该数据，并从脚本中删除参数化函数。

**名称**：可以为每个替换提供一个名称。

**撤消替换**：您也可以撤消列表中显示的一个或多个替换。

要从快照窗口中参数化数据，请执行下列操作：

1 选择要参数化的数据，然后从右键单击菜单中选择 “创建参数”（仅可用于接收缓冲区）。将打开对话框：

2 在 “参数名”框中指定参数的名称。

3 选择要参数化的字符的范围。默认情况下， VuGen 会接受您在缓冲区中选择的数据范围。要选择显示在该对话框中的范围以外的范围，请单击 “选择范围”。将打开一个小对话框，指明选定的范围。

在缓冲区快照窗口中选择范围，然后单击 “完成”。

4 如果参数数据不是常量，而边界是一致的，则可以指定左右边界。

5 指定边界：

选中 “使用边界提取参数数据”复选框。 VuGen 将把 “脚本语句”部分中的函数从 **lrs_save_param** 更改为 **lrs_save_searched_string**。单击 “完成”。

单击 “边界”部分中 “左”框旁边的 “浏览”按钮。将打开小对话框，指明您在缓冲区中的选择。在缓冲区中选择边界，然后单击 “完成”。对于右边界，重复此步骤。

6 对 “脚本语句”部分中的参数做出需要的修改。例如，可以向 **lrs_save_param** 函数中添加 *_ex* 指定编码类型。有关这些函数的详细信息，请参阅《LoadRunner Function Reference》。

7 单击 “确定”，创建参数。 VuGen 在替换参数之前会要求您进行确认。单击 “是”。可以在 “参数”选项卡中查看所有的替换。

8 要跳至参数在其缓冲区中的原始位置，请选择该参数，然后单击 “转至源”。

9 要跳至选定替换的缓冲区位置，请选择该替换，然后单击 “转至”。

10 要删除整个参数，请在 “筛选器”框中选择该参数，然后单击 “删除参数”。

11 要撤消替换，请在 “参数”选项卡中选择该替换，然后单击 “撤消”。要撤消显示的参数的全部替换，请在 “参数”选项卡中选择该参数，然后单击 “全部撤消”。

12 撤消特定的替换后，标签将从 “已替换”变为 “找到”。要重新将参数化规则应用于被撤消替换的参数，请单击 “替换”或 “全部替换”。

13 要删除整个参数并撤消全部替换，请在 “筛选器”框中选择该参数，然后单击 “删除参数”。

14 选择 “Vuser” > “参数列表”，将数据分配给参数。

## 修改缓冲区名称

使用 *data.ws* 文件的脚本视图可以修改缓冲区的名称。如果在进行录制之后修改缓冲区名称，这将影响到 Vuser 脚本的回放。您可以使用缓冲区导航器在脚本视图或树视图中查看已重命名的缓冲区的内容。

如果在缓冲区中创建了书签，且其已不再可用， VuGen 会提示您在定义这些书签的缓冲区中删除它们。

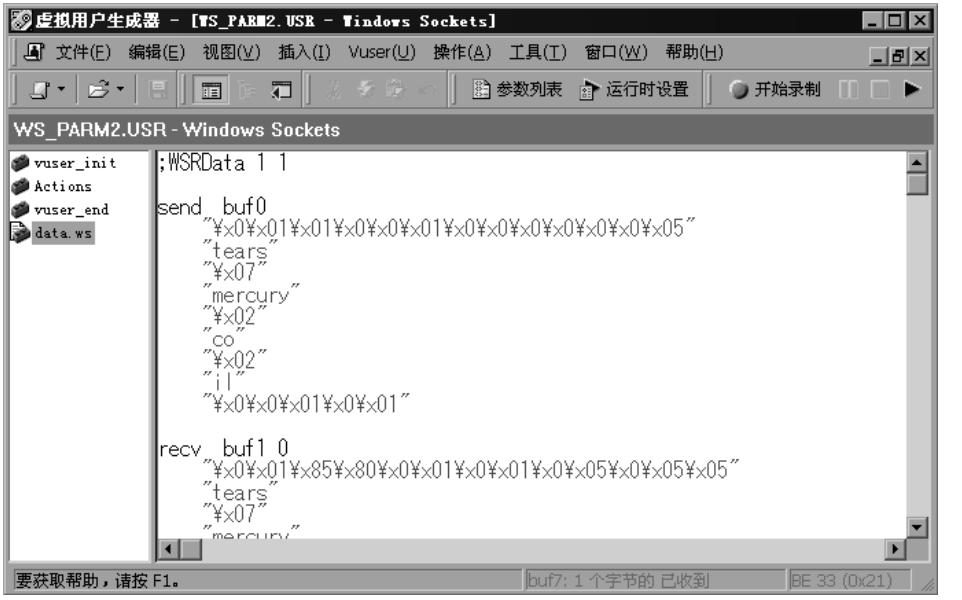
如果在缓冲区中创建了参数，且其已不再可用， VuGen 会提示您在定义这些参数的缓冲区中删除它们。删除参数时，将撤消全部替换，甚至是其他缓冲区中的替换。

在缓冲区导航器中查看重命名的缓冲区时， VuGen 会警告您该缓冲区中将禁止创建参数。

# 在脚本视图中查看 Windows 套接字数据

使用 VuGen 创建 Windows 套接字 Vuser 脚本时，您的操作将被录制到脚本的下列三个部分中：*vuser_init*、*Actions* 和 *vuser_end*。除了 Vuser 脚本外，VuGen 还将创建数据文件 *data.ws*，该文件包含在录制会话期间传输或收到的数据。通过在 VuGen 主窗口的 “数据文件” 框中选择 *data.ws* 文件，可以使用 VuGen 查看该数据文件的内容。

默认情况下，查看数据文件的选项可用于 Windows 套接字脚本。注意，只能在脚本视图中查看数据。




有几个 LRS 函数（例如 **lrs_receive** 和 **lrs_send**）可以处理服务器和客户端之间传输的实际数据。接收或传输的数据存储在数据缓冲区中，数据缓冲区可能会非常大。为了简化 Vuser 脚本的外观，实际数据存储在外部文件（而不是 C 文件）中。发生数据传输时，数据会从外部文件复制到临时缓冲区中。

外部文件 *data.ws* 包含所有临时缓冲区的内容。缓冲区的内容将存储为连续的记录。记录由标识符和缓冲区描述符标记，其中标识符用于指明数据是发送还是接收的。 LRS 函数使用缓冲区描述符访问数据。

该描述符具有下列格式之一：

recv buf*index* *接收的字节数*
send buf*index*

缓冲区索引从 0 （零）开始，后面的所有缓冲区都依次编号 （1、 2、 3 等），无论它们是发送还是接收缓冲区。

下例中，在 Vuser 会话期间录制了 **lrs_receive** 函数：

```
lrs_receive("socket1", "buf4", LrsLastArg)
```

在此例中， **lrs_receive** 处理了 *socket1* 上接收的数据。数据存储在第五个接收记录 (buf4) 中 （注意，索引号是从零开始的）。 *data.ws* 文件的相应部分显示该缓冲区及其内容。

```
recv buf4 39
     "\xff\xfb\x01\xff\xfb\x03\xff\xfd\x01"
     "\r\n"
     "\r\n"
     "SunOS UNIX (sunny)\r\n"
     "\r"
     "\x0"
     "\r\n"
     "\r"
     "\x0"
```

## 了解数据文件格式

*data.ws* 数据文件具有下列格式：

- 文件标头
- 缓冲区及其内容的列表

文件标头包括数据文件格式的内部版本号。当前版本为 2。如果试图访问格式版本为 1 的数据文件中的数据，则会发出错误消息。

```
;WSRData 2 1
```

每个记录之前都有一个标识符，指明数据是接收的还是发送的，标识符后接缓冲区描述符和接收的字节数 （仅适用于 **lrs_receive**）。缓冲区描述符包含用于标识缓冲区的数字。例如，

```
recv buf5   25
```

指明缓冲区包含接收的数据。记录编号为 5，指明此接收操作是第六次数据传输（索引从零开始），而接收的数据为二十五个字节。

如果数据是 ASCII 格式，则描述符后接套接字所传输的实际 ASCII 数据。

如果数据是 EBCDIC 格式，则必须经过查找表进行转换。有关设置转换表的信息，请参阅第 246 页的 “设置 WinSock 录制选项”其 ASCII 值 （转换后）可打印的 EBCDIC 数据将显示为 ASCII 字符。如果该 ASCII 值对应于不可打印的字符，则 VuGen 将显示原始的 EBCDIC 值。

```
recv buf6 39
     "\xff\xfb\x01\xff\xfb\x03\xff\xfd\x01"
     "\r\n"
     "SunOS UNIX (sunny)\r\n"
```

下一段显示典型数据文件的标头、描述符和数据：

```
;WSRData 2 1

send buf0
     "\xff\xfd\x01\xff\xfd\x03\xff\xfb\x03\xff\xfb\x18"

recv buf1 15
     "\xff\xfd\x18\xff\xfd\x1f\xff\xfd"
     "#"
     "\xff\xfd"
     "'"
     "\xff\xfd"
     "$"

send buf2
     "\xff\xfb\x18"
```

# 查看十六进制格式的缓冲区数据

VuGen 包含一个实用程序，通过该实用程序可以以下列方式查看数据段：以十六进制和 ASCII 格式显示数据，同时指明数据的偏移量。

要在查看器窗口中显示数据，请选中该数据，然后按 F7 键。如果选择的文本少于四个字符， VuGen 将以短格式显示该数据，显示其十六进制、十进制和八进制表示形式。

您可以在 *conv_frm.dat* 文件中自定义短格式，如第 272 页的 “设置显示格式” 所述。

如果选择的文本多于四个字符， VuGen 则会以长格式在数列中显示数据。您可以通过修改 *conv_frm.dat* 文件自定义长格式，如第 272 页的 “设置显示格式” 所述。

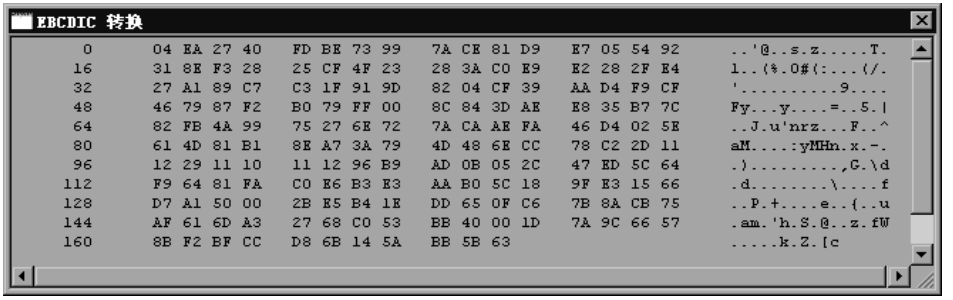
在默认格式下，第一列显示从标记部分开始的字符偏移量。第二列显示数据的十六进制表示形式。第三列以 ASCII 格式显示数据。显示 EBCDIC 数据时，所有不可打印的 ASCII 字符 （例如 /n）都以点线表示。

F7 查看器实用程序对参数化特别有用。通过它可以确定要保存到参数的数据的偏移量。

要确定特定字符的偏移量，请执行下列操作：

**1** 查看 *data.ws*，并从缓冲区的开始处选择数据。

**2** 按 F7 键显示该数据和字符偏移量。由于选择了四个以上的字符，因此数据以长格式显示。




**3** 查找要在 ASCII 数据中关联的值。此例中，我们将关联数字 13546（UNIX 会话期间的进程 ID），它从第 31 个字符开始，最后一个字符在第二行。

**4** 在 **lrs_save_param_ex** 函数中使用偏移量值，以便关联进程 ID 的值。详细信息，请参阅第 8 章 “关联语句”。

# 设置显示格式

您可以指定 VuGen 如何在查看器 (F7) 窗口中显示缓冲区数据。 *lrun/dat* 目录中的 *conv_frm.dat* 文件包含下列显示参数：

**LongBufferFormat**：用于显示五个或更多字符的格式。用 nn 代表偏移量， XX 代表十六进制数据， aa 代表 ASCII 数据。

**LongBufferHeader**：长缓冲区格式中每个缓冲区前面的页眉。

**LongBufferFooter**：长缓冲区格式中每个缓冲区之后的页脚。

**ShortBufferFormat**：用于显示四个或更少字符的格式。可以使用标准的转义序列和转换字符。

支持的转义序列字符有：

| | |
|---|---|
| \a | 铃声 （警告） |
| \b | Backspace 键 |
| \f | 换页 |
| \n | 换行 |
| \r | 回车 |
| \t | 水平制表符 |
| \v | 垂直制表符 |
| \' | 单引号标记 |
| \" | 双引号标记 |
| \\ | 反斜杠 |
| \? | 文本问号 |
| \ooo | ASCII 字符 – 八进制 |

支持的转换字符有：

| | |
|---|---|
| %a | ASCII 表示形式 |
| %BX | 大端 （网络顺序）十六进制 |
| %BO | 大端 （网络顺序）八进制 |
| %BD | 大端 （网络顺序）十进制 |
| %LX | 小端十六进制 |
| %LO | 小端八进制 |
| %LD | 小端十进制 |

**AnyBufferHeader**：每个缓冲区前面的页眉。

**AnyBufferFooter**：每个缓冲区后面的页脚。

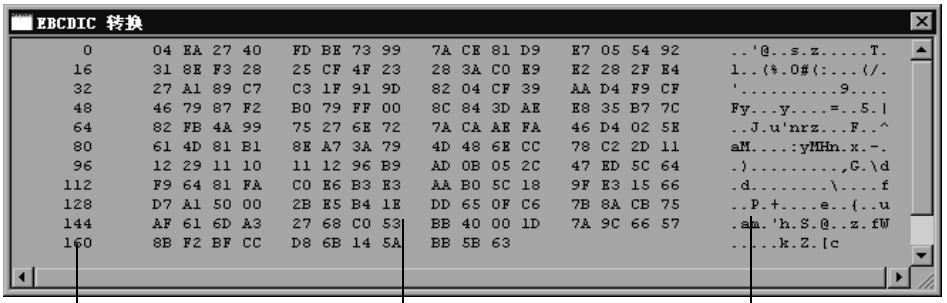
**NonPrintableChar**：表示不可打印的 ASCII 字符的字符。

**PrintAllAscii**：设置为 1 可以强制打印不可打印的 ASCII 字符。

在默认设置下会设置长格式和短格式，且将指定点线代表不可打印的字符。

```
[BufferFormats]
LongBufferFormat=nnnnnnnn    XX XX XX XX  XX XX XX XX  XX XX XX
XX  XX XX XX XX    aaaaaaaaaaaaaaaa\r\n
LongBufferHeader=
LongBufferFooter=
ShortBufferFormat=ASCII:\t\t\t%a\r\n\t\tNetwork Order\t\tLittle
Endian\r\n\t\t (Big
Edian)\r\nHex:\t\t%BX\t\t%LX\r\nOctal:\t\t%BO\t\t%LO\r\nDecim
al:\t%BD\t\t%LD\r\n
AnyBufferHeader=
AnyBufferFooter=----------------------------------------------------------------------
\r\n
NonPrintableChar=.
PrintAllAscii=0
```

默认的 LongBufferFormat 显示为：



默认的 ShortBufferFormat 显示为：

## 调试提示

VuGen 提供了几种方式，通过它们可以调试脚本。您可以查看各种输出日志和窗口以便了解执行期间发出的详细消息。

尤其对于 Windows 套接字 Vuser 脚本， VuGen 提供有关缓冲区不匹配的其他信息。缓冲区不匹配可以指明接收的缓冲区大小 （回放期间生成）与预期的缓冲区 （录制期间生成）的不同。但是，如果接收的缓冲区和预期的缓冲区大小相同，即使其内容不同也不会发出不匹配消息。此信息有助于查找系统或 Vuser 脚本中的问题。

可以在执行日志中查看缓冲区不匹配信息。如果执行日志不可见，请选择 “视图” > “输出” 显示它。

注意，缓冲区不匹配并不总是指明有问题。例如，如果缓冲区包含不重要的数据，例如以前的登录时间，则此类不匹配可以忽略。

```
不匹配 （预期 54 个字节，实际接收 58 个字节）
预期的缓冲区为：
==================
\r\n 最后登录时间：9 月 2 日，星期三， 10:30:18 于
acme.mercury.c\r\n
==================
接收的缓冲区为：
==================
\r\n 最后登录时间：9 月 10 日，星期四， 11:19:50 于
dolphin.mercury.c\r\n
```

但是，如果预期的缓冲区和接收的缓冲区的大小有非常大的差异，则可能指明您的系统有问题。请在相应的缓冲区中检查数据的差异。

为了能够确定不匹配是否重要，您必须完全了解应用程序。

# 手动关联 WinSock 脚本

VuGen 提供关联 Vuser 脚本的用户界面。使用动态数据时需要关联。 WinSock Vuser 脚本的一个常见问题是动态端口，即其号码被动态分配的端口。当某些应用程序总是使用同一个端口时，其他应用程序会使用下一个可用的端口。如果您试图回放脚本，而录制的端口不再可用，则您的测试将失败。要克服此问题，必须执行关联，即保存实际的运行时值并在脚本中使用它们。

您可以使用关联函数手动关联 Vuser 脚本，该函数将把动态值保存到参数。通过 **lrs_save_param 和 lrs_save_param_ex** 函数，可以根据数据在接收的缓冲区中的偏移量将其保存到参数。通过高级关联函数 **lrs_save_searched_string**，可以指定边界并指明以何种匹配的模式保存到参数从而指派数据。下例将描述使用 **lrs_save_param_ex** 函数进行的关联。有关使用其他关联函数的信息，请参阅《LoadRunner Function Reference》。

**要关联 WinSock Vuser 语句，请执行下列操作：**

**1** 将 **lrs_save_param_ex** 语句插入到脚本中要保存缓冲区内容的位置。您可以保存用户、静态或接收这三种类型的缓冲区。

lrs_save_param_ex （**套接字、类型、缓冲区、偏移量、长度、编码、参数**）；

**2** 引用参数。

通过在 VuGen 主窗口的 “数据文件” 框中选择 *data.ws* 文件，查看缓冲区内容。查找要使用已保存缓冲区的内容来替换的数据。用尖括号 (<>) 中的参数名替换该值的所有实例。

在下例中，用户执行了 *telnet* 会话。该用户使用 *ps* 命令确定进程 ID (PID)，并根据该 PID 终止应用程序。

```
frodo:/u/jay>ps
  PID TTY      TIME CMD
14602 pts/18    0:00 clock
14569 pts/18    0:03 tcsh

frodo:/u/jay>kill 14602
[3]    Exit 1                clock
frodo:/u/jay>
```

执行期间，过程中的 PID 是不同的 （UNIX 为每次执行都分配唯一的 PID），因此，终止已录制的 PID 将是无效的。要克服此问题，请使用 **lrs_save_param_ex** 函数将当前的 PID 保存到参数。用参数替换常量。

**3** 在 *data.ws* 文件中，确定接收数据的缓冲区为 buf47。

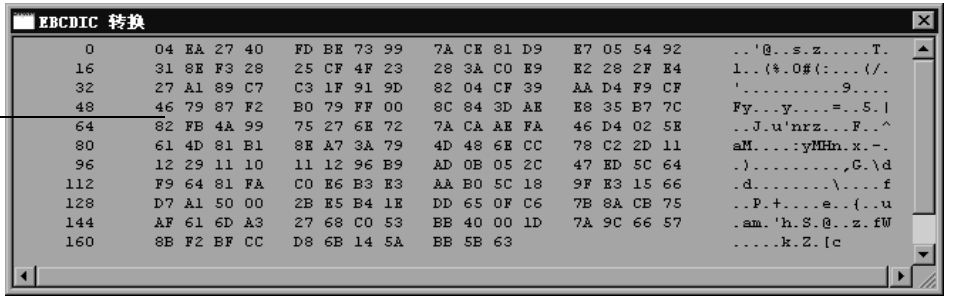
```
recv  buf47 98
      "\r"
      "\x00"
      "\r\n"
      "   PID TTY      TIME CMD\r\n"
      " 14602 pts/18   0:00 clock\r\n"
      " 14569 pts/18   0:02 tcsh\r\n"
      "frodo:/u/jay>"
.
.
.
send buf58
      "kill 14602"
```

**4** 在 *Actions* 部分中，确定 buf47 使用的套接字。在此例中是 *socket1*。

```
lrs_receive("socket1", "buf47", LrsLastArg);
```

**5** 确定要保存的数据字符串的偏移量和长度。突出显示整个缓冲区，然后按 F7。*PID* 的偏移量为 11，其长度为 5 字节。有关显示数据的其他信息，请参阅第 268 页的“了解数据文件格式”

*行中第一个字符的偏移量*



**6** 将 **lrs_save_param_ex** 函数插入 Actions 部分，置于相应缓冲区的 **lrs_receive** 函数之后。在此实例中，缓冲区为 *buf47*。 PID 保存到参数 *param1*。使用 **lr_output_message** 将参数打印到输出。

```
lrs_receive("socket1", "buf79", LrsLastArg);
lrs_save_param("socket1", "user ", buf47, 11, 5, ascii, param1);
lr_output_message ("param1:%s", lr_eval_string("<param1>"));
lr_think_time(10);
lrs_send("socket1", "buf80", LrsLastArg);
```

**7** 在数据文件 *data.ws* 中，确定需要使用参数 *PID* 替换的数据。

```
send buf58
     "kill 14602"
```

**8** 用以尖括号括起来的参数替换该值。

```
send buf58
     "kill <param1>"
```

# 第 VI 部分

## 自定义 Vuser 脚本

# 23

# 创建自定义 Vuser 脚本

除了录制会话以外，您还可以创建自定义的 Vuser 脚本。您既可以使用 LoadRunner API 函数，也可以使用标准的 C、 Java、 VB、 VBScript 或 Javascript 代码。

本章描述：

- C Vuser
- Java Vuser
- VB Vuser
- VBScript Vuser
- JavaScript Vuser

***下列信息适用于所有的自定义 Vuser 脚本：C、 JavaScript、 Java、 VB 和 VBScript。***

## 关于创建自定义 Vuser 脚本

通过 VuGen 可以在脚本中编写您自己的函数，而不用录制实际会话。您可以使用 LoadRunner API 或标准的编程函数。通过 LoadRunner API 函数可以收集关于 Vuser 的信息。例如，可以使用 Vuser 函数来度量服务器性能、控制服务器负载、添加调试代码或者检索参与场景的 Vuser 的运行时信息。

本章描述了如何在 VuGen 编辑器内编写 Vuser 脚本，从而合并应用程序的库或类。

您也可以通过在 Visual C 和 Visual Basic 环境中进行编程来开发 Vuser 脚本。在这些环境中，您可以在导入 LoadRunner 库时，使用开发应用程序开发 Vuser 脚本。详细信息，请参阅第 67 章 “在 Visual Studio 中创建 Vuser 脚本”。

要创建自定义脚本，首先应创建一个骨架脚本。骨架脚本包含脚本的三个主要部分：*init*、 *actions* 和 *end*。这些部分为空，需要手动地向其中插入函数。

您可以为下列编程语言创建空脚本：

- C
- Java
- Visual Basic
- VBScript
- JavaScript

---

**注意**：当使用 JavaScript 和 VBScript Vuser 时，在脚本中用到的 COM 对象必须完全的兼容。这使下列情况成为了可能：一个应用程序操纵另一个应用程序中的对象，或者公开对象以便操纵它们。

---

## C Vuser

在 C Vuser 脚本中，可以放置任何符合标准 ANSI 约定的 C 代码。要创建空的 C Vuser 脚本，请在 “新建虚拟用户” 对话框中的 “自定义” 类别下选择 “C Vuser”。 VuGen 将创建一个空脚本：

```
Action1()
{

    return 0;
}
```

您可以在所有使用 C 函数的 Vuser 脚本类型中使用 C Vuser 函数。

有关 C 的引用及其语法，以及常用 C 函数的示例，另请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

## 使用 C 函数的规则

所有标准 ANSI-C 约定都适用于 C Vuser 脚本，包括控制流和语法。与在其他 C 程序中的操作类似，可以向脚本中添加注释和条件语句。可以使用 ANSI C 约定声明和定义变量。

用于运行 Vuser 脚本的 C 解释器接受标准的 ANSI C 语言。它不支持对 ANSI C 的任何 Microsoft 扩展。

向 Vuser 脚本中添加任何 C 函数前，请注意下列限制：

- Vuser 脚本不能将其中某个函数的地址作为回调传递给库函数。
- Vuser 脚本不支持 *stdargs*、 *longjmp* 和 *alloca* 函数。
- Vuser 脚本不支持结构参数或返回类型。支持指向结构的指针。
- 在 Vuser 脚本中，字符串为只读。任何写入字符串的尝试都将生成访问冲突。
- 不返回 int 的 C 函数必须转换。例如， extern char * strtok();

### 调用 libc 函数

在 Vuser 脚本中，可以调用 *libc* 函数。不过，由于用于运行 Vuser 脚本的解释器不支持任何对 ANSI C 的 Microsoft 扩展，因此无法使用 Microsoft 的包含文件。可以在必要时编写自己的原型，也可要求 Mercury Interactive 客户支持部门给您发送包含 *libc* 函数原型的兼容 ANSI 的包含文件。

### 链接模式

用于运行 Vuser 脚本的 C 解释器使用 “惰性”链接模式，因为不需要在运行开始就定义函数，只要在使用函数前对其进行定义即可。例如：

```
lr_load_dll("mydll.dll");
    myfun();  /* 在 mydll.dll 中定义 – 可以在
                    加载 myfun.dll 后立即直接调用。 */
```

# Java Vuser

Java Vuser 脚本中可以放置任何标准 Java 代码。要创建空的 Java Vuser 脚本，请在 “新建虚拟用户”对话框中的 “自定义”类别下选择 “Java Vuser”。VuGen 将创建空 Java 脚本：

```
import lrapi.lr;

public class Actions
{

    public int init() {
        return 0;
    }


    public int action() {
        return 0;
    }


    public int end() {
        return 0;
    }
}
```

注意，对于 Java 类型的 Vuser，仅可以编辑 *Actions* 类。在 Actions 类中，有三种方法：*init*、*action* 和 *end*。可以在 *init* 方法中放置初始化代码、在 *action* 方法中放置业务流程以及在 *end* 方法中放置清理代码。

您还可以在 Corba-Java 和 RMI-Java Vuser 脚本中使用 Java Vuser 函数。

## VB Vuser

您可以创建空的 Visual Basic Vuser 脚本，在其中可以放置 Visual Basic 代码。通过此脚本类型可以将 Visual Basic 应用程序集成到 LoadRunner 中。要创建空的 VB Vuser 脚本，请在 “新建虚拟用户”对话框中的 “自定义”类别下选择 “VB Vuser”。 VuGen 将创建空 VB 脚本：

```
Public Function Actions() As Long

        '"TO DO: 在此处放置您的操作代码



    Actions = lr.PASS
End Function
```

VuGen 将创建三部分：*vuser_init*、 *action* 和 vuser_*end*。每一部分都包含一个 VB 函数，分别为 *Init*、*Actions* 和 *Terminate*。正如 *TO DO* 注释所指明的，可以将代码放置在这些函数内。

还有一个可以在 VuGen 中看到的附加部分：*global.vba* 文件，其中包括 LoadRunner 和 VB 应用程序的对象和变量的全局声明。

# VBScript Vuser

您可以创建空的 VBScript Vuser 脚本，在其中可以放置 VBScript 代码。通过此脚本类型可以将 VBScript 应用程序集成到 LoadRunner 中。要创建空的 VBScript Vuser 脚本，请在 “新建虚拟用户”对话框的 “自定义”类别中选择 “VB Script Vuser”。 VuGen 将创建空 VBScript Vuser 脚本：

```
Public Function Actions()

        '"TO DO: 在此处放置您的操作代码



    Actions = lr.PASS
End Function
```

VuGen 将创建三部分：*vuser_init*、 *action* 和 vuser_*end*。每一部分都包含一个 VBScript 函数，分别是 *Init*、 *Actions* 和 *Terminate*。正如 *TO DO* 注释所指明的，可以将代码放置在这些函数内。

还有一个可以在 VuGen 中看到的附加部分：*global.vbs* 文件，它为 LoadRunner 和 VBScript 创建对象。例如，下段代码即可创建标准 LoadRunner 对象：

```
Set lr = CreateObject("LoadRunner.LrApr")
```

# JavaScript Vuser

您可以创建空的 JavaScript Vuser 脚本，在其中放置 JavaScript 代码。通过此脚本类型可以将现有的 javascript 应用程序集成到 LoadRunner 中。要创建空的 JavaScript Vuser 脚本，请在 “新建虚拟用户” 对话框的 “自定义” 类别中选择 “JavaScript Vuser”。

```
function Actions()
{
    //"TO DO: 在此处放置业务流程 / 操作代码



    return(lr.PASS);
}
```

VuGen 将创建三部分：*vuser_init*、 *action* 和 vuser_*end*。每一部分都包含一个 JavaScript 函数，分别为 *Init*、 *Actions* 和 *Terminate*。正如 *TO DO* 注释所指明的，可以将代码放置在这些函数内。

还有一个可以在 VuGen 中看到的附加部分：*global.js* 文件，它为 LoadRunner 和 JavaScript 创建对象。例如，下段代码即可创建标准 LoadRunner 对象：

```
var lr = new ActiveXObject("LoadRunner.LrApr")
```

# 24

# 编写 Java 脚本

VuGen 支持协议级的 Java 类型用户。本章解释如何通过编程创建 Java Vuser 脚本。有关通过录制创建 Java Vuser 脚本的信息，请参阅有关 Corba-Java、 RMI-Java、 EJB 或 Jacada 类型协议的章节。

本章描述在 Java 中如何使用 *Java* Vuser 编写 Vuser 脚本：

- 创建 Java Vuser
- 编辑 Java Vuser 脚本
- LoadRunner 的 Java API
- 使用 Java Vuser 函数
- 设置 Java 环境
- 运行 Java Vuser 脚本
- 将脚本作为包的一部分编译和运行
- 编写提示

***下列信息适用于 Java、EJB 测试、Corba-Java、RMI-Java 以及 Jacada Vuser 脚本。***

## 关于编写 Java 脚本

要准备用 Java 代码编写 Vuser 脚本，可以使用 *Java*、*Corba-Java* 或 *RMI-Java* 类型的 Vuser。这些 Vuser 类型支持协议级 Java。 Vuser 脚本由 Java 编译器编译，它支持全部标准 Java 约定。例如，可通过在文本前面加上两个正斜杠 “//” 来插入注释。

有关 Corba、 RMI、 EJB 和 Jacada Vuser 的章节解释如何通过录制创建脚本。要准备通过编程编写 Java 编码的脚本，请参阅下列部分。

创建兼容 Java 的 Vuser 脚本的第一步，就是新建一个 Java Vuser 类型的 Vuser 脚本模板。然后将需要的 Java 代码编写或粘贴到脚本模板中。可以添加 LoadRunner Java Vuser 函数，增强脚本并将参数参数化以便在循环期间使用不同的值。

Java Vuser 脚本作为可伸缩的多线程应用程序运行。如果脚本中包含自定义类，请确保代码线程安全。非线程安全的代码可能导致结果不准确。对于非线程安全的代码，请将 Java Vuser 作为进程运行。这样就会为每个进程创建一个独立的 Java 虚拟机，导致脚本伸缩性差。

准备好脚本之后，将其作为独立于 VuGen 的测试运行。Java 编译器（Sun 的 Javac）可以检查脚本中是否有错误并编译该脚本。确认脚本可正常运行后，可将该脚本集成到 LoadRunner 场景中。

## 创建 Java Vuser

创建兼容 Java 的 Vuser 脚本的第一步是创建 Java Vuser 模板。

**要创建 Java Vuser 脚本，请执行下列操作：**

1. 打开 VuGen。
2. 选择“文件”>“新建”或单击“新建”按钮。将打开“新建虚拟用户”对话框。
3. 从“选择 Vuser 类型”列表中选择“自定义”>“Java Vuser”，单击“确定”。VuGen 将显示一个空白 Java Vuser 脚本。
4. 在左框中单击 *Actions* 部分，显示 *Actions* 类。

# 编辑 Java Vuser 脚本

生成空模板以后，可以插入需要的 Java 代码。使用该类型的 Vuser 脚本时，需要将全部代码放在 *Actions* 类中。要查看 Actions 类，请单击左窗格中的“Actions”。 VuGen 将在右窗格中显示其内容。

```
import lrapi.*;
public class Actions
{
    public int init() {
        return 0;
    }

    public int action() {
        return 0;
    }

    public int end() {
        return 0;
    }
}
```

Actions 类包含三个方法：*init*、 *action* 和 *end*。下表显示每个方法包含的内容以及执行的时间。

| 脚本方法 | 用于模拟的内容 | 执行时间 |
|---|---|---|
| *init* | 登录到服务器 | Vuser 已初始化 （已加载） |
| *action* | 客户端活动 | Vuser 处于 “正在运行”状态 |
| *end* | 注销过程 | Vuser 完成或停止 |

### Init 方法

将全部登录过程和一次性配置设置放在 *init* 方法中。 *init* 方法仅执行一次（当 Vuser 开始运行脚本时）。以下 *init* 方法示例将初始化一个小程序。

```
import org.omg.CORBA.*;
import org.omg.CORBA.ORB.*;
import lrapi.lr;

// 公共函数：init
   public int init() throws Throwable {

      // 初始化 Orb 实例……
      MApplet mapplet = new MApplet("http://chaos/classes/", null);
      orb = org.omg.CORBA.ORB.init(mapplet, null);
…
```

### Action 方法

将全部 Vuser 操作放在 *action* 方法中。根据在运行时设置中设置的循环数执行 *action* 方法。有关循环设置的详细信息，请参阅第 9 章 “配置运行时设置”。以下 *action* 方法示例将检索并打印 Vuser ID。

```
public int action() {
     lr.message("vuser:" + lr.get_vuser_id() + " xxx");
          return 0;
 }
```

### End 方法

在 *end* 方法中，放置希望 LoadRunner 在场景结束时执行的代码，如从服务器注销、清理环境等。 *end* 方法仅执行一次 （当 Vuser 完成运行脚本时）。在下例中，*end* 方法关闭并将 *end* 消息打印到执行日志。

```
public int end() {
      lr.message("End");
      return 0;
  }
```

# LoadRunner 的 Java API

LoadRunner 为访问 Vuser 函数提供了特定的 Java API。这些函数都是 lrapi.lr 类的静态方法。以下部分列出了 LoadRunner 的 Java Vuser 函数。有关每个函数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。注意，新建 Java Vuser 脚本时， *import lrapi.** 已插入脚本。

### 事务函数

| | |
|---|---|
| **lr.declare_transaction** | 声明一个事务 |
| **lr.start_transaction** | 标记事务的开始 |
| **lr.end_transaction** | 标记事务的结束 |

### 命令行分析函数

| | |
|---|---|
| **lr.get_attrib_double** | 检索脚本命令行中使用的 *double* 类型变量 |
| **lr.get_attrib_long** | 检索脚本命令行中使用的 *long* 类型变量 |
| **lr.get_attrib_string** | 检索脚本命令行中使用的字符串 |

### 信息性函数

| | |
|---|---|
| **lr.value_check** | 检查参数的值 |
| **lr.user_data_point** | 录制用户定义的数据示例 |
| **lr.get_group_name** | 返回 Vuser 组的名称 |
| **lr.get_host_name** | 返回执行 Vuser 脚本的负载生成器的名称 |
| **lr.get_master_host_name** | 返回运行 LoadRunner Controller 的计算机的名称 |
| **lr.get_object** | 捕获 Java 对象并将其转储到数据文件。（仅用于 Corba-Java） |
| **lr.get_scenario_id** | 返回当前场景的 ID |
| **lr.get_vuser_id** | 返回当前 Vuser 的 ID |

### 字符串函数

| | |
|---|---|
| **lr.deserialize** | 展开对象以表示其 ASCII 组件 |
| **lr.eval_string** | 用当前值替换参数 |
| **lr.eval_data** | 用字节值替换参数 |
| **lr.eval_int** | 用整型值替换参数 |
| **lr.eval_string** | 用字符串替换参数 |
| **lr.next_row** | 指示使用指定参数的下一行数据 |
| **lr.save_data** | 将字节另存为参数 |
| **lr.save_int** | 将整数另存为参数 |
| **lr.save_string** | 将以 NULL 结尾的字符串保存到参数 |

### 消息函数

| | |
|---|---|
| **lr.debug_message** | 向输出窗口发送调试消息 |
| **lr.enable_redirection** | 启用将标准消息和错误重定向到日志文件（作为标准输出和标准错误） |
| **lr.error_message** | 向 Vuser 日志文件和输出窗口发送错误消息以及位置详细信息 |
| **lr.get_debug_message** | 检索当前消息类 |
| **lr.log_message** | 向 Vuser 日志文件发送消息 |
| **lr.message** | 向输出窗口发送消息 |
| **lr.output_message** | 向日志文件和输出窗口发送消息和位置信息 |
| **lr.redirect** | 将字符串重定向到文件 |
| **lr.set_debug_message** | 设置调试消息类 |
| **lr.vuser_status_message** | 向 Controller 窗口中的 “Vuser 状态” 区域发送消息 |

### 运行时函数

| | |
|---|---|
| **lr.declare_rendezvous** | 在 Vuser 脚本中声明集合 |
| **lr.peek_events** | 指示可以暂停 Vuser 脚本的位置 |
| **lr.rendezvous** | 在 Vuser 脚本中设置集合点 |
| **lr.think_time** | 暂停脚本执行，模拟实际用户在操作之间暂停思考的时间 |

要使用其他 Java 类，请在脚本的开头将其导入，如下所示。记住向类路径中添加类目录或相关 jar 文件。确保其他类的线程安全性和可伸缩性。

```
import java.io.*;
import lrapi.*;

public class Actions
{
...
}
```

## 使用 Java Vuser 函数

可以使用 Java Vuser 函数增强脚本，方法是：

- 插入事务
- 插入集合点
- 获取 Vuser 信息
- 发出输出消息
- 模拟用户思考时间
- 处理命令行参数

## 插入事务

可以定义事务以度量服务器的性能。每个事务度量服务器响应指定请求所用的时间。这些请求可以是简单任务或复杂任务。场景运行期间和运行之后，可以使用 LoadRunner 的联机监视器和图来分析每个事务的性能。

还可以指定事务状态：lr.PASS 或 lr.FAIL。可以允许 LoadRunner 确定事务是否成功，或者也可以将其集成到条件循环中。例如，可以在代码中检查是否有特定的返回代码。如果代码正确，则发布 lr.PASS 状态。如果代码错误，则发布 lr.FAIL 状态。

**要标记事务，请执行下列操作：**

1. 将 lr.start_transaction 插入到脚本中您希望开始度量任务计时的点。
2. 将 lr.end_transaction 插入到脚本中您希望停止度量任务的点。使用出现在 lr.start_transaction 函数中的事务名称。
3. 为事务指定需要的状态：lr.PASS 或 lr.FAIL.

```
public int action() {

for(int i=0;i<10;i++)
  {
     lr.message("action()"+i);
     lr.start_transaction("trans1");
     lr.think_time(2);
     lr.end_transaction("trans1",lr.PASS);
  }
     return 0;
}
```

## 插入集合点

要模拟客户端 / 服务器系统的较重用户负载，请通过创建集合点同步 Vuser，使其在同一时刻执行任务。当某个 Vuser 到达集合点时， Controller 会将其保留，直到参与该集合的全部 Vuser 都到达。

可通过将集合点插入 Vuser 脚本来指定会合位置。

要插入集合点，请执行下列操作：

**1** 将 lr.rendezvous 函数插入到脚本中您希望 Vuser 执行集合的点。

```
public int action() {

  for(int i=0;i<10;i++)
   {
      lr.rendezvous("rendz1");
      lr.message("action()"+i);
      lr.think_time(2);
   }
   return 0;
}
```

## 获取 Vuser 信息

可以向 Vuser 脚本中添加下列函数以检索 Vuser 信息：

| | |
|---|---|
| **lr.get_attrib_string** | 返回包含命令行参数值或运行时信息的字符串，例如 Vuser ID 或负载生成器名 |
| **lr.get_group_name** | 返回 Vuser 组的名称 |
| **lr.get_host_name** | 返回执行 Vuser 脚本的负载生成器的名称 |
| **lr.get_master_host_name** | 返回运行 LoadRunner Controller 的计算机的名称 |
| **lr.get_scenario_id** | 返回当前场景的 ID |
| **lr.get_vuser_id** | 返回当前 Vuser 的 ID |

在下例中， **lr.get_host_name** 函数将检索运行 Vuser 的计算机的名称。

```
String my_host = lr.get_host_name( );
```

有关上述函数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

## 发出输出消息

运行场景时， Controller 的输出窗口将显示有关脚本执行的信息。可以在 Vuser 脚本中包含向 Controller 发送错误和通知消息的语句。 Controller 将在输出窗口显示这些消息。例如，可以插入显示客户端应用程序当前状态的消息。还可将这些消息保存到文件。

---

**注意**：不要从事务内发送消息。这样做会延长事务执行时间，并且可能扭曲实际事务结果。

---

可以在 Vuser 脚本中使用下列消息函数：

| | |
|---|---|
| **lr.debug_message** | 向输出窗口发送调试消息 |
| **lr.log_message** | 向 Vuser 日志文件发送消息 |
| **lr.message** | 向输出窗口发送消息 |
| **lr.output_message** | 向日志文件和输出窗口发送消息和位置信息 |

在下例中， **lr.message** 将向输出窗口发送指示循环次数的消息。

```
for(int i=0;i<10;i++)
  {
     lr.message("action()"+i);
     lr.think_time(2);
  }
```

有关消息函数的详细信息，请参阅第 294 页的 “消息函数”或 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

可以指示 LoadRunner 将 Java 标准输出和标准错误流重定向到 VuGen 执行日志。需要在 Vuser 脚本中粘贴现有 Java 代码，或使用预置类（包括 System.out 和 System.err 调用）时，该操作尤其有用。在执行日志中，标准输出消息标为蓝色，而标准错误显示为红色。

下例显示了如何使用 lr.enable_redirection 将特定消息重定向到标准输出和标准错误。

```
lr.enable_redirection(true);

System.out.println("This is an informatory message…"); // 已重定向
System.err.println("This is an error message…"); // 已重定向

lr.enable_redirection(false);

System.out.println("This is an informatory message…"); // 未重定向
System.err.println("This is an error message…"); // 未重定向
```

---

**注意：**将 **lr.enable_redirection** 设置为 “true” 时，它将覆盖全部以前的重定向。要恢复以前的重定向，请将该函数设置为 “false”。

---

有关该函数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

### 模拟用户思考时间

用户在执行两个连续操作期间等待的时间称为 “思考时间”。 Vuser 使用 lr.think_time 函数模拟用户思考时间。在下例中， Vuser 将在两个循环之间等待两秒。

```
for(int i=0;i<10;i++)
  {
     lr.message("action()"+i);
     lr.think_time(2);
  }
```

可以使用脚本中的思考时间设置，或这些值的某个系数。要配置 LoadRunner 处理思考时间函数的方式，请打开 “运行时设置” 对话框。详细信息，请参阅第 9 章 “配置运行时设置”。

有关 lr.think_time 函数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

## 处理命令行参数

运行脚本时，可以通过指定命令行参数，在运行时向 Vuser 脚本传递值。可以在 LoadRunner Controller 中的脚本路径后面插入命令行选项。通过下面的三个函数可以读取命令行参数，然后向 Vuser 脚本传递值：

| | |
|---|---|
| **lr.get_attrib_double** | 检索双精度浮点类型参数 |
| **lr.get_attrib_long** | 检索长整型参数 |
| **lr.get_attrib_string** | 检索字符串 |

命令行应该采用下列两种格式之一，在这两种格式中，参数及其值在脚本名后面成对列出：

```
script_name   -argument argument_value   -argument argument_value
```

```
script_name   /argument argument_value   /argument argument_value
```

下例显示了用来在 pc4 计算机上重复五次 *script1* 的命令行字符串：

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

有关命令行分析函数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。有关在命令行中包含参数的详细信息，请参阅《LoadRunner Controller 用户指南 (Windows)》。

## 设置 Java 环境

运行 Java Vuser 脚本之前，请确保在所有运行 Vuser 的计算机上正确设置环境变量 PATH 和 CLASSPATH：

- 要编译和回放脚本，必须完全安装 JDK 版本 1.1、 1.2 或 1.3。只安装 JRE 是不够的。最好不要在一台计算机上安装多个 JDK 或 JRE。如果可能，请卸载所有不必要的版本。
- PATH 环境变量必须包含 JDK/bin 项。
- 对于 JDK 1.1.x， CLASSPATH 环境变量必须包括 *classes.zip* 路径 (*JDK/lib*) 和全部 LoadRunner 类 (*loadrunner/classes*)。
- Java Vuser 使用的所有类必须位于类路径中 （在计算机的 CLASSPATH 环境变量中设置，或在 “运行时设置” 的 “类路径” 节点的 “类路径项” 列表中设置。

## 运行 Java Vuser 脚本

Java Vuser 脚本与 C Vuser 脚本的不同之处在于，它们是先编译再执行，而 C Vuser 脚本是解释性的。 Vugen 在 JDK 安装内查找 *javac* 编译器，并在脚本内编译 Java 代码。该阶段由 Vugen 窗口底部的 “正在编译…” 状态消息来指示。如果在编译期间出错，则这些错误将在执行日志中列出。要转到脚本中引起错误的代码，请双击包含该错误所在行的编号的错误消息。解决该错误并重新运行脚本。

如果编译成功，则 “正在编译…” 状态消息将更改为 “正在运行…”， VuGen 开始执行脚本。重新运行脚本时，如果没有对脚本进行任何更改， VuGen 将直接运行脚本，而不再重新编译。要进一步调试脚本，可以使用断点以及应用分步选项的动态运行类型执行。

---

**注意**：如果正在脚本中调用 JNDI 扩展，则在尝试将 Vuser 作为线程运行时可能会遇到问题。发生这种情况的原因是 JNDI 要求每个线程有自己的上下文类加载程序。为了作为线程运行，可以向 *init* 部分的开头添加以下行，指示每个 Vuser 使用自己的上下文类加载程序运行：

```
DummyClassLoader.setContextClassLoader();
```

---

## 将脚本作为包的一部分编译和运行

创建 Java Vuser 脚本时，可能需要使用其他类中的方法，而这些类中的类或方法是受保护的。如果尝试编译该类型的脚本，则在编译阶段将收到错误信息，指示无法访问该方法。要确保脚本可以访问这些方法，请在脚本顶部插入包含这些方法的包名，与在标准 Java 程序中的操作类似 – < 包名 >。在下例中，脚本定义了由一个路径构成的 *just.do.it* 包：

```
package just.do.it;

import lrapi.*;
public class Actions
{
    :
}
```

在上例中， Vugen 在 Vuser 目录下自动创建了 *just/do/it* 目录层次结构，并将 *Actions.java* 文件复制到 *just/do/it/Actions.java*，允许它用相关包编译。注意，该包语句必须在脚本的首行中，类似 Java （除注释外）。

## 编写提示

编写 Java Vuser 脚本时，可以将预置代码段粘贴到脚本中，或导入预置类以便调用其方法。如果 Vuser 需要在 Controller 下作为线程运行 （出于可伸缩性原因），则需要确保所有导入代码是线程安全的。

线程安全通常易于实现，但难于检测。在仅有有限数量的 Vuser 时， Java Vuser 可以在 Vugen 和 Controller 下零缺陷运行。用户很多时就会出现问题。非线程安全的代码通常是使用静态类成员的结果，如下例所示：

```
import lrapi.*;
public class Actions
{
  private static int iteration_counter = 0;

    public int init() {
        return 0;
    }

    public int action() {
        iteration_counter++;
        return 0;
    }

    public int end() {
        lr.message("Number of Vuser iterations:"+iteration_counter);
        return 0;
    }
}
```

运行一个 Vuser 时， *iteration_counter* 成员可以准确确定执行的循环数。当多个 Vuser 在单个虚拟机上作为线程一起运行时，所有线程共享静态类成员 *iteration_counter*，导致计数错误。所有 Vuser 循环的总数都被计算在内。

如果知道代码非线程安全仍要将其导入脚本，可以将 Vuser 作为进程运行。有关将 Vuser 作为线程或进程运行的详细信息，请参阅第 9 章 “配置运行时设置”。

运行简单 Java Vuser 脚本时，它通常由单个线程 （主线程）构成。只有主线程可以访问 LoadRunner Java API。如果 Java Vuser 衍生辅助工作线程，则使用 LoadRunner API 可能引起难以预计的后果。因此，建议仅在主线程中使用 LoadRunner Java API。注意，该限制也会影响 **lr.enable_redirection** 函数。

下例说明了 LR API 在哪里可以使用，在哪里不可以使用。执行日志中的第一条日志消息指示 “标志” 的值为 *false*。虚拟机接着衍生出一个新的线程 *set_thread*。该线程运行并将 “标志” 设置为 *true*，但不会向日志发送消息，即便存在对 **lr.message** 的调用。最终日志消息显示该线程内的代码已被执行，而且 “标志” 已被设置为 *true*。

```
boolean flag = false;

public int action() {
  lr.message("Flag value:"+flag);
  Thread set_thread = new Thread(new Runnable();{
        public void run() {
          lr.message("LR-API NOT working!");
          try { Thread.sleep(1000); } catch(Exception e) {}
          flag = true;
        }
  });
  set_thread.start();
  try { Thread.sleep(3000); } catch(Exception e) {}
  lr.message("Flag value:"+flag);
  return 0;
}
```

# 第 VII 部分

## 分布式组件协议

# 25

# 录制 COM Vuser 脚本

许多 Windows 应用程序都直接或通过库调用使用基于 COM 的函数。使用 VuGen 可以录制一个脚本，模拟访问 COM 服务器的 COM 客户端。得到的脚本称为 COM Vuser 脚本。还可以使用 Visual Basic 外接程序创建 COM Vuser 脚本。有关 Visual Basic 外接程序的详细信息，请参阅第 67 章 “在 Visual Studio 中创建 Vuser 脚本”。

第 26 章 “了解 COM Vuser 脚本”介绍了 VuGen COM 脚本的工作原理并提供了简要的函数参考。

本章描述：

- COM 概述
- COM Vuser 入门
- 选择要录制的 COM 对象
- 设置 COM 录制选项

***下列信息仅适用于 COM Vuser 脚本。***

## 关于录制 COM Vuser 脚本

录制 COM 客户端应用程序时， VuGen 将生成若干函数，描述 COM 客户端 – 服务器活动。录制的脚本包含接口声明、 API 调用和对方法的实例调用。每个 COM 函数都以 **lrc** 前缀开头。

在 VuGen 的主窗口中，可以查看和编辑录制的脚本。该窗口显示会话过程中录制的 COM API/ 方法调用，通过它，可以以可视方式跟踪应用程序 COM/DCOM 调用。

您可以指明用于创建 Vuser 脚本的编程语言 – C 或 Visual Basic 脚本。详细信息，请参阅第 4 章 “设置脚本生成首选项”。

# COM 概述

本部分将对 COM 技术进行简要介绍。通过本部分的介绍，您将了解 COM Vuser 脚本的入门知识。详细信息，请参阅 Microsoft Developer 扭 Network (MSDN) 和其他文档。

COM （组件对象模型）是一种开发可重用的软件组件 （“插件”）的技术。通过 DCOM （分布式 COM）可以在远程计算机中使用 COM 组件。 Microsoft Transaction Server (MTS)、 Visual Basic 和 Explorer 均使用 COM/DCOM 技术。因此，您正在测试的应用程序可能间接使用了 COM 技术，尽管您还不了解它。您可能需要在 Vuser 脚本中包含某些 （但肯定不是所有）应用程序使用的 COM 调用。

### 对象、接口和类型库

COM 对象为二进制代码模块。每个 COM 对象都实现一个或多个接口，以便与客户端程序进行通信。您需要了解这些接口以便跟踪 Vuser 脚本中的 COM 调用。类型库被用作访问 COM 接口方法和参数的引用，它包含 COM 对象和接口的说明。每个 COM 类、接口和类型库都由全局唯一标识符 (GUID) 标识。

### COM 接口

COM 接口提供了一组相关方法的集合。例如， **Clock** 对象可能包含 **Clock**、**Alarm** 和 **Timer** 接口。每个接口都包含一个或多个方法。例如， **Alarm** 接口可能包含 **AlarmOn** 和 **AlarmOff** 方法。

每个接口还可能包含一个或多个属性。有时，执行同一函数需要调用方法、设置或获取属性值。例如，您可以将 **Alarm Status** 属性设置为 **On** 或调用 **AlarmOn** 方法。

一个 COM 对象可以支持许多接口。 **IUnknown** 接口由所有组件实现并用于查找其他接口。许多组件还实现了 **IDispatch** 接口，该接口可以公开对象的所有其他接口和方法，以便在脚本语言中实现 COM 自动化。

### COM 类上下文和位置透明度

COM 对象可以作为客户端程序在同一计算机上运行，也可以在远程服务器上运行。应用程序创建的 COM 对象可以位于本地库、本地进程或远程计算机 （“远程对象代理服务器”）中。 COM 对象的位置 （称作 “上下文”）对应用程序可以是透明的。多数用户应用 LoadRunner 来检查远程服务器上的负载。因此，由远程对象代理服务器访问的对象通常与这些测试最具相关性。

### COM 数据类型

COM 还提供了几个特殊的数据类型，包括安全数组、BSTR 字符串和变量。可能需要使用这些数据类型执行调试、参数化和类似任务。

## COM Vuser 入门

本部分描述了开发 COM Vuser 脚本的过程。

**要开发 COM Vuser 脚本，请执行下列操作：**

**1 使用 VuGen 录制基础脚本**

启动 VuGen 并新建一个 Vuser 脚本。将 Vuser 的类型指定为 COM。选择要录制的应用程序，并设置录制选项。要设置脚本相关的录制选项，请参阅第 4 章 “设置脚本生成首选项”。要设置 COM 特定选项和筛选器，请参阅第 312 页的 “设置 COM 录制选项”。使用应用程序录制典型操作。

有关录制的详细信息，请参阅第 3 章 “使用 VuGen 进行录制”。

**2 优化对象筛选器**

使用生成的日志文件对筛选器中要录制的对象进行优化选择。详细信息，请参阅下一部分 “选择要录制的 COM 对象”。

**3 增强脚本**

通过将事务、集合点以及控制流结构插入到脚本中，增强 Vuser 脚本。

详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

**4 定义参数 （可选）**

为脚本中录制的固定值定义参数。通过使用参数替换固定值，可以用不同的值多次重复同一业务流程。

详细信息，请参阅第 7 章 “定义参数”。

**5 配置运行时设置**

运行时设置控制脚本执行期间 Vuser 的行为。这些设置包括循环、日志和计时信息。

详细信息，请参阅第 9 章 “配置运行时设置”。

**6 从 VuGen 运行脚本**

在 VuGen 中保存并运行脚本，以验证其是否能够正确运行。

详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

创建 COM Vuser 脚本后，将其集成到 Windows 平台上的场景中。有关将 Vuser 脚本集成到场景中的详细信息，请参阅 《LoadRunner Controller 用户指南》。

# 选择要录制的 COM 对象

要测试的应用程序可能使用了大量的 COM 对象。其中，可能只有少数几个对象实际产生负载，并对负载测试有重要意义。因此，在录制 COM 应用程序之前，应选择要为负载测试录制的对象。通过 VuGen，可以在类型库中浏览对象，所浏览的对象均为 VuGen 能够在本地计算机和网络中其他计算机上读取的对象。

## 决定要使用的对象

有多种方法可以决定应包含在测试中的 COM 对象。确定软件使用的远程对象。如果不能确定选择哪些对象，请使用默认筛选器。筛选器的 “环境”分支包含对三个对象集 （ADO、 RDS 和远程）的调用，这三个对象集可能在远程服务器上生成负载。

也可以检查实际调用以优化筛选器。录制测试后，可以保存文件并在 VuGen 创建的 *data* 目录中查找名为 *lrc_debug_list_<nnn>.log* 的文件。其中， *nnn* 为进程编号。该日志文件包含由录制的应用程序调用的所有 COM 对象的列表，无论录制筛选器是否包含该对象。录制过程应该只包含在服务器上生成负载的调用。

以下是一个 Visual Basic 库的本地 COM 示例：

```
Y类 JetES {039EA4C0-E696-11D0-878A-00A0C91EC756}
从类型库 “JET Expression Service Type Library”
（{2358C810-62BA-11D1-B3DB-00600832C573} 4.0 版）加载
```

不应添加该对象。

同样， OLE DB 和 Microsoft Windows Common Control 均为本地对象。因此，对于以下所示的不产生服务器负载的类和库，不应进行录制：

```
类 DataLinks {2206CDB2-19C1-11D1-89E0-00C04FD7A829}
从类型库 “Microsoft OLE DB Service Component 1.0 Type Library”
（{2206CEB0-19C1-11D1-89E0-00C04FD7A829} 1.0 版）中加载

类 DataObject {2334D2B2-713E-11CF-8AE5-00AA00C00905}
从类型库 “Microsoft Windows Common Controls 6.0 (SP3)”
（{831FDD16-0C5C-11D2-A9FC-0000F8754DA1} 2.0 版）中加载
```

但是，下面的示例表则指明了应该录制的类：

```
类 Order {B4CC7A90-1067-11D4-9939-00105ACECF9A}
从类型库 “FRS”
（{B4CC7A8C-1067-11D4-9939-00105ACECF9A} 1.0 版）中加载
```

对 **FRS** 库中 （用于同 VuGen 一起安装的 flight_sample 中的实例）类的调用将使用服务器容量，因此应该对其进行录制。

如果 COM 对象本身调用其他 COM 对象，则所有调用将在类型信息日志文件中列出。例如，每当应用程序调用 **FRS** 类函数时， **FRS** 库都将调用 ActiveX 数据对象 (ADO) 库。如果筛选器中列出了这样多个函数构成的函数链，则 VuGen 将仅录制初始化该链的第一个调用。如果同时选择了 **FRS** 和 **ADO** 调用，则仅录制 **FRS** 调用。另一方面，如果在筛选器中仅选择 **ADO** 库，则将录制 **ADO** 库调用。通常录制对链中第一个远程对象的调用最为简单。但在某些情况下，应用程序可能使用多个不同 COM 对象中的方法，但所有这些方法都使用单个对象产生服务器负载，因此您仅能录制最后一个公用对象。

### 可以选择的对象

VuGen 仅能录制可以读取其类型库的对象。如果系统中未安装类型库，或 VuGen 找不到它们，则 “录制选项”对话框中将不列出这些 COM 对象。如果应用程序使用了这些对象， VuGen 将无法识别他们，并将在文件中将其标识为 **INoTypeInfo**。

### 可以排除的接口

对于每个对象，“录制选项”对话框将显示类型库中列出的所有接口，可以在此指定包含还是排除每个对象。而 **ADO**、 **RDS** 和远程对象可以作为组包含在筛选器中。筛选器将不显示这些环境中的单个对象或它们的接口。从类型库中包含的对象可能还具有类型库中未列出的接口，因此在 “录制选项”对话框中将无法显示这些接口。生成 VuGen 脚本后，可以在脚本中标识这些接口，并从 VuGen 生成的 interfaces.h 文件中获取它们的 GUID。可以使用该信息排除如下接口。

## 设置 COM 录制选项

使用 “COM 录制选项”对话框设置筛选和 COM 脚本选项。可以使用联机浏览器在注册表、文件系统或 Microsoft Transaction Server (MTS) 中找到类型库。

详细信息，请参阅：

- 筛选对象
- 设置 COM 脚本选项

### 筛选对象

通过筛选器选项，可以指明 VuGen 应录制哪些 COM 对象。可以从环境和库中选择对象。

使用筛选器选项可以设置默认筛选器或创建其他筛选器。您可以按环境和类型库筛选录制会话。

### DCOM 配置文件

- **默认筛选器：**录制 COM Vuser 脚本时使用的默认筛选器。
- **新建筛选器：**基于默认环境设置的全新筛选器。注意，在使用新筛选器设置进行录制之前，必须为其指定一个名称。

### DCOM 侦听器设置

DCOM 侦听器设置显示类型库的树层次结构。可以展开树结构显示类型库中的所有可用类。也可以展开类树显示该类支持的所有接口。

要排除一个类型库，请清除库名称旁边的复选框。这将排除此类型库中的所有类。展开树结构后，可以通过清除项目旁边的复选框排除单个类或接口。

一个接口可以由各种类以不同的方式实现。在排除由尚未排除的其他类实现的接口时，将打开一个对话框，询问是否还要在所有实现它的类中排除该接口。

注意，清除接口旁边的复选框与在 “排除的接口”对话框中选择接口是等效的。

- **环境：**要录制的环境：ADO 对象、 RDS 对象和远程对象。清除不希望录制的对象。
- **类型库：**类型库 *.tlb* 或 *.dll* 文件，表示要录制的 COM 对象。所有 COM 对象都有一个表示它们的类型库。可以从注册表、 Microsoft Transaction Server 或文件系统中选择一个类型库。

**类型库：**VuGen 在该对话框的下半部分显示每个类型库的下列信息。

- **类型库：**类型库 （tlb 文件）的名称。
- **路径：**类型库的路径。
- **Guid：**类型库的全局唯一标识符。

### 设置筛选器

本部分描述了如何设置筛选器。

**要选择录制的 COM 对象，请执行下列操作：**

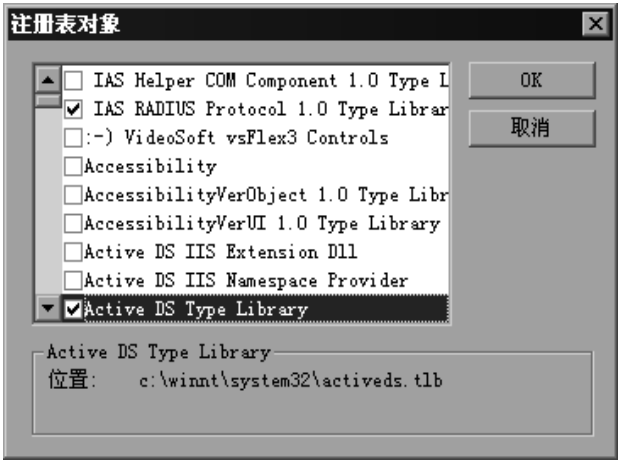
1 从主菜单中选择 “工具” > “录制选项”，或在 “开始录制” 对话框中单击 “选项”。将 [ 打开一个显示 “录制选项” 树的对话框。选择 “COM/DCOM：筛选器” 节点。

展开 “环境” 子树显示 “ADO”、“RDS” 和 “远程对象” 列表。筛选器还包含一个初始为空的 “类型库” 树。您可以添加类型库，如以下步骤中所述。

默认情况下，所有环境都被选中，且筛选器中包含对它们任意对象的调用。清除 “ADO”、“RDS” 或 “远程对象” 旁边的复选框将其从筛选器中排除。

2 单击 “添加” 添加其他 COM 类型库，并如下所述选择要浏览的位置：注册表、文件系统或 MTS。

3 选择 “浏览注册表” 显示本地计算机注册表中的类型库列表。




选中所需库旁边的复选框，然后单击 “确定”，可以复选多个库。

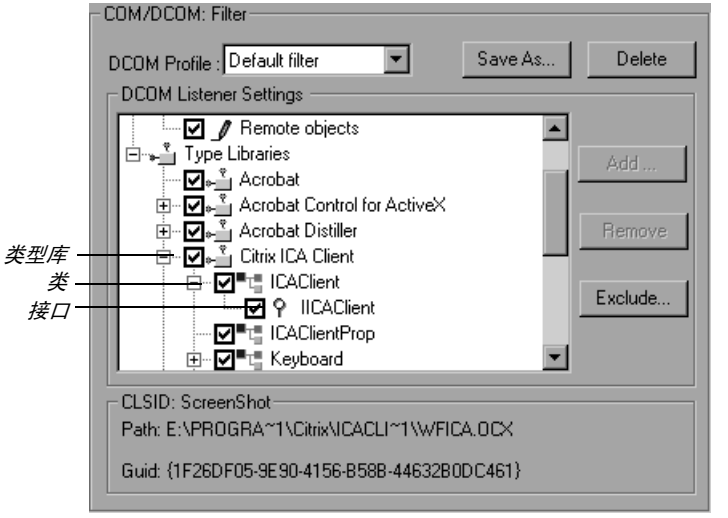
4 要添加文件系统中的类型库，单击 “添加”，然后选择 “浏览文件系统”。

选择所需的文件，然后单击 “确定”。

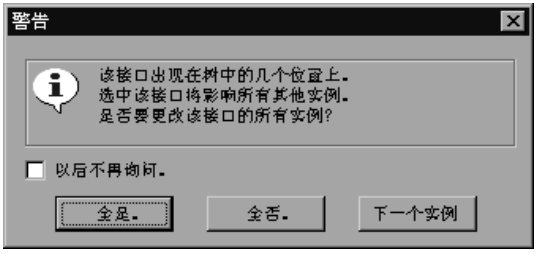
**5** 该类型库显示在类型库列表中后，可以展开树显示该类型库中的所有可用类。您可以展开类树显示该类支持的所有接口。

要排除类型库，请清除库名称旁边的复选框。这将排除此类型库中的所有类。展开树结构后，可以通过清除项目旁边的复选框排除单个类或接口。

注意，清除接口旁边的复选框与在 “排除的接口”对话框中选择接口是等效的。

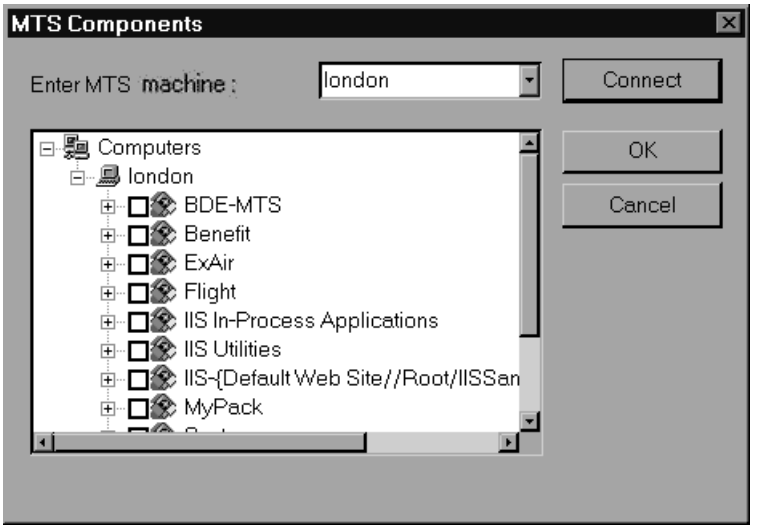



**6** 一个接口可以由各种类以不同的方式实现。在排除由尚未排除的其他类实现的接口时， VuGen 将显示下列警告：




如果选中 “不要再问我”并关闭该对话框，则每当更改一个对象中接口的状态时，该筛选器中所有其他类的所有接口实例的状态将自动更改。单击 “全是”更改所有其他类中该接口的所有实例的状态，单击 “全否”保持所有其他实例的状态不变。单击 “下一个实例”查看使用该接口的下一个类。

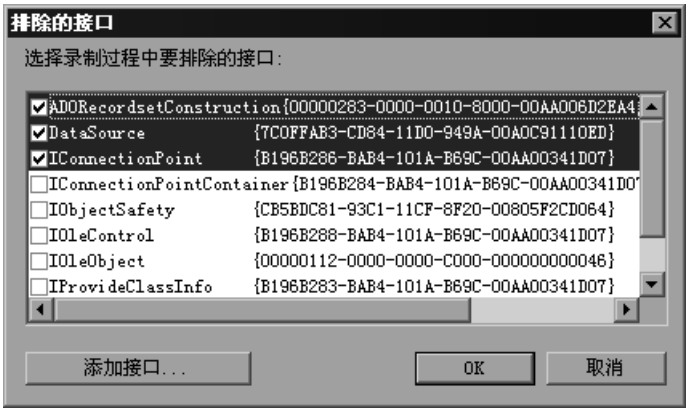
**7** 要添加 Microsoft Transaction Server 中的组件，单击 “添加”，然后选择 “浏览 MTS”。将出现 “MTS 组件”对话框，提示输入 MTS 服务器的名称。




键入 MTS 服务器的名称，然后单击 “连接”。注意，要录制 MTS 组件，计算机上应安装 MTS 客户端。

从可用软件包列表中选择一个或多个 MTS 组件包，然后单击 “添加”。类型库列表中显示该软件包后，可以从中选择特定组件。

**8** 除禁用和启用树视图中的接口录制外，还可以在 “录制选项”对话框中单击 “排除”，在筛选器中包含或排除接口 （无论该接口源自何处）。注意，在类型库树层次结构中，还可以通过清除项目旁边的复选框排除类和接口。



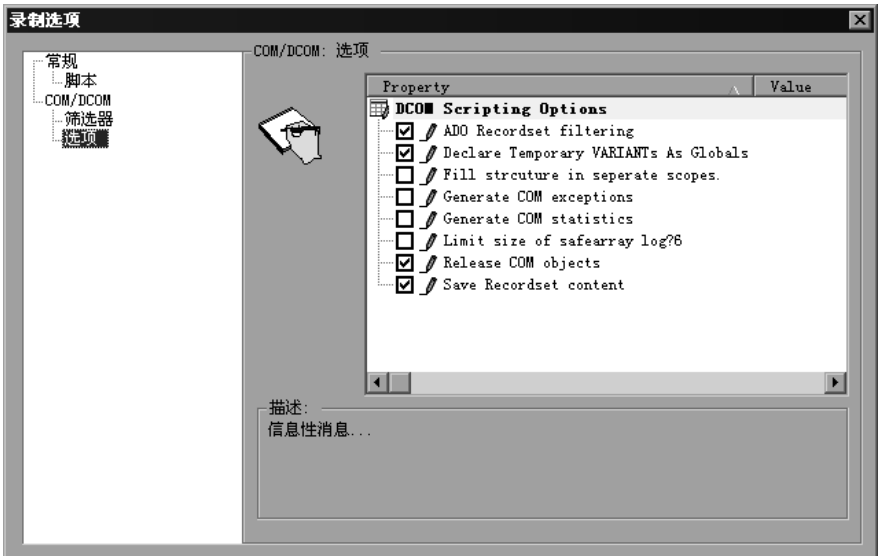

选中的接口列表将被排除。也可以添加列表中未列出的接口。单击 “排除的接口”对话框中的 “添加接口 ...”并输入该接口的 GUID 编号 （接口 ID）和名称。您可以从 VuGen 创建的 interfaces.h 文件中复制 GUID，该文件位于 VuGen 屏幕左侧列的选择树中。使用 “添加接口 ...”功能可以排除由脚本调用但未在筛选器的任何位置列出的多余接口。

**9** 完成筛选器修改后，单击 “确定”保存并关闭该对话框。单击 “另存为”保存一个 “新筛选器”，或使用新名称保存一个现有筛选器。在随后的录制中可以选择已保存的筛选器。“默认筛选器”最初提供了默认设置。

## 设置 COM 脚本选项

可以为 COM 录制会话设置其他有关对象处理、日志生成和 VARIANT 定义的选项。

DCOM 脚本选项适用于所有编程语言。通过这些设置，可以为 DCOM 方法和接口的处理配置脚本选项。




**ADO 记录集筛选**：将多个记录集操作合并到单行 fetch 语句中。（默认情况下启用）

**保存记录集内容**：将记录集内容存储在网格中以便在 VuGen 中查看多个记录集。（默认情况下启用）

**生成 COM 异常**：生成在录制过程中引发异常的 COM 函数和方法。（默认情况下启用）

**释放 COM 对象**：记录已不使用的 COM 对象的释放。（默认情况下禁用）

**限制 SafeArray 日志的大小**：将每个 COM 调用在 safearray 日志中打印的元素个数限制为 16。（默认情况下启用）

**生成 COM 统计信息**：生成录制时间性能统计和概要信息。（默认情况下禁用）

**声明临时变量为全局变量**：将临时变量类型定义为全局，而不是局部变量。（默认情况下禁用）

**要设置 COM/DCOM 选项，请执行下列操作：**

1. 从主菜单中选择 “工具” > “录制选项”，或在 “开始录制” 对话框中单击 “选项 ...”。 VuGen 将打开 “录制选项” 树。选择 “COM/DCOM: 选项” 节点。
2. 单击选项旁边的复选框启用所需的选项。
3. 单击 “确定” 保存设置并退出。

# 26

# 了解 COM Vuser 脚本

本章详细介绍了 VuGen 为 COM 客户端通信生成的脚本，并对函数调用和示例进行了说明。有关 COM Vuser 脚本的基本入门信息，请参阅第 25 章 “录制 COM Vuser 脚本”。

本章描述：

- 了解 VuGen COM 脚本结构
- 检查 VuGen COM 脚本示例
- 在脚本中扫描关联

***下列信息仅适用于 COM Vuser 脚本。***

## 关于 COM Vuser 脚本

在录制 COM 客户端通信时， VuGen 通过调用 COM API 函数和接口方法创建脚本。此外，您还可以编写 COM 类型转换函数。每个函数调用都包含一个 **lrc** 前缀，如 **lrc_CoCreateInstance** 或 **lrc_long**。本章概述了 COM API 和类型转换函数。有关各函数的语法和示例，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

接口方法调用应遵循以下名称和语法格式：

**lrc_< 接口名称 >_< 方法名称 >( 实例 , ...);**

注意，实例始终是第一个传递的参数。

各 COM 组件的供应商通常会为接口函数提供文档。

VuGen 为每个 COM Vuser 脚本创建了下列各项：

- interfaces.h 文件中的接口指针和其他变量声明。
- 可录制到 vuser_init、 action 或 vuser_end 部分中的函数调用。
- 一个 user.h 文件，包含 Vuser 脚本到低级调用的转换。

录制脚本后，可以在 VuGen 屏幕左侧的树中，选择任意的文件进行查看。

# 了解 VuGen COM 脚本结构

VuGen COM 脚本采用了特殊的构造方式，以符合 COM 接口的要求。

### 接口方法

接口方法调用应遵循以下名称和语法格式：

**lrc_< 接口名称 >_< 方法名称 >( 实例 ,...);**

注意，实例始终是第一个传递的参数。

各 COM 组件的供应商通常会为接口函数提供文档。

### 接口指针

interfaces.h 文件定义了接口指针和其他要在脚本中使用的变量。每个接口都有一个接口 ID (IID)，接口 ID 唯一标识接口。

接口定义的格式为：

**< 接口类型 >*< 接口名称 > = 0; // “{< 接口类型的 IID>} ”**

在下例中，接口类型为 **IDispatch**，接口实例的名称为 **IDispatch_0**，**IDispatch** 类型的 **IID** 为长数字字符串：

```
IDispatch* IDispatch_0= 0;//"{00020400-0000-0000-C000-
000000000046}"
```

### Vuser 脚本语句

COM Vuser 脚本由创建对象实例、检索接口指针以及调用接口方法的代码组成。每个用户操作都可能生成一个或多个 COM 调用。 VuGen 将每个 COM 调用都编码为语句组。每个这样的语句组都位于单独的位置，由大括号括起。多个不同的语句通过分配值并执行类型转换为执行主调用作准备。例如，创建对象所需的调用组可能类似于以下形式：

```
{
GUID pClsid = lrc_GUID("student.student.1");
IUnknown * pUnkOuter = (IUnknown*)NULL;
unsigned long dwClsContext = lrc_ulong("7");
GUID riid = IID_IUnknown;
lrc_CoCreateInstance(&pClsid, pUnkOuter, dwClsContext, &riid,
(void**)&IUnknown_0, CHECK_HRES);
}
```

### 错误检查

每个 COM 方法或 API 调用都返回一个错误值。根据原始录制过程中的调用是否成功， Vugen 将设置一个标志，指示是否在回放过程中检查错误。该标志为函数调用的最后一个参数，可以是下列值：

| | |
|---|---|
| **CHECK_HRES** | 如果函数在录制过程中通过，且在回放过程中应进行错误检查，将插入该值 |
| **DONT_CHECK_HRES** | 如果函数在录制过程中失败，且在回放过程中不应进行错误检查，将插入该值 |

## 检查 VuGen COM 脚本示例

本部分举例说明了 VuGen 如何模拟 COM 客户端应用程序。

### 基本 COM 脚本操作

基本操作包括：

- 实例化对象
- 检索接口指针
- 调用接口方法

每种类型的操作都在单独的作用域中执行。

### 实例化对象

要使用 COM 对象，应用程序必须先将其实例化，并获取该对象接口的指针。

**VuGen 执行下列操作实例化对象：**

1 VuGen 调用 **lrc_GUID** 获取该对象的唯一 ProgID，并将其存储在 *pClsid* 中：

```
GUID pClsid = lrc_GUID("student.student.1");
```

**pClsid** 是对象的全局唯一 CLSID，它由 **ProgID** *"student.student.1"* 转换而来

2 如果未知的接口指针指向集合对象，则 VuGen 将检索指向该对象的指针，或者将其设置为 NULL：

```
IUnknown * pUnkOuter = (IUnknown*)NULL;
```

3 VuGen 设置要创建的对象的上下文：

```
unsigned long dwClsContext = lrc_ulong("7");
```

**dwClsContext** 包含进程、本地、远程或这些位置组合中的对象的上下文。

4 VuGen 设置一个变量，保存请求的接口 ID，本例中为 **IUnknown**：

```
GUID riid = IID_IUnknown;
```

**riid** 包含 **IUnknown** 接口的接口 ID。

5 准备好输入参数后， **lrc_CoCreateInstance** 调用将使用以上语句中定义的参数创建一个对象，并返回一个指向 **IUnknown** 接口的指针，用于下一阶段。

```
lrc_CoCreateInstance(&pClsid, pUnkOuter, dwClsContext, &riid,
(void**)&IUnknown_0, CHECK_HRES);
```

以上部分准备并介绍了输入参数。由于调用成功， VuGen 将插入 **CHECK_HRES** 值，设置用户模拟过程中的错误检查。该调用返回一个 IUnknown_0 中指向 IUnknown 接口的指针，供后面的调用使用。

### 检索接口

创建对象后， VuGen 将只能访问 **IUnknown** 接口。 VuGen 将使用 **IUnknown** 接口与对象进行通信。这可以通过 **IUnknown** 标准接口的 **QueryInterface** 方法来完成。 VuGen 方法调用中的第一个参数是接口实例。本例中，它是前面由 **CoCreateInstance** 返回的 **IUnknown_0** 指针。**QueryInterface** 调用要求输入要检索的接口的 ID，并返回指向该接口的指针。

**要获取指向接口的指针，请执行下列操作：**

1 首先， VuGen 设置一个参数 **riid**，该参数为 Istudent 接口的 ID：

```
GUID riid = IID_Istudent;
```

2 如果 Istudent 对象包含这样的接口，则调用 **QueryInterface** 将返回一个指向 **Istudent** 接口的指针：

```
lrc_IUnknown_QueryInterface(IUnknown_0, &riid, (void**)&Istudent_0,
CHECK_HRES);
```

### 使用接口设置数据

以下是一个使用接口方法设置数据的示例。假设在应用程序中，用户需要输入一个名称。这将激活一个方法以设置该名称。 VuGen 使用两个语句录制这一操作。一个语句用于设置名称字符串，另一个语句设置名称属性。

**要设置整个函数调用，请执行下列操作：**

1 首先， VuGen 设置一个变量 (**Prop Value**)，存储该字符串。该参数的类型为 BSTR （COM 文件中使用的一种字符串类型）：

```
BSTR PropValue = lrc_BSTR("John Smith");
```

在随后的阶段，您可能要将该调用参数化 （使用参数替换 “John Smith”），以便在每次运行 Vuser 脚本时能够使用不同的名称。

2 然后， VuGen 调用 **Istudent** 接口的 **Put_Name** 方法输入该名称：

```
lrc_Istudent_put_name(Istudent_0, PropValue, CHECK_HRES);
```

### 使用接口返回数据

从应用程序返回数据不同于输入数据，因为您可能要存储这些值，并在以后的参数化调用中将其用作输入值。

**以下是应用程序在检索数据时， VuGen 可能执行的操作的示例：**

**1** 创建一个适当类型 （本例中为 BSTR）的变量，以存储属性值：

```
BSTR pVal;
```

**2** 使用本例中 **Istudent** 接口的 **get_name** 方法将属性值 （本例中为一个名称）存储于前面创建的 **pVal** 变量中：

```
lrc_Istudent_get_name(Istudent_0, &pVal, CHECK_HRES);
```

**3** 然后， VuGen 生成一条语句，以保存该值：

```
//lrc_save_BSTR("param-name",pVal);
```

该语句被注释掉。您可以删除注释并将 *param-name* 更改为一个具有有意义名称的变量，以存储该值。 Vugen 将使用该变量保存由上一个调用返回的 **pVal** 的值。在随后对其他方法的调用中，可以将该变量用作参数化输入。

### IDispatch 接口

大多数 COM 对象都具有特定的接口。其中的多数对象还实现一个名为 **IDispatch** 的通用接口， VuGen 将以特殊方式转换该接口。 **IDispatch** 是一个用于公开 COM 对象的所有其他接口和方法的 “超级接口”。 VuGen 脚本中的 **IDispatch:Invoke** 方法调用是由 **lrc_Disp** 函数实现的。这些调用与其他接口调用的构造方式略有不同。

**IDispatch** 接口的 **Invoke** 方法可以执行一个方法，可以获取一个属性值，也可以设置一个属性的值或引用值。在标准的 **IDispatch:Invoke** 方法中，这些不同的用法由 *wflags* 参数标识。在 VuGen 实现中，它们在调用方法、设置或获取属性的不同过程调用中实现。

例如，用于激活 ***GetAgentsArray*** 方法的 **IDispatch** 调用可能类似于以下形式：

```
retValue = lrc_DispMethod1((IDispatch*)IDispatch_0, "GetAgentsArray",
/*locale*/1033, LAST_ARG, CHECK_HRES);
```

以上调用中的参数包括：

| | |
|---|---|
| **IDispatch_0** | 这是对前面的 **IUnknown:Queryinterface** 方法调用返回的 **IDispatch** 接口的指针 |
| **GetAgentsArray** | 这是要调用的方法的名称。 VuGen 将通过后台操作，由该名称获取方法 ID |
| **1033** | 这是语言区域设置 |
| **LAST_ARG** | 该标志用于通知 **IDispatch** 接口再没有其他的参数 |
| **CHECK_HRES** | 因为调用在录制过程中成功，所以此标志打开对 HRES 的检查 |

此外，可能还存在另一个参数 **OPTIONAL_ARGS**。该参数指示除标准参数以外， VuGen 还要发送一些可选参数。每个可选参数都由一对参数 ID/ 参数名称和值组成。例如：

```
{
        GUID riid = IID_IDispatch;
        lrc_IOptional_QueryInterface(IOptional_0, &riid,
(void**)&IOptional_0, CHECK_HRES);
    }
    {
        VARIANT P1 = lrc_variant_short("47");
        VARIANT P2 = lrc_variant_short("37");
        VARIANT P3 = lrc_variant_date("3/19/1901");
        VARIANT var3 = lrc_variant_scode("4");
        lrc_DispMethod((IDispatch*)IOptional_0, "in_out_optional_args",
/*locale*/1024, &P1, &P2, OPTIONAL_ARGS, "#3", &P3, "var3", &var3,
LAST_ARG, CHECK_HRES);
```

“LRC 函数参考”部分详细介绍了使用 **IDispatch** 接口的不同 **lrc_Disp** 方法。

### 类型转换和数据提取

如上例所示，许多 COM 参数都被定义为变量。为提取这些值， VuGen 使用一些从对应的 COM 函数中派生的转换函数。第 27 章 “了解 COM Vuser 函数”中提供了一个完整列表。前面已经介绍了如何使用 **lrc_DispMethod1** 调用检索名称字符串数组：

```
VARIANT retValue = lrc_variant_empty();
retValue = lrc_DispMethod1((IDispatch*)IDispatch_0, "GetAgentsArray",
/*locale*/1033, LAST_ARG, CHECK_HRES);
```

下例介绍了 VuGen 如何从 **retValue** （一个字符串数组变量）中获取字符串。

首先， VuGen 从该变量中提取 BSTR 数组：

```
BstrArray array0 = 0;
array0 = lrc_GetBstrArrayFromVariant(&retValue);
```

在下例中，所有值都存储在 **array0** 中， VuGen 为您提供了用于提取数组元素以便稍后在参数化中使用的代码：

```
//GetElementFrom1DBstrArray(array0, 0); // 值：Alex
//GetElementFrom1DBstrArray(array0, 1); // 值：Amanda
....
```

VuGen 具有大量类型转换函数和从变量中提取常规类型的函数。第 27 章“了解 COM Vuser 函数”中详细介绍了这些函数，您也可以参阅 《Online Function Reference》。

## 在脚本中扫描关联

VuGen 提供了一个关联实用程序，帮助您修复脚本，确保成功回放。它执行下列步骤：

- 扫描可能的关联
- 插入适当的关联函数将结果保存到参数
- 使用参数替换语句值

可以在整个脚本或在脚本中的特定位置执行自动关联。

本部分描述了如何确定需要关联的语句。如果您已经知道要关联的值，请转到下一部分了解关联特定值的说明。

**要扫描并使用自动关联关联脚本，请执行下列操作：**

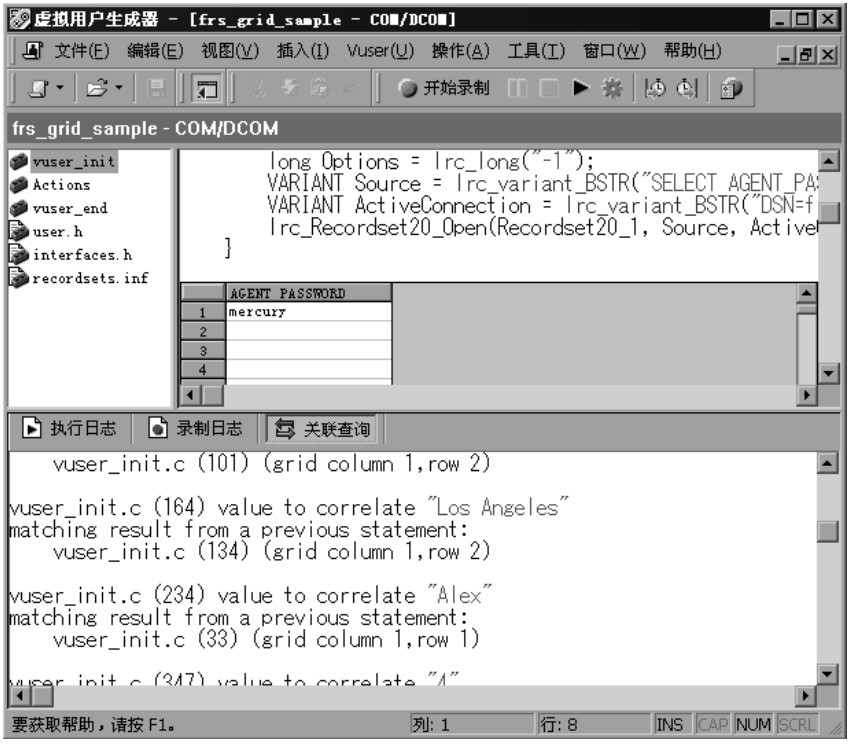
**1** 打开 “输出”窗口。

选择 “视图” > “输出”，将在该窗口底部显示 “输出”选项卡。检查 “执行日志”文件夹中是否存在错误。这些错误通常可以通过关联校正。

**2** 选择 “Vuser” > “扫描关联”。

VuGen 将扫描整个脚本并在 “关联查询”选项卡中列出所有可能要关联的值。

在下例中， VuGen 在 lrc_variant_BSTR ( “SELECT...” ) 语句中找到了几个可能需要关联的值。



3 关联值。

在 “关联查询” 选项卡中，双击要关联的结果。该结果位于消息的第三行，并显示 “网格 列 x，行 x”。 VuGen 将光标置于脚本中该值指示的网格位置。

1 在网格中，选择要关联的值，然后选择 “Vuser” > “创建关联”。 VuGen 将提示输入结果值的参数名称。

2 指定一个名称，或接受默认名称。单击 “确定” 继续。 VuGen 插入相应的关联语句 (**lrc_save_<** *类型* **>**)，将结果保存到参数。

3 单击 “是” 确认关联。

将显示一个消息，询问您是否在脚本中搜索该值出现的所有位置。

4 单击 “否” 仅替换选定语句中的值。

5 要搜索其他出现该值的位置，请单击 “是”。

将打开 “搜索和替换” 对话框。确认所有替换，包括原始语句。完成所需的替换后，单击 “取消” 关闭 “搜索和替换” 对话框。

VuGen 使用参数引用替换语句值。注意，如果选择取消关联， VuGen 还将清除在上一步中创建的语句。

## 关联已知值

如果已经知道了要关联的值，请执行下列操作：

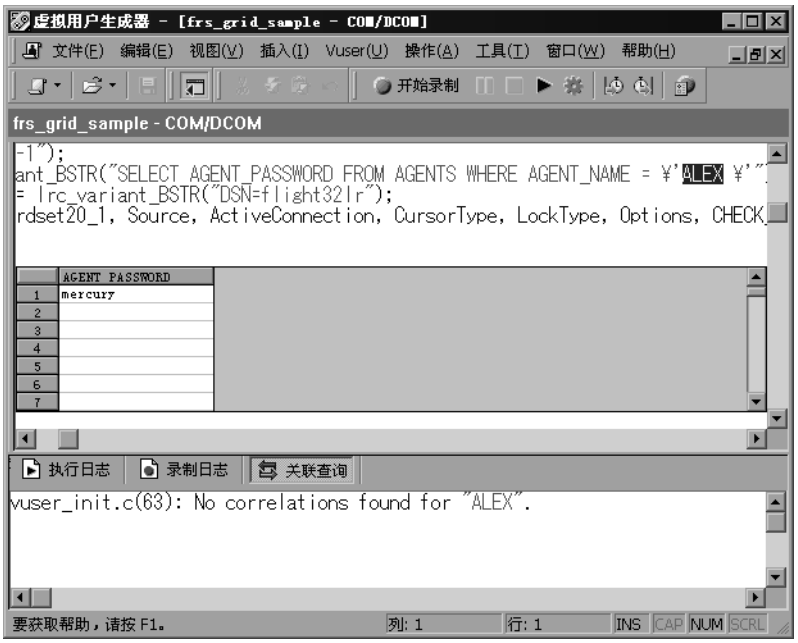
**关联特定值：**

1 找到并选择 （不带引号）要关联的值。

2 选择 “Vuser” > “扫描关联 ( 光标处 )”。

VuGen 扫描该值并列出脚本中与该值匹配的所有结果。“关联查询” 选项卡中列出了关联值。

在下例中， VuGen 找到一个与 “Alex” 匹配的结果值。




在 “关联查询” 选项卡中，双击要关联的结果。该结果位于消息的第三行，并显示 “网格 列 x，行 x”。 VuGen 将光标置于脚本中该值指示的网格位置。

**3** 在网格中，选择要关联的值，然后选择 “Vuser” > “创建关联”。 VuGen 将提示输入结果值的参数名称。

**4** 指定一个名称，或接受默认名称。单击 “确定” 继续。 VuGen 插入相应的关联语句 (**lrc_save_**< *类型*>)，将结果保存到参数。

```
lrc_save_rs_param (Recordset20_0, 1, 1, 0, “Saved_AGENT_NAME”);
```

5 单击 “是”确认关联。

将显示一个消息，询问您是否在脚本中搜索该值出现的所有位置。

6 单击 “否”仅替换选定语句中的值。

7 要搜索其他出现该值的位置，请单击 “是”。

将打开 “搜索和替换”对话框。确认所有替换，包括原始语句。完成所需的替换后，单击 “取消”关闭 “搜索和替换”对话框。

VuGen 使用参数引用替换语句值。注意，如果选择取消关联， VuGen 还将清除在上一步中创建的语句。

# 27

# 了解 COM Vuser 函数

COM Vuser 函数模拟运行 COM 应用程序的用户的操作。

本章描述：

- 创建实例
- IDispatch 接口调用方法
- 类型分配函数
- 变量类型
- 从引用到变量的分配
- 参数化函数
- 从变量中提取
- 将数组分配给变量
- 数组类型和函数
- 字节数组函数
- ADO 记录集函数
- 调试函数
- VB 集合支持

***下列信息仅适用于 COM Vuser 脚本。***

# 关于 COM Vuser 函数

每个 VuGen COM 函数都以 lrc 前缀开头。 VuGen 录制本部分中列出的 COM API 和方法调用。也可以对 lrc 类型的转换调用进行手动编程。有关 lrc 函数的语法和示例，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

您可以指明用于创建 Vuser 脚本的编程语言 – C 或 Visual Basic 脚本。详细信息，请参阅第 4 章 “设置脚本生成首选项”。下列部分描述为 C 语言类型的虚拟用户脚本生成的函数。

## 创建实例

从相应的 COM 函数中派生出的几个用于创建和释放对象的函数：

| | |
|---|---|
| **lrc_CoCreateInstance** | 创建对象实例并返回未知的接口 |
| **lrc_CreateInstanceEx** | 创建远程计算机上对象的实例，并能够返回多个接口 |
| **lrc_CoGetClassObject** | 获取指定类的类工厂。随后可使用类工厂创建该类的多个对象 |
| **lrc_Release_Object** | 释放不再使用的 COM 对象 |

## IDispatch 接口调用方法

下列调用使用 **Invoke** 方法并在 **Invoke** 的 **wflags** 参数中设置不同的标志值以调用 **IDispatch** 接口：

| | |
|---|---|
| **lrc_DispMethod** | 使用 IDispatch:Invoke 方法调用接口的方法 |
| **lrc_DispMethod1** | 使用 IDispatch 接口调用方法并获取同一名称的属性 |
| **lrc_DispPropertyGet** | 使用 IDispatch 接口获取属性 |
| **lrc_DispPropertyPut** | 使用 IDispatch 接口设置属性 |
| **lrc_DispPropertyPutRef** | 使用 IDispatch 接口通过引用设置属性 |

## 类型分配函数

要补充 VuGen 自动录制的函数，可以在脚本中手动编写类型分配函数。类型转换函数将字符串数据分配给指定类型。函数名称为：

**lrc_< 类型名称 >**

其中， **<** 类型名称 **>** 可以是下列数据类型之一：

| | |
|---|---|
| **ascii_BSTR** | ascii BSTR |
| **bool** | 布尔 |
| **BSTR** | BSTR |
| **BYTE** | 字节 |
| **char** | 字符变量 |
| **currency** | 货币 |
| **date** | 日期 |
| **double** | 双精度型 |
| **dword** | 双字 |
| **float** | 浮点型数字 |
| **GUID** | 返回指定对象的 GUID |
| **hyper** | 超整型 |
| **int** | 整型 |
| **long** | 长整型 |
| **short** | 短整型 |
| **uint** | 无符号整型 |
| **ulong** | 无符号长整型 |
| **uhyper** | 无符号 64 位超整型 |
| **ushort** | 无符号短整型 |

## 变量类型

变量可以包含任何类型的信息。例如，变量可以是字符串或双字数组。变量还可以是变量数组。 VuGen 可以将字符串数据转换为各种变量类型。函数命名为：

**lrc_variant_< 类型名称 >**

其中， < 类型名称 > 可以是下列任何类型：

| | |
|---|---|
| **ascii BSTR** | ascii BSTR 变量 |
| **bool** | 布尔型变量 |
| **BSTR** | BSTR 变量 |
| **BYTE** | 无符号字符 (BYTE) 变量 |
| **char** | 字符变量 |
| **CoObject** | **IUnknown** 接口指针 |
| **currency** | 货币变量 |
| **date** | 日期变量 |
| **DispObject** | **IDispatch** 接口指针 |
| **float** | 浮点型变量 |
| **int** | 整型变量 |
| **long** | 长整型变量 |
| **scode** | 错误代码变量 |
| **short** | 短整型变量 |
| **uint** | 无符号整型变量 |
| **ulong** | 无符号长整型变量 |
| **ushort** | 无符号短整型变量 |

除了变量类型转换函数以外，还有三个用于新建变量的函数：

| | |
|---|---|
| **lrc_variant_empty** | 创建一个空变量 |
| **lrc_variant_null** | 创建一个 null 变量 |
| **lrc_variant_variant_by_ref** | 新建一个包含现有变量的变量 |

## 从引用到变量的分配

VuGen 可以向存储在变量中的引用分配变量。函数命名为：

**lrc_variant_< 类型名称 >_by_ref**

其中， < 类型名称 > 可以是下列任何类型：

| | |
|---|---|
| **ascii BSTR** | ascii BSTR 变量 |
| **bool** | 布尔型变量 |
| **BSTR** | BSTR 变量 |
| **BYTE** | BYTE 变量 |
| **char** | 字符变量 |
| **CoObject** | **IUnknown** 接口指针 |
| **currency** | 货币变量 |
| **date** | 日期变量 |
| **DispObject** | **IDispatch** 接口指针 |
| **float** | 浮点型变量 |
| **int** | 整型变量 |
| **long** | 长整型变量 |
| **scode** | scode 变量 |
| **short** | 短整型变量 |
| **uint** | 无符号整型变量 |
| **ulong** | 无符号长整型变量 |
| **ushort** | 无符号短整型变量 |
| **from_variant** | 从变量中检索变量 |

## 参数化函数

参数化函数将指定类型的值保存到字符串参数。参数化函数的语法如下所示：

**lrc_save_< 类型名称 >**

**lrc_save_VARIANT_< 类型名称 >**

将给定 < 类型名称 > 的变量保存为变量。

**lrc_save_VARIANT_< 类型名称 >_by_ref**

将给定 < 类型名称 > 的变量保存为变量中的引用。

该值从 < 类型名称 > 转换为字符串。它存储在参数中。 VuGen 会将这些语句注释掉。要使用它们，请将参数名称更改为有意义的名称并删除语句的注释符号。接下来，可以使用参数作为随后调用的输入。 < 类型名称 > 可以是下列类型之一：

| | |
|---|---|
| **ascii_BSTR** | ascii BSTR |
| **bool** | 布尔 |
| **BSTR** | BSTR |
| **BYTE** | 字节 |
| **char** | 字符类型 |
| **currency** | 货币 |
| **date** | 日期 |
| **double** | 双精度型 |
| **dword** | 双字 |
| **float** | 浮点型数字 |
| **hyper** | 超整型 |
| **int** | 整型 |
| **long** | 长整型 |
| **uint** | 无符号整型 |
| **ulong** | 无符号长整型 |
| **short** | 短整型 |

| | |
|---|---|
| **uhyper** | 无符号超整型 |
| **ushort** | 无符号短整型 |
| **VARIANT** | 变量 |

VuGen 还添加了一个用于参数化 COM 脚本的 save 语句，以处理在网格中进行关联的请求。

## 从变量中提取

有几个用于从变量中提取数据的函数：

| | |
|---|---|
| **lrc_CoObject_from_variant** | 从变量中提取指向 IUnknown 接口的指针 |
| **lrc_CoObject_by_ref_from_variant** | 从变量中的引用提取指向 IUnknown 接口的指针 |
| **lrc_DispObject_from_variant** | 从变量中提取指向 IDispatch 接口的指针 |
| **lrc_DispObject_by_ref_from_variant** | 从变量中的引用提取指向 IDispatch 接口的指针 |

## 将数组分配给变量

下列函数将数组转换为变量：

| | |
|---|---|
| **lrc_variant_< 类型名称 >Array** | 将 < 类型名称 > 类型数组分配给变量 |
| **lrc_variant_< 类型名称 >Array_by_ref** | 将 < 类型名称 > 类型数组分配给变量，其中数组由引用传递 |

## 数组类型和函数

VuGen COM 支持安全数组的函数：

| | |
|---|---|
| **Create<n>D< 类型名称 >Array** | 创建类型名称中指定类型的 n 维数组 |
| **Destroy< 类型名称 >Array** | 释放类型名称中指定类型的数组 |
| **GetElementFrom<n>D< 类型名称 >Array** | 从 SafeArray 中检索指定类型的元素 |
| **PutElementIn<n>D< 类型名称 >Array** | 将元素存储在相应类型的数组中 |
| **lrc_Get< 类型名称 >ArrayFromVariant** | 从变量中提取类型名称中指定类型的数组 |
| **lrc_Get< 类型名称 >Array_by_refFromVariant** | 从变量中的指针引用提取类型名称中指定类型的数组 |
| **Fill<n>DbyteArray** | 使用在指定的 n-1 索引处开始的缓冲区填充字节数组的最后一维 |

在以上函数中， < 类型名称 > 可以是下列数据类型之一：

| | |
|---|---|
| **Bstr** | BSTR |
| **Byte** | 字节 （无符号字符） |
| **Char** | 字符数组 |
| **CoObject** | **IUnknown** 接口 |
| **Currency** | 货币 (CY) |
| **Date** | 日期变量 |
| **DispObject** | **IDispatch** 接口 |
| **Double** | 双精度型 |
| **Dword** | 双字 |
| **Error** | scode 错误 |
| **Float** | 浮点型数字 |

| | |
|---|---|
| **Int** | 整型 |
| **Long** | 长整型 |
| **Short** | 短整型 |
| **UInt** | 无符号整型 |
| **ULong** | 无符号长整型 |
| **UShort** | 无符号短整型 |
| **Variant** | 变量类型 |

## 字节数组函数

有两组函数仅用于填充和检索字节数组中的数据。

| | |
|---|---|
| **Fill<n>DByteArray** | 使用在指定的 n-1 索引处开始的缓冲区填充字节数组的最后一维 |
| **GetBufferFrom<n>DByteArray** | 从 n 维字节数组的最后一维获取在指定的 n-1 索引处的缓冲区 |

**lrc_CreateVBCollection** 调用为 Visual Basic 集合 （该集合为变量安全数组）提供特殊支持。 VuGen 将该集合作为接口进行处理。首次遇到该集合时， VB 将使用 **lrc_CreateVBCollection** 创建一个 “接口”。因此，它可以引用接口地址处的数据。

## ADO 记录集函数

下列是 ADO 记录集函数

| | |
|---|---|
| **lrc_FetchRecordset** | 在记录集中移动指针 |
| **lrc_FetchRecordsetUntillEOF** | 取记录，直到记录集结尾 |
| **lrc_RecordsetWrite** | 更新 ADO 记录集中的字段 |
| **lrc_RecordsetAddColumn** | 向记录集中添加一个新列 |
| **lrc_RecordsetDeleteColumn** | 从记录集中删除一列 |

## 调试函数

**lrc_print_variant** 函数打印变量内容。

## VB 集合支持

**lrc_CreateVBCollection** 函数创建 Visual Basic 集合对象。

# 28

# 开发 Corba-Java Vuser 脚本

通过 VuGen 可以录制用 Java 编写并使用 Corba 的应用程序或小程序。可以运行录制的脚本，也可以使用标准的 Java 库函数以及 LoadRunner 特有的 Java 函数来增强该脚本。

本章描述：

- 录制 Corba-Java Vuser
- 使用 Corba-Java Vuser 脚本
- 在 Windows XP 和 Windows 2000 Server 上录制

***下列信息适用于 Corba-Java Vuser 脚本。***

## 关于 Corba-Java Vuser 脚本

可以使用 VuGen 来录制 CORBA （通用对象请求代理体系结构） Java 应用程序或小程序。 VuGen 将创建一个完全使用 Java 编写的脚本，该脚本通过 LoadRunner 特有的 Java 函数得以增强。录制完后，可以通过使用 JDK 库或自定义类的标准 Java 代码来增强或修改该脚本。

准备好脚本后，可以在 VuGen 中以独立模式运行它。 Sun 的标准 Java 编译器 *javac.exe* 将检查脚本中是否有错误，并对其进行编译。确认脚本可正常运行后，可将该脚本合并到 LoadRunner 场景中。

在通过录制和手动增强来创建脚本时，与 Java Vuser 脚本相关的所有规则和限制都适用。此外，该脚本中使用的所有特定的类（例如， *org.omg.CORBA.ORB*）都必须位于执行该脚本的计算机上，并由 *classpath* 环境变量指示。有关函数语法和系统配置的重要信息，请参阅第 24 章“编写 Java 脚本”。在 Windows XP 和 2000 Server 上录制时，请遵循第 349 页的 “在 Windows XP 和 Windows 2000 Server 上录制”中的规则。

下面几章介绍 Java 录制选项、运行时设置以及关联功能。

# 录制 Corba-Java Vuser

录制 Corba Vuser 前，请验证应用程序或小程序函数能否在录制计算机上正常运行。

确保在运行 LoadRunner 的计算机上正确安装了 Sun 的 JDK 版本 （只有 JRE 是不够的）。录制脚本前必须完成该安装。验证是否按照 JDK 安装说明设置了 *classpath* 和 *path* 环境变量。

有关所需环境设置的详细信息，请参阅第 24 章 “编写 Java 脚本”。

**要开始录制，请执行下列操作：**

**1** 选择 “文件” > “新建”，然后从 “分布式组件” 组中选择 “Corba-Java”。将打开 “开始录制” 对话框。

**2** 从供应商列表中选择 Corba 供应商。

**3** 在 “应用程序类型” 框中，选择适当的值。

**Java 小程序**，用于通过 Sun 的 AppletViewer 录制 Java 小程序。

**Java 应用程序**，用于录制 Java 应用程序。

**Netscape 或 IExplore**，用于录制浏览器中的小程序。

**可执行文件 / 批处理文件**，用于录制从批处理文件中启动的小程序或应用程序。

**侦听器**，用于指示 VuGen 等待批处理文件，该批处理文件用于在进行录制之前初始化配置并运行应用程序。该模式需要使用 jdk1.2.x 及更高版本将系统变量 _JAVA_OPTIONS 定义为 *Xrunjdkhook*。（对于 jdk 1.1.x，请定义环境变量 *_classload_hook=JDKhook*。）

**4** 如果是从网络下载 Corba 类，则请在 “供应商类” 框中，选择 “网络”。否则如果是在本地加载 Corba 类 （如 JDK 1.2 及更高版本），则仅支持 “本地”。

**5** 根据下表指定其他参数：

| 应用程序类型 | 要设置的字段 |
|---|---|
| Java 小程序 | 小程序路径、工作目录 |
| Java 应用程序 | 应用程序主类、工作目录、应用程序参数 |
| IExplore | IExplore 路径、 URL |
| Netscape | IExplore 路径、 URL |
| 可执行文件 / 批处理文件 | 可执行文件 / 批处理文件、工作目录 |
| 侦听器 | N/A |

注意，仅当应用程序必须知道工作目录的位置 （例如，读取属性文件或写入日志文件）时，才需要指定工作目录。

**6** 要设置录制选项 （如 JVM 的命令行参数），请单击 “选项”。有关设置录制选项的信息，请参阅第 14 章 “设置 Java 录制选项”。

**7** 在 “录制到操作” 框中，选择与要录制到的方法相对应的部分。 Actions 类包含三个方法：*init*、*action* 和 *end*，分别对应于 vuser_init、 Actions 和 vuser_end 部分。下表显示要包括在每个方法中的内容以及执行每个方法的时间。

| Actions 类中的方法 | 录制到操作 | 模拟的情形 ... | 执行期间 ... |
|---|---|---|---|
| *init* | *vuser_init* | 登录到服务器 | 初始化 |
| *action* | *Actions* | 客户端活动 | 正在运行 |
| *end* | *vuser_end* | 注销过程 | 已完成或已停止 |

**注意：** 确保在 *vuser_init* 部分中导入 org.omg.CORBA.ORB 函数，以便它不会在每次循环时重复执行。

**8** 单击 “确定”开始录制。 VuGen 将启动应用程序，并将自身最小化，同时打开进度条和浮动录制工具栏。该进度工具栏显示类在加载时所用的名称。这指明已激活了 Java 录制支持。

**9** 在应用程序内执行典型操作。可使用浮动工具栏在录制过程中切换方法。

**10** 录制典型的用户操作后，从浮动工具栏中选择 *vuser_end* 方法。

执行注销过程。 VuGen 会将该过程录制到脚本的 *vuser_end* 方法中。

**11** 单击 “录制”工具栏上的 “停止录制”。 VuGen 脚本编辑器将显示录制的所有语句。

**12** 单击 “保存”以保存该脚本。将打开 “保存测试”对话框 （仅限于新的 Vuser 脚本）。指定脚本名称。

# 使用 Corba-Java Vuser 脚本

Corba 特有的脚本通常都有完善的模式。第一部分包括 ORB 初始化和配置。第二部分指明 Corba 对象的位置。第三部分包括服务器对 Corba 对象的调用。最后部分包括用于关闭 ORB 的关闭过程。注意，该模式并不是强制性的，其中每个部分都可以在脚本中多次出现。

在以下代码段中，该脚本将初始化 ORB 实例，并执行绑定操作以获取 Corba 对象。请注意 VuGen 是如何导入所有必需的类的。

```
import org.omg.CORBA.*;
import org.omg.CORBA.ORB.*;
import lrapi.lr;

public class Actions {

  // 公共函数：init
  public int init() throws Throwable {

  // 初始化 Orb 实例 ...
  MApplet mapplet = new MApplet("http://chaos/classes/", null);
  orb = org.omg.CORBA.ORB.init(mapplet, null);

  // 绑定到服务器 ...
  grid = grid_dsi.gridHelper.bind("gridDSI", "chaos");
  return lr.PASS;
}
```

org.omg.CORBA.ORB 函数用于与 ORB 建立连接。因此，它只能被调用一次。运行多个循环时，请将该函数置于 *init* 部分中。

在以下部分中， VuGen 将录制对网格 Corba 对象执行的操作。

```
// 公共函数：action
public int action() throws Throwable {

   grid.width();
   grid.height();
   grid.set(2, 4, 10);
   grid.get(2, 4);

   return lr.PASS;
}
```

VuGen 会在该会话结尾录制 ORB 的关闭过程。在整个录制代码中使用的变量位于 *end* 方法之后和 *Actions* 类的右花括号之前。

```
// 公共函数：end
 public int end() throws Throwable {

        if (lr.get_vuser_id() == -1)
            orb.shutdown();

        return lr.PASS;
}

// 变量部分
    org.omg.CORBA.ORB orb;
    grid_dsi.grid grid;
}
```

注意， ORB 关闭语句是针对本产品自定义的。该自定义语句可防止关闭一个 Vuser 即关闭其他所有 Vuser 的情况。

# 在 Windows XP 和 Windows 2000 Server 上录制

在 Windows XP 和 Windows 2000 Server 上录制时，Java 插件可能与 VuGen 的录制器不兼容。要确保正常运行，请在安装 java 插件后以及录制脚本前执行以下过程。

**要针对 Corba-Java 或 Rmi-Java 录制配置计算机，请执行下列操作：**

**1** 从 “控制面板”中打开 “Java Plug-in”。选择 “开始” > “设置” > “控制面板”，然后打开 “Java Plug-in”组件。将打开 “初级”选项卡。

**2** 清除 “启用 Java 插件”复选框，再单击 “应用”。然后再次选中 “启用 Java 插件”复选框，并单击 “应用”。

**3** 打开 “浏览器”选项卡。

**4** 清除 “Microsoft Internet Explorer”复选框，并单击 “应用”。然后再次选中 “Microsoft Internet Explorer”复选框，并单击 “应用”。

# 29

# 开发 RMI-Java Vuser 脚本

通过 VuGen 可以录制那些用 Java 编写并使用 RMI 的应用程序或小程序。可以运行录制的脚本，或者使用标准 Java 库函数和特定于 LoadRunner 的 Java 函数增强脚本。

本章描述：

- 录制 RMI over IIOP
- 录制 RMI Vuser
- 使用 RMI Vuser 脚本

***下列信息适用于 RMI-Java Vuser 脚本。***

## 关于开发 RMI-Java Vuser 脚本

使用 VuGen 可以录制 RMI （远程方法调用） Java 应用程序或小程序。 VuGen 将创建一个纯粹的 Java 脚本，并通过特定于 LoadRunner 的 Java 函数增强该脚本。完成录制后，可以使用 JDK 库或自定义类，通过标准的 Java 代码来增强或修改脚本。

准备好脚本后，可以在 VuGen 中以独立模式运行该脚本。Sun 的标准 Java 编译器 *javac.exe* 将检查脚本中是否有错误，并对其进行编译。验证脚本可正常运行后，可将脚本合并到 LoadRunner 场景中。

通过录制和手动增强的方式创建脚本时，所有与 Java Vuser 脚本相关的规则和限制都适用。此外，脚本中使用的任何特定类都必须存在于执行 Vuser 的计算机上，并通过 *classpath* 环境变量指明。有关函数语法和系统配置的重要信息，请参阅第 24 章 “编写 Java 脚本”。

在 Windows XP 和 2000 Server 上进行录制时，请按照第 349 页的 “在 Windows XP 和 Windows 2000 Server 上录制”中指明的规则进行操作。

## 录制 RMI over IIOP

开发 Internet ORB 间协议 (IIOP) 技术的目的是为了在万维网上实现 CORBA 解决方案。与只支持文本传输的 HTTP 不同， IIOP 使浏览器和服务器可以交换数组之类的复杂对象。

以前，某些服务只能从 RMI 客户端或 CORBA 客户端进行访问，而 *RMI over IIOP* 技术则使单个客户端访问这些服务成为可能。此技术中融合了与 RMI 结合使用的 JRMP 协议以及与 CORBA 结合使用的 IIOP。通过 *RMI over IIOP*， CORBA 客户端可以访问 *Enterprise Java Beans* (EJB) 等新技术以及其他 J2EE 标准。

VuGen 完全支持使用 *RMI over IIOP* 协议录制和回放 Vuser。根据所录制的内容，可以利用 VuGen 的 RMI 录制器创建能够最佳地模拟实际用户的脚本：

- **纯粹的 RMI 客户端：**录制使用本机 JRMP 协议进行远程调用的客户端
- **RMI over IIOP 客户端：**录制使用 IIOP 协议而不是 JRMP 进行编译 （为了实现与 CORBA 服务器的兼容）的客户端应用程序。

## 录制 RMI Vuser

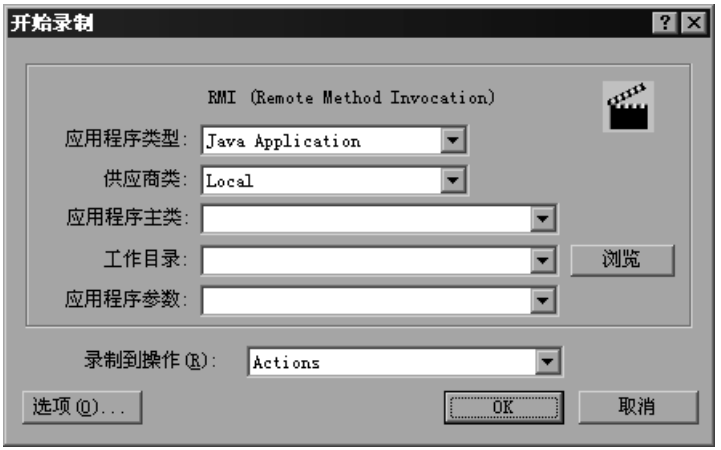
录制 RMI Vuser 之前，请验证应用程序或小程序能否在录制计算机上正常运行。

确保在运行 LoadRunner 的计算机上已正确安装了 Sun 发布的 JDK 版本（仅有 JRE 是不够的）。必须先完成此安装，然后才能录制 Vuser 脚本。验证 *classpath* 和 *path* 环境变量是否已按照 JDK 安装说明进行设置。

开始录制之前，验证环境是否已正确配置。确保 classpath 中存在所需的类，并且已安装了完整的 JDK。有关所需环境设置的详细信息，请参阅第 24 章 “编写 Java 脚本”。

注意，录制过程中从 VuGen 加载小程序或应用程序所需要的时间要比单独加载它们 （独立于 LoadRunner）长几秒钟。

1 要开始录制，请选择 “文件” > “新建”，然后从 “分布式组件” 组中选择 “RMI- Java”。将打开 “开始录制” 对话框。




2 在 “应用程序类型” 框中，选择合适的值。

**Java 小程序**：通过 Sun 的小程序查看器录制 Java 小程序。

**Java 应用程序**：录制 Java 应用程序。

**Netscape 或 IExplore**：在浏览器中录制小程序。

**可执行程序 / 批处理文件**：录制从批处理文件中启动的小程序或应用程序。

**侦听器模式**：指示 VuGen 在录制之前等待批处理文件初始化配置和运行应用程序。此模式要求您将系统变量 _JAVA_OPTIONS 定义为使用 jdk1.2.x 和更高版本的 *--Xrunjdkhook*。（对于 jdk 1.1.x，请定义环境变量 *_classload_hook=JDKhook*。）

3 在 “供应商类” 框中，选择 “网络” 或 “本地”。

4 按照下表指定其他参数：

| 应用程序类型 | 要设置的字段 |
|---|---|
| Java 小程序 | 小程序路径、工作目录 |
| Java 应用程序 | 应用程序主类、工作目录、应用程序参数 |
| IExplore | IExplore 路径、 URL |
| Netscape | Netscape 路径、 URL |

| 应用程序类型 | 要设置的字段 |
|---|---|
| 可执行程序 / 批处理文件 | 可执行程序 / 批处理文件、工作目录 |
| 侦听器 | N/A |

注意，仅当应用程序必须了解工作目录位置的情况下 （例如，读取属性文件或写入日志文件），工作目录才是必需的。

**5** 要设置录制选项 （例如， JVM 的命令行参数），请单击 “选项”。有关设置录制选项的信息，请参阅第 14 章 “设置 Java 录制选项”。

**6** 在 “录制到操作”框中，选择与要录制到的方法相对应的部分。 Actions 类包含三个方法：*init*、*action* 和 *end*，分别对应于 vuser_init、 Actions 和 vuser_end 部分。下表显示了每个方法包含的内容以及每个方法的执行时间。

| Actions 类中的方法 | 录制到操作 | 用于模拟 ... | 执行时间 ... |
|---|---|---|---|
| *init* | *vuser_init* | 登录到服务器 | 初始化 |
| *action* | *Actions* | 客户端活动 | 运行时 |
| *end* | *vuser_end* | 注销过程 | 完成或停止时 |

**7** 单击 “确定”开始录制。 VuGen 将启动应用程序，将自身最小化，然后打开一个进度栏和浮动的录制工具栏。进度工具栏显示正在加载的类的名称。这指明 Java 录制支持已激活。

**8** 在应用程序中执行典型操作。使用浮动工具栏在录制过程中切换方法。

**9** 录制典型的用户操作之后，从浮动工具栏中选择 *vuser_end* 部分。

执行注销过程。 VuGen 将该过程录制到脚本的 *end* 方法中。

**10** 单击录制工具栏上的 “停止录制”。 VuGen 脚本编辑器会显示录制的所有语句。

**11** 单击 “保存”保存脚本。将打开 “保存测试”对话框 （仅限于新的 Vuser 脚本）。指定脚本名。

## 使用 RMI Vuser 脚本

本部分描述了特定于 RMI Vuser 的 Java Vuser 脚本的元素。RMI 不具有构造（如 CORBA 中一样），而是使用 Serializable Java 对象。第一部分执行命名注册表初始化和配置。下一部分在找到 Java 对象 （Remote 和 Serializable）并转换其类型后生成。后面的部分由服务器对这些 Java 对象的调用组成。与 CORBA 不同的是， RMI 中没有特定的关闭部分。注意，对象可能会在脚本中出现多次。

在以下代码段中，找到了一个命名注册表。紧接着是一个查找操作，用于获取特定的 Java 对象。然后，可以使用该对象，并执行 **set_sum**、 **increment** 和 **get_sum** 之类的调用。以下代码段还显示了 VuGen 如何导入所有必要的 RMI 类。

```
Import java.rmi.*;
Import java.rmi.registry.*;

:
:

// 公共函数：action
public int action() throws Throwable {

    _registry = LocateRegistry.getRegistry("localhost",1099);

    counter = (Counter)_registry.lookup("Counter1");

    counter.set_sum(0);
    counter.increment();
    counter.increment();
    counter.get_sum();

    return lr.PASS;
    }
 :
```

录制 RMI Vuser 时，脚本中可能包含多个对 **lr.deserialize** 的调用，以便对所有相关的对象进行反序列化。生成 **lr.deserialize** 调用的原因是，传递给下一个调用的对象无法与前面的任何调用的返回值相关联。因此， VuGen 会录制其状态，并在回放期间使用 **lr.deserialize** 调用表示这些值。反序列化在 VuGen 将对象作为参数传递给调用之前完成。详细说明，请参阅第 180 页的 “使用序列化机制”。

# 第 VIII 部分

## 电子商务协议

# 30

# 开发 FTP Vuser 脚本

通过 VuGen 可以直接访问 FTP 服务器，以模拟网络活动。

本章描述：

➤ 使用 FTP 函数

***下列信息仅适用于 FTP Vuser 脚本。***

## 关于开发 FTP Vuser 脚本

FTP 协议是一种低级协议，通过它可以模拟使用 FTP 服务器的用户的操作。

对于 FTP，可以模拟用户登录到 FTP 服务器、传输文件以及注销。要创建脚本，可以录制 FTP 会话或手动输入 FTP 函数。

录制 FTP 会话时， VuGen 将生成模拟邮件客户端的操作的函数。如果通过多个协议 （例如 FTP、 HTTP 以及邮件协议）进行通信，则可以录制所有这些协议。有关指定多个协议的说明，请参阅第 3 章 “使用 VuGen 进行录制”。

要为 FTP 协议创建脚本，请在电子商务类别中选择 FTP 协议类型。要开始录制，请单击 “录制”按钮并执行针对 FTP 服务器的典型操作。有关创建和录制脚本的详细信息，请参阅第 3 章 “使用 VuGen 进行录制”

创建虚拟用户脚本之后，可以将其集成到 Windows 或 UNIX 平台上的场景中。有关在场景中集成虚拟用户脚本的详细信息，请参阅 《LoadRunner Controller 用户指南》。

# 使用 FTP 函数

您可以指示用于创建 Vuser 脚本的编程语言。详细信息，请参阅第 4 章 “设置脚本生成首选项”。以下部分描述了为 C 语言类型的虚拟用户脚本生成的函数。

FTP Vuser 脚本函数录制文件传输协议 (FTP)。每个 FTP 函数都以 **ftp** 前缀开头。

多数 FTP 函数都成对出现 （一个用于全局会话，另一个用于指示特定邮件会话）。要将该操作应用于所有会话，请使用没有 **ex** 后缀的版本。要将该操作应用于特定会话，请使用带有以 **ex** 为后缀的会话标识符的版本。例如， **ftp_logon** 可以全局登录到 FTP 服务器，而 **ftp_logon_ex** 只能登录到特定会话的 FTP 服务器。

| 函数名 | 描述 |
|---|---|
| **ftp_delete[_ex]** | 从 FTP 服务器删除文件 |
| **ftp_dir[_ex]** | 在 FTP 服务器上运行 dir 命令 |
| **ftp_get[_ex]** | 从 FTP 服务器取文件 |
| **ftp_get_last_error** | 检索从 FTP 服务器接收到的最后一个错误 |
| **ftp_get_last_error_id** | 检索从 FTP 服务器接收到的最后一个错误的 ID |
| **ftp_logon[_ex]** | 执行登录到 FTP 服务器 |
| **ftp_logout[_ex]** | 执行从 FTP 服务器注销 |
| **ftp_mkdir[_ex]** | 在 FTP 服务器计算机上创建目录 |
| **ftp_put[_ex]** | 将文件放到 FTP 服务器上 |
| **ftp_rendir[_ex]** | 重命名 FTP 服务器计算机上的目录 |
| **ftp_rmdir[_ex]** | 删除 FTP 服务器计算机上的目录 |

对于 **ftp_get[_ex]**、**ftp_put[_ex]** 和 **ftp_dir[_ex]** 函数，可以设置用于精确模拟 FTP 会话的属性：

**PATH**：要上传到 FTP 服务器上的文件。（仅用在未指定 **MSOURCE_PATH** 时）

**MPATH**：指定要上传到 FTP 服务器上的多个文件。（不是 **ftp_dir**）

**TARGET_PATH** （可选）：用于将文件放到服务器计算机上的路径和文件名。（仅限 **ftp_put**）

**LOCAL_PATH** （可选）：用于将文件放到本地计算机上的路径和文件名。（仅限 **ftp_get**）

**MODE** （可选）：检索模式 ASCII 或 BINARY （默认）。

**PASSIVE** （可选）：将与服务器的通信设置为 PASSIVE 传输模式。

有关这些函数的详细语法信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

在下例中， **ftp_delete** 函数将 *test.txt* 文件从 FTP 服务器上删除。

```
Actions()
{
        ftp_logon("FTP",
                "URL=ftp://user:pwd@ftp.merc-int.com",
                "LocalAddr=ca_server:21",
                LAST);

        ftp_delete("Ftp_Delete",
            "PATH=/pub/for_jon/test.txt", ENDITEM,
            LAST);

        ftp_logout();
    return 1;
}
```

# 31

# 开发 LDAP Vuser 脚本

通过 VuGen 可以模拟与 LDAP 服务器之间的通信。

本章描述：

- 使用 LDAP 函数
- 定义可分辨名称项

***下列信息仅适用于 LDAP Vuser 脚本。***

## 关于开发 LDAP Vuser 脚本

LDAP（轻型目录访问协议）是用于访问目录列表的协议。 LDAP 目录由很多 LDAP 项组成。每个 LDAP 项都是具有可分辨名称 (DN) 的属性的集合。有关 DN 的详细信息，请参阅定义可分辨名称项部分。

LDAP 目录项以层次结构的形式排列，该结构能够反映政治、地理和 / 或组织边界。代表国家 / 地区的项显示在树的顶部。这些项下面是代表州或国家组织的项。它们之下可能是代表人员、组织单位、打印机、文档或其他任何内容的项。

VuGen 将录制在 LDAP 服务器上的通信。它将使用模拟您操作的函数创建一个脚本。此过程包括登录和注销服务器、添加和删除项以及查询项。

要为 LDAP 协议创建脚本，请在 “电子商务” 类别中选择 LDAP 协议类型。要开始进行录制，请选择 “Vuser” > “开始录制”，然后对 LDAP 服务器执行典型操作。有关录制过程的详细信息，请参阅第 3 章 “使用 VuGen 进行录制”。

创建虚拟用户脚本之后，可以将其集成到 Windows 或 UNIX 平台上的场景中。有关在场景中集成虚拟用户脚本的详细信息，请参阅 《LoadRunner Controller 用户指南》。

# 使用 LDAP 函数

您可以指明用于创建 Vuser 脚本的编程语言。详细信息，请参阅第 4 章 “设置脚本生成首选项”。以下部分描述为 C 语言类型虚拟用户脚本生成的函数。

LDAP Vuser 脚本函数可以模拟 LDAP 协议。每个 LDAP 函数都以 **mldap** 前缀开头。

所有 LDAP 函数都是成对出现的 （一个用于全局会话，另一个可指明特定的会话）。要将操作应用于所有会话，请使用没有 **ex** 后缀的版本。要将操作应用于特定的会话，请使用带有以 **ex** 为后缀的会话标识符的版本。例如，使用 **mldap_logon** 全局登录到 LDAP 服务器，而使用 **mldap_logon_ex** 登录到特定会话的 LDAP 服务器。

| 函数名 | 描述 |
|---|---|
| **mldap_add** | 向 LDAP 目录添加项 |
| **mldap_add_ex** | 向特定会话的 LDAP 目录添加项 |
| **mldap_delete** | 删除项或属性 |
| **mldap_delete_ex** | 删除特定会话的项或属性 |
| **mldap_get_attrib_name** | 获取属性名 |
| **mldap_get_attrib_name_ex** | 获取特定会话的属性名 |
| **mldap_get_attrib_value** | 获取当前项的属性值 |
| **mldap_get_attrib_value_ex** | 获取特定会话当前项的属性值 |
| **mldap_get_next_entry** | 显示下一个搜索结果 |
| **mldap_get_next_entry_ex** | 显示指定会话的下一个搜索结果 |
| **mldap_logon** | 执行登录到 LDAP 服务器的操作 |
| **mldap_logon_ex** | 执行登录到特定会话的 LDAP 服务器的操作 |
| **mldap_logoff** | 执行从 LDAP 服务器注销的操作 |
| **mldap_logoff_ex** | 执行从特定会话的 LDAP 服务器注销的操作 |
| **mldap_modify** | 修改项的属性值 |
| **mldap_modify_ex** | 修改特定会话的项的属性值 |
| **mldap_rename** | 重命名项 |

| 函数名 | 描述 |
| --- | --- |
| **mldap_rename_ex** | 重命名特定会话的项 |
| **mldap_search** | 在 LDAP 服务器上执行搜索 |
| **mldap_search_ex** | 在特定会话的 LDAP 服务器上执行搜索 |

有关这些函数的详细语法信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

在下例中，用户将登录到 LDAP 服务器 ldap1。首先将添加一个项，然后将 OU 属性从 Sales 重命名为 Marketing。

```
Action()
{
        // 登录到 LDAP 服务器
        mldap_logon("Login",
                "URL=ldap://johnsmith:tiger@ldap1:80",
                LAST);

        // 添加 Sally R. Jones 的项
        mldap_add("LDAP Add",
                "DN=cn=Sally R. Jones,OU=Sales, DC=com",
                "Name=givenName", "Value=Sally", ENDITEM,
                "Name=initials", "Value=R", ENDITEM,
                "Name=sn", "Value=Jones", ENDITEM,
                "Name=objectClass", "Value=contact", ENDITEM,
                LAST);

        // 将 Sally 的 OU 重命名为 Marketing
        mldap_rename("LDAP Rename",
                  "DN=CN=Sally R. Jones,OU=Sales,DC=com",
                  "NewDN=OU=Marketing",
                  LAST);

        // 从 LDAP 服务器注销
        mldap_logoff();
    return 0;
}
```

# 定义可分辨名称项

LDAP API 通过其可分辨名称 (DN) 引用对象。 DN 是相对可分辨名称 (RDN) 的序列，用逗号分隔。

一个 RDN 就是一个带有关联值的属性，采用**属性 = 值**的形式。属性名不区分大小写。下表列出了最常见的 RDN 属性类型。

| 字符串 | 属性类型 |
|---|---|
| **DC** | domainComponent |
| **CN** | commonName |
| **OU** | organizationalUnitName |
| **O** | organizationName |
| **STREET** | streetAddress |
| **L** | localityName |
| **ST** | stateOrProvinceName |
| **C** | countryName |
| **UID** | userid |

以下是可分辨名称的示例。

```
DN=CN=John Smith,OU=Accounting,DC=Fabrikam,DC=COM
DN=CN=Tracy White,CN=admin,DC=corp,DC=Fabrikam,DC=COM
```

下表列出了不能在属性值中使用的保留字符。

| 字符 | 描述 |
| --- | --- |
| | 字符串开始处的空格或 # 字符 |
| | 字符串末尾的空格字符 |
| , | 逗号 |
| + | 加号 |
| " | 双引号 |
| \ | 反斜杠 |
| < | 左尖括号 |
| > | 右尖括号 |
| ; | 分号 |

要将保留字符作为属性值的一部分，它的前面必须带有作为转义符的反斜杠 (\)。如果属性值包含其他保留字符，如等号 (=) 或非 UTF-8 字符，则必须以十六进制格式对其进行编码，即反斜杠后接两个十六进制数字。

下列是包含转义符的 DN 的示例。第一个示例是带有嵌入逗号的组织单位名称；第二个示例是包含回车的值。

```
DN=CN=Bitwise,OU=Docs\, Support,DC=Fabrikam,DC=COM
DN=CN=Before\0DAfter,OU=Test,DC=North America,DC=Fabrikam,DC=COM
```

# 32

# 创建 Web Vuser 脚本

可以使用 VuGen 来开发 Web Vuser 脚本。 VuGen 通过录制您使用客户端浏览器时执行的操作来创建 Vuser 脚本。

本章描述：

- 介绍 Web Vuser
- 了解 Web Vuser 技术
- Web Vuser 脚本入门
- 录制 Web 会话
- 将 Web Vuser 脚本转换为 Java

***以下信息适用于 Web (HTML/HTTP)、 SOAP 和 PeopleSoft8 Vuser 脚本。***

## 关于开发 Web Vuser 脚本

可以使用 VuGen 来开发 Web Vuser 脚本。浏览某个站点并执行典型的用户操作时， VuGen 会录制您的操作并生成 Vuser 脚本。运行该脚本时，生成的 Vuser 将模拟一个访问 Internet 的用户。 PeopleSoft8 协议与 Web 相同，另外还具有支持 UTF-8 字符编码的功能。

创建 Vuser 脚本后，可使用 VuGen 以独立模式运行该脚本。如果执行成功，便可以将此 Vuser 脚本集成到场景中。有关如何将 Vuser 脚本集成到场景中的详细信息，请参阅 《LoadRunner Controller 用户指南》。

# 介绍 Web Vuser

假设您有一个网站，其中显示您公司的产品信息。潜在客户会访问该网站。您希望确保任何客户查询的响应时间小于某个指定值（例如， 20 秒），即使在大量用户 （例如 200 个）同时访问该网站时也是如此。可以使用 Vuser 来模拟这一情况，其中由 Web 服务器负责处理并发的信息请求。每个 Vuser 可以：

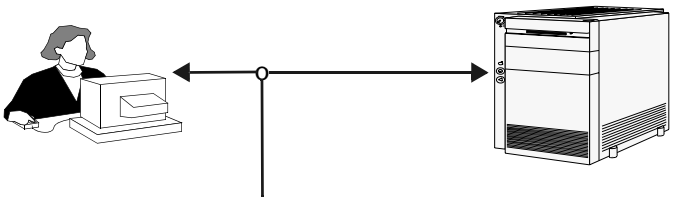
- 加载主页
- 导航到包含产品信息的页面
- 提交查询
- 等待服务器的响应

可以将数百个 Vuser 分布在可用的测试计算机上，每个 Vuser 都使用服务器的 API 来访问服务器。这样，您便可以度量服务器在许多用户产生的负载下的性能。

包含对服务器 API 的调用的程序称为 Vuser 脚本。该脚本模拟浏览器应用程序以及浏览器执行的所有操作。使用 Controller 可以将该脚本分配给多个 Vuser。Vuser 执行该脚本并在 Web 服务器上模拟用户负载。

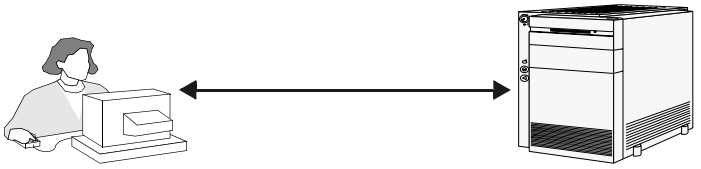
# 了解 Web Vuser 技术

VuGen 通过录制浏览器和 Web 服务器之间的活动来创建 Web Vuser 脚本。VuGen 监控系统的客户端 （浏览器），并跟踪所有发送到服务器以及从服务器接收的请求。



*客户端运行浏览器应用程序。* *VuGen 录制脚本。* *服务器接收和发送请求。*

在 VuGen 中或从 LoadRunner Controller 运行录制的 Vuser 脚本时， Vuser 将与服务器直接通信，无需依赖客户端软件。而 Vuser 脚本则通过 API 函数直接执行对 Web 服务器的调用。



*Web 虚拟用户执行 API 调用。* *服务器接收和发送请求。*

# Web Vuser 脚本入门

本部分概述了开发 Web Vuser 脚本的过程。

**要开发 Web Vuser 脚本，请执行下列操作：**

**1 使用 VuGen 新建一个脚本。**

选择 “文件” > “新建” 或单击 “新建” 按钮，从电子商务类别中新建一个 Web (HTTP/HTML) 脚本 （使用单协议模式或多协议模式）。

有关新建脚本的详细信息，请参阅第 3 章 “使用 VuGen 进行录制”。

**2 设置录制选项。**

设置录制选项。有关设置常用的 Internet 录制选项的信息，请参阅第 35 章 “为 Internet 协议设置录制选项” 有关特定于 Web 的录制选项的信息，请参阅第 36 章 “为 Web Vuser 设置录制选项”。

**3 录制浏览器会话。**

录制您在浏览网站时执行的操作。

有关新建脚本的详细信息，请参阅第 3 章 “使用 VuGen 进行录制”。

**4 增强录制的 Vuser 脚本。**

通过插入事务、集合点、检查和服务步骤来增强 Vuser 脚本。

详细信息，请参阅第 39 章 “在负载下验证网页”、第 40 章 “修改 Web 和无线 Vuser 脚本” 和第 41 章 “配置 Web Vuser 脚本的关联规则”。

**5 定义参数 （可选）。**

为录制到脚本中的固定值定义参数。通过用参数替换固定值，您可以用不同的值多次重复执行相同的 Vuser 操作。

详细信息，请参阅第 7 章 “定义参数”。

**6 配置运行时设置。**

运行时设置可控制脚本执行过程中 Vuser 的行为。这些设置包括常规运行时设置（循环、日志、思考时间和常规信息）和与 Web 相关的设置 （代理服务器、网络和 HTTP 详细信息）。

详细信息，请参阅第 9 章 “配置运行时设置”。

**7 执行关联。**

扫描 Vuser 脚本以查找关联，并使用 VuGen 的某个机制来实现关联。详细信息，请参阅第 41 章 “配置 Web Vuser 脚本的关联规则”或第 42 章 “录制后关联 Vuser 脚本”。

**8 使用 VuGen 运行并调试 Vuser 脚本。**

从 VuGen 运行 Vuser 脚本以验证该脚本是否能够正确运行。详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”和第 44 章 “使用报告调试 Vuser 脚本”。

创建 Vuser 脚本后，请将其集成到 LoadRunner 场景中。有关将 Vuser 脚本集成到场景中的详细信息，请参阅 《LoadRunner Controller 用户指南》。

## 录制 Web 会话

录制 Web 会话时，VuGen 会监控您在 Web 浏览器中执行的所有操作。您的活动可以包括超链接跳转 （包括超文本和超图形）以及表单提交。录制时， VuGen 会将录制的操作保存在 Web Vuser 脚本中。

您创建的每个 Vuser 脚本至少包含三个部分：*vuser_init*、一个或多个 *Actions* 以及 *vuser_end*。在录制过程中，可以选择让 VuGen 将录制的函数插入到脚本中的哪个部分。 *vuser_init* 和 *vuser_end* 部分通常用于录制服务器登录和注销过程，当运行具有多个循环的 Vuser 脚本时，这些过程不会重复执行。因此，应该将 Web 会话录制到 Actions 部分中，以便为每个循环重复执行完整的浏览器会话。

在录制过程中， VuGen 会将函数以及函数的资源放在并发组中。并发组表示同时加载到页面中的链接和资源。例如，浏览器经常会在加载第一个图像的同时，开始加载第二个或第三个图像。并发组中的语句以 **web_concurrent_start** 函数开始，并以 **web_concurrent_end** 函数结束。在回放过程中，如果 LoadRunner 遇到 **web_concurrent_start** 语句，它将注册这些函数，但直到该组以 **web_concurrent_end** 结束时才会执行这些函数。有关并发组函数的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

# 将 Web Vuser 脚本转换为 Java

VuGen 提供了一个实用程序，可用来将为 Web Vuser 创建的脚本转换为用于 Java Vuser 的脚本。这样，您还可以创建可同时用于 Web 和 Java 的混合 Vuser 脚本。

**要将 Web Vuser 脚本转换为 Java Vuser 脚本，请执行下列操作：**

1. 创建一个空的 Java Vuser 脚本并保存。
2. 创建一个空的 Web Vuser 脚本并保存。
3. 使用标准的 HTML/HTTP 录制功能录制 Web 会话。
4. 回放该 Web Vuser 脚本。如果该脚本能够正确回放，请将整个脚本剪切并粘贴到文本文档中，然后将它保存为 *.txt* 文本文件。在该文本文件中，将所有 Web 类型的参数大括号 “{}” 修改为 Java 类型的 “< >”。
5. 打开 DOS 命令窗口，并进入 <LOADRUNNER>/dat 目录。
6. 键入以下命令：

   <LOADRUNNER>\bin\sed -f web_to_java.sed *filename* > *outputfilename*

   其中 *filename* 是您以前保存的文本文件的完整路径和文件名， *outputfilename* 是输出文件的完整路径和文件名。
7. 打开输出文件，将其内容复制到 Java Vuser 脚本的操作部分中的所需位置。如果要将这些内容复制到空的自定义 Java 模板 （Java Vuser 类型）中，请按如下所示修改包含 public int action() 的那一行：

   public int action() throws Throwable

   对于录制的 Java 用户 （RMI 和 Corba），会自动完成此更改。

   像处理普通的 Java 脚本那样参数化和关联 Vuser 脚本，然后运行该脚本。

# 33

# 使用 Web Vuser 函数

使用 VuGen 可开发 Web Vuser 脚本。当您操作客户端浏览器时， VuGen 通过录制来创建 Vuser 脚本。

本章描述：

- 添加和编辑函数
- Web 函数列表
- 在树视图中查看脚本
- 在脚本视图中查看 Vuser 脚本

***下列信息适用于 Web (HTML/HTTP)、 SOAP、无线和 PeopleSoft8 Vuser 脚本。***

## 关于 Web Vuser 函数

所开发的用来模拟浏览器或工具包与 Web 服务器之间的 Internet 通信的函数称为 *Web Vuser 函数*。每个 Web Vuser 函数都有一个 **web** 前缀。一些函数在录制脚本时生成；其他函数必须手动插入到脚本中。

有关 Internet 协议函数 （Web、无线等）的详细信息和示例，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

VuGen 可以使用两种方式来显示 Web 或无线 Vuser 脚本：

- 作为 Vuser 脚本的基于图标的表示形式。这是默认的视图，并且称作 “树视图”（不可用于 WAP Vuser）。详细信息，请参阅第 383 页的 “在树视图中查看脚本”。
- 作为 Vuser 脚本的基于文本的表示形式。这称作 “脚本视图”。详细信息，请参阅第 385 页的 “在脚本视图中查看 Vuser 脚本”。

# 添加和编辑函数

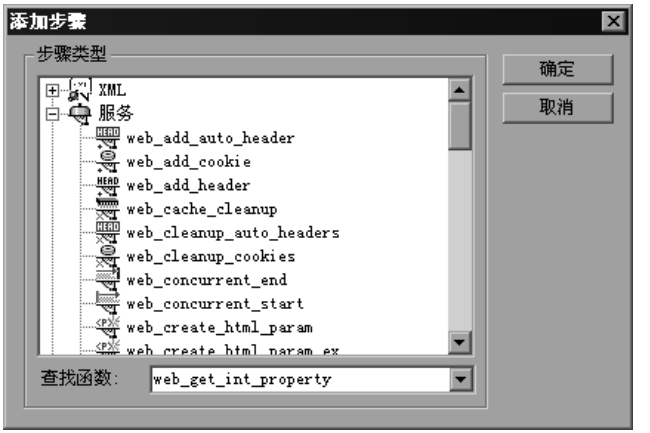
许多 Web Vuser 函数是在浏览器或工具包会话期间录制的。

在录制期间，可以手动添加常规 Vuser 函数，如事务、集合、注释和日志函数。详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

本节描述在 “树视图”和 “脚本视图”中，如何在录制期间和录制之后添加和编辑 Web Vuser 函数。

**要向现有的 Vuser 脚本中添加新函数，请执行下列操作：**

**1** 选择 “插入” > “新建步骤”。将打开 “添加步骤”对话框。




**2** 选择需要的函数，然后单击 “确定”。大部分 Web Vuser 函数位于 “服务”类别下。将打开该函数的 “属性”对话框。通过每个函数的 “属性”对话框可指定函数的参数。

3 指定属性并单击 “确定”。 Vugen 将在光标位置插入函数及其参数。

通过打开 “属性”对话框并修改参数值，可以编辑现有步骤。这仅对于支持树视图的协议有效 （不可用于 WAP）。

**要编辑现有步骤，请执行下列操作：**

1 在树视图中，从右键单击菜单中选择 “属性”。将打开该函数的 “属性”对话框。

2 根据需要修改参数值，然后单击 “确定”。

## Web 函数列表

那些表示在 Internet 上进行的通信的 Web Vuser 函数以 **web** 前缀开头。 Web 函数的分类如下：

- 操作函数
- **身份验证函数**
- **缓存函数**
- 检查函数
- **连接定义函数**
- 并发组函数
- Cookie 函数
- 关联函数
- 筛选器函数
- **标头函数**
- **代理服务器函数**
- 其他函数

### 操作函数

在录制 Web Vuser 脚本时， VuGen 将生成下列操作函数，并且将它们插入到脚本中：

| | |
|---|---|
| **web_custom_request** | 允许您使用 HTTP 支持的任何方法来创建自定义 HTTP 请求 |
| **web_image** | 在定义的图像上模拟鼠标单击 |
| **web_link** | 在定义的文本链接上模拟鼠标单击 |
| **web_submit_data** | 执行 “无条件”或 “无上下文”的表单提交 |
| **web_submit_form** | 模拟表单的提交 |
| **web_url** | 加载由 “URL”属性指定的 URL |

### 身份验证函数

| | |
|---|---|
| **web_set_certificate** | 使 Vuser 使用在 Internet Explorer 注册表中列出的特定证书 |
| **web_set_certificate_ex** | 指定证书和密钥文件的位置和格式信息 |
| **web_set_user** | 指定 Web 服务器的登录字符串和密码，用于 Web 服务器上已验证用户身份的区域 |

### 缓存函数

| | |
|---|---|
| **web_cache_cleanup** | 清除缓存模拟程序的内容 |

### 检查函数

| | |
|---|---|
| **web_find** | 在 HTML 页内搜索指定的文本字符串 |
| **web_global_verification** | 在所有后面的 HTTP 请求中搜索文本字符串 |
| **web_image_check** | 验证指定的图像是否存在于 HTML 页内 |
| **web_reg_find** | 在后面的 HTTP 请求中注册对 HTML 源或原始缓冲区中文本字符串的搜索 |

### 连接定义函数

| | |
|---|---|
| **web_disable_keep_alive** | 禁用 Keep-Alive HTTP 连接 |
| **web_enable_keep_alive** | 启用 Keep-Alive HTTP 连接 |
| **web_set_connections_limit** | 设置 Vuser 在运行脚本时可以同时打开连接的最大数目 |

### 并发组函数

| | |
|---|---|
| **web_concurrent_end** | 标记并发组的结束 |
| **web_concurrent_start** | 标记并发组的开始 |

## Cookie 函数

| | |
|---|---|
| **web_add_cookie** | 添加新的 Cookie 或修改现有的 Cookie |
| **web_cleanup_cookies** | 删除当前由 Vuser 存储的所有 Cookie |
| **web_remove_cookie** | 删除指定的 Cookie |

## 关联函数

| | |
|---|---|
| **web_create_html_param** | 将 HTML 页上的动态信息保存到参数中。（LR 6.5 及更低版本） |
| **web_create_html_param_ex** | 基于包含在 HTML 页内的动态信息创建参数 （使用嵌入边界）。（LR 6.5 及更低版本） |
| **web_reg_save_param** | 基于包含在 HTML 页内的动态信息创建参数 （不使用嵌入边界） |
| **web_set_max_html_param_len** | 设置已检索的动态 HTML 信息的最大长度 |

## 筛选器函数

| | |
|---|---|
| **web_add_filter** | 设置在下载时包括或排除 URL 的条件 |
| **web_add_auto_filter** | 设置在下载时包括或排除 URL 的条件 |
| **web_remove_auto_filter** | 禁用对下载内容的筛选 |

## 标头函数

| | |
|---|---|
| **web_add_auto_header** | 向所有后面的 HTTP 请求中添加自定义标头 |
| **web_add_header** | 向下一个 HTTP 请求中添加自定义标头 |
| **web_cleanup_auto_headers** | 停止向后面的 HTTP 请求中添加自定义标头 |
| **web_remove_auto_header** | 停止向后面的 HTTP 请求中添加特定的标头 |
| **web_revert_auto_header** | 停止向后面的 HTTP 请求中添加特定的标头，但是生成隐性标头 |
| **web_save_header** | 将请求和响应标头保存到变量中 |

## 代理服务器函数

| | |
|---|---|
| **web_set_proxy** | 指定将所有后面的 HTTP 请求定向到指定的代理服务器 |
| **web_set_proxy_bypass** | 指定 Vuser 直接访问 （即不通过指定的代理服务器访问）的服务器列表 |
| **web_set_proxy_bypass_local** | 指定 Vuser 对于本地 (Intranet) 地址是否应该避开代理服务器 |
| **web_set_secure_proxy** | 指定将所有后面的 HTTP 请求定向到服务器 |

## 回放函数

| | |
|---|---|
| **web_set_max_retries** | 设置操作步骤的最大重试次数 |
| **web_set_timeout** | 指定 Vuser 等待执行指定任务的最长时间 |

### 其他函数

| | |
|---|---|
| **web_convert_param** | 将 HTML 参数转换成 URL 或纯文本 |
| **web_get_int_property** | 返回有关上一个 HTTP 请求的特定信息 |
| **web_report_data_point** | 指定数据点并将其添加到测试结果中 |
| **web_set_option** | 在非 HTML 资源的编码、重定向和下载区域中设置 Web 选项 |
| **web_set_sockets_option** | 设置套接字的选项 |

### 控制类型函数

除了 Web Vuser 函数以外，下列控制函数也可能会出现在 Vuser 脚本中：

| | |
|---|---|
| **lr_start_transaction** | 为性能分析标记事务的开始 |
| **lr_end_transaction** | 为性能分析标记事务的结束 |
| **lr_rendezvous** | 在 Vuser 脚本中设置集合点 |
| **lr_think_time** | 暂停 Vuser 脚本中命令之间的执行 |

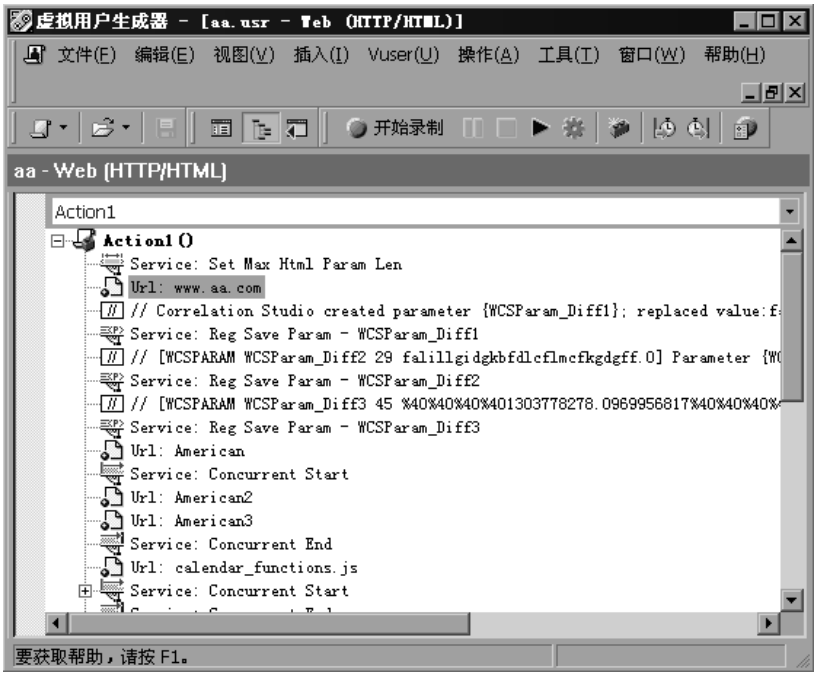
有关向脚本中添加常规 Vuser 函数的详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

## 在树视图中查看脚本

树视图以基于图标的视图显示 Vuser 脚本。每个 Vuser 函数都由图标表示。该视图不可用于 WAP Vuser。

**要显示示 Web Vuser 脚本的树视图，请执行下列操作：**

➤ 从 VuGen 主菜单中，选择 “视图” > “树视图”，或者单击 “按树查看脚本”按钮。 Vuser 脚本将以基于图标的树视图显示。如果已经处于树视图中，该菜单项将被禁用。




Vuser 脚本的树视图由表示 Vuser 的操作或 Vuser 脚本中的步骤的图标组成。步骤类型由步骤名前面的字符串指明。例如：

VuGen 中支持下列步骤类型：

| 图标类型 | 描述 |
|---|---|
| 服务 | “服务步骤”是一个函数，它不会在 Web 应用程序上下文中进行任何更改。更确切地说，服务步骤执行自定义任务 （如设置代理服务器）、提供授权信息以及发出自定义的标头 |
| URL | 在键入 URL 或者使用书签访问特定网页时，“URL”图标将被添加到 Vuser 脚本中。每个 URL 图标代表 Vuser 脚本中的一个 **web_url** 函数。 URL 图标的默认标签是目标页 URL 的最后一部分 |
| 链接 | 在录制期间单击超文本链接时， VuGen 将添加一个**链接**图标。每个“链接”图标代表 Vuser 脚本中的一个 **web_link** 函数。该图标的默认标签是超文本链接的文本字符串 |
| 图像 | 在录制期间单击超图形链接时， VuGen 将向 Vuser 脚本中添加一个“图像”图标。每个“图像”图标代表 Vuser 脚本中的一个 **web_image** 函数。如果 HTML 代码中的图像具有 ALT 属性，则该属性将用作图标的默认标签。如果 HTML 代码中的图像不具有 ALT 属性，那么 SRC 属性的最后一部分将用作图标的标签 |
| 提交表单 / 提交数据 | 在录制期间提交表单时， VuGen 会添加“提交表单”或“提交数据”步骤。该步骤的的默认标签是用于处理表单的可执行程序的名称 |
| 自定义请求 | 在录制 VuGen 无法识别为任何标准操作 （如 URL、链接、图像或表单提交）的操作时， VuGen 将向 Vuser 脚本中添加“自定义请求”。这适用于非标准 HTTP 应用程序 |

---

**注意：**仅在选择要以基于 HTML 的模式录制的选项时，才会录制链接、图像和表单提交步骤。详细信息，请参阅第 36 章 “为 Web Vuser 设置录制选项”。

---

# 在脚本视图中查看 Vuser 脚本

要查看和编辑 Web Vuser 脚本的基于文本的表示形式，请选择脚本视图。

**要显示 Web Vuser 脚本的脚本视图，请执行下列操作：**

从 VuGen 主菜单中，选择 “视图” > “脚本视图”，或者单击 “按文本查看脚本”图标。 Vuser 脚本将以基于文本的脚本视图显示。如果已经处于脚本视图中，该菜单项将被禁用。

在脚本视图中，可以查看由浏览器应用程序生成的函数，并且可以根据需要对脚本进行更改。

---

**注意：**如果在脚本视图中对 Vuser 脚本进行更改， VuGen 将在 Vuser 脚本的树视图中进行相应的更改。如果 VuGen 无法理解所做的基于文本的更改， VuGen 将无法将脚本视图转换成树视图。

---

# 34

# 录制 Web/WinSock 和 SOAP Vuser 脚本

通过 VuGen 可以创建 Web/WinSock 双协议 Vuser 脚本，使其模拟访问 Web 和 Windows 套接字的应用程序。该协议普遍用于 Palm HotSync 进程。

本章描述：

- Web/WinSock Vuser 脚本入门
- 设置浏览器和代理服务器录制选项
- 设置 Web 捕获录制选项
- 录制 Web/WinSock 会话
- 录制 Palm 应用程序

***下列信息仅适用于 Web/Windows 套接字双协议、SOAP 和 Palm Vuser 脚本。***

# 关于录制 Web/WinSock Vuser 脚本

通过 VuGen 的 Web/WinSock 双协议类型可以成功地录制非 HTML Web 应用程序。 VuGen 同时使用 Web 和 Windows 套接字协议函数录制这些应用程序，并创建能够模拟对网页和套接字活动进行的访问的脚本。该协议的常见应用是对使用 Palm OS 协议的手持式管理器的 HotSync 进程进行录制。VuGen 将录制数据的传输并生成相关的函数。注意，不会录制 Palm 的无线数据传输。

当您运行双协议脚本时， Vuser 将模拟 Web 浏览器、非 HTML 应用程序和 Web 服务器之间的活动。通过双协议功能可以针对 Web 协议和 WinSock 协议仅录制一次，这样可以避免任何重复调用。 VuGen 将两个协议的录制进行同步，并且创建一个同时包含 Web 和 WinSock Vuser 函数的脚本。

最好使用多协议脚本录制 Web 和 WinSocket 会话，同时指定 Web 和 WinSock 协议。但是，多协议模式不支持 UDP 套接字，所以如果需要录制 UDP 套接字，请使用本章中讨论的 “双 Web/WinSock Vuser”。

WinSock 函数以低级代码形式表示录制会话期间的套接字活动。每个 WinSock 函数都以 **lrs** 前缀开头，并且与套接字、数据缓冲区和环境有关。通过在 VuGen 的左窗格中选择 *data.ws*，还可以查看会话期间发送和接收的实际数据。注意，在该模式中不支持 UDP 类型套接字的录制。

Web 函数以 **web** 前缀开头。这些函数与标准 Web 操作有关，如转到 URL (**web_url**)、提交数据 (**web_submit_data**) 和添加 Cookie (**web_add_cookie**)。

有关 WinSock 函数和 Web 函数的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

录制双协议脚本后，可以通过在脚本视图中修改脚本的文本来编辑它。注意，Web/WinSock 脚本不支持树视图和 “快照”窗口 （对于标准 Web Vuser 脚本可用）。

对 Web/WinSock Vuser 脚本中的值进行关联的方式与在单一协议脚本中一样。但是，必须根据 Web 关联过程来关联 Web 函数，并且必须根据其过程来关联 WinSock 函数。有关关联 Web 函数的信息，请参阅第 41 章 “配置 Web Vuser 脚本的关联规则”。有关关联 WinSock 函数的详细信息，请参阅第 8 章 “关联语句”。

# Web/WinSock Vuser 脚本入门

本部分概述使用 VuGen 开发双协议 Web/WinSock Vuser 脚本的过程。

**要开发 Web/WinSock Vuser 脚本，请执行下列操作：**

**1 使用 VuGen 录制基本脚本。**

调用 VuGen 并新建 Vuser 脚本。指定 Web/Winsocket 双协议作为 Vuser 的类型。选择要录制的应用程序并设置 Web 和 WinSock 录制选项。录制应用程序的典型操作。

详细信息，请参阅第 390 页的 “设置浏览器和代理服务器录制选项”。

**2 增强脚本。**

通过将事务、集合点和控制流结构插入到脚本中，增强 Vuser 脚本。

详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

**3 定义参数 （可选）。**

为录制到脚本中的固定值定义参数。通过用参数替换固定值，可以使用不同的值多次重复执行相同的业务流程。

详细信息，请参阅第 7 章 “定义参数”。

**4 关联语句 （可选）。**

通过关联语句，可以在后面的语句中使用某个业务流程的结果。

详细信息，请参阅第 8 章 “关联语句”或第 41 章 “配置 Web Vuser 脚本的关联规则”。

**5 配置运行时设置。**

运行时设置用于控制脚本执行期间 Vuser 的行为。这些设置包括循环、日志和计时信息。

详细信息，请参阅第 9 章 “配置运行时设置”。

**6 从 VuGen 运行脚本。**

保存然后从 VuGen 运行该脚本，以验证其是否运行正常。

详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

创建 Vuser 脚本后，请将其集成到场景中。详细信息，请参阅 《LoadRunner Controller 用户指南》。

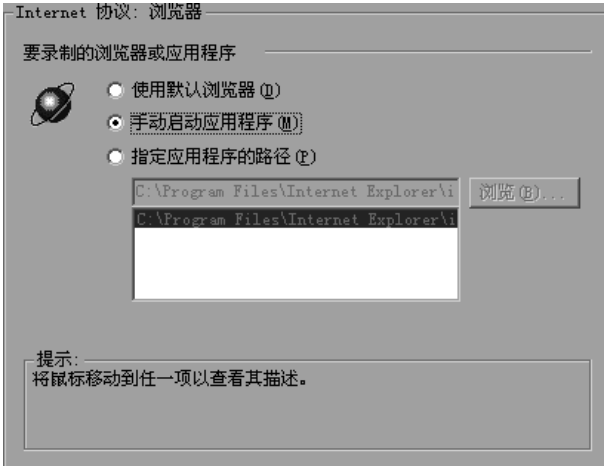
# 设置浏览器和代理服务器录制选项

在录制脚本之前，设置 Web 和 WinSock 录制选项。可以在下列区域中设置 Web 录制选项：浏览器、代理服务器、录制信息和关联。设置 WinSock 录制选项可以排除套接字、设置思考时间阈值和指定转换表。本部分描述浏览器和代理服务器录制选项。有关其他 Internet 协议录制选项的信息，请参阅第 36 章“为 Web Vuser 设置录制选项”。有关 WinSock 录制选项的信息，请参阅第 21 章“开发 WinSock Vuser 脚本”。

要打开录制选项，请选择“工具”>“录制选项”，或者单击“开始录制”对话框中的“选项”按钮。

## 设置浏览器录制选项

通过浏览器录制选项可以指定 VuGen 在录制 Vuser 脚本时使用哪一个浏览器。




在“浏览器”选项卡上，选择下列三个选项之一。注意，这些选项仅在禁用 Web 捕获时才相关（请参阅第 393 页的“设置 Web 捕获录制选项”）。如果启用 Web 捕获，将始终启动“要录制的程序”字段中的应用程序。

➤ **使用默认浏览器**：指示 VuGen 使用录制计算机上的默认 Web 浏览器。将忽略“开始录制”对话框的“要录制的程序”字段中的应用程序。但是，即使不使用也必须在该字段中输入一个值。使用该选项可录制 Active-X 应用程序或 Java 模板。

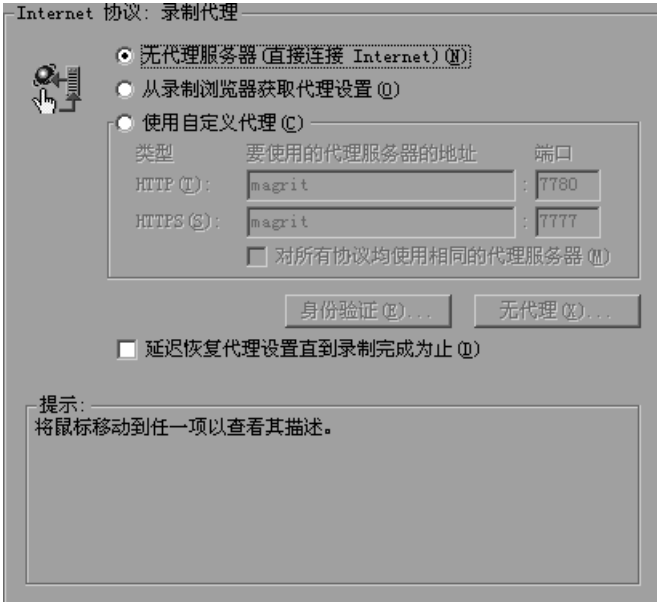
➤ **手动启动应用程序：**指示 VuGen 不要在开始录制时自动启动应用程序（在该情况下是指浏览器）。在“开始录制”对话框的“要录制的程序”字段中指定浏览器的路径。开始录制时， VuGen 将启动它，并且提示您修改代理服务器设置（请参阅第 391 页）。将该选项应用于独立的应用程序或调用浏览器的应用程序。

➤ **指定应用程序的路径：**指示 VuGen 自动启动特定的应用程序。从列表中选择应用程序及其路径，或者单击“浏览”按钮找到需要的应用程序。将忽略“开始录制”对话框的“要录制的程序”字段中的应用程序。但是，必须在该字段中输入一个值，即使不使用它。通过该选项可以使用非浏览器应用程序或默认浏览器之外的浏览器。

## 指定录制代理服务器设置

如果将录制选项设置为手动启动浏览器（请参阅上一部分），并且不启用 Web 捕获，您可能需要调整代理服务器设置。因为不是自动调用浏览器，所以无法指示 VuGen 从录制浏览器获取代理服务器设置。

相反，选择“始终使用到 Internet 的直接连接”选项。



开始录制之后， VuGen 将产生一条消息，指明应该更改浏览器的代理服务器设置以及更改哪些设置。

如果单击 “确定”而未修改浏览器的设置， VuGen 将仅录制应用程序而不是浏览器操作。要设置代理服务器设置，请中止录制并设置浏览器设置。

**修改代理服务器设置：**

➤ 对于 Netscape，请选择 “编辑” > “首选项” > “高级” > “代理服务器” > “手动代理服务器配置”，然后输入 *localhost*（小写）作为主机名，并输入上面对话框中提供的端口号。

➤ 对于 Internet Explorer，选择 “工具” > “Internet 选项” > “连接” > “局域网设置”，然后选择 “使用代理服务器”。输入 *localhost*（小写）作为主机名，并输入上面对话框中提供的端口号。

有关其他 Web 录制选项的信息，请参阅第 36 章 “为 Web Vuser 设置录制选项”。有关 WinSock 录制选项的信息，请参阅第 21 章 “开发 WinSock Vuser 脚本”。

# 设置 Web 捕获录制选项

当 VuGen 录制 Web/WinSock Vuser 的脚本时，它将修改浏览器的代理服务器设置。VuGen 将通过重新配置的代理服务器端口来定向所有 HTTP 和 HTTPS 请求。通过代理服务器端口定向 Web 请求之后， VuGen 会将它们定向到“录制代理服务器”选项卡中指定的端口。所有不是通过指定的代理服务器端口发送或接收的请求都将录制为 WinSock 函数而不是 HTTP Web 请求。录制之后，VuGen 将还原所有原始的代理服务器设置。

某些应用程序会产生 Web 事件，但不支持代理服务器配置，例如某些 Java 小程序。 VuGen 不能为这些应用程序设置所需的内部代理服务器设置。因此，这些应用程序不会录制为 Web 事件，并且事件将被录制为 WinSock 请求，使它们可读性更差并且更不直观。有关如何录制应用程序及其启动的信息，请参阅第 390 页的 “设置浏览器录制选项”。

通过 Web 捕获设置可以捕获或保存通常被录制为 WinSock 函数或 Web 函数的事件。在启用捕获选项时， VuGen 将在特定的端口等待事件，将它们标记为 Web 事件，然后生成适当的 Web 函数。这会产生更加易读和直观的脚本。

需要指定 VuGen 应该侦听 Web 事件的端口。该端口上的所有通信都将作为 Web 事件 （由 Web Vuser 函数表示）进行处理。可以使用默认端口 80 （对于 HTTP）和 443 （对于 HTTPS），或者可以指定任何 IP：端口组合。 VuGen 支持通配符组合，以便包括特定主机上的所有端口。

例如，207.232.15.30:* 表示主机 207.232.15.30 上的所有端口。207.232.*.*:80 项表示 207.232 域中所有计算机上的标准端口 80。注意，不能在 IP 地址部分内混合使用数字和通配符。例如， 207.2*.32.9 是无效项。

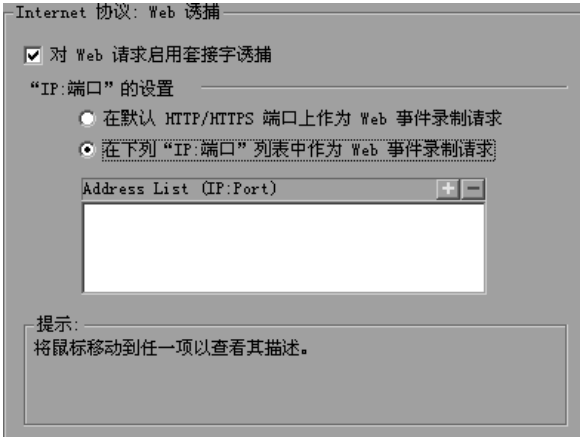
要确定是否启用 Web 捕获，请首先执行录制会话。查看数据文件 *data.ws*。如果看到被录制为 WinSock 缓冲区数据的 HTTP 或 HTTPS 数据，这可能表示请求是通过其他端口进行的。在该实例中，应该启用 Web 捕获允许 VuGen 为这些请求生成 Web 函数。

在手动启动要录制的应用程序 （而不是通过浏览器进行录制）时，该选项尤其有用。有关手动启动应用程序的信息，请参阅第 390 页的 “设置浏览器录制选项”。

注意，在启用 Web 捕获时，将忽略指定端口上的所有 Windows 套节字通信。

**要设置 Web 捕获录制选项，请执行下列操作：**

1 选择 “工具” > “录制选项”，然后选择 “Web 捕获” 节点。




2 要启用 Web 事件捕获，请选中 “启用 Web 请求的套接字捕获” 复选框。

3 要捕获默认端口上的 Web 事件，请选择 “将发往默认 HTTP/HTTPS 端口的请求作为 Web 事件录制”。

4 要捕获默认端口以外端口上的 Web 事件，请选择 “将发往下列 IP：端口列表的请求作为 Web 事件录制”。单击 “+” 号可以向该列表中添加新的 IP：端口项。单击 “-” 号可以删除现有项。可以如上一部分中所述使用通配符。

5 单击 “确定”，保存设置并关闭对话框。

## 录制 Web/WinSock 会话

录制双协议会话的方式与录制独立的 Web 和 Windows 套接字 Vuser 类似。在录制双协议会话时， VuGen 将监视您在 Web 浏览器或应用程序内执行的所有操作，并且生成适当的 Web 函数或 WinSocket 函数。

所创建的每个脚本包含三个部分：*vuser_init*、 *Actions* 和 *vuser_end*。在录制期间，可以选择 VuGen 将录制的函数插入到其中的脚本部分。 *vuser_init* 和 *vuser_end* 部分通常用于录制服务器登录和注销过程，它们在您运行具有多次循环的脚本时不会重复执行。因此，应该在 *Actions* 部分进行录制，以便对每次循环重复执行完整的浏览器会话。

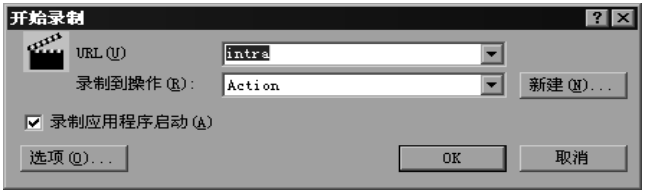
**要录制 Web/WinSock 会话，请执行下列操作：**

1 打开录制浏览器，然后将主页设置为要录制的 URL。

2 选择 “开始” > “程序” > “LoadRunner” > “虚拟用户生成器”。将打开 VuGen 主窗口。

3 新建 Web/WinSock 脚本。选择 “文件” > “新建”或单击 “新建”按钮。

4 从 “电子商务”文件夹中选择 “Web/Winsocket 双协议”，然后单击 “确定”。VuGen 将打开骨架 Vuser 脚本并显示 “开始录制”对话框。




5 单击 “选项”，设置套接字、浏览器、代理服务器或其他高级设置的录制选项。如果要使用浏览器进行录制，请指定浏览器。如果要录制非浏览器应用程序 （如流式数据），请将 “浏览器录制选项”设置为 “手动启动浏览器”。对于手动启动，请将代理服务器选项设置为 “始终使用到 Internet 代理服务器的直接连接”，然后将浏览器的代理服务器设置修改为 “localhost”。有关设置这些录制选项的其他信息，请参阅第 390 页的 “设置浏览器录制选项”。

6 单击 “浏览”，选择要录制的程序。注意，仅当在录制选项 （“浏览器”节点）中指定 “手动启动浏览器”时，才使用该项。在 “要录制的程序”框中指定非浏览器应用程序的路径和名称。如果要使用浏览器进行录制，将忽略该项；但是，您必须在该框中输入一个值。

7 从 “录制到操作”列表中，选择要在其中开始录制的部分。

8 单击 “确定”，启动应用程序并开始录制。将显示浮动录制工具栏。

**注意**：在录制 Web Vuser 脚本时，仅可以运行单个 Netscape Navigator 实例。因此，如果 Netscape Navigator 在开始录制之前已经运行， VuGen 将提示您关闭该浏览器。这样 VuGen 可以自己打开 Netscape 浏览器。

**9** 执行需要的业务流程。您单击的每个链接都将向脚本中添加一个 **web_url** 函数。您提交的每个表单都将向 Vuser 脚本中添加一个 **web_submit_form** 函数。非浏览器应用程序操作将被录制为套接字数据。

在录制期间，可以使用 VuGen 浮动工具栏插入事务、集合点和即时文本检查。详细信息，请参阅以下内容。有关插入文本检查或图像检查的详细信息，请参阅第 39 章 “在负载下验证网页”。

**10** 执行所有需要的用户进程之后，单击浮动录制工具栏上的 “停止录制”按钮。VuGen 将还原 VuGen 主窗口。

**11** 选择 “文件” > “保存”或者单击 “保存”按钮，保存 Vuser 脚本。在 “保存测试”对话框中指定文件名和位置，然后单击 “保存”。

录制之后，可以通过将事务、集合点和控制流结构插入到脚本中，编辑 Vuser 脚本。详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

修改脚本之后，可以使用重新生成 Vuser 实用程序恢复到原来录制的脚本版本。该实用程序仅重新生成 WinSock 语句；它不会影响 Web 语句。详细信息，请参阅第 39 页的 “重新生成 Vuser 脚本”。

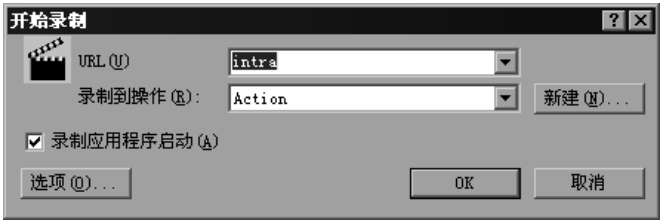
## 录制 Palm 应用程序

基于 Palm 的应用程序提供了两种与远程服务器进行通信的方式：Cradle 和无线。停靠在 Cradle 上的 Palm 应用程序通过 HotSync 服务，直接在 Internet 上与其服务器进行通信。通过 VuGen 可以捕获所有通过 Palm 的 HotSyncall 服务传送的流量。由于许多应用程序使用 HTTP 作为传输层与其服务器进行通信，所以生成的脚本都类似于 Web，并且继承了与 Web 相同的语法和功能。在极少数情况下，流量将通过专有协议传送。该专有流量也将被录制，并且在脚本中表示为 WinSock 函数。

要录制 Palm 应用程序，请执行下列操作：

**1** 新建脚本。选择 “文件” > “新建”，或者单击 “新建” 按钮。

**2** 从 “电子商务” 文件夹中选择 “Palm”，然后单击 “确定”。 VuGen 将打开骨架 Vuser 脚本并显示 “开始录制” 对话框。




**3** 指定 HotSync.Exe 作为要录制的应用程序，然后单击 “确定”。

确保在从 VuGen 启动 HotSync.Exe 之前，它尚未运行。

**4** 在 Cradle 上设置 Palm Pilot，并且与应用程序进行交互。

注意，可能需要按 Palm Pilot 上的 “HotSync” 按钮，才能启动 Palm 和服务器之间的通信。

**5** 执行所有需要的用户进程之后，请单击浮动录制工具栏上的 “停止录制” 按钮。 VuGen 将还原 VuGen 主窗口。

**6** 选择 “文件” > “保存” 或者单击 “保存” 按钮，保存 Vuser 脚本。在 “保存测试” 对话框中指定文件名和位置，然后单击 “保存”。

录制之后，可以通过将事务、集合点和控制流结构插入到脚本中，编辑 Vuser 脚本。详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

该脚本将用 Web 协议和 WinSock 协议的组合来表示。通过 HTTP 进行的所有 Palm 流量都将以 **web_url** 语句和 **web_submit_data** 请求来表示。专用协议将由对 WinSock 函数的调用来表示。

# 35

# 为 Internet 协议设置录制选项

对于在 Internet 上使用的协议，可以自定义与 Internet 相关的录制选项。

本章描述：

- 使用代理服务器设置
- 设置高级录制选项
- 设置录制场景

***下列信息仅适用于 Web、无线和 Oracle NCA 协议。***

## 关于为 Internet 协议设置录制选项

VuGen 可创建模拟真实 Internet 环境的 Vuser 脚本。

录制前，可以配置与代理服务器相关的 VuGen 录制选项以及脚本生成首选项。

还可以为 Web Vuser 脚本设置特定于某一协议的录制选项。详细信息，请参阅“录制选项”一章中与相应协议相关的部分。

注意，可以使用下列几种方式打开“录制选项”对话框：

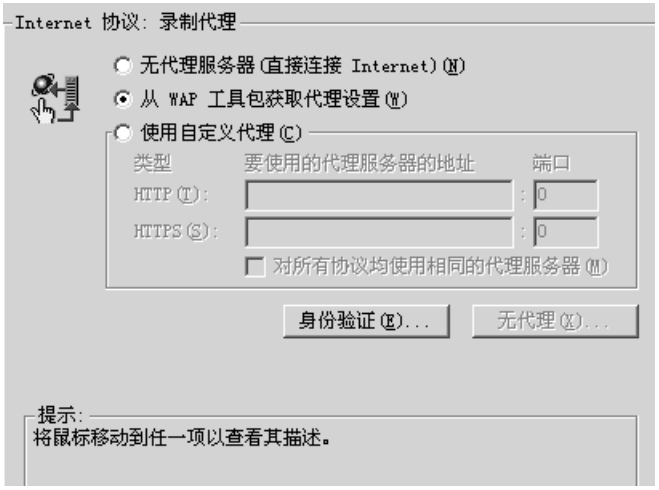
- 工具栏按钮：
- 键盘快捷键：Ctrl+F7
- “工具”菜单：选择“工具”>“录制选项”

# 使用代理服务器设置

代理服务器是指驻留在客户端（例如 Web 浏览器）和 Web 服务器之间的服务器。它将侦听所有发送到服务器的请求，并尝试实现这些请求。使用代理服务器的原因有两个：改善性能和筛选请求。为改善性能，它会将某个用户访问过的网页存储起来，以使其他用户无需再次访问服务器即可查看这些网页。管理员还可以使用代理服务器对浏览器中可以查看的内容进行筛选。

要使用代理服务器，请在浏览器的首选项中指定该代理服务器的名称或 IP 地址。通常， Internet 服务提供商建议其用户使用代理服务器进行连接，并且各个公司也要求其职员通过代理服务器访问 Internet。

默认情况下， VuGen 使用录制浏览器中的代理服务器设置。 VuGen 还可以自定义录制会话的代理服务器设置。如果事先知道用户不通过代理服务器而直接访问 Internet，或者用户使用特定的代理服务器而不是浏览器的默认设置，则可以对代理服务器设置进行自定义。要自定义设置，请在 “录制选项”树中选择 “Internet 协议：录制代理服务器”节点，然后修改录制代理服务器设置。



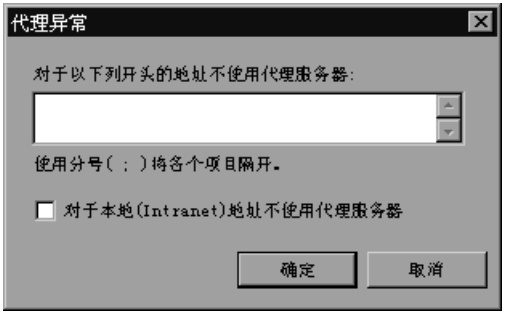
可以选择下列代理服务器选项之一：

- **无代理服务器 （直接连接 Internet）**：总是直接连接到 Internet。该选项是指不使用代理服务器而直接进行连接。该选项通常对应于 Internet Explorer 设置中的 “自动检测设置”。
- **从录制浏览器获取代理服务器设置**：使用录制浏览器中的代理服务器设置。这是默认选项。该选项不适用于 Web/WinSock Vuser。
- **使用自定义代理服务器**。在录制期间使用指定的代理服务器。可以为所有非安全 HTTP 站点指定一个代理服务器，并为所有安全的 (HTTPS) 站点指定另一个代理服务器。该选项只有在未选中上面两个选项的情况下才会启用。

  如果 HTTP 和 HTTPS 代理服务器是相同的，则只需指定 HTTP 地址和端口，并选择 “对所有协议使用同一代理服务器”选项。

某些代理服务器需要使用用户名和密码进行身份验证。如果通过需要身份验证的代理服务器录制会话，则请单击 “身份验证”按钮，并在 “代理服务器身份验证”对话框中输入相关的 “用户名”和 “密码”。

要指定希望 VuGen 直接访问的主机名或 IP 地址，而不使用代理服务器，请单击 “无代理服务器”按钮。将打开 “代理服务器异常”对话框。



键入希望 VuGen 直接访问的地址。使用分号分隔各个地址。

要使 VuGen 在访问本地 (Intranet) 地址时不使用代理服务器，请选择 “对于本地 (Intranet) 地址不使用代理服务器”选项。

### 还原代理服务器设置

如果为录制指定的代理服务器设置不同于计算机的常规浏览器设置， VuGen 将还原原始浏览器设置。默认情况下， VuGen 将在所启动的浏览器读取设置之后立即还原原始的代理服务器设置。要仅在停止录制后还原原始的代理服务器设置，请选中 “延迟还原代理服务器设置直到录制完成”复选框。该选项仅适用于 Internet Explorer。

最好立即还原代理服务器设置以确保计算机的安全。如果在录制后还原设置，则安全性会降低，但如果以后要读取代理服务器设置，则这一点是必需的。例如，在小程序、 ActiveX 控件和多窗口应用程序中录制 HTTP 操作时。

## 设置高级录制选项

使用 “Internet 协议 : 高级”设置在下列区域中设置录制选项：

- Internet 首选项录制选项
- 设置录制场景

### Internet 首选项录制选项

通过 “Internet 首选项”，可以自定义思考时间、重置上下文、保存快照以及生成 **web_reg_find** 函数等区域中的代码生成设置。注意，在多协议模式下，其中的某些选项不可用。

**录制思考时间：**（仅适用于单协议）思考时间用来模拟实际用户在各操作之间等待的时间，从而影响 Vuser 在运行脚本时使用录制的思考时间的方式。要录制用户思考时间，请选择 “录制思考时间”。可以通过定义 “思考时间阈值”来指定要录制为思考时间的用户等待时间的最小值。例如，可以将思考时间阈值设置为 5，这样，如果用户等待的时间小于五秒，将不录制思考时间。该选项仅适用于 Web 和无线 Vuser。

**重置每个操作的上下文：**（仅适用于 Web 和 Oracle NCA）该设置在默认情况下是启用的，用于通知 VuGen 重置各操作之间的所有 HTTP 上下文。通过重置上下文，可以使 Vuser 更准确地模拟开始浏览会话的新用户。该选项将重置 HTML 上下文，以便始终在操作开始时录制无上下文的函数。该选项还会清除缓存并重置用户名和密码。

**完全跟踪录制日志：**（仅适用于单协议）该设置将在录制期间创建跟踪日志。该日志由 Mercury Interactive 客户支持内部使用，并且在默认情况下是禁用的。

**在本地保存快照资源：**该选项指示 VuGen 在录制和回放期间保存快照资源的本地副本。通过该功能，可以使 VuGen 更准确地创建快照，并且更快地显示这些快照。

**为页标题生成 web_reg_find 函数：**（仅适用于 Web 和 Oracle NCA）通过该选项，可以为所有 HTML 页标题生成 **web_reg_find** 函数。 VuGen 将添加页标题标记中的字符串，并将其用作 **web_reg_find** 的参数。

选择 “为子帧生成 web_reg_find 函数”，以便为所录制页面的所有子帧中的页标题生成 **web_reg_find** 函数。

**支持字符集：**

**UTF-8**：通过该选项可支持 UTF-8 编码。该选项指示 VuGen 将非 ASCII 的 UTF-8 字符转换为本地计算机上的编码，以便在 VuGen 编辑器中正确显示它们。如果启用 UTF-8 支持选项，则无法录制非 UTF-8 字符集的站点。

**EUC-JP**：对于日文版 Windows 的用户，请选择该选项以支持使用 EUC-JP 字符编码的网站。该选项指示 VuGen 将 EUC-JP 字符转换为本地计算机上的编码，以便在 VuGen 编辑器中正确显示它们。 VuGen 会将所有 EUC-JP （日文版 UNIX）字符串转换为本地计算机上的 SJIS （日文版 Windows）编码，并在脚本中添加 **web_sjis_to_euc_param** 函数。（仅适用于 Kanji）

# 设置录制场景

还可以通过在下列区域中指定录制场景来进一步自定义录制：

- 录制自定义标头
- 筛选内容类型
- 指定非资源的内容类型

## 录制自定义标头

Web Vuser 会自动将多个标准 HTTP 标头随每个提交至服务器的 HTTP 请求一起发送。单击 “标头”以指示 VuGen 录制其他 HTTP 标头。可以使用下面三种模式：“不录制标头”、“录制列表中的标头”或 “录制不在列表中的标头”。在第一种模式下工作时， VuGen 不录制任何标头。在第二种模式下， VuGen 仅录制选中的自定义标头。如果指定 “录制不在列表中的标头”模式， VuGen 将录制除选中的标头之外的所有自定义标头以及其他危险标头。

下列标准标头称为危险标头：Authorization、 Connection、 Content-Length、 Cookie、 Host、 If-Modified-Since、 Proxy-Authenticate、 Proxy-Authorization、 Proxy-Connection、 Referer 和 WWW-Authenticate。除非在标头列表中将它们选中，否则将不会录制这些标头。默认选项为 “不录制标头”。

在 “录制列表中的标头”模式下， VuGen 将在脚本中为检测到的每个已选中标头插入一个 **web_add_auto_header** 函数。该模式是录制危险标头的理想模式，这种标头除非明确声明，否则将不会被录制。在 “录制不在列表中的标头”模式下， VuGen 将在脚本中为录制期间检测到的每个未选中标头插入一个 **web_add_auto_header** 函数。

要确定需要录制哪些自定义标头，可以执行一个录制会话，指示 VuGen 录制所有标头 （请参阅如下过程）。然后，可以决定录制哪些标头，不录制哪些标头。

在该示例中， *Content-type* 标头已在 “录制列表中的标头”模式下被指定。VuGen 已检测到该标头并向脚本中添加了以下语句：

```
web_add_auto_header("Content-Type","application/x-www-form-urlencoded");
```

指示服务器该应用程序的 Content-type 为 x-www-form-urlencode。

要控制自定义标头的录制，请执行下列操作：

1 在 “录制选项”树中，选择 “Internet 协议 : 高级”节点。

2 单击 “标头”。将打开 “标头”对话框。

3 使用下列方法之一：

- 要指示 VuGen 不录制任何标头，请选择 “不录制标头”。
- 要仅录制特定的标头，请选择 “录制列表中的标头”并在标头列表中选择所需的自定义标头。注意，标准标头 （例如 *Accept*）在默认情况下是选定的。
- 要录制所有标头，请选择 “录制不在列表中的标头”并且不选择列表中的任何项目。
- 要仅排除特定的标头，请选择 “录制不在列表中的标头”并选择需要排除的标头。

4 单击 “还原列表”可将列表还原为对应的默认列表。“录制列表中的标头”和 “录制不在列表中的标头”都有各自对应的默认列表。

5 单击 “确定”接受设置并关闭 “标头”对话框。

### 筛选内容类型

通过 VuGen，可以筛选所录制脚本的内容类型。可以指定要在脚本中录制或排除的内容类型。可以使用下列三种模式：“不筛选内容类型”、“排除列表中的内容类型”或 “排除不在列表中的内容类型”。在第一种模式下工作时， VuGen 不筛选任何内容类型。在第二种模式下， VuGen 只排除选定的内容类型。如果

指定 “排除不在列表中的内容类型”， VuGen 将筛选除选中的内容类型之外的所有内容类型。默认情况下，不激活任何筛选器。

例如，如果只关注网站上的文本和图像，则可以选择 “排除不在列表中的内容类型”，并指定类型 *text/html*、 *image/gif* 和 *image/jpeg*。 VuGen 将录制所有 HTML 页面和图像，并排除 *text/css*、 *application/x-javascript* 或站点上出现的其他资源。

**要在录制期间筛选内容，请执行下列操作：**

1 在 “录制选项”树中，选择 “Internet 协议 : 高级”节点。

2 单击 “内容类型”。将打开 “内容类型筛选器”对话框。

3 使用下列方法之一：

- 要指示 VuGen 不筛选任何内容，请选择 “不筛选内容类型”。
- 要只排除特定的内容类型，请选择 “排除列表中的内容类型”并从列表中选择所需的内容类型。
- 要只包含特定的内容类型，请选择 “排除不在列表中的内容类型”并选择要包含的内容类型。

4 单击 “还原列表”可将列表还原为对应的默认列表。“排除列表中的内容类型”和 “排除不在列表中的内容类型”都有各自对应的默认列表。

5 单击 “确定”接受设置并关闭 “内容类型筛选器”对话框。

### 指定非资源的内容类型

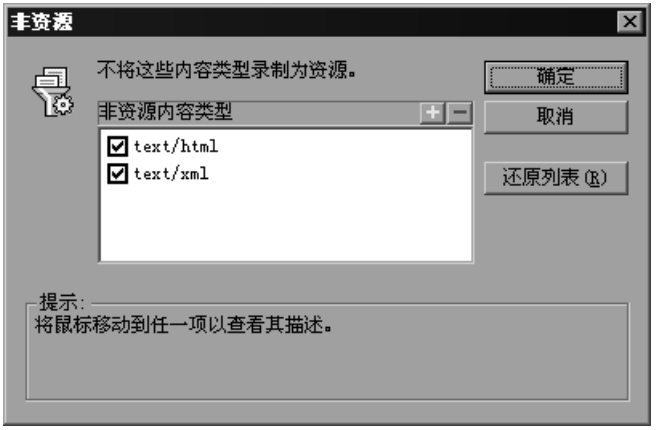
录制脚本时， VuGen 将指示在回放期间是否使用 **web_url** 函数中的 **Resource** 属性检索资源。如果将 **Resource** 属性设置为 0，则会在执行脚本期间检索资源。如果将 **Resource** 属性设置为 1，则 Vuser 会跳过资源类型。

```
web_url("nav_tpo.gif",
        "URL=http://graphics.aa.com/images/navimg/nav_tpo.gif",
        "Resource=1",
        "RecContentType=image/gif",
        "Referer=http://www.im.aa.com/American?BV_EngineID=...",
        "Mode=HTML",
        LAST);
```

可以排除特定的内容类型，使其不作为资源进行处理。例如，可以指示 VuGen 不将 *gif* 类型的资源作为资源来处理，从而无条件地下载该资源。当 VuGen 遇到 *gif* 类型的资源时，它会将 **Resource** 属性设置为 0，以指示 VuGen 在回放期间无条件地下载 gif 资源。

**要指定不应作为资源来录制的内容，请执行下列操作：**

1. 在 “录制选项”树中，选择 “Internet 协议 : 高级”节点。
2. 单击 “非资源”可打开一个对话框，并显示不应作为资源来录制的内容类型列表。



3 单击 “+” 号可向列表中添加内容类型。单击 “-” 号可删除现有项。

4 选中要启用的项旁边的复选框。

5 单击 “还原列表” 可将列表还原为默认列表。

6 单击 “确定” 接受设置并关闭 “非资源” 列表。

# 36

# 为 Web Vuser 设置录制选项

录制 Web 会话前，可以自定义录制选项。

本章描述：

- 指定用于录制的浏览器
- 选择录制级别
- 设置基于 HTML 的高级选项
- 设置基于 URL 的高级选项
- 设置录制级别

***下列信息仅适用于 Web Vuser 脚本。***

## 关于设置录制选项

通过 VuGen 可以录制用户在网站上执行的典型进程，从而生成 Web Vuser 脚本。

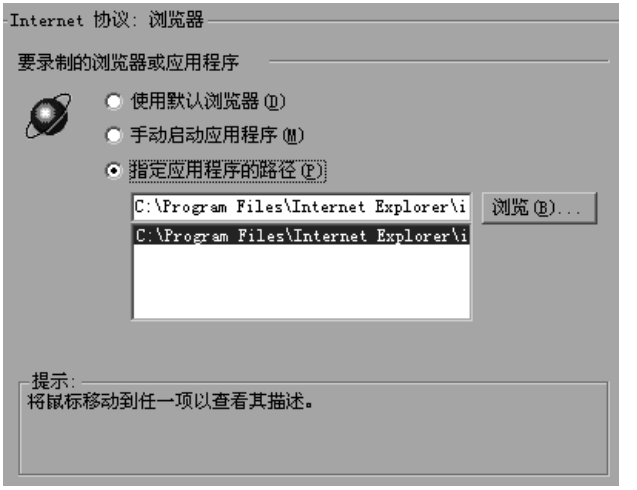
在录制前，可以配置录制选项，指定要录制的信息以及用于录制的浏览器或客户端，并指定脚本内容。

可以设置常见的 Internet 协议录制选项，如代理服务器设置和其他高级设置。详细信息，请参阅第 35 章 “为 Internet 协议设置录制选项”。

还可以为 Web Vuser 脚本设置关联录制选项。详细信息，请参阅第 41 章“配置 Web Vuser 脚本的关联规则”。

# 指定用于录制的浏览器

可以指定录制 Web Vuser 脚本时 VuGen 使用的浏览器。可以使用 “录制选项” 树中的 “Internet 协议：浏览器”节点指定浏览器的位置。

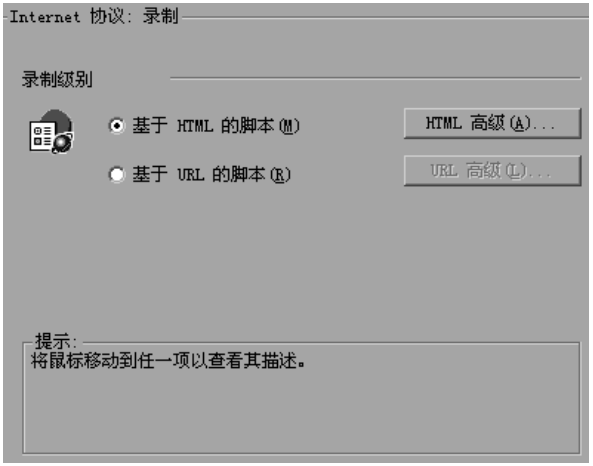



下列是可用的 “浏览器”选项：

- **使用默认浏览器**，可指示 VuGen 在录制计算机上使用默认的 Web 浏览器。
- **手动启动应用程序**，可指示 VuGen 在开始录制时不启动浏览器。启动录制会话后，必须手动启动浏览器或应用程序。
- **指定应用程序的路径**，可指示 VuGen 使用您指定的浏览器或应用程序。从路径列表中选择一个路径，或者单击 “浏览”按钮查找所需的应用程序。

## 选择录制级别

通过 VuGen 可以选择录制级别，从而指定要录制的信息，以及生成脚本时使用的函数。选择的录制级别取决于您的需要和环境。可用级别有 “基于 HTML 的脚本”和 “基于 URL 的脚本”。




“基于 HTML 的脚本”级别可为每个 HTML 用户操作生成单独的步骤。非 HTML 元素的录制取决于处理非 HTML 生成的元素的设置 （见下）。另请参阅第 412 页的 “设置基于 HTML 的高级选项”。

“基于 URL 的脚本”录制级别可捕获所有作为用户操作的结果发送到服务器的 HTTP 请求。该录制模式甚至可以捕获非 HTML 应用程序，如小程序和非浏览器应用程序。基于 URL 的脚本的可伸缩性更强，用于创建负载测试更加有效，该脚本包含有关用户操作的更详细的信息，但是不如基于 HTML 的脚本直观。另请参阅第 417 页的 “设置基于 URL 的高级选项”。

按照下列提示确定要选择的录制级别。

- 对于浏览器应用程序，请使用基于 HTML 的级别。
- 对于非浏览器应用程序，请使用基于 URL 的级别。

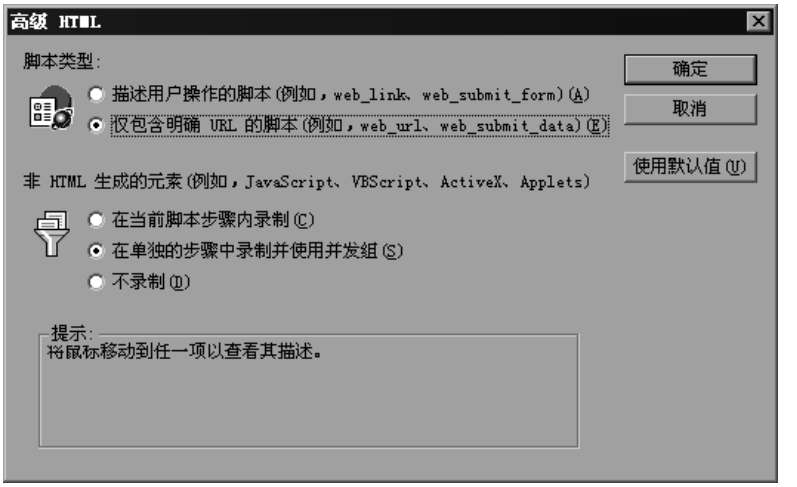
可以在录制时切换录制级别和高级录制选项。混合录制级别的选项可供高级用户用于调整性能。

## 设置基于 HTML 的高级选项

“基于 HTML”选项（默认录制级别）可指示 VuGen 在当前网页的上下文中录制 HTML 操作。它并不会在录制会话期间录制所有资源，而是在回放期间将其下载。

通过 VuGen 可以在下列方面为基于 HTML 的级别设置高级选项：

- 指定脚本类型
- 处理非 HTML 生成的元素




### 指定脚本类型

在基于 HTML 的级别中，可以指定脚本的类型：

- 描述用户操作的脚本
- 仅包含明确 URL 的脚本

第一个选项 （“描述用户操作的脚本”）是默认选项。它可以生成直接响应所进行的操作的函数。它可以创建 URL (**web_url**)、链接 (**web_link**)、图像 (**web_image**) 和表单提交 (**web_submit_form**) 函数。产生的脚本非常直观，并且类似上下文相关的录制。

```
/* 基于 HTML 的模式 - 描述用户操作的脚本 *//
...
web_url("Click Here For Additional Restrictions",
        "URL=http://www.im.aa.com/American...restrictions.html",
        "TargetFrame=",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://www.im.aa.com/American?...,
        "Snapshot=t4.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);
...
```

第二个选项 （“仅包含明确 URL 的脚本”）将所有链接、图像和 URL 录制为 **web_url** 语句，或者，如果是表单，则录制为 **web_submit_data**。它不会生成 **web_link**、 **web_image** 和 **web_submit_form** 函数。产生的脚本的直观性较差。如果站点中有许多链接具有相同的链接文本，该模式十分有用。如果使用第一个选项录制站点，它会录制链接的序号 （实例），但如果使用第二个选项录制，将按每个链接的 URL 将其列出。这便于实现该步骤的参数化和关联。

以下段说明了选定 “仅包含明确 URL 的脚本”时录制的会话：

```
/* 仅包含明确 URL 的基于 HTML 的脚本 *//
...
web_url("Click Here For Additional Restrictions",
        "URL=http://www.im.aa.com/American...restrictions.html",
        "TargetFrame=",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://www.im.aa.com/American?...
        "Snapshot=t4.inf",
        "Mode=HTML",
        LAST);

web_url("buttonhelp.gif",
        "URL=http://www.im.aa.com/American?BV_EngineID...,
        "TargetFrame=aamain",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://www.im.aa.com/American?...
        "Snapshot=t5.inf",
        "Mode=HTML",
        LAST);
...
```

### 处理非 HTML 生成的元素

许多网页包含非 HTML 元素，如小程序、XML、ActiveX 元素或 javascript。这些非 HTML 元素通常包含或检索它们自己的资源。例如，从录制的网页调用的 javascript *js* 文件可能会加载多个图像。小程序可以加载外部文本文件。使用下列选项，可以控制 VuGen 录制非 HTML 生成的元素的方式。

有下列选项可用：

- 在当前脚本步骤内录制 （默认选项）
- 使用并发组在单独的步骤中录制
- 不录制

第一个选项 （“在当前脚本步骤内录制”）不会为每个非 HTML 生成的资源生成新函数。它将所有资源作为为非 HTML 元素生成的 **web_url** 语句的参数列出。资源 （**web_url** 的参数）由 **EXTRARES** 标志表示。在下例中， **web_url** 函数列出了在页面上加载的所有非 HTML 生成的资源：

```
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);
```

第二个选项 （“使用并发组在单独的步骤中录制”）为每一个非 HTML 生成的资源新建一个函数，它不会将它们作为项目包括在页面的 **web_url** 函数中。为某个资源生成的所有 **web_url** 函数都被放置在一个并发组（用 **web_concurrent_start** 和 **web_concurrent_end** 括起）中。

下例中，在选定该选项的情况下录制了上面的会话。为小程序和使用小程序加载的文本文件生成了一个 **web_url** 函数：

```
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);
```

第三个选项 （“不录制”）指示 VuGen 不要录制由非 HTML 元素生成的任何资源。

注意，当您在基于 HTML 的模式中工作时， VuGen 会在 **web_url** 语句中插入 **TargetFrame** 属性。 VuGen 使用该信息在运行时浏览器中正确显示网页和测试结果报告。

```
web_url("buttonhelp.gif",
        "URL=http://www.im.aa.com/American?BV_EngineID=...",
        "TargetFrame=aamain",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://www.im.aa.com/American?BV_EngineID=...l",
        "Snapshot=t5.inf",
        "Mode=HTML",
        LAST);
```

录制基于 URL 的模式时， VuGen 会录制页面上所有帧的内容，因此忽略了 TargetFrame 属性。

### 设置基于 URL 的高级选项

“基于 URL”模式选项指示 VuGen 录制来自服务器的所有请求和资源。它会自动将每个 HTTP 资源录制为 URL 步骤 （**web_url** 语句），或者如果是表单，则录制为 **web_submit_data**。它不会生成 **web_link**、 **web_image** 和 **web_submit_form** 函数，也不会录制帧。

通过 VuGen 可以在下列方面为 URL 录制模式设置高级选项：

- 资源处理
- 浏览器缓存

### 资源处理

在基于 URL 的录制中， VuGen 会捕获作为 HTTP 请求的结果下载的所有资源。默认情况下，该选项处于启用状态，并且 VuGen 会录制并发组（用 **web_concurrent_start** 和 **web_concurrent_end** 括起）中 URL 后面的资源。资源包括图像文件、导入文件和 *js* 文件等。如果禁用该选项，会将资源作为单独的 **web_url** 步骤列出，但是不会将其标记为并发组。

以下段说明了在禁用 “在资源的源 HTML 页后为其创建并发组”选项时录制的会话。

```
web_concurrent_start (NULL);
...
web_url("spacer.gif",
        "URL=http://graphics.aa.com/images/spacer.gif",
        "Resource=1",
        "RecContentType=image/gif",
        "Referer=http://www.im.aa.com/American?BV_EngineID...",
        "Mode=HTTP",
        LAST);

web_url("calendar_functions.js",
        "URL=http://www.im.aa.com/travelp/calendar_functions.js",
        "Resource=1",
        "RecContentType=application/x-javascript",
        "Referer=http://www.im.aa.com/American?BV_Operation=...l",
        "Mode=HTTP",
        LAST);
...
web_concurrent_end (NULL);
```

注意，该脚本包含 *gif* 和 *js* 文件。该模式还包含其他图形文件和导入文件 （如 *imp*、 *txt*）或级联样式表 (*css*) 文件。

### 浏览器缓存

浏览器缓存会将最近查看过的页存储在计算机的内存中，以便减少访问网页所需的时间。默认情况下，“启用缓存”选项处于禁用状态， VuGen 在录制期间直接从服务器检索所有页，并且不使用浏览器缓存。

但是，某些应用程序无法在没有缓存的情况下运行。要使用缓存并且只直接从服务器检索最近修改过的页，请选择 “启用缓存”选项。

**If-Modified-Since** HTTP 标头是一种请求，客户端通过该请求检查自上次下载以来是否在服务器端修改了缓存的资源。如果资源已修改，则客户端会再次将它下载到缓存。否则，服务器会返回 HTTP 状态代码 304 （未修改）。如果禁用了缓存， **If-Modified-Since** 标头会被取消并且 VuGen 会直接从服务器检索所有页。在这种模式下， VuGen 会删除 **If-None-Match** 请求标头，以及 **Last-Modified**、 **Expires** 和 **Etag** 响应标头。如果浏览器没有收到上述任何响应标头，则它不会将图像存储在缓存中。

注意，可以使用 “高级标头”选项手动控制该标头 （请参阅第 404 页的 “录制自定义标头”）。

### 清除浏览器缓存

默认情况下，如果启用了浏览器缓存，则 VuGen 会在录制前清除缓存。这意味着缓存中的所有项均会到期，因此浏览器必须直接从服务器检索它们。

清除缓存需要 VuGen 直接从其网站访问所有页面，即使最近访问过该页也是如此。如果您在录制重复访问某个站点的 Vuser，则可以选择在录制前不清除浏览器缓存。

要指示 VuGen 在录制前不清除浏览器缓存，请清除 “在录制前清除缓存”复选框。注意，只有在使用 Internet Explorer 进行录制时，才适用该选项。

### 生成自定义请求

录制非浏览器应用程序时，可以指示 VuGen 将所有 HTTP 请求录制为自定义请求。 VuGen 会为所有请求生成 **web_custom_request** 函数，无论请求的内容是什么：

```
web_custom_request("www.aa.com",
        "URL=http://www.aa.com/",
        "Method=GET",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=",
        "Snapshot=t1.inf",
        "Mode=HTTP",
        LAST);
```

### 启用 EUC 编码的网页

（该选项只适用于日语 Windows。）使用非 Windows 标准字符集时，可能需要执行代码转换。字符集是从一组字符到一组整数的映射。该映射为给定字母表生成了唯一的字符和整数的组合。扩展 UNIX 代码 (EUC) 和 Shift 日本工业标准 (SJIS) 是用于在网站上显示日语的非 Windows 标准字符集。

Windows 使用 SJIS 编码，而 UNIX 使用 EUC 编码。当 Web 服务器位于 UNIX 计算机上，并且客户端是 Windows 时，由于编码方法的不同，网站中的字符不能在客户端上正确显示。这会影响 Vuser 脚本中 EUC 编码的日语字符的显示。

在录制期间， LoadRunner 会通过网页的 HTTP 标头检测它的编码。如果 HTTP 标头中没有显示有关字符集的信息，则它会检查 HTML 元标记。如果该页不向 HTTP 标头或元标记发送字符集信息，则 LoadRunner 不会检测 EUC 编码。

如果您事先知道网页是使用 EUC 编码的，则可以指示 LoadRunner 在录制期间使用正确的编码。要录制使用 EUC 编码的页面，请启用 “录制”选项卡中录制选项的 “EUC”选项 （只对于日语 Windows 是可见的）。

如果启用 “EUC”选项，将强制 LoadRunner 录制使用 EUC 编码的网页，即使它不使用 EUC 编码也是如此。因此，应仅当 LoadRunner 无法从 HTTP 标头或 HTML 元标记检测编码，以及您事先知道该页使用 EUC 编码时启用该选项。

在录制期间， LoadRunner 会从 Web 服务器接收 EUC 编码的字符串并将它转换为 SJIS。 SJIS 字符串保存在脚本的 Action 函数中。但是，为了成功回放，必须先将字符串重新转换为 EUC，然后再将其发送回 Web 服务器。因此， LoadRunner 会在 Action 函数前添加一个 **web_sjis_to_euc_param** 函数，该函数可将 SJIS 字符串重新转换为 EUC。

在下例中，用户会转至 EUC 编码的网页并单击某个链接。 LoadRunner 先录制 Action 函数，然后向脚本中 Action 函数前添加 **web_sjis_to_euc_param** 函数。

```
web_sjis_to_euc_param("param_link","Search");

web_link("LinkStep","Text={param_link}");
```

## 设置录制级别

本部分描述用于设置录制级别及其高级选项的过程。注意，您可以在录制时切换录制级别和高级录制选项。

**要设置录制选项，请执行下列操作：**

1 选择 “工具” > “录制选项” 打开 “录制选项”。在 “录制选项” 树中选择 “Internet 协议：录制” 节点。

2 选择录制模式：“基于 HTML” 或 “基于 URL” 模式。

3 对于基于 HTML 的录制，请单击 “HTML 高级”，为脚本类型和非 HTML 元素的处理设置其他选项。

选择脚本类型。

选择处理非 HTML 资源的方法。（请参阅第 412 页的 “设置基于 HTML 的高级选项”）。

4 对于基于 URL 的录制，请单击 “URL 高级”，为资源处理和缓存启用设置其他脚本选项。

选择 “在资源的源 HTML 页后为其创建并发组”，启用资源录制并将它们标记为并发组 （用 **web_concurrent_start** 和 **web_concurrent_end** 括起）。

选择 “启用缓存” 在录制期间使用浏览器缓存。如果启用该选项，请清除 “在录制前清除缓存” 复选框，指示 VuGen 不清除缓存和使用以前访问过的页。

选择 “仅使用 web_custom_request”，将所有 HTTP 请求生成为 **web_custom_request** 函数。

5 有关这些选项的详细信息，请参阅第 417 页的 “设置基于 URL 的高级选项”。

# 37

# 配置 Internet 运行时设置

录制 Internet 协议 Vuser 脚本之后，需要配置其运行时设置。

本章描述：

- 设置代理服务器选项
- 设置浏览器仿真属性
- 设置 Internet 首选项

***下列信息适用于所有“Internet 协议”Vuser 类型，如 Web、 Wireless、 Oracle NCA 和 Real。***

## 关于 Internet 运行时设置

开发 Internet 协议 Vuser 脚本之后，需要设置运行时设置。

有关适用于所有 Vuser 的常规运行时设置的信息，请参阅第 9 章 “配置运行时设置”。有关网络速度运行时设置的信息，请参阅第 10 章 “配置网络运行时设置”。

通过 Internet 运行时设置，可以配置 Internet 环境以便 Vuser 可以准确模拟实际用户。可以为 “代理服务器”、“浏览器”及其他高级首选项设置 Internet 运行时设置。

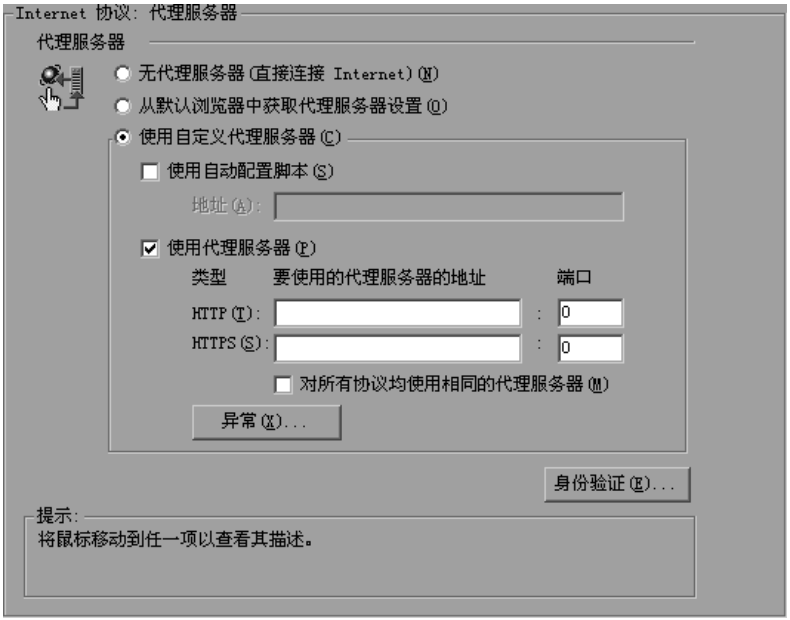
从 “运行时设置”对话框中，可设置与 Internet 相关的运行时设置。单击适当的节点可指定所需的设置。

注意，还可以从 LoadRunner Controller 中修改运行时设置。在 “Controller”窗口中，单击 “设计”选项卡，再单击 “运行时设置”按钮。

**注意：**通过使用 Vuser 函数设置的运行时设置将覆盖通过 “运行时设置”对话框设置的相应设置。有关使用 Vuser 函数的详细信息，请参阅第 33 章“使用 Web Vuser 函数”。

## 设置代理服务器选项

使用 “运行时设置”树的 “Internet 协议 : 代理服务器”节点，可设置与代理服务器相关的设置：




默认情况下， Vuser 使用在 Web 录制选项中用于录制的浏览器的代理服务器设置。建议对录制和回放使用相同的设置。有关 “代理服务器录制”选项的信息，请参阅第 36 章 “为 Web Vuser 设置录制选项”。

在运行时设置中可以使用下列代理服务器选项。

- **无代理服务器**：所有 Vuser 都始终使用与 Internet 的直接连接。这意味着在不使用代理服务器的情况下进行连接。
- **从默认浏览器中获取代理服务器设置**：所有 Vuser 都使用所在计算机上默认浏览器的代理服务器设置。
- **使用自定义代理服务器**：所有 Vuser 都使用自定义代理服务器。可以提供实际代理服务器的详细信息，或者提供启用自动配置的代理服务器自动配置脚本（*.js* 格式的 pac 文件）的路径。（请参阅第 426 页的 “设置自动代理服务器配置”）

  要提供服务器的详细信息，请指定服务器的 IP 地址或服务器名和端口。可以为所有 HTTP 站点指定一台代理服务器，而为所有 HTTPS （安全）站点指定另一台代理服务器。

  提供代理服务器信息之后，可以指定代理服务器的身份验证信息，并指明代理服务器规则的例外。

---

**注意**：要指示 Vuser 在回放期间等待代理服务器响应，并且假设代理服务器不支持基本身份验证，请添加以下语句：
web_set_sockets_option("PROXY_INITIAL_BASIC_AUTH", "0");

---

## 身份验证

如果代理服务器要求对每个 Vuser 进行身份验证，请使用该对话框输入相应的密码和用户名。

**用户名**：输入 Vuser 将用于访问代理服务器的用户名。

**密码**：输入 Vuser 访问代理服务器所需的密码。

### 例外

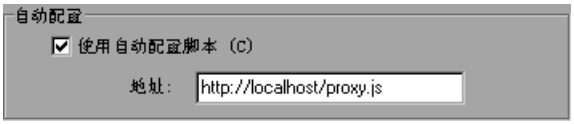
可以指定所有 Vuser 都使用指定的代理服务器。在这种情况下，如果有希望 Vuser 直接访问 （即不使用代理服务器）的任何 URL，请在文本框中输入这些 URL 的列表。

**不对以以下字母开头的地址使用代理服务器：**输入要从代理服务器中排除的地址。使用分号将各个项目隔开。

**对于本地 (Intranet) 地址不使用代理服务器：**选中该复选框从代理服务器中排除本地地址，例如 Intranet 中的地址。

### 设置自动代理服务器配置

大多数浏览器都支持 “自动代理服务器配置” 功能。通过该功能可以指定包含代理服务器分配信息的 Javascript 文件 （扩展名为 *.js*）。该脚本将根据 URL 指示浏览器何时访问代理服务器以及何时直接连接到站点。此外，该脚本还可以指示浏览器对特定的地址使用特定的代理服务器，而对其他地址使用另一台代理服务器。要指定 Internet Explorer (IE) 中的配置脚本，请选择 “工具” > “Internet 选项”，然后选择 “连接” 选项卡。单击 “局域网设置” 按钮。在 “局域网 (LAN) 设置” 对话框中，选择 “使用自动配置脚本” 选项，并指定脚本的位置。 VuGen 即可支持自动代理服务器配置。可以为自动代理服务器配置指定 JavaScript，以便当 LoadRunner 运行测试时，可以使用 JavaScript 文件中的规则。



要跟踪 Vuser 的行为，请在文本执行期间生成日志，然后查看 “执行日志” 选项卡或 *mdrv.log* 文件。该日志将显示为每个 URL 使用的代理服务器。以下示例中， VuGen 对 URL australia.com 使用直接连接，而对 URL http://www.google.com 则使用代理服务器 *aqua*。

```
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
...l
Action1.c(17): t=1578ms: 168-byte request headers for
"http://www.google.com/" (RelFrameId=1)
Action1.c(17): GET http://www.google.com/ HTTP/1.1\r\n
```

### 设置代理服务器运行时设置

以下部分讨论配置代理服务器运行时设置所需的步骤。

**要设置代理服务器设置，请执行下列操作：**

1 打开 “运行时设置”。单击 VuGen 工具栏中的 “运行时设置” 按钮，或选择 “Vuser” > “运行时设置”。

2 选择所需的代理服务器选项 “无代理服务器”、“从默认浏览器中获取代理服务器设置” 或 “使用自定义代理服务器”。

3 如果指定了自定义代理服务器，请：

- 指明 HTTP 和 HTTPS 代理服务器的 IP 地址
- 要使用 JavaScript 文件指明代理服务器，请选择 “使用自动配置脚本”选项并指定脚本位置。可以指定以 http:// 开头的 Web 位置 （例如，*http://hostname/proxy.js*），也可以指定文件服务器上的位置 （例如，*C:\temp\proxy.js*）。

4 要指定希望 Vuser 直接访问 （而不使用代理服务器）的 URL，请单击 “异常”，然后提供这些 URL 的列表。在 “异常”对话框中，还可以指定直接访问本地 (Intranet) 地址。

5 如果代理服务器要求进行身份验证，请单击 “身份验证”，然后输入相应的密码和用户名。

6 选中 “对所有协议使用同一代理服务器”复选框，指示 Vuser 对所有 Internet 协议 （HTTP、 HTTPS）都使用同一代理服务器，而不是为安全站点指定特定的服务器。

## 设置浏览器仿真属性

使用 “运行时设置”树中的 “浏览器 : 浏览器仿真”节点，可设置测试环境的浏览器属性。

### 浏览器属性

可以在下列区域中设置浏览器属性：

- 要模拟的用户代理浏览器
- 模拟浏览器缓存
- 下载非 HTML 资源
- 每次循环模拟一个新用户

还可以设置高级选项，对较新资源进行高速缓存和检查。

### 要模拟的用户代理浏览器

每当 Vuser 向 Web 服务器发送请求时，相应的请求总是包含 HTTP 标头。第一行文本包含一个动词 （通常为 “GET” 或 “POST”）、资源名 （例如 “pclt/default.htm”）和协议版本 （例如 “HTTP/1.0”）。后面的行中包含属性名形式的 “标头信息”、一个冒号和某个值。该请求以空行结束。

所有 Internet Vuser 标头都包含用于标识所模拟的浏览器 （或无线工具包）类型的 “用户代理” 标头。例如，

用户代理：Mozilla/3.01Gold (WinNT; I)

将浏览器标识为在 Intel 计算机上的 Windows NT 操作系统中运行的 Netscape Navigator Gold 版本 3.01。

从 “浏览器仿真” 节点中单击 “更改…”，以指定要包含在标头中的浏览器信息。可以指定 Web Vuser 模拟任一标准浏览器。此外，对于非浏览器 HTTP 应用程序，可以指定 HTTP 客户端以匹配特定用户的应用程序。在这种情况下，必须输入包含在所有后续 HTTP 标头中的 “自定义用户代理” 字符串。默认情况下， Vuser 使用 Internet Explorer 4.0 浏览器代理。

注意该选项未指明回放浏览器 – 用于播放脚本的浏览器。它只影响发送到服务器的 HTTP 标头的用户代理属性。

### 模拟浏览器缓存

该选项指示 Vuser 用高速缓存模拟浏览器。高速缓存用于保留经常访问的文档的本地副本，从而减少与网络连接的时间。默认情况下，启用高速缓存模拟。禁用高速缓存时，LoadRunner 仍然每个页面图像只下载一次。从 Controller 中运行多个 Vuser 时，每个 Vuser 都使用自己的高速缓存，并从该高速缓存中检索图像。如果禁用了该选项，则所有 Vuser 在模拟浏览器时都没有可用的高速缓存。

可以修改运行时设置以匹配 Internet Explorer 的浏览器设置：

| 浏览器设置 | 运行时设置 |
|---|---|
| 每次访问页时 | 选择 “模拟浏览器缓存”并启用 “每次访问该页时检查已存储页的较新版本” |
| 每次启动 Internet Explorer 时 | 仅选择 “模拟浏览器缓存” |
| 自动 | 仅选择 “模拟浏览器缓存” |
| 从不 | 选择 “模拟浏览器缓存”并禁用 “每次访问该页时检查已存储页的较新版本” |

---

**注意：**与常规浏览器高速缓存不同，指定给 Vuser 的高速缓存仅模拟图形文件的存储。该高速缓存不存储文本或其他与网页相关联的页面内容。

---

还可以设置下列两个浏览器高速缓存选项：

**缓存要求内容 (HTML) 的 URL：**该选项指示 VuGen 仅高速缓存需要 HTML 内容的 URL。进行分析、验证或关联时可能需要这些内容。选择该选项时，将自动缓存 HMTL 内容。要定义需要缓存的其他内容类型，请单击 “高级”。（这将增加虚拟用户的内存使用量。）默认情况下禁用该选项。详细信息，请参阅 “高速缓存要求内容的 URL - “高级””。

**每次访问该页时检查已存储页的较新版本：**该设置将指示浏览器检查指定 URL 的较新 （与存储在高速缓存中的内容相比）版本。启用该选项时， VuGen 将向 HTTP 标头中添加 If-modified-since 属性。该选项可以保证始终显示最新版本的页面，但在场景执行期间将产生更多的通信流量。默认情况下，浏览器不检查较新的资源，因此禁用该选项。配置该选项以匹配要模拟的浏览器中的设置。

### 下载非 HTML 资源

该选项指示 Vuser 在回放期间访问网页时加载图形图像。其中包括同页面一起录制的图形图像，还包括未明确随同页面一起录制的图形图像。当实际用户访问网页时，需要等待图像加载。因此，如果打算测试整个系统 （包括终端用户时间），请启用该选项 （默认情况下启用）。要提高性能并且不模拟实际用户，请禁用该选项。

---

**注意**：如果在图像检查中遇到了差异，请禁用该选项，因为每次访问网页时，一些图像会随之改变 （例如，广告横幅）。

---

### 每次循环模拟一个新用户

指示 VuGen 将各个循环之间的所有 HTTP 上下文重置为 *init* 部分结束时相应的状态。使用该设置， Vuser 可以更准确地模拟开始浏览会话的新用户。它将删除所有 Cookie，关闭所有 TCP 连接 （包括 Keep-Alive 连接），清除模拟浏览器的高速缓存，重置 HTML 帧层次结构 （帧编号将从 1 开始）并清除用户名和密码。默认情况下启用该选项。

## 高速缓存要求内容的 URL – “高级”

通过 “高级”对话框可以指定要存储在高速缓存中的 URL 内容类型。可以从 “运行时设置” – “浏览器 : 浏览器仿真”节点中访问该对话框。

注意，不支持同时对多个组的高级设置进行更改 – 请分别编辑每个组的设置。

**要添加内容类型，请执行下列操作：**

1. 启用 “指定除 HTML 页以外还要求其他内容的 URL”选项。
2. 单击加号可以添加其他内容类型，例如 text/plain、 text/xml、 image/jpeg 和 image/gif。请在文本框中输入内容名。
3. 要从列表中删除内容类型，请选择该类型，然后单击减号。

# 设置 Internet 首选项

使用 “运行时设置”树中的 “Internet 协议 : 首选项”节点，设置与下列区域相关的设置：

- 图像和文本检查
- 生成 Web 性能图
- 高级 Web 运行时选项

### 图像和文本检查

通过 “启用图像和文本检查”复选框，可以让 Vuser 在回放期间执行验证检查（方法是执行验证函数）：web_find 或 **web_image_check**。该选项仅适用于在基于 HTML 的模式下录制的语句。运行验证检查的 Vuser 使用的内存要比不执行检查的 Vuser 使用的内存多 （默认情况下禁用）。

### 生成 Web 性能图

指示 Vuser 收集用于创建 Web 性能图的数据。在测试执行期间使用联机监视器以及在测试执行之后使用 Analysis 来查看每秒点击次数图、每秒页数图和 “每秒响应字节数 （吞吐量）” 图。使用 Analysis 可在测试执行之后查看组件细分图。选择 Vuser 要收集的图数据的类型。

---

**注意**：如果不使用 Web 性能图，请禁用这些选项以节省内存。

---

### 高级 Web 运行时选项

**WinInet 回放** （而非套接字）：指示 VuGen 使用 WinInet 回放引擎。 VuGen 有两个 HTTP 回放引擎：基于套接字 （默认值）或基于 WinInet。 WinInet 是由 Internet Explorer 使用的引擎，它支持集成到 IE 浏览器中的所有功能。 WinInet 回放引擎的局限性在于它是不可伸缩的，而且不支持 UNIX。此外，使用线程时，WinInet 引擎不能准确模拟调制解调器的速度和连接数。

VuGen 专用的基于套接字的回放是一种可伸缩以便进行负载测试的轻型引擎。使用线程时也非常准确。基于套接字的引擎的局限性在于它不支持 SOCKS 代理服务器。如果在该类型的环境中进行录制，请使用 WinInet 回放引擎。

**自动事务名中的文件和行号**：通过向事务名中添加文件名和行号，可为自动事务创建唯一的事务名 （默认情况下启用）。

---

**注意**：该选项会将其他信息放置到日志文件中，因此要求更多的内存。

---

**非关键项目错误 （如警告所述）**：该选项为在某一项 （该项对于负载测试并不关键，例如无法下载的图像或 Java 小程序）失败的函数返回警告状态。默认情况下启用该选项。如果已确定环境中的非关键性错误会失败，则可以禁用该选项。可以通过向非资源列表中添加内容类型来将该类型设置为关键类型。详细信息，请参阅第 407 页的 “指定非资源的内容类型”。

**出现错误时激活快照**：该选项用于在出现错误时生成快照。通过显示 Vuser 日志并双击发生错误的行，可以在 Controller 中查看快照。

**回放期间保存所有快照资源的本地副本**：指示 VuGen 将快照资源保存到本地计算机上的文件中。该功能可以使运行时查看器更准确地创建快照以及更快速地显示快照。

## Internet 首选项的其他选项

可以为 Internet 首选项设置下列高级选项：DNS 缓存、 HTTP 版本、 Keep-Alive HTTP 连接、HTTP 请求连接超时、HTTP 请求接收超时、网络缓冲区大小以及步骤下载超时。

**DNS 缓存**：指示 Vuser 从域名服务器中解析主机的 IP 地址值之后，将该地址保存到高速缓存中。这就节省了以后调用同一服务器的时间。在 IP 地址更改的情况下，如同特定的负载平衡技术一样，请确保禁用该选项以防止 Vuser 使用高速缓存中的值。（默认情况下启用）

**HTTP 版本**：指定要使用的 HTTP 版本：版本 1.0 或 1.1。每当 Vuser 向 Web 服务器发送请求时，都会将该信息包含在 HTTP 请求标头中。默认值为 HTTP 1.1。HTTP 1.1 支持下列功能：

- 持续连接 – 请参阅下面的 “Keep-Alive HTTP 连接”。
- HTML 压缩 – 请参阅 “执行 HTML 压缩”。
- 虚拟宿主 – 共享同一 IP 地址的多个域名。

**Keep-Alive HTTP 连接**：*Keep-Alive* 是用于建立持续连接或连续连接的 HTTP 扩展的术语。这些长期活动的 HTTP 会话允许在同一 TCP 连接上发送多个请求。这就提高了 Web 服务器和客户端的性能。

Keep-Alive 选项仅适用于支持 Keep-Alive 连接的 Web 服务器。该设置指定运行 Vuser 脚本的所有 Vuser 都启用 Keep-Alive HTTP 连接。（默认情况下为 YES）

**由资源引起的步骤超时是警告**：如果由于资源未在超时间隔内加载而引起超时，将发出警告而不是错误。对于非资源， VuGen 总是发出错误。（默认情况下为 NO）

**分析 HTML 内容类型**：要求使用 HTML 时，只有当响应属于指定的内容类型时才会分析该响应：HTML (*text\html*)、 TEXT （任何文本）或 ANY （任何内容类型）。注意 text/xml 不会被分析为 HTML。默认值为 “TEXT”。

**接受服务器端压缩**：向服务器指明回放可以接受压缩数据。注意，通过接受压缩数据，可能会显著增加 CPU 消耗。

**HTTP 请求连接超时 （秒）**：Vuser 终止之前在步骤内等待特定 HTTP 请求连接的时间 （秒）。超时为服务器保持稳定并响应用户提供了机会。默认值为 120 秒。

**HTTP 请求接收超时 （秒）**：Vuser 终止之前在步骤内等待接收特定 HTTP 请求的响应时间 （秒）。超时为服务器保持稳定并响应用户提供了机会。默认值为 120 秒。

超时设置主要用于以下高级用户：这些用户已确定可接受的超时值应该随环境而异。大多数情况下，默认设置应该足够长。如果服务器在合理的时间内并未做出响应，请检查其他与连接相关的问题，不要设置太长的超时，否则可能会导致脚本不必要地等待。

**步骤下载超时 （秒）**：Vuser 在终止脚本中的某个步骤之前将要等待的时间。该选项可用于模拟等待页面的时间不超过 x 秒的用户行为。

**网络缓冲区大小**：设置用于接收 HTTP 响应的缓冲区的最大大小。如果该数据的大小超过了指定的大小，则服务器将按块发送数据，从而增加了系统开销。从 Controller 中运行多个 Vuser 时，每个 Vuser 都使用自己的网络缓冲区。该设置主要用于以下高级用户：这些用户已确定网路缓冲区的大小可能影响其脚本的性能。默认值为 12K 字节。

**验证重试时的固定思考时间 （秒）**：自动向 Vuser 脚本中添加思考时间，以便模拟输入验证信息 （用户名和密码）的用户。该思考时间将包含在事务时间中。默认值为 0。

## 获取调试信息

运行 Vuser 脚本时，执行信息将显示在 LoadRunner Controller 的 “输出”窗口中。使用 “常规”运行时设置的 “日志”节点，可控制发送到 “输出”窗口和日志文件中的信息量。（请参阅 “配置日志运行时设置”）

调试信息包括：

- 日志信息
- 事务故障
- 网关的连接状态 – 连接、断开连接和重定向。（仅适用于 WAP）

要获取有关调试的详细信息，请编辑 *default.cfg* 文件。查找 “WEB”部分并将 “LogFileWrite”标志设置为 *1*。生成的跟踪文件会记录脚本执行过程中的所有事件。

执行负载测试时，请确保清除了 “LogFileWrite”标志，以防止 LoadRunner 创建大型跟踪文件来浪费资源。

# 执行 HTML 压缩

支持 HTTP 1.1 的浏览器可以解压缩 HTML 文件。服务器压缩文件以便进行传输，大大降低了数据传输所需的带宽。

要在 VuGen 中启用压缩，请将函数

```
web_add_auto_header("Accept-Encoding", "gzip");
```

添加到脚本的开头。要验证服务器是否发送了压缩数据，请在 “执行”日志的服务器响应部分中搜索字符串 *Content -Encoding:gzip*。该日志还显示了解压缩前后的数据大小。

压缩会对大型数据传输操作产生更大的影响 – 数据越大，压缩产生的影响也就越大。使用较大数据时，还会增加网路缓冲区大小 （请参阅 ““网络缓冲区大小”选项”），以便获取单个块中的数据。

# 38

# 检查网页内容

录制 Web Vuser 脚本之后，可以配置运行时设置检查页面内容。

本章描述：

- 设置内容检查运行时设置
- 定义内容检查规则

***下列信息仅适用于 Web Vuser 类型。***

## 关于检查网页内容

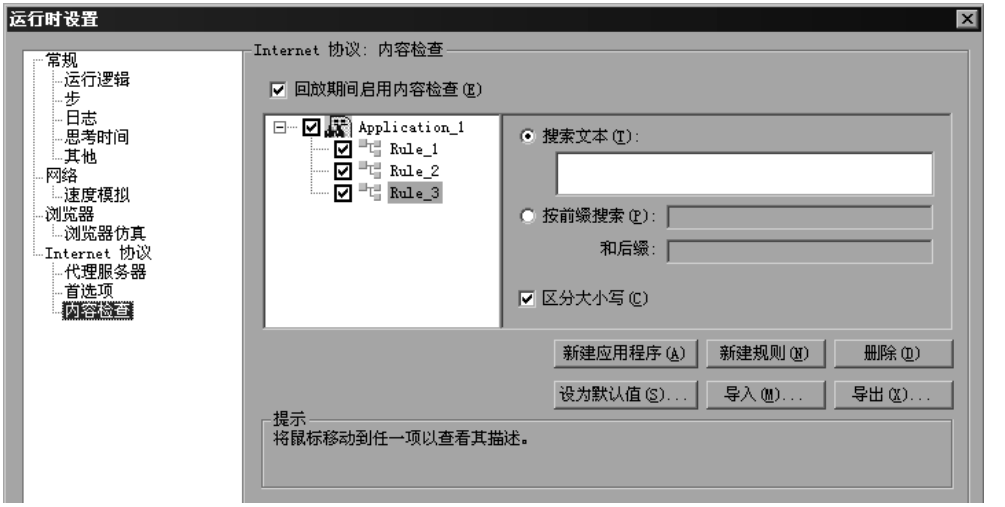
通过 VuGen 的内容检查机制，可以在页面内容中检查某个特定的字符串。这对于检测非标准错误非常有用。在正常操作情况下，当应用程序服务器发生故障时，浏览器将显示常规 HTTP 错误页，指明错误的性质。 VuGen 可以识别这些标准的错误页，并将其视为错误，从而使脚本运行失败。但某些应用程序服务器会显示自己的错误页，这些错误页不会被 VuGen 检测到并视为错误页。该错误页由服务器发出，并包含设置了格式的文本字符串，说明发生错误。

例如，假定发生错误时应用程序显示一个自定义页面，其中包含文本 “ASP 错误”。您可以指示 VuGen 在所有返回的页面中查找该文本。如果 VuGen 检测到该字符串，它将使回放失败。注意， VuGen 将搜索页面的正文，而不是标题。

# 设置内容检查运行时设置

可以使用 “Internet 协议：内容检查”运行时设置指定要搜索的内容。可以使用不同的规则定义多个应用程序的内容。下列部分讨论：

- ➤ 了解内容规则
- ➤ 定义内容检查规则




## 了解内容规则

可以使用内容检查运行时选项在页面内容中检查某个特定字符串。这对于检测非标准错误非常有用。在正常操作情况下，当应用程序服务器发生故障时，浏览器将显示常规 HTTP 错误页，指明错误的性质。 VuGen 可以识别这些标准的错误页，并将其视为错误，从而使脚本运行失败。但某些应用程序服务器会显示自己的错误页，这些错误页不会被 VuGen 检测到并视为错误页。该错误页由服务器发出，并包含设置了格式的文本字符串，说明发生了错误。

例如，假定发生错误时应用程序显示一个自定义页面，其中包含文本 “ASP 错误”。您可以指示 VuGen 在所有返回的页面中查找该文本。如果 VuGen 检测到该字符串，它将使回放失败。注意， VuGen 将搜索页面的正文，而不是标题。

注意，不支持对多个组的内容检查设置进行全局更改 – 请单独编辑每个组的设置。

**回放期间启用内容检查：**回放期间启用内容检查。（默认情况下已启用）注意，即使在定义应用程序之后，仍然可以通过禁用该选项使其不能应用于特定的测试运行。

### 规则信息

该右窗格包含要查找的文本的匹配条件。可以指定实际的文本或者该文本的前缀和后缀。

**搜索文本：**要搜索的字符串文本。

**按前缀和后缀搜索：**要搜索的字符串的前缀和后缀。

**区分大小写：**搜索时区分大小写。

## 添加和删除应用程序与规则

**新建应用程序**：自动将新应用程序添加到左窗格的应用程序列表中。默认名称是 Application_*index*（从 **Application_1** 开始）。新建组之后，单击“新建规则”向该组添加规则。要修改应用程序的名称，请选择其名称并单击它。

**新建规则**：在右窗格中显示规则条件，通过该条件，可以为当前选定的应用程序输入新规则。这些规则以标准 xml 文件的形式与脚本一起存储。可以导出规则文件并与其他用户共享，也可以将其导入到其他计算机。

**删除**：删除选定的规则或应用程序。

### 导入和导出规则

**导入 / 导出**：导入或导出规则文件。扩展名为 *xml* 的规则文件存储应用程序和规则。可以导出该文件，以便在其他计算机上使用。也可以导入其他规则文件。如果要导入一个规则，而选定规则与现有规则相冲突，则 VuGen 会发出警告，指明该规则是冲突规则。此时，您可以选择是将目前导入的规则与以前脚本中创建的规则合并，还是覆盖当前规则。单击“导出”时， VuGen 将打开“选择要导出的应用程序”对话框。

### 将规则设置为默认值

**设为默认值**：内容检查规则有三种：安装、默认和每脚本。安装规则将在产品安装期间自动提供。默认规则适用于计算机上执行的所有脚本。每脚本规则是为当前脚本定义的规则。修改或添加规则时，所做的更改仅适用于当前脚本。要指示 VuGen 向默认规则列表中添加规则，使其适用于该计算机上的所有脚本，请单击“设为默认值”。

使用多个脚本或执行产品升级时，默认规则和脚本规则之间可能会发生冲突。VuGen 将询问您是否要合并这些规则。如果合并这些规则 （建议），则该规则将被添加到应用程序的规则列表中。

该操作仅影响已在应用程序列表 （左窗格）中启用的应用程序。如果当前脚本中没有标记为 “已启用”的应用程序，则不会在默认文件中将任何应用程序标记为 “已启用”。单击 “是”可覆盖默认文件。单击 “否”可取消操作并保留原始默认文件。

这些规则将以标准 xml 文件的形式进行存储。可以导出规则文件并与其他用户共享，也可以将其导入到其他计算机。

**单击 “设为默认值”（并确认覆盖）后， VuGen 将执行下列操作：**

1. 将默认文件中的所有应用程序标记为 “已禁用”。
2. 对于当前脚本中标记为 “已启用”的应用程序，则会执行合并或复制操作，具体情况取决于应用程序是否存在。如果应用程序存在，则会将当前脚本的规则与默认文件的规则合并。如果默认文件中没有该应用程序，则仅将规则复制到默认文件中。
3. 在默认文件中，将脚本中已启用的应用程序标记为 “已启用”。如果当前脚本中没有标记为 “已启用”的应用程序，则不会在默认文件中将任何应用程序标记为 “已启用”。

#### 使用默认值

从默认文件中导入规则。单击该按钮时， VuGen 将打开一个对话框，其中包含应用程序及其默认设置的列表。可以选择导入还是修改这些规则。如果与现有规则冲突， VuGen 将发出警告，指明该规则是冲突规则。也可以将默认文件中定义的规则与当前定义的规则合并。

要为所有应用程序使用默认设置，请单击 “使用默认值”，从默认文件中导入定义。将打开一个对话框，其中包含应用程序及其默认设置的列表。可以选择导入还是修改这些定义。如果与现有规则冲突， VuGen 将发出警告，指明该规则是冲突规则。可以使用活动规则合并或覆盖默认文件中定义的规则。

### 定义内容检查规则

使用 “运行时设置”树中的 “Internet 协议：内容检查”节点可以定义检查网页内容的规则。

要定义内容检查规则，请执行下列操作：

1. 打开运行时设置，然后选择 “Internet 协议：内容检查”节点。
2. 选择 “回放期间启用内容检查”选项。
3. 单击 “新建应用程序”，向要检查其内容的应用程序列表中添加新项。
4. 单击 “新建规则”，为现有应用程序添加规则。每台应用程序服务器都可以有一个或多个规则。通过在左窗格中清除或选中规则旁边的复选框可以启用或禁用相关规则。
5. 要搜索实际的文本字符串，请选择 “搜索文本”，并指定要搜索的文本。建议指定的文本尽可能具体。例如，不要使用术语 “错误”，而使用 “ASP 错误”或者特定于应用程序的文本。
6. 要搜索字符串前后的文本，请选择 “按前缀搜索”，并指定前缀和后缀。
7. 要指定搜索时区分大小写，请选中 “区分大小写”复选框。
8. 要将规则设置为默认值，指明其适用于该计算机上的所有脚本，请选择该规则或应用程序并单击 “设为默认值”。
9. 要导出规则文件，请单击 “导出”，并指定保存位置。
10. 要导入规则文件，请单击 “导入”，并查找该文件。
11. 要删除应用程序或规则，请选择该应用程序或规则，并单击 “删除”。
12. 要为所有应用程序使用默认设置，请单击 “使用默认值”。将打开一个对话框，其中包含应用程序及其默认设置的列表。可以在有冲突时选择是覆盖规则还是合并规则。

# 39

# 在负载下验证网页

可以向 Web Vuser 脚本中添加 Web 检查以确定在运行 Vuser 脚本时，服务器是否返回了正确的网页。

本章描述：

- 添加文本检查
- 使用其他文本检查方法
- 添加图像检查
- 定义其他属性
- 使用常规表达式

***下列信息仅适用于 Web Vuser 脚本。***

# 关于负载下的验证

通过 VuGen 可以向 Web Vuser 脚本中添加 Web 检查。 Web 检查可验证网页上是否存在特定的对象。该对象可以是文本字符串、图像或 Java 小程序。

通过 Web 检查可以确定当许多 Vuser 访问您的网站时，网站是否能够正常运行 – 即服务器是否返回正确的网页。当站点承受许多用户所带来的负载时 （此时服务器更有可能返回不正确的页面），这一点尤其重要。

例如，假设网站显示有关全球主要城市气温的信息。您使用 VuGen 创建了一个访问网站的 Vuser 脚本。在下面的脚本段中， Vuser 将访问站点 *www.aa.com* 上的超文本链接。

浏览器显示与该 URL 关联的页面。然后， Vuser 将对该网页执行文本检查。例如，如果单词 *Specials* 出现在该页面上，检查就将通过。如果 *Specials* 由于某种原因 （例如，服务器没有返回正确的页面）没有出现，则检查将失败。

虽然在录制脚本以及单个 Vuser 执行脚本时，服务器可能显示正确的页面，但是当服务器处于许多 Vuser 带来的负载下时，可能不会返回正确的页面。服务器可能超负荷，因此可能返回无意义的或不正确的 HTML 代码。或者，在某些服务器超负荷的场合中，服务器可能返回 *500 Server Error* 页。在上述两种情况下，都可以插入检查来确定服务器是否返回了正确的页面。

---

**注意**：Web 检查将增加 Vuser 的工作，因此您可能需要对每个负载生成器运行较少的 Vuser。仅当有经验表明服务器有时返回不正确的页面时，才应该使用 Web 检查。

---

在录制 Vuser 脚本期间或之后，可以定义 Web 检查。通常，在录制时 （此时要检查的网页可见）定义检查更加方便。

在添加 Web 检查时， VuGen 会将 Web 检查图标添加到 Vuser 脚本的树视图中的当前步骤中。Web 检查图标总是在关联的步骤下缩进显示。在运行 Vuser 脚本时， VuGen 将对执行该步骤后显示的网页进行检查。

**注意：**只有在启用了检查并且脚本以 HTML 模式运行时，Vuser 才会在脚本执行期间进行 Web 检查。要启用检查，请在 “运行时设置”对话框的 “首选项”选项卡中，选择 “启用图像和文本检查”选项。详细信息，请参阅第 9 章 “配置运行时设置”。

VuGen 将使用多个不同的 Web 检查图标，每个图标代表不同的检查类型：

| Web 检查图标 | 描述 |
| --- | --- |
| 文本 | 文本检查是对网页上的指定文本字符串进行的搜索 |
| 图像 | 图像检查是对网页上的指定图像进行的搜索 |

本章描述如何使用 VuGen 在树视图中添加 Web 检查。有关在基于文本的脚本视图中向脚本中添加检查的信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

# 添加文本检查

通过 VuGen 可在网页上添加搜索文本字符串的检查。可以在录制期间或录制之后添加文本检查。

在创建文本检查时，需要定义检查的名称、检查范围、要检查的文本和搜索条件。

**要在录制之后添加文本检查，请执行下列操作：**

**1** 在 VuGen 主窗口中，右键单击与要对其执行检查的网页相应的步骤。从弹出菜单中选择 “在之后插入”。将打开 “添加步骤”对话框。

---

**注意：**在 Web 浏览器录制会话期间， VuGen 主窗口可能会最小化。要在录制期间添加文本检查，请还原 VuGen 主窗口。

---

**2** 在 “步骤类型”树中，展开 “Web 检查”。

**3** 选择 “文本检查”，然后单击 “确定”。将打开 “文本检查属性”对话框。请确保 “规格”选项卡可见。

**4** 在 “搜索”框中，键入要验证其存在与否的字符串。 ABC 图标表示尚未为 “搜索”框中的字符串分配参数。有关分配参数的详细信息，请参阅第 7 章 “定义参数”。

**5** 要相对于邻近文本指定搜索字符串的位置，请选中 “其右侧”或 “其左侧”复选框。然后，在适当的字段中键入文本。例如，要验证字符串 “support@mercuryinteractive.com”是否出现在单词 “e-mail:”的右侧，请选中 “其右侧”，然后在 “其右侧”框中键入 “e-mail:”。

ABC 图标表示尚未为 “其右侧”或 “其左侧”框中的字符串分配参数。有关分配参数的详细信息，请参阅第 7 章 “定义参数”。

**6** 命名文本检查。单击 “常规” 选项卡，然后在 “步骤名” 框中键入文本检查的名称。使用一个以后容易识别该检查的名称。

**7** 属性表显示其他用于定义检查的属性。

清除 “仅查看活动属性” 复选框可以查看活动和非活动属性。要启用某个属性，请单击该属性名左侧的单元格。在 “值” 列中为属性分配一个值。

有关分配属性值的详细信息，请参阅第 454 页的 “定义其他属性”。

**8** 单击 “确定” 接受设置。代表新文本检查的图标将被添加到脚本中的关联步骤中。在脚本视图中，“文本检查” 图标显示为 **web_find** 函数。

**要在录制期间添加文本检查，请执行下列操作：**

**1** 使用鼠标标记所需的文本。

**2** 单击录制工具栏上的 “插入文本检查” 图标。

## 使用其他文本检查方法

除了使用 web_find 函数外，还可以使用两个其他的增强函数来搜索 HTML 页内的文本：

- web_reg_find
- web_global_verification

**web_reg_find** 函数是注册类型函数。它将注册对 HTML 页上的文本字符串进行的搜索。注册意味着它不会立即执行搜索 – 仅在执行下一个操作函数 （如 **web_url**）之后，才会执行检查。注意，如果正在使用并发函数组，则 **web_reg_find** 函数仅在分组结束后才会执行。

该函数与 web_find 函数的不同之处在于：它并不局限于基于 HTML 的脚本（请参见 “录制选项” > “录制” 选项卡）。该函数还具有其他属性 （如实例）通过该属性可以确定文本出现的次数。在执行标准文本搜索时， **web_reg_find** 是首选函数。

对于 **web_reg_find**，下列属性可用：

- Text：要搜索的文本字符串。该属性必须是非空的、以 Null 结尾的字符串。该搜索机制区分大小写；要忽略大小写，请在边界后面添加 “/IC”。在文本后指定 “/BIN”（或者选中步骤属性中的 “二进制” 复选框）可指定二进制数据。使用格式 “Text= 字符串”。

可以不指定 “Text”，改而指定下列两个属性：

- TextPfx：要搜索的文本字符串的前缀。要忽略大小写，请在边界后添加 “/IC”。在文本后指定 “/BIN” 可指定二进制数据。使用格式 “TextPfx= 字符串”。
- TextSfx：要搜索的文本字符串的后缀。要忽略大小写，请在边界后添加 “/IC”。在文本后指定 “/BIN” 可指定二进制数据。使用格式 “TextSfx= 字符串”。
- Search：搜索文本的位置。可用的值包括 Headers、 BODY、 NORESOUCE 或 ALL。默认值是 BODY。使用格式 “Search= 值”。（可选）
- SaveCount：找到的匹配项的个数。要使用该属性，请指定 SaveCount=param_name，其中， param_name 是存储了以 Null 结尾的 ASCII 值的变量。（可选）

➤ Fail：找不到字符串时的处理方法。可用的值包括 Found、 NotFound 和 None。 *Found* 表示在找到文本时出现故障（如 “Error”）。 *Not Found* 表示在找不到文本时出现故障。在指定 SaveCount 属性时，默认值是 None （没有故障）。在忽略 SaveCount 属性时，默认值是 NotFound。注意，不能明确地将值 None 分配给 Fail 属性。

还可以从函数的 “属性”对话框中设置上述属性。进入树视图，选择相应的函数，然后从右键单击菜单中选择 “属性”。将打开一个对话框，可以在其中输入所有参数值。

在以下示例中， **web_reg_find** 函数将搜索文本字符串 “Welcome”。如果找不到该字符串，函数失败并且脚本执行将停止。

```
web_reg_find("Text=Welcome", "Fail=Found", LAST);
web_url("Step", "URL=...", LAST);
```

通过 **web_global_verification** 函数可搜索整个业务流程的数据。与 **web_reg_find** （它仅应用于下一个操作函数）相反，该函数将应用于*所有*后续操作函数，如 **web_url**。默认情况下，搜索的范围是 NORESOURCE，此时仅搜索 HTML 正文，而排除标题和资源。 **web_global_verification** 函数非常适合于检测未包含在 HTTP 状态代码中的应用程序级错误。该函数不局限于基于 HTML 的脚本 （请参见 “录制选项” > “录制”选项卡）。

# 添加图像检查

通过 VuGen 可添加在网页上搜索图像的用户定义的检查。图像可以由 ALT 属性、SRC 属性或这两者来标识。

可以在录制期间或录制之后添加用户定义的图像检查。录制之后，可以在脚本中编辑任何现有的图像检查。

**要添加图像检查，请执行下列操作：**

**1** 在 VuGen 主窗口中，右键单击与要对其执行检查的网页相应的步骤。从弹出菜单中选择 “在之后插入”。将打开 “添加步骤”对话框。

---

**注意：**在 Web 浏览器录制会话期间， VuGen 主窗口可能会最小化。要在录制期间添加图像检查，请还原 VuGen 主窗口。

---

**2** 在 “步骤类型”树中，展开 “Web 检查”。

**3** 选择 “图像检查”，然后单击 “确定”。将打开 “图像检查属性”对话框。请确保 “规格”选项卡可见。

**4** 选择一种标识图像的方法：

- 要使用图像的 ALT 属性来标识图像，请选中 “替换图像名 (ALT 属性 )”复选框，然后键入 ALT 属性。在运行脚本时， Vuser 将搜索具有指定的 ALT 属性的图像。
- 要使用图像的 SRC 属性来标识图像，请选中“图像服务器文件名 (SRC 属性 )”复选框，然后键入 SRC 属性。在运行脚本时， Vuser 将搜索具有指定的 SRC 属性的图像。

ABC 图标表示尚未为 ALT 或 SRC 属性分配参数。有关分配参数的详细信息，请参阅第 7 章 “定义参数”。

---

**注意**：如果将 ALT 属性和 SRC 属性复选框都选中， Vuser 将搜索同时具有指定的 ALT 属性和指定的 SRC 属性的图像。

---

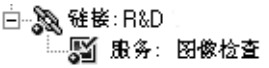
**5** 要命名图像检查，请单击“常规”选项卡。在“步骤名”框中，键入图像检查的名称。使用一个以后容易识别该检查的名称。

**6** 属性表显示其他用于定义检查的属性。

清除“仅查看活动属性”复选框可以查看活动和非活动属性。要启用某个属性，请单击该属性名左侧的单元格。在“值”列中为属性分配一个值。

有关分配属性值的详细信息，请参阅第 454 页的“定义其他属性”。

**7** 单击“确定”以接受设置。代表新图像检查的图标将被添加到 Vuser 脚本中的关联步骤中。




—— *图像检查*

# 定义其他属性

可以指定插入到 Vuser 脚本中的每个 Web 检查的其他属性。在检查属性对话框的 “常规” 选项卡上的属性表中设置其他属性。

**要设置其他属性，请执行下列操作：**

1 右键单击要编辑其属性的 Web 检查，然后从弹出菜单中选择 “属性”。将打开适当的检查属性对话框。请确保 “常规” 选项卡可见。

2 清除 “仅查看活动属性” 复选框以查看所有可用属性。

3 要启用某个属性，请单击该属性名左侧的单元格。红色的复选标记将出现在属性旁边。

4 在 “值” 列中为属性分配一个值：

- **Frame**：键入检查对象所在的帧的名称。
- **AssignToParam**：选择 “YES”，允许分配参数。选择 “NO”，禁用该功能。默认值是 NO。
- **MatchCase**：选择 “YES”，执行区分大小写的搜索。选择 “NO”，执行不区分大小写的搜索。默认值是 NO。
- **OnFailure**：选择 “Abort” 可在检查失败时中止整个 Vuser 脚本。VuGen 将中止 Vuser 脚本，而不管在运行时设置中已经设置了哪种错误处理方法。选择 “Continue” 可以让在运行时设置中定义的错误处理方法来确定检查失败时是否中止脚本。默认值是 Continue。有关定义错误处理方法的详细信息，请参阅第 9 章 “配置运行时设置”。
- **Expect**：选择 “NotFound” 表示如果 Vuser 找不到指定的检查对象，该检查成功。选择 “Found” 表示如果 Vuser 找到指定的检查对象，该检查成功。默认值是 Found 。
- **Repeat**：选择 “YES” 可搜索指定的检查对象的多个实例。选择 “NO” 可在找到指定的检查对象的一个实例后立即结束检查。Vuser 脚本将转而执行下一个步骤。当搜索可能含有检查对象的多个实例的网页时，该选项很有用。默认值是 YES。
- **Report**：选择 “Always” 可以始终在 “执行” 日志中查看检查结果的详细描述。选择 “Failure” 仅在检查失败时查看详细的检查结果。选择 “Success” 仅在检查成功时查看详细的检查结果。默认值是 Always。

ABC 图标表示尚未为该属性值分配参数。单击该图标可以分配参数。详细信息，请参阅第 7 章 “定义参数”。

# 使用常规表达式

在添加文本检查时，可以将值类型指定为 “常规表达式”。使用常规表达式可增加检查的灵活性和适应性。

在常规表达式中，将按字面意义搜索任何不属于下述特殊字符的字符。当特殊字符前面带有反斜杠 (\) 时， Vuser 将搜索字面字符。

下列选项可用于创建常规表达式：

- 匹配任一单个字符
- 匹配某个范围内的任一单个字符
- 匹配一个或多个特定的字符

## 匹配任一单个字符

问号 (?) 用于指示 VuGen 搜索任一单个字符。例如，

```
welcome?
```

与 welcomes、welcomed 或后面跟随空格或任何其他单个字符的 welcome 相匹配。一系列问号表示未指定的字符序列。序列的长度等于问号个数。

## 匹配某个范围内的任一单个字符

为了匹配某个范围内的单个字符，可以使用方括号 ([ ])。例如，要搜索日期 1968 或 1969，请写：

```
196[89]
```

可以使用连字符 (-) 来表示实际范围。例如，要匹配二十世纪六十年代的任何年，请写：

```
196[0-9]
```

如果连字符显示为括号内的第一个或最后一个字符，或者位于插字号 (^) 之后，则不表示范围。

插字号 (^) 指示 VuGen 匹配除该字符串中指定的字符之外的任何字符。例如：

```
[^A-Za-z]
```

匹配任何非字母字符。仅当插字号在括号内首次出现时，才具有这一特殊含义。

注意，在括号内，下列字符取字面意义：

| |
|---|
| 句点 (.) |
| 星号 (*) |
| 左方括号 ([) |
| 反斜杠 (\) |

如果右方括号是范围内的第一个字符，它也取字面意义。例如：

```
[]g-m]
```

匹配右方括号和 g 到 m。

### 匹配一个或多个特定的字符

星号 (*) 指示 VuGen 匹配前面字符的零个或多个实例。例如：

```
Q*
```

使 VuGen 匹配 Q、 QQ、 QQQ 等等。

# 40

# 修改 Web 和无线 Vuser 脚本

录制 Web 或无线 Vuser 脚本之后，可使用 VuGen 修改录制的脚本。您可以添加新步骤以及编辑、重命名和删除现有步骤。

本章描述：

- 向 Vuser 脚本中添加步骤
- 从 Vuser 脚本中删除步骤
- 修改操作步骤
- 修改控制步骤
- 修改服务步骤
- 修改 Web 检查 （仅限于 Web）

***下列信息适用于 Web 和无线 Vuser 脚本。***

## 关于修改 Web 和无线 Vuser 脚本

录制浏览器或工具包会话后，可以通过编辑步骤的属性或者添加和删除步骤，在 VuGen 中修改录制的脚本。

可以在基于图标的树视图或基于文本的脚本视图中进行修改。有关这两种查看模式的详细信息，请参阅第 32 章 “创建 Web Vuser 脚本”。

本章描述如何使用 VuGen 在树视图中修改脚本。有关在基于文本的脚本视图中修改脚本的信息，请参阅 《Online Function Reference》。（“帮助” > “函数参考”）

### 添加二进制数据

要将二进制编码数据包含在 HTTP 请求的主体中，请使用以下格式：

\x[char1][char2]

这表示 [char1][char2] 的十六进制值。

例如， \x24 是 16*2+4=36，这是一个 $ 号，而 \x2B 是一个 + 号。

请勿使用单字符十六进制序列。例如， \x2 不是有效的序列，而 \x02 是有效序列。

## 向 Vuser 脚本中添加步骤

除了 VuGen 在浏览器或工具包录制会话期间录制的步骤以外，还可以向录制的脚本中添加步骤。

**要向 Vuser 脚本中添加步骤，请执行下列操作：**

1. 在脚本的树视图中，选择要在其前面或后面添加新步骤的步骤。
2. 选择 “插入” > “新建步骤”，在选定的步骤之后插入一个步骤，或者从右键单击菜单中选择 “在之后插入” 或 “在之前插入”。将打开 “添加步骤” 对话框。

3 从 “步骤类型”树或从 “查找函数”列表中选择要添加的步骤类型。

4 单击 “确定”。将另外打开一个对话框，提示您输入有关要添加的步骤的信息。该对话框根据要添加的步骤的类型而不同。

有关使用这些对话框的详细信息，请参阅适当的部分，如下所列：

| 要添加… | 请参阅… |
|---|---|
| LoadRunner 函数 | 第 6 章 “增强 Vuser 脚本” |
| 服务步骤 | 第 476 页的 “修改服务步骤” |
| Web 检查 | 第 477 页的 “修改 Web 检查 （仅限于 Web）” |
| 事务 | 第 473 页的 “修改事务” |
| 集合点 | 第 475 页的 “修改集合点” |
| 思考时间步骤 | 第 475 页的 “修改思考时间” |
| URL 步骤 | 第 460 页的 “修改 URL 步骤” |
| 链接步骤 | 第 462 页的 “修改超文本链接步骤 （仅限于 Web）” |
| 图像步骤 | 第 463 页的 “修改图像步骤 （仅限于 Web）” |
| 提交表单步骤 | 第 465 页的 “修改提交表单步骤 （仅限于 Web）” |
| 提交数据步骤 | 第 468 页的 “修改提交数据步骤” |
| 自定义请求步骤 | 第 471 页的 “修改自定义请求步骤” |
| 用户定义的步骤 | 第 6 章 “增强 Vuser 脚本” |

## 从 Vuser 脚本中删除步骤

录制浏览器或工具包会话之后，可以使用 VuGen 从 Vuser 脚本中删除任何步骤。

**要从 Vuser 脚本中删除步骤，请执行下列操作：**

1 在 Vuser 脚本的树视图中，右键单击要删除的步骤，然后从弹出菜单中选择 “删除”。

2 单击 “确定”，确认要删除该步骤。

该步骤将从脚本中删除。

# 修改操作步骤

操作步骤表示录制期间的用户操作，也就是到新 URL 的跳转或 Web 上下文中的更改。

在录制期间，操作步骤 （在 Vuser 脚本的树视图中由 “操作” 图标表示）被自动添加到脚本中。录制之后，可以修改所录制的操作步骤。

本部分包括：

- 修改 URL 步骤
- 修改超文本链接步骤 （仅限于 Web）
- 修改图像步骤 （仅限于 Web）
- 修改提交表单步骤 （仅限于 Web）
- 修改提交数据步骤
- 修改自定义请求步骤

## 修改 URL 步骤

在键入 URL 或者使用书签访问特定的网页时，会将 URL 步骤添加到 Vuser 脚本中。

可以修改的属性包括步骤的名称、 URL 的地址、目标帧和录制模式。

默认情况下， VuGen 基于录制它所用的下列模式运行 URL 步骤：*HTML* 或 *HTTP* （没有资源）。有关录制模式的信息，请参阅第 411 页的 “选择录制级别”。

### 设置回放模式

在 URL 步骤的 “属性” 对话框中，可以修改模式设置，指示 LoadRunner 用录制的模式以外的模式执行脚本。要自定义回放模式，请选中 “录制模式” 复选框。可用的回放模式包括：

**HTML:** 自动下载所有资源和图像，并且存储随后步骤所需的 HTTP 信息。这非常适用于含有 Web 链接的脚本。

**HTTP:** 在回放期间，不下载该步骤的任何资源。仅下载由函数明确表示的资源。

还可以指明某个特定步骤不应该计为资源。例如，如果有一个表示要跳过的特定图像的步骤，可以指示 VuGen 排除该资源类型。详细信息，请参阅第 418 页的 “资源处理”。

**要修改 URL 步骤的属性，请执行下列操作：**

**1** 在 Vuser 脚本的树视图中，选择要编辑的 URL 步骤。URL 步骤将使用 URL 图标显示。

**2** 单击 VuGen 工具栏上的 “属性”按钮。将打开 “URL 步骤属性”对话框。

**3** 要更改步骤名，请在 “步骤名”框中键入一个新名称。录制期间的默认名称是 URL 的最后一部分。

**4** 在 “URL”框中，键入 URL 步骤所访问的网页的 Web 地址 (URL)。“ABC”图标表示尚未为该 URL 分配参数。有关分配参数的详细信息，请参阅第 7 章 “定义参数”。

**5** 在 “目标帧”列表中，选择下列值之一：

**_TOP**：替换整页

**_BLANK**：打开新窗口

**_PARENT**：替换最后一个 （更改的）帧的父项

**6** 要自定义回放模式，请选中 “录制模式”复选框。

选择需要的模式：HTML 或 HTTP。

**7** 要拒绝将某个项作为资源下载，请清除 “资源”复选框。

**8** 单击 “确定”关闭 “URL 步骤属性”对话框。

## 修改超文本链接步骤 （仅限于 Web）

单击超文本链接时，超文本链接步骤将被添加到 Web Vuser 脚本中。仅当选择相应的选项以便用 “基于 HTML 的脚本”模式进行录制时，才录制该步骤。详细信息，请参阅第 36 章 “为 Web Vuser 设置录制选项”。

可以修改的属性包括步骤的名称、标识超文本链接的方式以及超文本链接的位置。

**要修改超文本链接步骤的属性，请执行下列操作：**

**1** 在 Vuser 脚本的树视图中，选择要编辑的超文本链接步骤。超文本链接步骤将使用 “超文本链接”图标显示。

**2** 从右键单击菜单中选择 “属性”。将打开 “链接步骤属性”对话框。

3 要更改步骤名，请在 “步骤名” 框中键入一个新名称。录制期间的默认名称是超文本链接的文本字符串。

4 属性表显示用于标识链接的属性。

清除 “仅查看活动属性” 复选框查看活动和非活动属性。要启用某个属性，请单击该属性名左侧的单元格。在 “值” 列中为该属性分配一个值：

- **文本**：超文本链接的准确字符串
- **帧**：链接所在帧的名称
- **TargetFrame**：目标帧：

  **_TOP**：替换整页

  **_BLANK**：打开新窗口

  **_PARENT**：替换最后一个 （更改的）帧的父项

- **序号**：当所有其他属性都与网页上的一个或多个其他链接相同时，唯一标识该链接的数字。详细信息，请参阅 《Online Function Reference》。

“ABC” 图标表示尚未为该链接属性值分配参数。有关分配参数的详细信息，请参阅第 7 章 “定义参数”。

5 单击 “确定” 关闭 “链接步骤属性” 对话框。

### 修改图像步骤 （仅限于 Web）

在单击超图形链接时，图像步骤将被添加到 Vuser 脚本中。仅当您选择相应的选项以便用 HTML （上下文相关）模式进行录制时，才会录制该步骤。详细信息，请参阅第 36 章 “为 Web Vuser 设置录制选项”。

可以修改的属性包括步骤的名称、标识超图形链接的方式以及超图形链接的位置。

要修改图像步骤的属性，请执行下列操作：

1 在 Vuser 脚本的树视图中，选择要编辑的图像步骤。图像步骤将使用 “图像”图标显示。

2 从右键单击菜单中选择 “属性”。将打开 “图像步骤属性”对话框。

3 要更改步骤名，请在 “步骤名”框中键入一个新名称。录制期间的默认名称是图像的 ALT 属性。如果图像没有 ALT 属性，那么 SRC 属性的最后一部分将用作默认名称。

4 属性表将显示用于标识链接的属性。

清除 “仅查看活动属性”复选框查看活动和非活动属性。要启用某个属性，请单击该属性名左侧的单元格。在 “值”列中为属性分配一个值：

- **ALT**：图像的 ALT 属性
- **SRC**：图像的 SRC 属性
- **MapName**：与图像有关的映射的名称。仅适用于客户端图像映射。
- **AreaAlt**：要单击的区域的 ALT 属性。仅适用于客户端图像映射。

- **AreaOrdinal**：要单击的区域的序列号。仅适用于客户端图像映射。
- **帧 :** 图像所在帧的名称
- **TargetFrame**：目标帧：

  **_TOP**：替换整页

  **_BLANK**：打开新窗口

  **_PARENT**：替换最后一个 （更改的）帧的父项

  **_SELF**：替换最后一个 （更改的）帧

- **序号**：当所有其他属性都与网页上的一个或多个其他图像相同时，唯一标识该图像的数字。详细信息，请参阅 《Online Function Reference》。
- **XCoord, YCoord**：在图像上单击鼠标时所在位置的坐标

“ABC”图标表示尚未为该链接属性值分配参数。有关分配参数的详细信息，请参阅第 7 章 “定义参数”。

**5** 单击 “确定”，关闭 “图像步骤属性”对话框。

### 修改提交表单步骤 （仅限于 Web）

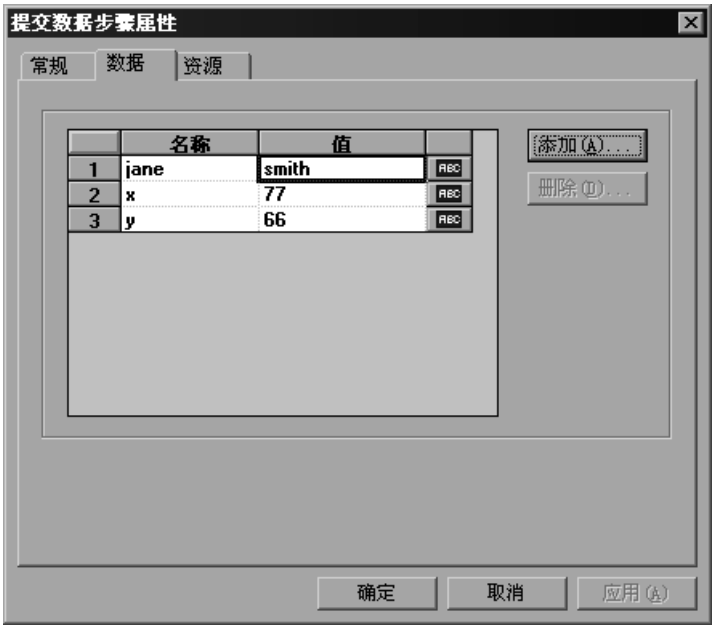
在提交表单时，提交表单步骤将被添加到 Vuser 脚本中。仅当您选择相应的选项以便用 HTML （上下文相关）模式进行录制时，才会录制该步骤。详细信息，请参阅第 36 章 “为 Web Vuser 设置录制选项”。

可以修改的属性包括步骤的名称、表单位置、标识表单提交的方式和表单数据。

要修改提交表单步骤的属性，请执行下列操作：

1 在 Vuser 脚本的树视图中，选择要编辑的提交表单步骤。提交表单步骤将使用“提交表单”图标显示。

2 从右键单击菜单中选择 “属性”。将打开 “提交表单步骤属性”对话框。请确保已经选择了 “数据”选项卡。




- “名称”列将列出表单上的所有数据参数。
- “值”列显示数据参数的相应值输入。
- 类型列包含一个图标。最初，所有值都是常数或非参数化的值，并且具有一个“ABC”图标。如果给数据值分配参数，如第 7 章 “定义参数”中所述，“ABC”图标将更改为表格图标。

3 要编辑数据参数，请双击它激活单元格内的光标。然后键入新值。

**4** 要向表单提交中添加新的数据参数，请单击 “添加”。将打开 “添加数据”对话框。

**5** 键入数据参数的 “名称”和 “值”，然后单击 “确定”。

**6** 要删除参数，请选择它并单击 “删除”。

**7** 要更改提交表单步骤的名称，请单击 “常规”选项卡。

**8** 要更改步骤名，请在 “步骤名”框中键入一个新名称。录制期间的默认名称是用于处理表单的可执行程序的名称。

**9** 属性表将显示用于标识表单提交的属性。

清除 “仅查看活动属性” 选项查看活动和非活动属性。要启用某个属性，请单击该属性名左侧的单元格。在 “值” 列中为属性分配一个值：

- **操作**：执行表单的操作所用的地址
- **帧 :** 表单提交所在帧的名称
- **TargetFrame**：目标帧：

  **_TOP**：替换整页

  **_BLANK**：打开新窗口

  **_PARENT**：替换最后一个 （更改的）帧的父项

  **_SELF**：替换最后一个 （更改的）帧

- 序号 : 当所有其他属性都与同一网页上的一个或多个其他表单相同时，唯一标识该表单的数字。详细信息，请参阅 《Online Function Reference》。（“帮助” > “函数参考”）

“ABC” 图标表示尚未为该提交表单步骤属性值分配参数。有关分配参数的详细信息，请参阅第 7 章 “定义参数”。

**10** 单击 “确定”，关闭 “提交表单步骤属性” 对话框。

### 修改提交数据步骤

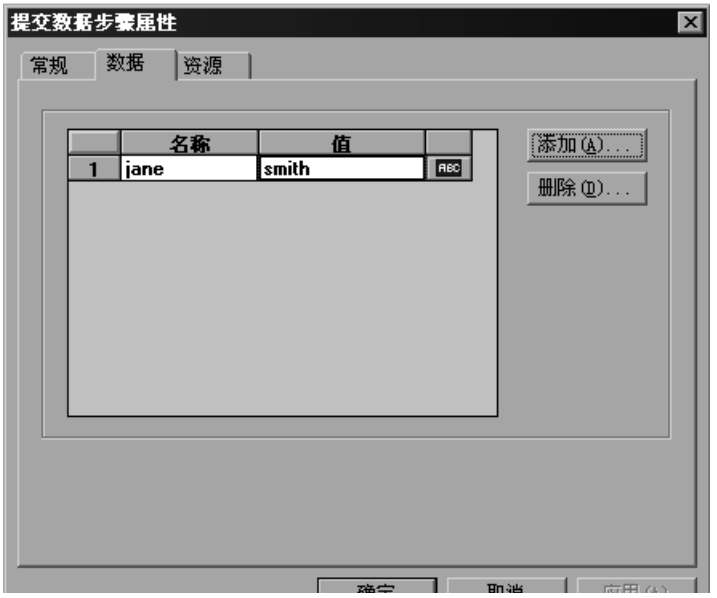
提交数据步骤表示将数据表单提交到网站以便进行处理。这与提交表单步骤不同，因为不需要让表单上下文执行该请求。

可以修改的属性包括步骤的名称、方法、操作、目标帧和表单上的数据项。

**要修改提交数据步骤的属性，请执行下列操作：**

**1** 在 Vuser 脚本的树视图中，选择要编辑的提交数据步骤。提交数据步骤将使用 “提交数据” 图标显示。

**2** 从右键单击菜单中选择 “属性”。将打开 “提交数据步骤属性”对话框。请确保 “数据”选项卡可见。




- “名称”列将列出表单上的所有数据参数。这包括所有隐藏的字段。
- “值”列显示数据参数的相应值输入。
- 类型列包含一个图标。最初，所有值都是常数或非参数化的值，并且具有一个 “ABC”图标。如果给数据值分配参数，如第 7 章 “定义参数”中所述，“ABC”图标将更改为表格图标。

**3** 要编辑数据参数，请双击它激活单元格内的光标。然后键入新值。

**4** 要添加新数据，请单击 “添加”。将打开 “添加数据”对话框。

5 键入数据参数的 “名称” 和 “值”，然后单击 “确定”。

6 要删除参数，请选择它并单击 “删除”。

7 要更改提交数据步骤的名称，请单击 “常规” 选项卡。

8 要更改步骤名，请在 “步骤名” 框中键入一个新名称。

9 在 “方法” 下，单击 “POST” 或 “GET”。默认方法是 “POST”。

10 在 “操作” 框中，键入执行数据提交操作所要使用的地址。“ABC” 图标表示尚未为该操作分配参数。有关分配参数的详细信息，请参阅第 7 章 “定义参数”。

11 从列表中选择一个 “目标帧”：

**_TOP**：替换整页

**_BLANK**：打开新窗口

**_PARENT**：替换最后一个 （更改的）帧的父项

**_SELF**：替换最后一个 （更改的）帧

12 要自定义回放模式，请选择 “录制模式” 选项。选择需要的模式：HTML 或 HTTP。详细信息，请参阅第 460 页的 “设置回放模式”。

13 要拒绝将某个项作为资源下载，请清除 “资源”复选框。

14 要指定编码类型，如 *multipart/www-urlencoded*，请选中 “编码类型”复选框并指定编码方法。

15 要编码 URL 中的 “@”，请选择 “编码 @ 符号”。

16 单击 “确定”关闭 “提交数据步骤属性”对话框。

### 修改自定义请求步骤

自定义请求表示使用 HTTP 所支持的任何方法进行的针对 URL 的自定义 HTTP 请求。自定义请求步骤是没有上下文的。

可以修改的属性包括步骤的名称、方法、 URL、目标帧和主体。

VuGen 具有一项功能，可用来将自定义请求主体字符串转换成 C 格式字符串。例如，如果将 XML 树或大量数据复制到自定义请求的 “主体”区域中，可以容易地将字符串转换成 C 格式，以便可以将它合并到当前函数中。它将插入必需的转义序列字符，并且删除字符串中的换行符。

要修改自定义请求步骤的属性，请执行下列操作：

1 在 Vuser 脚本的树视图中，选择要编辑的自定义请求步骤。自定义请求步骤将使用 “自定义请求”图标显示。

2 从右键单击菜单中选择 “属性”。将打开 “自定义请求属性”对话框。

3 要更改步骤名，请在 “步骤名”框中键入一个新名称。录制期间的默认名称是 URL 的最后一部分。

4 在 “方法”框中，键入 HTTP 所支持的任何方法。例如， GET、 POST 或 HEAD。

5 在 “URL”框中，键入所请求的 URL。

6 从列表中选择一个 “目标帧”：

**_TOP**：替换整页

**_BLANK**：打开新窗口

**_PARENT**：替换最后一个 （更改的）帧的父项

**_SELF**：替换最后一个 （更改的）帧

7 在 “主体”框中，键入请求的主体或者粘贴所需的文本。如果选中 “二进制数据”复选框，文本将被视为二进制数据而不是 ASCII 数据。有关使用二进制数据的详细信息，请参阅 《Online Function Reference》。（“帮助” > “函数参考”）

8 对于粘贴到 “主体”框中的字符串，请选中文本并从右键单击菜单中选择 “转换为 C 格式”。

9 要自定义回放模式，请选择 “录制模式”选项。选择需要的模式：HTML 或 HTTP。详细信息，请参阅第 460 页的 “设置回放模式”。

10 要拒绝将某个项作为资源下载，请清除 “资源”选项。

11 要指定编码类型，如 *multipart/www-urlencoded*，请选择 “编码类型”并指定编码方法。

12 单击 “确定”关闭 “自定义请求属性”对话框。

## 修改控制步骤

控制步骤表示在负载测试期间使用的控制。控制步骤包括事务、集合点和思考时间。

在录制期间或录制之后，将控制步骤 （在 Vuser 脚本的树视图中由 “控制”图标表示）添加到脚本中。

本部分包括：

- 修改事务
- 修改集合点
- 修改思考时间

### 修改事务

事务是要度量其服务器响应时间的任务或操作集。

可以修改的属性包括事务的名称 （开始事务和结束事务）及其状态 （仅限于结束事务）。

**要修改开始事务控制步骤，请执行下列操作：**

1 在 Vuser 脚本的树视图中，选择要编辑的开始事务控制步骤。开始事务控制步骤将使用 “开始事务”图标显示。

2 从右键单击菜单中选择 “属性”。将打开 “开始事务”对话框。

3 要更改事务名，请在 “事务名”框中键入一个新名称，然后单击 “确定”。

**要修改结束事务控制步骤，请执行下列操作：**

1 在 Vuser 脚本的树视图中，选择要编辑的结束事务控制步骤。结束事务控制步骤将使用 “结束事务”图标显示。

2 从右键单击菜单中选择 “属性”。将打开 “结束事务”对话框。

3 从 “事务名”列表中选择要结束的事务的名称。

4 从 “事务状态”列表中选择事务状态：

**LR_PASS**：返回 “succeed”返回代码

**LR_FAIL**：返回 “fail”返回代码

**LR_STOP**：返回 “stop”返回代码

**LR_AUTO**：自动返回检测到的状态

详细信息，请参阅 《Online Function Reference》。（“帮助” > “函数参考”）

5 单击 “确定”关闭 “结束事务”对话框。

### 修改集合点

通过集合点可以同步 Vuser 以便恰好在同一时刻执行任务。

可以修改的属性为集合点的名称。

**要修改集合点，请执行下列操作：**

1 在 Vuser 脚本的树视图中，选择要编辑的集合点。集合点将使用 “集合”图标显示。

2 从右键单击菜单中选择 “属性”。将打开 “集合”对话框。

3 要更改集合名，请在 “集合名”框中键入一个新名称，然后单击 “确定”。

### 修改思考时间

思考时间可模拟实际用户在操作之间等待的时间。在录制期间， VuGen 会在每个用户操作之后自动将思考时间添加到 Vuser 脚本中，前提是该操作和后面的操作之间的时间超过预定义的阈值 （大约为四秒）。

可以修改的属性是思考时间 （以秒为单位）。

**要修改思考时间，请执行下列操作：**

1 在 Vuser 脚本的树视图中，选择要编辑的思考时间步骤。思考时间步骤将使用 “思考时间”图标显示。

2 从右键单击菜单中选择 “属性”。将打开 “思考时间”对话框。

3 在 “思考时间” 框中键入思考时间，然后单击 “确定”。

---

注意：在 VuGen 中或从 Controller 运行 Web Vuser 脚本时，可以指示 Vuser 按录制参数回放思考时间或者忽略录制的思考时间。详细信息，请参阅第 9 章 “配置运行时设置”。

---

## 修改服务步骤

服务步骤是执行自定义任务 （如设置代理服务器、提交授权信息和发出自定义标题）的函数。服务步骤不会对网站上下文进行任何更改。

在录制期间或录制之后，可以向脚本中添加服务步骤。

**要修改服务步骤的属性，请执行下列操作：**

1 在 Vuser 脚本的树视图中，选择要编辑的服务步骤。服务步骤将使用 “服务” 图标显示。

2 从右键单击菜单中选择 “属性”。将打开适当的 “服务步骤属性” 对话框。该对话框根据要修改的服务步骤的类型而不同。服务步骤的描述将显示在对话框的标题栏中。

---

注意：一些服务步骤函数没有参数。在这些情况下，将禁用 “属性” 菜单项。

---

3 键入或选择服务步骤所需的参数。有关每个函数的详细信息，请参阅 《Online Function Reference》。（“帮助” > “函数参考”）

4 单击 “确定”，关闭 “服务步骤属性” 对话框。

# 修改 Web 检查 （仅限于 Web）

Web 检查是一个函数，用于验证网页上是否存在特定的对象。对象可以是文本字符串或图像。

在录制期间和录制之后，可以向脚本中添加 Web 检查。

**要修改 Web 检查的属性，请执行下列操作：**

1 在 Vuser 脚本的树视图中，选择要编辑的 Web 检查。 Web 检查将使用 “Web 检查”图标显示。

—— *“图像检查”图标*

—— *“文本检查”图标*

2 从右键单击菜单中选择 “属性”。将打开适当的 “Web 检查属性”对话框。该对话框根据要修改的检查的类型而不同。

3 键入或选择检查所需的属性。详细信息，请参阅第 39 章 “在负载下验证网页”。

4 单击 “确定”关闭 “检查属性”对话框。

# 41

# 配置 Web Vuser 脚本的关联规则

通过 VuGen 的关联功能，可以通过将一个语句的结果用作另一语句的输入来链接 Vuser 函数。

本章描述录制期间关联语句的方法。其中讨论了：

- 了解关联方法
- 选择关联处理方法
- 测试规则
- 设置关联录制选项

***下列信息适用于 Web Vuser 脚本。***

# 关于关联语句

HTML 页中通常包含动态数据，每访问一次网站，这些数据就会随之进行改变。例如，某些 Web 服务器使用由当前日期和时间构成的链接。

录制 Web Vuser 脚本时，可能会将动态数据录制到脚本中。脚本将尝试把这些录制变量提供给 Web 服务器，但它们不再有效。 Web 服务器将拒绝这些变量，并发出错误信息。这些错误并不总是很明显，您可能只有通过仔细检查 Vuser 日志文件才能检测到它们。

如果在运行 Vuser 时遇到了错误，请在发生错误的位置检查脚本。大多数情况下，通过将一个语句的结果用作另一语句的输入的关联功能就可以解决该问题。

HTML 页中的动态数据可以是下列形式：

- 每次访问相关网页时都会随之改变的 URL
- 表单提交期间记录的域 （有时为隐藏）
- javascript cookie

### 事例 1

假设网页中包含一个超文本链接，其文本内容为：“Buy me now!”用 HTTP 数据录制脚本时， VuGen 将把该 URL 录制为：

"http://host//cgi-bin/purchase.cgi?date=170397&ID=1234"

由于 “*170397*”和 ID “*1234*”是在录制期间动态创建的，所以每个新浏览器会话都将重新创建日期和 ID。运行脚本时，链接 “Buy me now!”将不再与录制的同一 URL 关联，而是与一个新建的 URL 关联。因此 Web 服务器将无法检索该 URL。

### 事例 2

假定一个事例，用户将自己的姓名和帐户 ID 填写到表单中，然后提交表单。

提交表单时，会有一个唯一的序列号连同用户的数据一起发送到服务器。虽然此序列号包含在 HTML 代码的隐藏域中，但是 VuGen 仍会将其录制到脚本中。因为该序列号会随每个浏览器会话而改变，所以 LoadRunner 将无法成功地回放录制的脚本。

可以使用关联的语句解决上述两个事例中的困难。用一个或多个参数替换录制的脚本中的动态数据。脚本运行时， LoadRunner 将分配参数值。

# 了解关联方法

本章讨论使用内置规则或用户定义的规则的自动关联。要手动关联语句，或者要执行无线 Vuser 脚本的关联，请参阅第 502 页的 “执行手动关联”。

录制浏览器会话时，应首先尝试在 HTML 模式中录制。此模式可以降低关联的需求。有关各种录制模式的详细信息，请参阅第 411 页的 “选择录制级别”。

可以指示 LoadRunner 在录制期间或录制之后关联脚本中的语句。本章中描述的录制时解决方案可以在录制期间自动关联脚本中的语句。还可以使用 VuGen 的快照关联在录制之后关联脚本。有关在录制之后进行关联的详细信息，请参阅第 42 章 “录制后关联 Vuser 脚本”。

# 使用 VuGen 的关联规则

通过 VuGen 的关联引擎，可以使用下列机制之一在录制会话期间自动关联动态数据：

- 内置关联
- 用户定义规则关联

其他信息，请参阅第 484 页的 “添加匹配条件”和第 485 页的 “高级关联规则”。

### 内置关联

内置关联检测并关联支持的应用程序服务器的动态数据。大多数服务器都具有清晰的语法规则 （即上下文），创建链接和参考时将使用这些语法规则。例如，BroadVision 服务器创建的会话 ID 始终放置在相同的分隔符之间：“BV_SessionID=”位于左侧，而 “&”位于右侧。

```
BV_SessionID=@@@@1303778278.0969956817@@@@&
```

如果用支持的应用程序服务器录制会话，可以使用 VuGen 内置的现有规则之一。应用程序服务器可能具有多个规则。可以通过选中或清除规则旁的复选框来启用或禁用特定的规则。 VuGen 在右窗格中显示规则的定义。

如果是在不支持的应用程序服务器 （其上下文未知）上录制会话，并且不能确定任何关联规则，则可以使用 VuGen 的快照比较方法。结束录制之后，此方法将指导您完成整个关联过程。详细信息，请参阅第 42 章 “录制后关联 Vuser 脚本”。

### 用户定义规则关联

如果您的应用程序具有唯一的规则，并且可以很明确地确定这些规则，则您可以使用录制选项定义新规则。

用户定义的规则关联要求在录制会话之前定义关联规则。可以在 “录制选项”对话框中创建关联规则。规则中包含一些信息 （例如要关联的动态数据的边界）以及有关匹配的其他规格 （例如二进制、大小写区分和实例编号）。

您可以指示 VuGen 搜索条件的位置：

- 全部正文文本
- 链接 / 表单操作
- Cookie 标头
- 表单域值
- 插入 Cookie 函数

### 全部正文文本

“在所有正文文本中搜索参数”选项指示录制器在整个正文中，而不只是在链接、表单操作或 cookie 中搜索匹配值。这样，会使用指定的边界搜索文本。

### 链接 / 表单操作

“在链接和表单操作中搜索参数”方法指示 VuGen 在链接和表单类型操作范围内搜索要参数化的文本。此方法适用于您已了解其上下文规则的应用程序服务器。您可以在当前链接范围内定义左边界、右边界、可选右边界和左边界 （出现）的实例。

例如，假设需要用参数替换介于第二次出现的字符串 “sessionid=”与 “@”之间的任何文本。可在 “左边界”框中指定 “sessionid=”作为左边界，在 “右边界”框中指定 “@”作为右边界。由于要查找第二次出现，所以请在 “实例”框中选择 “第二次”。

如果右边界不一致，可以在 “可选右边界”框中指定一个可选的右边界。仅在不能唯一地确定指定的右边界时使用此值。

例如，假设网页中包含下列格式的链接：

"SessionID=122@page.htm"
"Page.htm@SessionID=122&test.htm"

指定单个右边界是不够的，因为右边界不一致 – 有时是 “@”，而有时是 “&”。这种情况下，请指定 “&”作为可选右边界。

左边界和右边界应唯一地标识字符串。请不要在边界内包含动态数据。还可以指定 “字符串结尾”或 “换行符”作为右边界，它们是下拉列表选项。如果找不到任何右边界，则自左边界至源字符串结尾处的文本将保存到参数。

注意，对于此选项，左边界和右边界都必须出现在脚本中显示的字符串中 – 由服务器返回这些边界是不够的。此限制不适用于其他操作类型。

### Cookie 标头

“从 Cookie 标头搜索参数”方法与前一种规则类似，除了它是从 cookie 文本（与录制日志中显示的完全相同）而不是从链接或表单操作中提取值。

此外，链接 / 表单操作规则仅对与规则边界相匹配的 URL 部分进行参数化。Cookie 规则在链接和操作表单域查找提取的值，并用参数替换该值，而不需要指定任何边界。

### 表单域值

“参数化表单域值”方法指示录制器将指定的表单域保存到参数。该方法将在表单的操作步骤之前创建一个参数，并将其放置在脚本内。对于此选项，需要指定域名。

### 插入 Cookie 函数

如果在缓冲区中检测到特定的字符串，“插入 web_reg_add_cookie 函数”方法将插入 **web_reg_add_cookie** 函数。该方法只为具有指定前缀的 cookie 添加函数。对于此选项，需要指定要搜索的文本，以及 cookie 前缀。

## 添加匹配条件

除了上述规则，还可以通过为字符串指定下列各项内容，进一步为关联定义匹配类型：

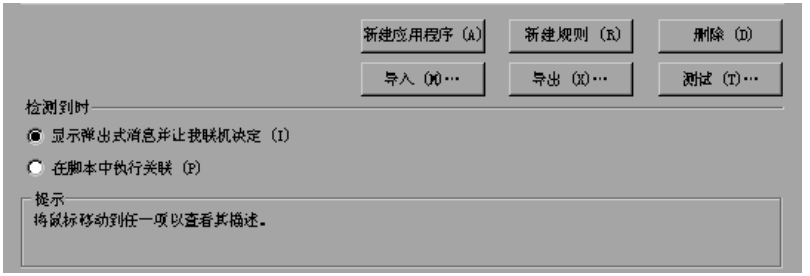
**参数前缀**：在所有基于此规则自动生成的参数中使用前缀。使用前缀可以避免覆盖现有的用户参数。此外，通过前缀还可以更容易地识别脚本中的参数。例如，在 Siebel-Web 中，有一个内置规则用于搜索 Siebel_row_id 前缀。

**区分大小写**：查找边界时区分大小写。

**使用 “#”代表任何数字**：用哈希符号替换所有数字。这将允许您查找除数值数字以外其余部分全部匹配的文本字符串。例如，如果左边界为 *Mercury193*，它将与 *Mercury284* 相匹配。所以可以在左边界框中，指定 *Mercury###*。

### 选择关联处理方法

启用关联时， VuGen 将查看是否存在适用于您的应用程序的规则。如果动态数据符合现有规则， VuGen 将根据对话框的 “如果检测到”部分中的下列设置准备执行关联：



- **显示弹出式消息并让我联机决定**：执行关联之前，在录制期间当检测到动态数据时发布消息。
- **在脚本中执行关联**：自动关联脚本内的语句。

通过 VuGen 还可以指定高级关联规则。

### 高级关联规则

通过 VuGen 可以指定下列高级关联规则：

**始终新建参数**：即使用参数替换的值在上一实例中并未更改，也要为此规则新建参数。如果 Web 服务器为每个网页指定了不同的值，则应设置此选项。例如，NetDynamics 服务器可能会逐页地更改会话 ID，以便将欺骗行为减至最低。

**仅在完全匹配时替换为参数**：仅当边界之间的文本与找到的值 （从第一个快照中）完全匹配时，才用参数替换录制的值。如果在该字符串之前或之后存在其他字符，则不会替换参数。例如，在表单提交过程中， VuGen 录制了边界 *aaa* 与 *bbb* 之间的字符 1234。 **web_submit_data** 的 Name 参数为 Name=aaa1234bbb。在此表单的后续提交过程中，如果 VuGen 找到了字符 *1234*（即 Name=1234），则仅用参数替换录制的值。如果输入了另一个值，则即使包含第一个字符串 （例如， Name=12345）， VuGen 也不会用参数替换该值，而是使用值 12345 来代替。

**反向搜索**：从字符串的结尾往回搜索左边界。

**左边界实例**：字符串 （不是正文）内出现左边界，导致该字符串被视为匹配值的次数。

**偏移量**：找到的值 （将保存到参数）的子字符串的偏移量。默认值为匹配的字符串的开头。注意，必须指定非负值。

**长度**：从偏移量开始，匹配的字符串 （将保存到参数）的子字符串的长度。如果禁用此选项，将使用默认值，并保存自指定的偏移量到匹配值结尾的字符串。

**可选右边界**：右边界的可选条件 （如果找不到以前指定的边界）。可以指定文本、字符串结尾或换行符。

# 测试规则

此部分适用于为具有已知上下文的服务器创建的用户定义规则。在 “关联规则” 对话框中定义新的规则之后，可以在录制会话之前通过将该规则应用于示例字符串来对其进行测试。可以在 “标记替换测试板” 中测试规则。要使用测试板，请执行下列操作：

1 从左窗格中选择规则，然后单击 “测试”。将打开 “标记替换测试板” 对话框。

2 在 “要替换的源字符串” 框中输入文本。

3 单击 “测试”。

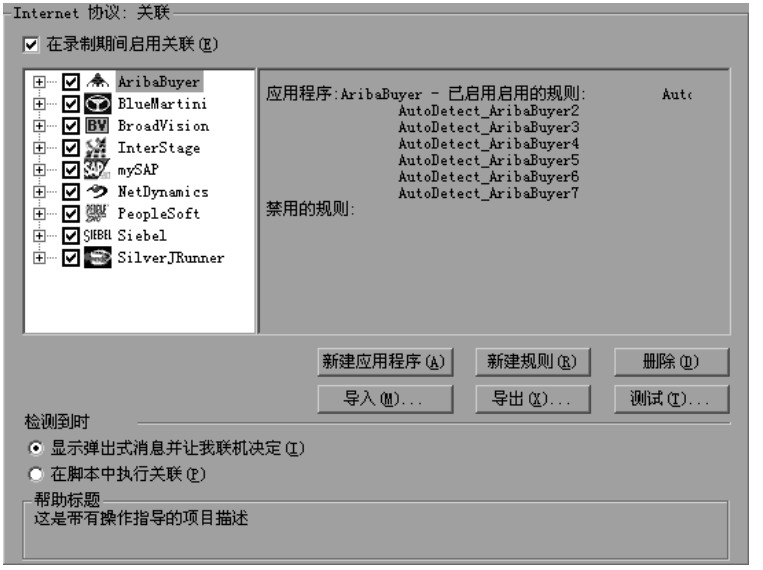
如果发生了替换，可以在 “替换结果” 框中看到已参数化的源文本，以及在 “适用规则” 框中看到适用的规则列表。

# 设置关联录制选项

要指示 LoadRunner 在录制期间关联语句，可以设置 “关联” 录制选项。可以在打开 Web Vuser 脚本之后但开始录制会话之前设置这些选项。

**要设置关联录制选项，请执行下列操作：**

**1** 在创建脚本之后但开始录制之前，选择 “工具” > “录制选项”，然后在 “录制选项” 树中选择 “Internet 协议：关联” 节点。




**2** 选择 “录制期间启用关联” 选项。

**3** 指出要应用关联规则的服务器。选中服务器名旁的复选框，为该服务器启用这些规则。要在服务器组内启用特定规则，请单击加号展开树，然后选择所需的规则。

**4** 要向现有服务器中添加新规则，请选择现有项之一，然后单击 “新建规则”。在右窗格中设置规则的属性。详细信息，请参阅第 488 页的 “设置关联规则”。

**5** 要为新应用程序添加一组规则，请单击 “新建应用程序”。再单击 “新建规则”，为该应用程序创建规则。

**6** 要修改现有规则的属性，请在左窗格中选择规则，然后在右窗格中对它们进行修改。

7 指出当 VuGen 检测到需要关联的值时应进行的操作：“显示弹出式消息”或“在脚本中执行关联”。默认情况下， LoadRunner 会显示弹出式消息。

8 要删除应用程序或规则，请选中它，然后单击 “删除”。删除选中的应用程序或规则之前， VuGen 会发出警告进行提示。

9 要导出一组关联规则，请单击 “导出”，然后将 *cor* 文件保存到所需的位置。要导入在先前的会话期间创建的一组关联规则，请单击 “导入”，然后从相应的位置打开文件。

10 单击 “确定”。

### 设置关联规则

可以使用关联选项添加、修改或删除规则。注意，还可以编辑为应用程序服务器环境自动创建的规则。

**要定义关联规则，请执行下列操作：**

1 单击现有规则，或者在左窗格中单击 “新建规则”。将在右窗格中显示 “关联规则”。

2 选择操作类型：链接或表单操作、 cookie、全部正文或表单域。

3 对于前三种类型，请在 “左边界”和 “右边界”框中指定数据的边界。

4 对于表单类型的操作，请指定域名。

**5** 选择需要的选项：“区分大小写”和 / 或“参数前缀”。指定参数前缀。要将所有数字转换为哈希符号 (#)，请选择“使用 # 代表任何数字”。

**6** 要设置高级规则，请单击“高级”。将打开“高级关联属性”对话框。

- 选择“始终新建参数”，则即使用参数替换的值在上一实例中并未更改，也会为规则新建参数。
- 选择“仅在完全匹配时替换为参数”，则仅当文本与找到的值完全匹配时才会用参数替换值。
- 选择“反向搜索”可以执行向后搜索。
- 选择“左边界实例”框可以指定所需的实例。
- 选择“偏移量”可以指定匹配值内字符串的偏移量。
- 选择“长度”可以指定匹配的字符串 （要保存到参数）的长度。此选项可以与“偏移量”选项结合使用。
- 可以在“可选右边界”框中另外指定右边界，或者从下拉列表中选择“字符串结尾”或“换行符”。

**7** 单击“测试规则”以测试刚刚定义的规则。相关信息，请参阅第 486 页的“测试规则”。

**8** 单击“确定”以保存规则并关闭对话框。

# 42

# 录制后关联 Vuser 脚本

如果在录制期间未执行关联，使用 VuGen 的内置 Web 关联机制可以在录制会话后关联 Vuser 脚本。

本章描述：

- 了解快照
- 设置 VuGen 关联
- 执行扫描关联
- 执行手动关联
- 定义动态字符串的边界

***下列信息仅适用于 Web 和无线以及 < 应用程序 >-Web Vuser 脚本。***

## 关于与快照关联

VuGen 为 Web Vuser 脚本提供了几种关联机制。第 41 章 “配置 Web Vuser 脚本的关联规则”。中讨论的自动方法在录制期间会检测动态值，并且可以用于立即关联这些值。如果禁用了自动关联，或者如果自动方法并未检测到所有差异，则可以使用本章中所述的 VuGen 内置关联机制来查找差异和关联值。还可以对那些仅部分关联的脚本使用此机制。

关联机制使用快照来跟踪脚本执行的结果。快照是网页的图形表示形式。VuGen 在录制期间创建基快照，并在每次执行脚本时生成新快照。可以将录制的快照与任一回放快照进行比较确定需要关联的值，以保证脚本成功执行。

Web 关联机制具有内置的比较实用程序，使用该实用程序可以查看快照之间的文本或二进制差异。然后可以逐一或者一次性全部关联这些差异。

如果 VuGen 的关联机制不足，或者对于不支持这些机制的协议（如 “无线”），请使用手动关联。详细信息，请参阅第 502 页的 “执行手动关联”。

## 了解快照

关联机制使用快照来跟踪脚本执行的结果。快照是网页的图形表示形式。VuGen 在录制期间创建基快照，并在每次执行脚本时生成新快照。可以比较快照及其 HTML 代码，以查找要运行脚本时需要关联的动态值。

快照文件存储在 script 目录下，扩展名为 *.inf*。录制期间创建的快照存储在 Vuser 脚本的 data 文件夹中。“回放”快照位于脚本的下列 Iteration 文件夹中：*Iteration1*、*Iteration2* 等（对于每组结果）。默认情况下， VuGen 将比较录制快照和第一个回放快照。不过，您可以选择其他快照进行比较。

如果没有显示选定步骤的录制快照，请检查下列可能的原因：

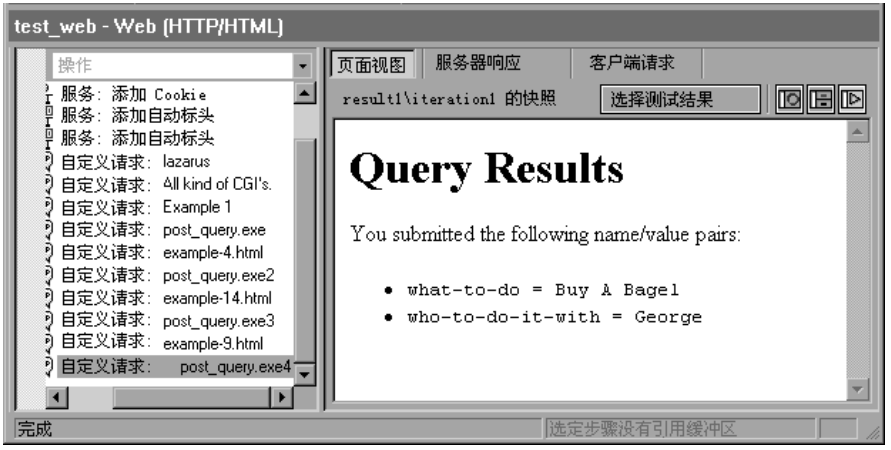
- 脚本是用 VuGen 6.02 版或更低版本录制的。
- 没有为特定类型的步骤生成快照。
- 导入的操作不包含快照。

如果没有显示选定步骤的回放快照，请检查下列可能的原因：

- 脚本是用 VuGen 6.02 版或更低版本录制的。
- 导入的操作不包含快照。
- Vuser 文件存储在只读目录中， VuGen 无法保存回放快照。
- 该步骤表示导航到某个资源。
- 为禁用快照的生成而关闭了下列选项：

  “工具” > “常规”选项 > “关联”选项卡 > “回放期间保存关联信息”。

默认情况下，在树视图中工作时， VuGen 不会在右窗格中显示所选步骤的快照。要隐藏或显示这些快照，请选择 “视图” > “快照”。




可以使用 “视图” > “快照” 展开菜单查看 “已录制” 快照、“已回放” 快照，也可以同时查看这两种快照。还可以使用快捷按钮显示所述的视图：

显示 “已录制” 快照的完整窗口

显示 “已录制” 快照和 “已回放” 快照的拆分窗口

显示 “已回放” 快照的完整窗口

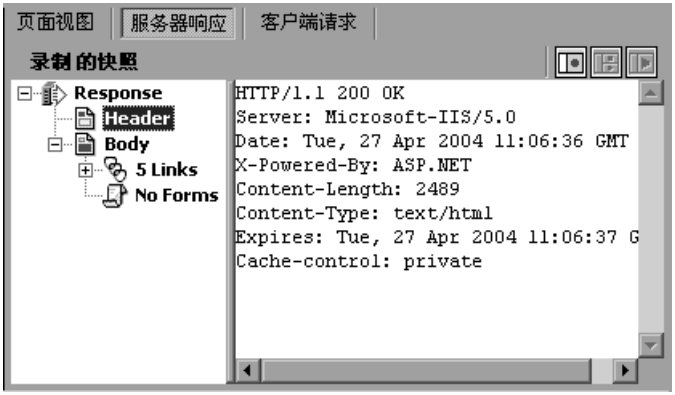
要确定快照文件的名称，请在 “脚本视图”（“视图” > “脚本视图”）中查看脚本。在下例中，快照信息由 *t1.inf* 代表。

```
web_url("www.aa.com",
        "URL=http://www.aa.com/",
        "Resource=0",
        "RecContentType=text/html",
        "SupportFrames=0",
        "Snapshot=t1.inf",
        LAST);
```

在 “快照” 窗口中，有下列选项卡：

**页面视图**：以 HTML （以它在浏览器中显示的方式）显示快照。此按钮对于 “录制” 快照和 “回放” 快照都可用。使用此视图可以确保您查看正确的快照。不过在此视图中，看不到需要关联的值。

**服务器响应**：显示快照的服务器响应 HTML 代码。此按钮对于 “已录制” 快照和 “已回放” 快照都可用。 HTML 视图还将显示左窗格中脚本的树层次结构，并对文档的组件进行了细分：页眉和带有标题、链接、表单等的正文。



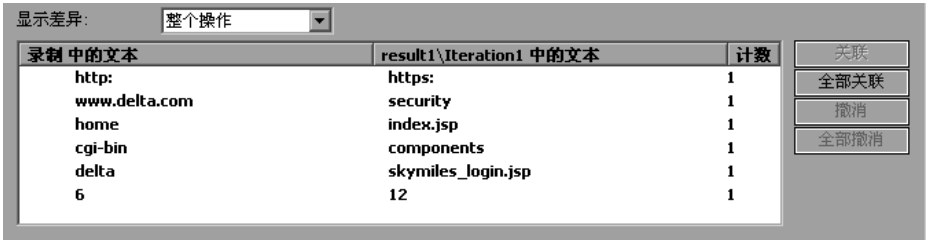

**客户端请求**：显示快照的客户端请求 HTML 代码。此按钮对于 “已录制” 快照和 “已回放” 快照都可用。 HTML 视图还将显示左窗格中脚本的树层次结构，并对文档的组件进行了细分：页眉和正文，以及它们的子组件。

**选择测试结果：**（仅适用于 “已回放”快照）使您可以选择某个 “已回放”快照与录制快照进行比较。此按钮将打开一个对话框，其中列出活动脚本的结果文件夹。“已回放”快照位于脚本的下列 Iteration 文件夹中：*Iteration1*、*Iteration2* 等 （对于每组结果）。

## 查看关联结果选项卡

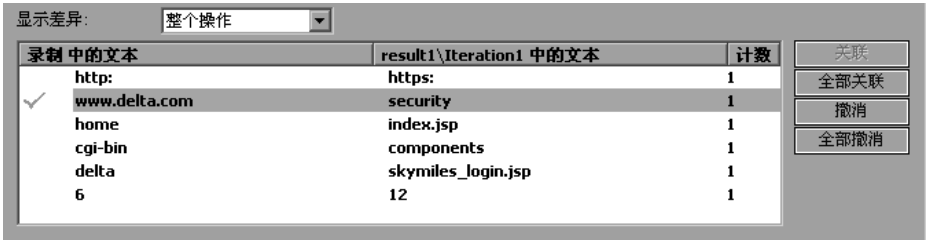
“关联结果”选项卡显示 “录制”快照与 “回放”快照间的差异。

如果指示 VuGen 扫描脚本中的关联，将打开 “输出”窗口并显示 “关联结果”选项卡。“关联结果”选项卡中将显示 “录制”快照与 “回放”快照间的差异。




通过从 “显示差异”列表框中选择所需的选项，可以显示脚本中的所有差异，也可以仅显示当前步骤中的差异。

关联的差异由最左侧一列中的复选标记表明。接下来的两列显示两个快照间的 HTML 差异。最右侧的 “计数”列指出已录制快照间的差异出现的次数。




检测到快照间的差异之后，可以每次关联一个差异 （“关联”），或者一次性关联所有差异 （“全部关联”）。 VuGen 还允许撤消特定的关联 （“撤消”）或所有的关联 （“全部撤消”）。

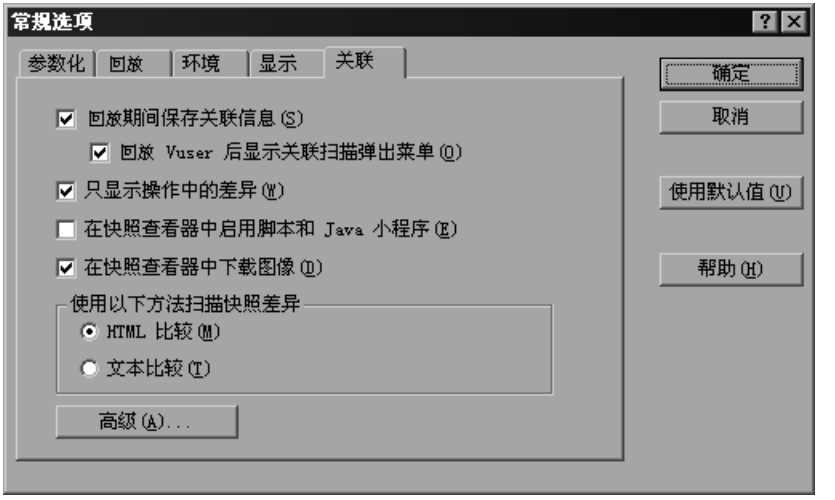
使用此机制对值进行关联时，VuGen 会向脚本中插入 **web_reg_save_param** 函数和注释，表明已为参数完成了关联。 VuGen 还会指出初始值。

```
// Correlation Studio 创建的参数 {WCSParam_Diff1}；替换后的值：
falillgidgkbfdlcflmcfkgdgff.0
web_reg_save_param("WCSParam_Diff1", "LB=BV_EngineID=", "RB=&",
"Ord=1", "Search=body", LAST);
web_url("American2",

"URL=http://www.im.aa.com/American?BV_EngineID={WCSParam_Diff1}
&BV_Operation=Dyn_Frame&form%25framespacing=0&BV_SessionID=%
40%40%40%401303778278.0969956817%40%40%40%40&form%25
destination=%2fnavguest.tmpl&form%25destination_type=template&for
m%25border=0&BV_ServiceName=American&form%25frameborder=no"
,
        "TargetFrame=",
        "Resource=0",
        "RecContentType=text/html",
        "SupportFrames=0",
        "Referer=http://www.im.aa.com/Ameri
        can?BV_Operation=Dyn_AAPage&ref
        erer=index.html&form%25referrer_site=None",
        "Snapshot=t3.inf",
        LAST);
```

# 设置 VuGen 关联

录制会话之前，请在 “常规”选项下设置 “关联”设置。这些选项指示 Vuser 在录制期间保存关联信息，以便在以后的阶段中使用。可以指定比较快照时要执行的比较类型：HTML 或文本。在 “高级”选项中，可以指出应将哪些字符视为分隔符。




**回放期间保存关联信息**：将回放信息保存为快照。可以将任一回放快照与录制快照进行比较。

**回放 Vuser 后显示关联扫描弹出菜单**：指示 VuGen 在扫描操作或关联前进行提示。回放后， VuGen 会询问是否要在操作中扫描关联。

**只显示操作中的差异**：出现在当前操作中的语句，即生成对 HTML 页而不是从服务器返回的 HTML 数据的调用的语句。大多数情况下，这些信息足够用于关联。稍候将在语句中使用并要求关联的任何动态数据都将显示在当前操作中。选中此选项 （默认）将显示当前操作中出现的差异。极少数情况下，您可能要创建所有操作中未显示的数据中的参数，这时需要禁用此选项。

**在快照查看器中启用脚本和 Java 小程序**：允许 VuGen 在快照窗口中运行小程序和 javascript。因为此功能将占用大量资源，所以默认情况下会禁用此选项。

**在快照查看器中下载图像**：指示 VuGen 在 “快照” 视图中显示图形。禁用此选项可以节省资源。此选项默认情况下是启用的。您也可以禁用此选项，例如当知道没有图形需要关联时。

**使用以下方法扫描快照差异**：选择比较方法

- **HTML 差异**：只显示 HTML 代码中的差异。
- **文本差异**：显示所有文本、 Html 和二进制差异。

---

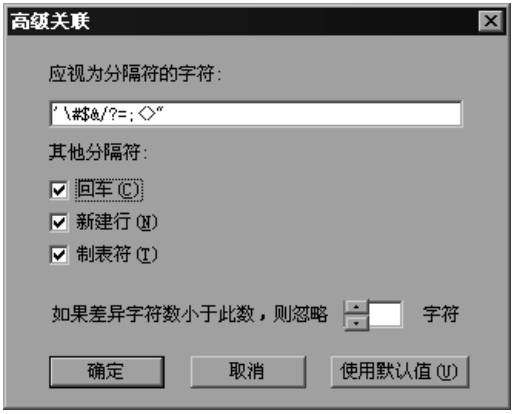
**注意**：大多数情况下，建议使用默认的 HTML 比较方法。如果脚本中包含非 HTML 标记，可以使用文本比较方法。

---

**高级**：将打开高级关联对话框。

### 高级关联对话框

通过此对话框可以指出应视为分隔符的字符。




**Html 中应视为分隔符的字符**：指定一个或多个非标准分隔符。

**其他分隔符**：可以指定标准分隔符，如回车符、换行符和制表符。要更改此设置，请清除分隔符旁的复选框。

**如果差异字符数小于此值，则忽略：**允许指定执行关联的阈值。 VuGen 在扫描进程期间比较已录制脚本与已执行脚本时，会检测差异。除非差异字符数大于或等于阈值，否则 VuGen 将不关联差异。默认值为 4 个字符。

### 设置关联首选项

录制会话前，需要配置关联首选项。

**要设置关联首选项，请执行下列操作：**

1. 选择 “选项” > “常规”，然后选择 “关联” 选项卡。
2. 要将回放信息保存为快照，请选中 “回放期间保存关联信息” 选项。可以将任一回放快照与录制快照进行比较。
3. 要指示 VuGen 在扫描脚本关联之前提示，请选中 “回放 Vuser 后显示关联扫描弹出菜单”。 VuGen 会在回放之后、扫描脚本之前提示您。
4. 要显示当前操作中出现的差异，请选择 “只显示操作中的差异”。要创建操作中未显示的数据的参数，请禁用此选项。
5. 选中 “在快照查看器中启用脚本和 Java 小程序” 可以允许 VuGen 在快照窗口中运行小程序和 javascript。
6. 要指示 VuGen 在 “快照” 视图中显示图形，请选中 “在快照查看器中下载图像” 选项。
7. 选择比较方法：“HTML 比较” 或 “文本比较”（仅适用于非 HTML 元素）。
8. 要设置分隔符字符，请单击 “高级” 以打开 “高级关联” 对话框。
9. 指定应视为分隔符的所有字符。
10. 在 “其他分隔符” 部分中选择所需选项，以指定一个或多个标准分隔符。
11. 在 “如果差异字符数小于此数目，则忽略” 框中为关联指定阈值。 VuGen 在扫描进程期间比较已录制脚本与已执行脚本时，会检测差异。除非差异字符数大于或等于阈值，否则 VuGen 将不关联差异。
12. 单击 “确定” 接受 “高级关联” 设置并关闭该对话框。
13. 在 “常规选项” 对话框中单击 “确定” 接受 “关联” 设置并关闭该对话框。

# 执行扫描关联

可以使用 VuGen 的快照窗口确定脚本中的哪些值是动态的，并且需要关联。下一部分描述如何自动扫描脚本中的差异，以及如何使用 VuGen 执行必需的关联。

**要扫描脚本中的关联，请执行下列操作：**

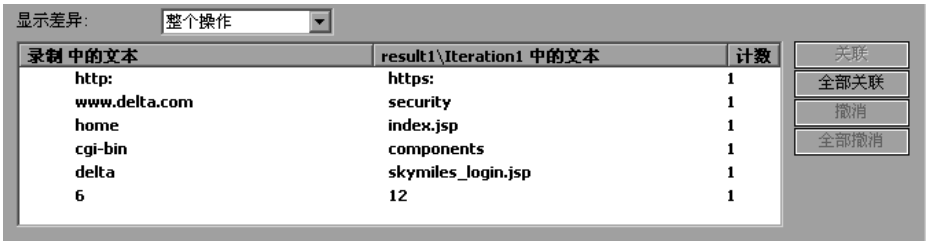
1. 打开脚本并在树视图 （“视图” > “树视图”）中查看该脚本。将显示快照（“视图” > “快照”）。
2. 从左窗格的树视图中选择脚本步骤。将在右窗格中打开 “录制”快照和第一个回放快照。

**3** 要使用除第一个以外的其他快照，请单击 “选择测试结果”。将打开一个对话框，显示包含快照文件的文件夹。这些文件夹通常都是位于脚本所在的文件夹下的 *result* 和 *Iteration* 文件夹。

**4** 要选择除脚本的子文件夹中的文件夹以外的其他文件夹中的快照文件，请单击 “选择文件夹”。浏览到所需的位置，然后单击 “确定”。

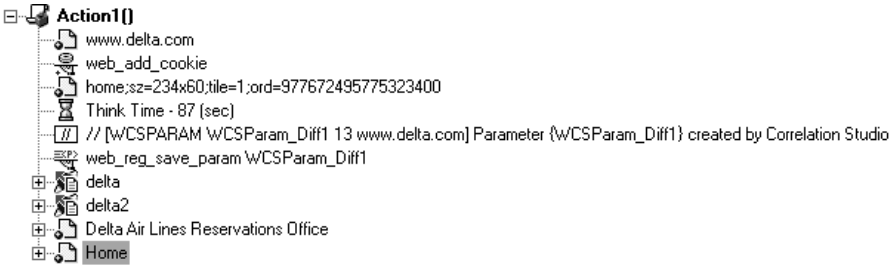
**5** 要查看 HTML 代码，请单击 “服务器请求”选项卡。展开 “正文”分支。要返回页面视图，请单击 “页面视图”选项卡。

**6** 选择 “Vuser” > “扫描关联”，或者单击 “扫描关联”按钮。 VuGen 将在脚本中扫描需要关联的动态值，并在 “关联结果”窗口中显示这些值。




**7** 要查看 Vuser 脚本的特定步骤中的差异，请在 VuGen 的树视图中选择该步骤，并在 “显示差异”列表框中选择 “当前步骤”。要查看所有差异，请在 “显示差异”列表框中选择 “整个脚本”。

**8** 要关联所有差异，请单击 “全部关联”。要关联特定差异，请选中该差异，然后单击 “关联”。 VuGen 会在已关联的差异旁放置一个绿色复选标记，并将 **web_reg_save_param** 函数插入到 Vuser 脚本中。




**9** 要撤消关联，请选中该差异，然后单击 “撤消”。要撤消所有关联，请单击 “全部撤消”。

**10** 选择 “文件” > “保存”以保存更改。

## 执行手动关联

对于 Web Vuser， VuGen 的自动或基于规则的关联通常关联脚本的动态函数，从而可以成功运行脚本。还可以使用 VuGen 快照比较在录制会话之后执行关联。（请参阅第 42 章 “录制后关联 Vuser 脚本”）。

对于无线 Vuser 和其他不适用自动关联的 Vuser 脚本， VuGen 也允许手动关联脚本。可以通过添加代码关联函数来手动关联脚本。可以将数据动态保存到参数的函数是 **web_reg_save_param**。

运行脚本时，web_reg_save_param 函数将扫描所访问的后续 HTML 页。如果指定左边界和 / 或右边界， VuGen 将搜索这些边界之间的文本。 VuGen 找到文本后，会将其分配给某个参数。

函数的语法如下所示：

int **web_reg_save_param** (const char **mpszParamName*, < *属性列表* >, LAST);

下表列出可用的属性。注意，属性值字符串 （例如 Search=all）不区分大小写。

| | |
|---|---|
| *NotFound* | 找不到边界并且生成了空字符串时的处理方法。默认值 “ERROR”表示找不到边界时 LoadRunner 应发出错误消息。如果设置为 “EMPTY”，则不会发出错误消息，并且脚本的执行将继续进行。注意，如果为脚本启用了 “出现错误时仍继续”，则即使将 NOTFOUND 设置为 “ERROR”，在找不到边界时脚本将仍然会继续执行，但会将错误消息写入扩展日志文件中 |
| *LB* | 参数或动态数据的左边界。此参数必须为非空的、以 null 结尾的字符串。边界参数区分大小写；要忽略大小写，请在边界之后添加 “/IC”。如果在边界之后指定 “/BIN”，则指定为二进制数据 |
| *RB* | 参数或动态数据的右边界。此参数必须为非空的、以 null 结尾的字符串。边界参数区分大小写；要忽略大小写，请在边界之后添加 “/IC”。如果在边界之后指定 “/BIN”，则指定为二进制数据 |
| *RelFrameID* | 与请求的 URL 相关的 HTML 页的层次结构级别。可能的值为 ALL 或数字 |
| *Search* | 搜索的范围 – 搜索已分隔的数据的位置。可能的值为 Headers （仅搜索页眉）、 Body （仅搜索正文数据，而不包括页眉）或 ALL （搜索正文和页眉）。默认值为 ALL |
| *ORD* | 此可选参数表示匹配的序号或出现的次数。默认序号为 1。如果指定 “All”，则会将参数值保存在数组中 |
| *SaveOffset* | 找到的值的子字符串偏移量，将保存到参数。默认值为 0。偏移量值必须为非负数 |

| | |
|---|---|
| *Savelen* | 找到的值的子字符串的长度（在指定的偏移量中），将保存到参数。默认值为 ñ1，表示直到字符串的末尾 |
| *Convert* | 要应用于数据的转换方法：<br>HTML_TO_URL：将 HTML 编码数据转换为 URL 编码数据格式<br>HTML_TO_TEXT：将 HTML 编码数据转换为纯文本格式 |

**要手动关联脚本，请执行下列操作：**

1. 标识包含动态数据的语句和表现数据边界的模式。请参阅第 507 页的 “定义动态字符串的边界”。
2. 在脚本中，用您自己的参数名替换动态数据。详细信息，请参阅下列内容。
3. 将 **web_reg_save_param** 函数添加到脚本中包含动态数据的语句之前。请参阅第 504 页的 “添加关联函数”或《Online Function Reference》。（“帮助”> “函数参考”）

## 用参数替换动态数据

首先标识实际动态数据。在录制语句中找到动态数据之后，搜索整个脚本中的动态数据，然后用参数进行替换。为参数赋予任一名称，并将其用大括号括起：*{param_name}*。每个脚本最多可以包括 64 个参数。

**要用参数替换动态数据，请执行下列操作：**

从 VuGen 主窗口中选择 “编辑” > “替换”，以显示 “搜索和替换”对话框。在整个脚本中搜索动态数据，然后用参数替换它们。

## 添加关联函数

可以通过插入 web_reg_save_param 语句保存脚本中的动态数据。此函数将指示 VuGen 创建一个参数，用于在回放期间保存动态数据的运行时值。

运行脚本时，web_reg_save_param 函数将扫描所访问的后续 HTML 页。它将搜索出现左边界后接任一字符串，再后接右边界的情况。找到这种情况时，VuGen 会将左边界与右边界间的字符串指定为函数参数中所命名的参数。找到指定的情况数后，web_reg_save_param 将不再搜索 HTML 页，而是继续脚本中的下一步操作。

## Web Vuser 关联示例

假设脚本中包含动态会话 ID：

```
web_url("FirstTimeVisitors",
      "URL=/exec/obidos/subst/help/first-time-visitors.html/002-8481703-
        4784428>Buy books for a penny ",
      "TargetFrame=",
      "RecContentType=text/html",
      "SupportFrames=0",
       LAST);
```

在上面的语句之前插入 web_reg_save_param 语句：

```
web_req_save_param ("user_access_number", "NOTFOUND=ERROR",
"LB=first-time-visitors.html/","RB=>Buy books for a penny", "ORD=6",
LAST);
```

实现关联语句之后，修改后的脚本将类似于如下所示，其中 user_access_number 是代表动态数据的参数的名称。

```
web_url("FirstTImeVisitors",
      "URL=/exec/obidos/subst/help/first-time-"
        "visitors.html/{user_access_number}Buy books for a penny ",
      "TargetFrame=",
      "RecContentType=text/html",
      "SupportFrames=0",
       LAST);
```

---

**注意：**对于后续 HTTP 请求，每个关联函数都会检索一次动态数据。如果稍后脚本中的另一 HTTP 请求生成了新的动态数据，则必须另外插入关联函数。

---

### 无线 Vuser 关联示例

假设脚本中包含 WAP 连接的动态会话 ID：

```
web_url("login.po;sk=IuZSuuRlHUMnpF-wpK8PzEpy(1YOSBSMy)",
    "URL=http://room33.com/portal/login.po;sk=IuZSuuRlHUMnpF-
      wpK8PzEpy(1YOSBSMy)",
    "Resource=0",
    "RecContentType=text/vnd.wap.wml",
    "Mode=HTML",
    LAST);
```

在上面的语句之前插入 web_reg_save_param 语句，并用参数替换动态值。在下面的示例中， **web_reg_save_param** 函数将登录 ID 字符串保存到名为 SK 的变量。该函数将保存二进制数据 （由 **RB/BIN** 属性表示），并将左边界设置为“sk=”。

```
web_reg_save_param(
   "SK",
"LB=sk=",
"RB/BIN=#login\\x00\\x01\\x03",
"Ord=1",
 LAST);


web_url("login.po;sk={SK}",
    "URL=http://room33.com/portal/login.po;sk={SK}",
    "Resource=0",
    "RecContentType=text/vnd.wap.wml",
    "Mode=HTML",
    LAST);
```

## 定义动态字符串的边界

可以使用下列准则确定和设置动态数据的边界：

- 始终分析 HTML 代码本身内、而不是录制的脚本内的动态数据的位置。
- 标识紧靠动态数据左侧的字符串。此字符串定义动态数据的左边界。
- 标识紧靠动态数据右侧的字符串。此字符串定义动态数据的右边界。
- **web_reg_save_param** 查找介于指定边界之间（但不包含边界）的字符，并保存开始于左边界之后一个字节且结束于右边界之前一个字节的信息。**web_reg_save_param** 不支持嵌入边界字符。例如，如果输入缓冲区是 {a{b{c}，“{”被指定为左边界，而 “}”被指定为右边界，第一个实例是 c，且再没有其他的实例 – 函数发现了右边界和左边界，但由于不允许嵌入边界，因此 “c”是唯一的有效的匹配值。

  默认情况下，任一边界字符的最大长度为 256 个字符。将 web_set_max_html_param_len 函数包含于脚本中可以增加允许的最大长度。例如，下面的函数可以将最大长度增加至 1024 个字符：
  web_set_max_html_param_len(“1024“);

# 43

# 测试 XML 页面

VuGen 的 Web Vuser 支持包含 XML 代码的网页。

本章描述：

- 将 XML 视为 URL 步骤
- 将 XML 作为自定义请求插入
- 查看 XML 自定义请求步骤

***以下信息仅适用于 Web Vuser 脚本。***

## 关于测试 XML 页

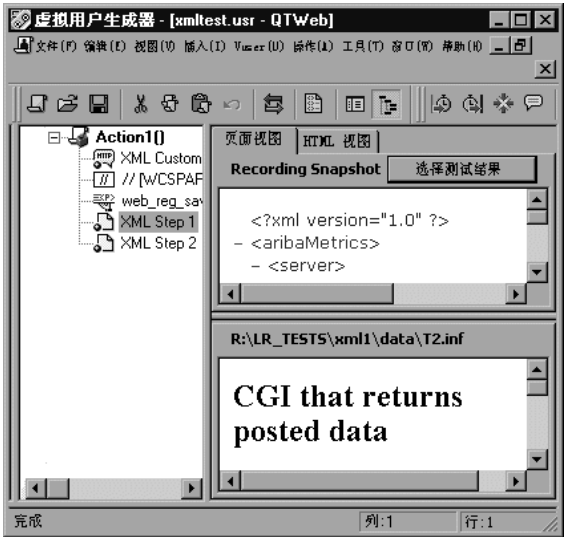
VuGen 支持录制和回放网页内的 XML 代码。

XML 代码可以作为常规 URL 步骤或者作为自定义请求出现在脚本中。 VuGen 将检测 HTML 并允许您查看每个文档类型定义 (DTD) 及其实体和属性。 DTD 使用颜色进行了区分，使您很容易识别元素。您还可以展开和折叠 DTD 的树视图。注意，即使没有将 **RecContentType** 属性中显示的 MIME 类型设置成 *text/xml*，VuGen 也可以检测到 XML。

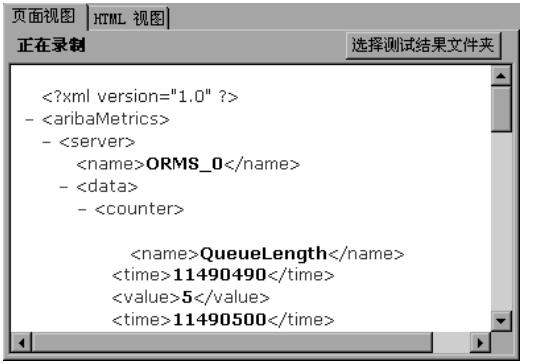
展开 DTD 后，可以将属性值参数化。还可以保存这些值，以便使用标准关联函数来执行关联。有关关联函数的详细信息，请参阅 《Online Function Reference》。（“帮助” > “函数参考”）

# 将 XML 视为 URL 步骤

一种测试含有 XML 代码的页的方法是使用 VuGen 来录制该页。可以按照录制标准网页的方式录制 XML 页。 VuGen 可以录制 DTD 及所有 XML 元素。它不会创建 XML 页的快照。而是对于每个 XML 步骤，在快照帧中显示 XML 代码。




VuGen 在快照帧中创建 DTD 的可展开的、用颜色进行区分的层次结构。单击“+”可展开某项，单击 “–”可折叠该项。 VuGen 将所有 XML 标记显示为褐色，将值显示为黑色。



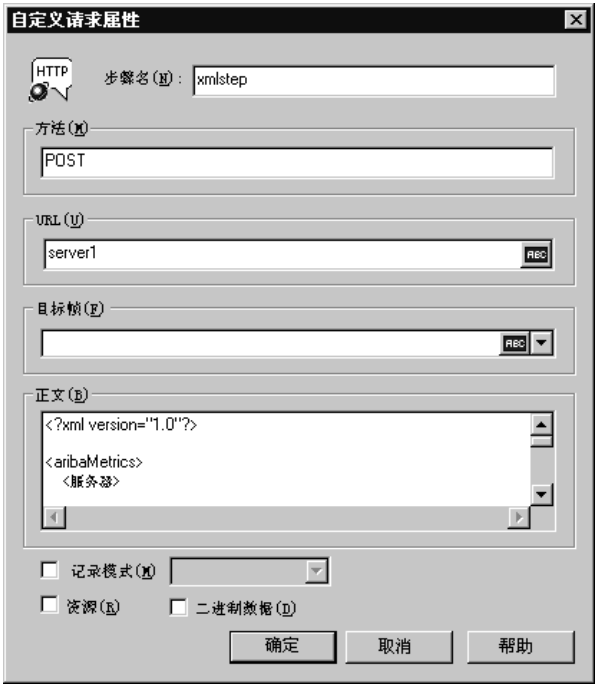

要用参数替换任何常量值，请选择一个值，右键单击该值，然后选择 “替换为新参数”。按照参数化的标准过程进行操作。详细信息，请参阅有关参数化的章节。

## 将 XML 作为自定义请求插入

还可以通过将 XML 代码作为自定义请求插入来测试 XML 页。在此模式中，“自定义请求”属性框将以文本或 XML 格式显示 DTD 的元素。

**要将 XML 代码作为自定义请求添加，请执行下列操作：**

1. 以树视图模式查看脚本，将光标放在所需的位置，然后选择 “插入” > “添加步骤”。将打开 “添加步骤”对话框。
2. 滚动到列表的底部，选择 “自定义请求”。单击 “确定”。将打开 “自定义请求属性”对话框。
3. 输入步骤名、方法 （GET 或 POST）、 URL 和目标帧 （可选）。
4. 从浏览器或编辑器复制 XML 代码并将其粘贴到 “自定义请求属性”框中的 “正文”部分。



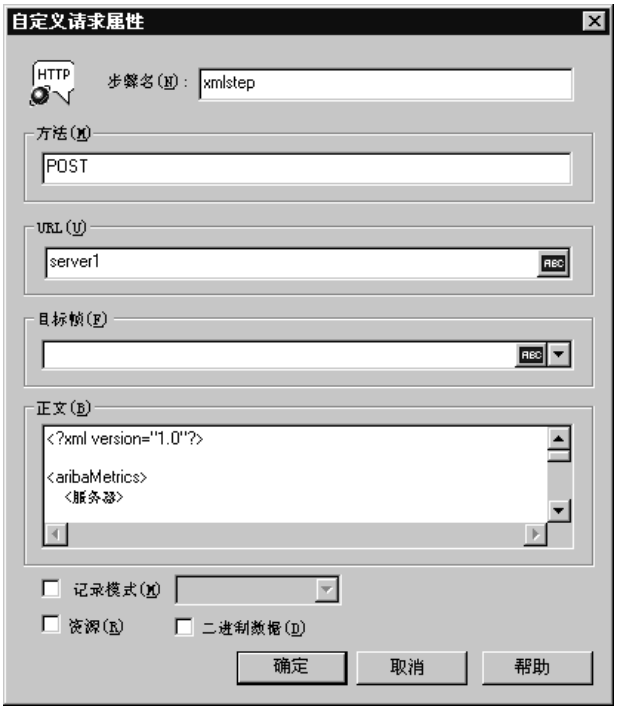

5. 选择适用的回放选项：“录制模式”、“资源”或 “二进制数据”。详细信息，请参阅第 40 章 “修改 Web 和无线 Vuser 脚本”。
6. 单击 “确定”。 VuGen 将把自定义请求步骤放入脚本中。

# 查看 XML 自定义请求步骤

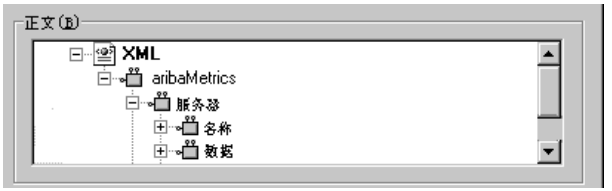
可以随时查看或修改作为自定义请求步骤实现的 XML 代码。VuGen 提供了查看器，可通过它来查看 DTD 的层次结构以及根据需要展开和折叠元素。

**要查看自定义请求步骤的 XML 代码，请执行下列操作：**

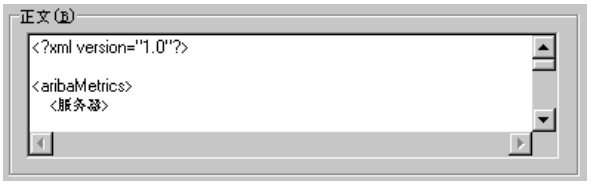
1. 在树视图模式中查看脚本，并选择所需的步骤。
2. 从右键单击菜单中选择 “属性”。将显示 “自定义请求属性”对话框。



对话框的底部将显示 XML 代码。如果将 **RecContentType** 属性设置为 *text/xml*，则默认情况下 VuGen 将以 XML 格式层次结构显示代码。在该模式中， XML 代码是不可编辑的。

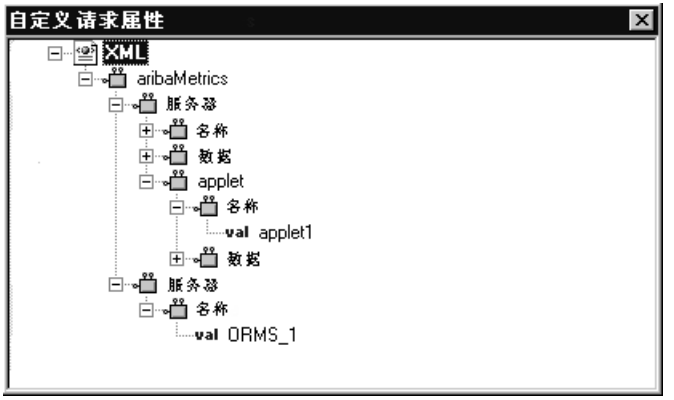



如果将 **RecContentType** 属性设置为除 *text/xml* 以外的任何其他类型，则 VuGen 将以纯文本格式显示代码。在该模式中， XML 代码是可以编辑的。




3 要在文本视图和 XML 视图之间切换，请从右键单击菜单中选择 “XML 视图” 或 “文本视图”。

4 如果是在 XML 视图中，则可以在较大的窗口中查看代码。从右键单击菜单中选择 “扩展视图”。要切换回对话框视图，请从右键单击菜单中选择 “普通视图”。



# 44

# 使用报告调试 Vuser 脚本

为了帮助调试 Web Vuser 脚本，您可以查看总结了脚本运行结果的报告。在 Web Vuser 脚本执行期间，VuGen 会生成报告，您在脚本执行完成后可以查看该报告。

本章描述：

- 了解 “结果概要”报告
- 筛选报告信息
- 管理执行结果

---

**注意**：要启用所有 VuGen Web 报告功能，建议您使用 Microsoft Internet Explorer 4.0 或更高版本。

---

***下列信息仅适用于 Web Vuser 脚本。***

# 关于使用报告调试 Vuser 脚本

当您使用 VuGen 调试 Web Vuser 脚本时，需要指定是否在脚本执行期间生成“结果概要”报告。“结果概要”报告包含 Vuser 访问的所有网页的详细信息以及 Vuser 执行的任何检查。在调试 Web Vuser 脚本时，检查该信息非常有用。有关使用 VuGen 运行 Vuser 脚本的详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

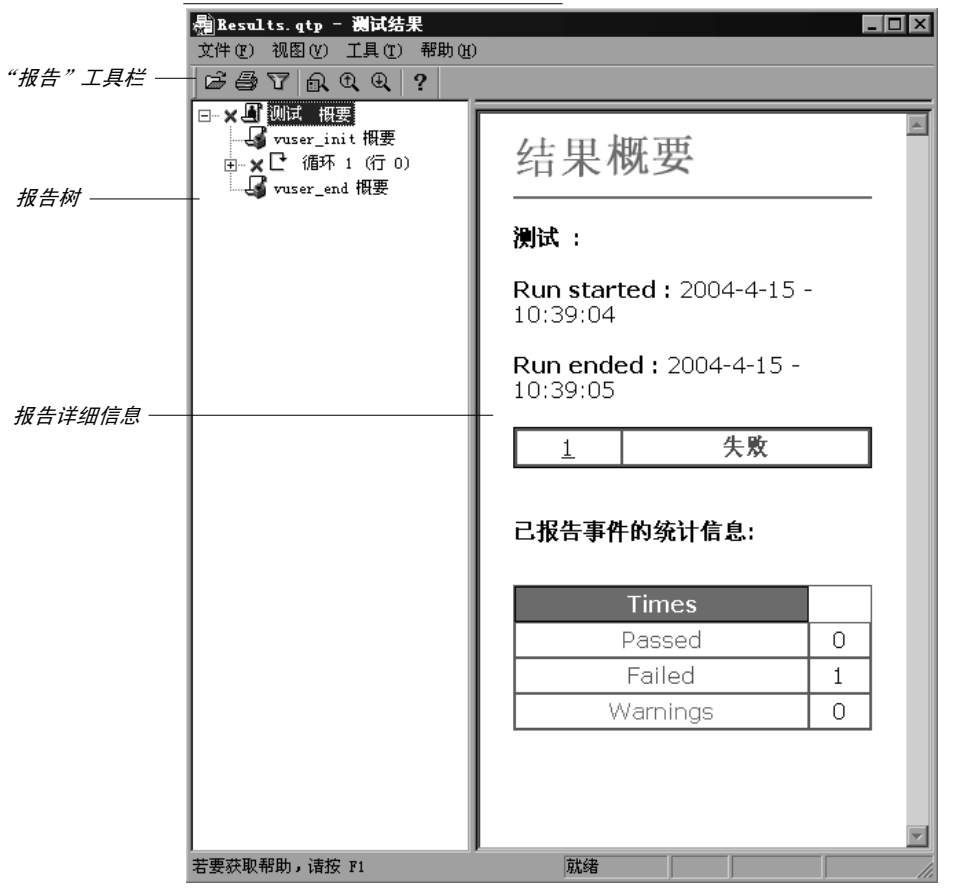
使用 VuGen 运行 Vuser 脚本之后，请查看 “结果概要”报告。

如果计算机上安装有 Microsoft Internet Explorer 4.0 或更高版本，VuGen 将以 VuGen 报告格式 （具有 *.qtp* 扩展名）生成结果，而您可以在虚拟用户生成器“报告”窗口中查看结果。这是建议的选项，因为 VuGen 的 “报告”窗口会提供更复杂的界面和其他功能。

设置 “显示”选项 （“工具” > “常规选项”）可以指定 VuGen 是否应该生成“结果概要”报告，如果是，在脚本执行之后该报告是否自动打开。有关设置“显示”选项的详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

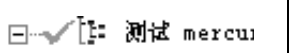
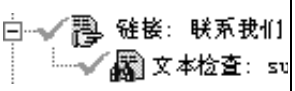
# 了解 “结果概要”报告

运行 Vuser 脚本之后，可以查看 “结果概要”报告。该报告将显示脚本执行结果的概要。




- 左侧窗格显示报告树 – 结果的图形表示形式。在报告树中，绿色复选标记表示成功的步骤，而红色 X 表示失败的步骤。
- 右侧窗格显示报告详细信息 – 脚本运行的全面概要以及报告树选定分支的其他信息。

选择报告树的分支可以查看该分支的信息。

| 选择分支 ... | 查看下列详细信息： |
| --- | --- |
| 测试名称<br>  | 脚本执行的全面结果概要 |
| 测试循环<br> | 特定循环的执行概要 |
| 测试步骤或检查<br>  | Vuser 脚本中选定步骤或检查的网页 |

可以折叠或展开报告树中的分支，以便更改报告树显示的详细信息的级别。

➤ 要折叠分支，请单击要折叠的分支左侧的折叠 (−) 号。报告树将隐藏分支的详细信息，并且折叠号将更改为展开 (+) 号。

➤ 要折叠报告树中的所有分支，请选择 “视图” > “全部折叠”。

➤ 要展开分支，请单击要展开的分支左侧的展开 (+) 号。报告树将显示分支的详细信息，并且展开号将更改为折叠 (−) 号。

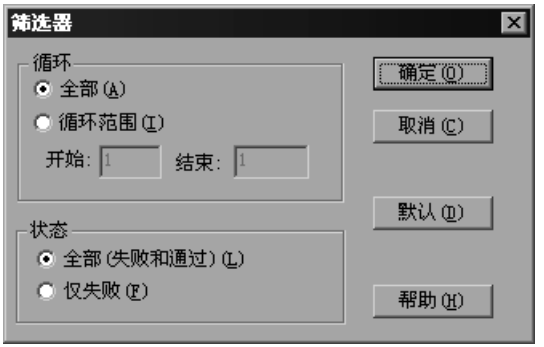
➤ 要展开报告树中的所有分支，请选择 “视图” > “全部展开”。

# 筛选报告信息

可以筛选显示在 “VuGen 结果概要” 报告中的信息。筛选器可以基于循环编号或者循环的状态。

**要筛选包含在报告中的信息，请执行下列操作：**

**1** 单击 “报告” 工具栏上的 “筛选器” 按钮，或者选择 “视图” > “筛选器”。将打开 “筛选器” 对话框。




**2** 设置需要的筛选器选项。默认的筛选器选项是 “全部”，如上例所示。

要将报告限制到指定范围的循环，请在 “循环” 部分中选择 “循环范围”，然后在 “从” 和 “到” 框中指定范围。

要将报告限制到失败的循环，请在 “状态” 部分中选择 “仅失败”。

**3** 单击 “确定” 接受设置并关闭 “筛选器” 对话框。

## 搜索结果

可以按照结果步骤的最后状态在 “测试结果”内搜索结果步骤：失败、通过、完成或警告。可以为搜索选择多个状态。

**要使用特定状态搜索步骤，请执行下列操作：**

**1** 选择 “工具” > “查找”，或者单击 “报告”工具栏上的 “查找”按钮。将打开 “查找”对话框。

**2** 选择要查找的步骤的状态 （一个或多个）。

**3** 选择搜索方向：“向上”或 “向下”。

**4** 单击 “查找下一个”。光标将跳至第一个匹配项。

**5** 要重复执行该搜索，请单击 “查找下一个”按钮。

## 管理执行结果

使用 “文件”菜单中的命令可以打开、打印和退出 “结果概要”报告。

有关设置 “结果概要”报告选项的详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”的第 134 页的 “使用 VuGen 的 Web Vuser 脚本调试功能”。

### 打开 “结果概要”报告

在运行 Web Vuser 脚本时， VuGen 会将 “结果概要”报告文件保存在脚本文件夹的结果子文件夹中。该报告文件的格式是：*script_name.qtp*。

要打开 “结果概要” 报告，请执行下列操作：

1 选择 “文件” > “打开”，或者单击 “报告” 工具栏上的 “打开” 按钮。将打开 “打开” 对话框。

2 选择要打开的报告文件的名称，然后单击 “打开”。

3 要打开最近查看的报告，请从 “文件” 菜单上的报告历史记录列表中选择它。

## 打印报告结果

可以打印 “测试结果概要” 报告。

要打印 “测试概要” 报告，请执行下列操作：

1 选择 “文件” > “打印”，或者单击 “报告” 工具栏上的 “打印” 按钮。将打开 “打印” 对话框。

2 从 “打印范围” 框中选择范围：

**全部**：打印整个报告。这包括循环中每个步骤的网页。

**选择**：打印 “报告” 树中的选定分支。

3 单击 “确定” 进行打印。

4 要更改打印机的设置选项，请选择 “文件” > “打印设置”，然后更改 “打印设置” 对话框中的设置。

## 关闭 “测试概要” 报告

要关闭 “测试概要” 报告，请选择 “文件” > “退出”。将关闭 “测试结果” 窗口。

# 45

# 有关 Web Vuser 的 Power User 提示

本章回答了 Web Vuser 的高级用户最常提出的一些问题。这些问题和答案分为下列几个部分：

- 安全性问题
- 处理 Cookie
- 运行时查看器 （联机浏览器）
- 浏览器
- 配置问题
- 兼容问题

***下列信息适用于 Web Vuser 脚本。***

## 安全性问题

**问题 1**：Web Vuser 同时支持安全 (HTTPS) 和不安全 (HTTP) 事务吗？

**解答**：是的， Web Vuser 同时支持安全 (HTTPS) 和不安全 (HTTP) 事务。

**问题 2**：Web Vuser 支持数字证书吗？

**解答**：是的， Web Vuser 支持客户端数字证书。出于安全目的，将数字证书作为电子邮件的附件。数字证书的最常见用途是验证发送邮件的用户是否是他或她所自称的人并且为接收者提供对答复进行编码的方法。

VUGen 在支持客户端证书时具有以下限制：

➤ **录制：**客户端证书始终取自 IE 数据库，而不管在录制期间所使用的实际浏览器是什么。因此，如果使用 IE 以外的其他浏览器或应用程序进行录制，必须首先从进行录制的浏览器中导出证书并将其导入 IE 中。在将证书导入 IE 中时，请确保可导出其私钥：

➤ **录制：**在较低版本的 VuGen （早于 LoadRunner 7.0）中，每当使用客户端证书时，都将生成 **web_set_certificate**。该函数仅有一个参数：证书列表中证书的序号。该函数仅可在 WinInet 模式下回放。

在较高版本的 VuGen （7.0 及更高版本）中，将生成 **web_set_certificate_ex**。该函数有一个附加参数 – 包含证书的文件的路径。该证书文件在录制期间自动生成并且与 Vuser 脚本保存在一起。每当使用 WinInet 回放模式时，都将使用第一个参数。对于套接字回放 （默认值），则使用第二个参数 （证书文件）。注意，如果无法转储特定的证书，例如，如果无法导出其私钥，将生成没有文件名的 **web_set_certificate_ex**。在这种情况下，只应使用 WinInet 回放模式。

**回放：**如果使用 **web_set_certificate_ex**，并且具有文件名参数，那么它只能用于套接字回放，并且在负载计算机上不需要任何自定义配置。如果使用 **web_set_certificate** 或者没有文件名的 **web_set_certificate_ex**，则它只能用于基于 WinInet 的回放。在这种情况下，需要安装您在录制计算机上拥有的所有证书，安装顺序与这些证书在证书列表中显示的顺序**相同**。可通过导出 / 导入来完成。

**问题 3：**当录制可访问支持 SSL 的站点的 Vuser 脚本时，弹出许多警告消息。这些消息应该出现吗？如果是这样，应该如何处理它们呢？

**解答：**为了能够录制对支持 SSL 的站点的访问， VuGen 提供自己的服务器证书而非原始的服务器证书。这将产生两个安全冲突：

➤ 颁发的证书不适用于用户要连接到的站点。

➤ 该证书是由未知颁发机构颁发的。

这些安全冲突会使录制浏览器弹出警告消息。

如果正在使用 Netscape 3.0 或更高版本，或者正在使用 Internet Explorer 4.0 或更高版本，那么可以选择忽略这些警告。您可以安全地忽略这些消息。

---

**注意：**弹出消息仅在录制脚本时出现，而不会在执行脚本时出现。

您可以禁止其中一些弹出消息出现，但不能禁止全部消息。

---

**问题 4**：我正在使用 IE 和 Netscape 之外的 Web 应用程序。当访问不具有受到认可的证书的安全站点时，该应用程序自动中止。能录制该应用程序吗？

**解答**：当您访问不具有受到认可的证书的安全站点时， IE 和 Netscape 会发出警告。某些浏览器和应用程序对于不受认可的证书不会发出警告，它们只是退出该安全站点。要录制这些站点，必须获取证书和密钥的 *pem* 文件，然后将其添加到 LoadRunner/bin 下的 *certs* 目录中。将 *pem* 文件用类似于现有项的格式列到 *index.txt* 文件中：含有主机名和端口的节名，后面跟着 *pem* 文件名。

```
[demoserver:443]
Certfile=xxx.pem
Keyfile=yyy.pem
```

**问题 5**：VuGen 支持 128 位加密吗？

**解答**：Netscape Communicator（4.5 及更高版本）和 Internet Explorer（5.0 及更高版本）都支持 128 位加密。目前， Netscape 仅支持 Verisign。

**问题 6**：VuGen 支持 Internet Explorer 的客户端证书吗？

**解答**：是的， VuGen 支持 Internet Explorer 的客户端证书。

**问题 7**：VuGen 支持 Netscape 的客户端证书吗？

**解答**：不， VuGen 仅支持 Internet Explorer 的客户端证书。如果您仅拥有 Netscape 证书，那么请首先从 Netscape 中导出所需的证书，然后将其导入到 Internet Explorer 中。请确保以相同的顺序导出和导入这些证书。您必须在每台将录制或运行需要证书的 Web Vuser 脚本的计算机上重复执行该过程。

**问题 8**：如果查看 Web Vuser 脚本，可以弄清楚 Vuser 访问的是常规 (HTTP) 服务器还是支持 SSL 的 (HTTPS) 服务器吗？

**解答**：有时可以。 Web Vuser 脚本不区分安全和非安全请求：图形 Vuser 脚本对安全请求和非安全请求使用相同的图标；基于文本的 Vuser 脚本对安全请求和非安全请求使用相同的函数。但是，如果 Vuser 脚本中的某个步骤包含 URL，那么您或许可以根据该 URL 辨别该步骤访问的是常规 (HTTP) 服务器还是支持 SSL 的 (HTTPS) 服务器。

**问题 9**：Web Vuser 支持哪些类型的身份验证？

**解答**：Web Vuser 支持基本身份验证和 NTLM 身份验证 （NT 质询响应身份验证）。

# 处理 Cookie

**问题 10**：当录制 Vuser 脚本时， VuGen 处理 Cookie 吗？

**解答**：VuGen 会自动处理所有通过 HTTP 标头设置的 Cookie。但是， VuGen 无法始终正确地处理由 JavaScript 或 *<meta->* 标记设置的 Cookie。详细信息，请参阅问题 14。

**问题 11**：当运行 Web Vuser 脚本时， Vuser 会重新使用在录制 Vuser 脚本时使用过的相同 Cookie 吗？

**解答**：可能会，也可能不会，这取决于 Cookie 的类型。 Cookies 可以分成两种类别：永久 Cookie 和会话 Cookie：

| | |
|---|---|
| *永久 Cookie* | 向 Web 服务器标识您的身份并且在有限的时间段内有效的纯文本字符串。永久 Cookie 存储在硬盘上。 |
| *会话 Cookie* | 仅在当前访问 （会话）期间向 Web 服务器标识您的身份的纯文本字符串。会话 Cookie 不会存储在硬盘上。 |

当录制 Web Vuser 脚本时， VuGen 会检测发送到浏览器的所有 Cookie。VuGen 会按如下方式区分永久 Cookie 和会话 Cookie：

| | |
|---|---|
| *永久 Cookie* | VuGen 将永久 Cookie 的详细信息直接录制成 Vuser 脚本。VuGen 使用 **web_add_cookie** 将永久 Cookie 包括在 Vuser 脚本中。当运行 Vuser 脚本时， Vuser 会根据需要使用这些永久 Cookie。 |
| *会话 Cookie* | VuGen 不保存在录制会话期间所使用的会话 Cookie。而是在录制时缓存会话 Cookie，然后在停止录制时丢弃它们。<br><br>当运行 Vuser 脚本时，Vuser 使用从 Web 服务器接收的新会话 Cookie。即， Vuser 不会重新使用在录制脚本时生成的相同会话 Cookie。这些会话 Cookie 存储在 Vuser Cookie 缓存中，然后在 Vuser 停止时被丢弃。Vuser 不会保存这些会话 Cookie。 |

**问题 12**：每个 Vuser 都有其自己唯一的 Cookie 缓存吗？

**解答**：是的，每个 Vuser 都有其自己唯一的 Cookie 缓存 – 会话 Cookie 不共享，即使 Vuser 在同一负载生成器上运行。

**问题 13**：在运行脚本之前，必须在录制的 Vuser 脚本中将 Cookie 参数化吗？

**解答**：有时需要。如问题 11 中所述，在录制脚本时， VuGen 会将永久 Cookie 复制到 Vuser 脚本中。当运行 Vuser 脚本时， Vuser 会使用所录制的永久 Cookie。如果每个 Vuser 都需要唯一的永久 Cookie，那么您需要在 Vuser 脚本中将 Cookie 参数化。

**问题 14**：Web Vuser 是否处理在 JavaScript 内设置的 Cookie？

**解答**：VuGen 会自动处理所有通过 HTTP 标头设置的 Cookie。但是， VuGen 无法始终正确处理由 HTML 页中包含的 JavaScript 设置的 Cookie。通过 JavaScript 设置的 Cookie 会在录制和回放期间导致独特的问题：

*录制*

VuGen 应该将永久 Cookie 而非会话 Cookie 录制到 Vuser 脚本中 （通过 **web_add_cookie** 语句）。但是，由于技术方面的限制，由 JavaScript 设置的所有 Cookie 都将由 VuGen 作为永久 Cookie 进行录制，即使这些 Cookie 是会话 Cookie

解决办法：录制 Vuser 脚本之后，请插入关联语句以关联所有设置会话 Cookie 的 **web_add_cookie** 语句。不要删除设置永久 Cookie 的 **web_add_cookie** 语句

*回放*

Web Vuser 不会运行嵌入到 HTML 页中的 JavaScript。因此，在 Vuser 运行时，不会创建由此类 JavaScript 创建的任何会话 Cookie

解决办法：录制 Vuser 脚本之后，将关联语句插入脚本中以确定相应的 Cookie。然后将 **web_add_cookie** 语句插入到 Vuser 脚本中以设置相应的 Cookie

**问题 15**：Vuser 可以在运行时操纵 Cookie 吗？

**解答**：是的，当 Vuser 运行时，该 Vuser 可以操纵存储在其 Cookie 缓存中的 Cookie。您可以在 Vuser 脚本中使用下列函数来操纵 Cookie 缓存：

- web_add_cookie()
- web_remove_cookie()
- web_cleanup_cookies()

有关上述函数的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

# 运行时查看器 （联机浏览器）

**问题 16**：运行时查看器如何显示网页？

**解答**：在运行 Web Vuser 脚本时，由 Vuser 访问的 Web 服务器会将信息下载到 Vuser 中。该信息的格式通常是 HTML。 Vuser 会将该信息保存到 Vuser 的结果目录中。每个网页都将以 HTML 格式保存为单独的 *.htm* 文件。当 Vuser 运行时，运行时查看器会加载保存在 Vuser 结果目录中的 *.htm* 文件，然后显示产生的网页。

**问题 17**：在使用运行时查看器时， JavaScript 错误频繁出现。是什么原因导致了这种情况，如何防止出现这种情况？

**解答**：在使用运行时查看器时，请确保未选中运行时浏览器 “选项”菜单中的 “启用脚本”选项。这将指示运行时查看器不运行任何 JavaScript，并且确保 JavaScript 错误不会再出现在运行时查看器中。

如问题 16 的答案中所述，当运行 Vuser 脚本时， VuGen 会保存由服务器返回的信息。运行时查看器将显示所保存的这一信息，而非显示由服务器直接返回的信息。

**问题 18**：运行时查看器可以显示哪些类型的数据？

**解答**：运行时查看器仅可以显示 HTML 页。它无法显示任何其他类型的信息。

**问题 19**：应该在负载生成器中安装哪些软件，才能够显示运行时查看器？

**解答**：由于运行时查看器使用 Internet Explorer ActiveX 控件，因此必须安装 Microsoft Internet Explorer 4.0 或更高版本以便使用运行时查看器。

**问题 20**：在运行 Vuser 脚本时，为什么运行时查看器不显示 Vuser 提交给 Web 服务器的数据？

**解答**：运行时查看器仅显示由服务器返回到 Vuser 的 HTML 页。运行时查看器不会显示 Vuser 提交给 Web 服务器的任何数据。详细信息，请参阅问题 16 的答案。

**问题 21**：运行时查看器能够正确地显示多窗口应用程序吗？

**解答**：不，当前运行时查看器不能正确地显示多窗口应用程序。

## 浏览器

**问题 22**：为什么建议我在计算机上安装 Internet Explorer 4.0 或更高版本，即使我总是使用 Netscape 来录制脚本？

**解答**：VuGen 严重依赖于 Microsoft Internet API，即 WinInet。这一点同时适用于录制和回放 Web Vuser 脚本。 WinInet.dll 是用于 Internet 连接的 Microsoft 基础结构。

LoadRunner 将安装 3.0 版的 WinInet.dll，除非计算机上已经安装了更新的版本。 3.0 版有许多限制。 4.0 版要高级得多，所以建议您安装 4.0 版以获得使用 Web Vuser 的最佳结果。安装 WinInet.dll 4.0 版的最简单的合法方式是安装 Internet Explorer 4.0 或更高版本。

**问题 23**：如果安装了 Internet Explorer 3.0 而未安装 Internet Explorer 4.0 或更高版本，哪些功能将无法使用？

**解答**：Internet Explorer 包括 WinInet.dll。您需要版本 4 的 WinInet.dll 文件以启用下列功能：

- 代理服务器身份验证
- NTLM 身份验证 （NT 质询响应身份验证）
- 数字证书
- 运行时浏览器
- 报告
- WAP 录制

**问题 24**：在录制时，必须使用标准浏览器，如 Netscape 或 Internet Explorer 吗？

**解答**：不，在录制 Web Vuser 脚本时，您可以使用所选择的任何浏览器。实际上，您不需要使用浏览器。您可以转而使用任何生成 HTTP(S) 请求的应用程序。应用程序的唯一要求是必须能够将代理服务器设置设为 localhost:7777，以便 VuGen 可以录制 HTTP(S) 请求。

**问题 25**：如何录制非标准 HTTP(S) 应用程序？

**解答**：执行以下过程：

1. 选择 “工具” > “录制选项”，然后单击 “浏览器” 节点。选择 “手动启动应用程序”。
2. 单击 “录制代理” 节点，然后选择 “使用自定义代理服务器” 选项。单击 “确定” 关闭 “录制选项”。
3. 单击 “开始录制” 按钮。 VuGen 会提示您提供所录制的应用程序需要的代理服务器设置。注意主机和端口名。
4. 单击 “取消”，然后打开 “录制选项” (CTRL+F7)。单击 “录制代理” 节点，然后在 “使用自定义代理服务器” 部分输入建议的代理服务器设置。单击 “确定” 关闭 “录制选项”。
5. 编辑正在录制的应用程序中的代理服务器设置。
6. 单击 “开始录制” 按钮，然后开始录制会话。
7. 在完成录制后关闭应用程序，然后还原原始代理服务器设置 （还原失败可能会使该应用程序无法正常运行）。

**问题 26**：VuGen 会修改录制浏览器中的代理服务器设置吗？

**解答**：是的。在开始录制 Web Vuser 脚本时， VuGen 会启动所指定的浏览器。然后， VuGen 会指示浏览器浏览 VuGen 代理服务器。为此， VuGen 会修改录制浏览器上的代理服务器设置。默认情况下， VuGen 会立即将代理服务器设置更改为 localhost:7777。录制之后，VuGen 会将原始代理服务器设置还原到该录制浏览器中。在 VuGen 进行录制的过程中，不得更改代理服务器设置。

**问题 27**：在进行录制的过程中，浏览器崩溃。我现在无法使用浏览器访问任何站点，即使不进行录制。为什么？

**解答**：问题 26 的答案描述了 VuGen 如何在录制期间更改浏览器中的代理服务器设置。如果在录制过程中浏览器崩溃， VuGen 可能无法还原浏览器的原始代理服务器设置。然后，浏览器将仍然具有 localhost:7777 这一设置，这将使浏览器无法访问任何站点。必须手动还原浏览器的原始代理服务器设置。

**问题 28**：VuGen 支持 Socks 代理服务器吗？

**解答**：是的， VuGen 确实支持 Socks 代理服务器。要使用 Socks 代理服务器，必须使用 Internet Explorer 而非 Netscape 作为录制浏览器。另外，请：

➤ 使用 Internet Explorer 4.0 或更高版本来定义 Socks 代理服务器。

在 Internet Explorer 中，选择 “工具” > “Internet 选项”。单击 “连接” 选项卡，然后在 “代理服务器” 组中单击 “高级”。在 “代理服务器设置” 对话框中，输入适当的 Socks 代理服务器设置。

该步骤适用于用来录制 Vuser 脚本的计算机，以及将运行访问 Socks 代理服务器的 Vuser 的所有计算机。

➤ 将 Internet Explorer 定义为默认浏览器。

通过将所有具有 *.htm* 扩展名的文件与 Internet Explorer 相关联，可以实现上述目的。

该步骤适用于用来录制 Vuser 脚本的计算机，以及将运行访问 Socks 代理服务器的 Vuser 的所有计算机。

➤ 在录制 Vuser 脚本时，指示 VuGen 从录制浏览器中获得代理服务器设置。

在 VuGen 中，选择 “工具” > “录制选项”。单击 “录制代理” 节点。选择 “从录制浏览器获取代理设置” 选项。

该步骤仅适用于用来录制 Vuser 脚本的计算机，而不适用于运行 Vuser 的计算机。

➤ 指示所有运行脚本的 Vuser 从默认浏览器获取代理服务器设置。

在 VuGen 中，选择 “Vuser” > “运行时设置”。单击 “代理服务器” 选项卡，然后选择 “从默认浏览器获取代理服务器设置” 选项。

该设置适用于所有运行 Vuser 脚本的 Vuser。

**问题 29**：如果安装了 Netscape 而未安装 Internet Explorer，可以显示执行报告吗？

**解答**：要使 VuGen 显示执行报告，需要有 Internet Explorer 4.0 或更高版本。

**问题 30**：我发现“并发连接数”运行时设置不再可用。仍然可以修改该设置吗？

**解答**：是的。可以使用 **web_set_sockets_options** 函数来修改该设置。要设置每个主机的最大连接数，请使用 MAX_CONNECTIONS_PER_HOST 标志并且给它分配需要的值。要设置全局连接数、每个 Vuser 的最大并发连接数，请使用 MAX_TOTAL_CONNECTIONS 标志并将它设置为需要的数字。使用 Internet Explorer 时，默认的并发连接数对于 HTTP 1.0 是四，对于 HTTP 1.1 是二。详细信息，请参阅函数参考中的 **web_set_sockets_options**。

## 配置问题

**问题 31**：我执行了快照比较并且结果非常不准确。

**解答**：选择“选项”>“常规”，打开“常规选项”对话框，然后选择“关联”选项卡。在“使用以下方法扫描快照差异”部分中，确保选择了“HTML 比较”选项而非“文本比较”选项。“文本比较”仅适用于非 HTML 的快照。

## 兼容问题

**问题 32**：可以在 UNIX 系统上回放录制的脚本吗？

**解答**：是的，在 UNIX 平台上支持回放。

# 第 IX 部分

## Enterprise Java Bean 协议

# 46

# 执行 EJB 测试

Enterprise Java Beans (EJB) 测试工具将生成测试 EJB 对象的脚本。

本章描述：

- 使用 EJB 检测器
- 了解 EJB Vuser 脚本
- 创建 EJB 测试 Vuser
- 设置 EJB 录制选项
- 运行 EJB Vuser 脚本

有关 EJB 测试的其他信息，请参阅附录 B “EJB 体系结构和测试”。

***下列信息仅适用于 EJB 测试 Vuser 脚本。***

## 关于 EJB 测试

VuGen 提供了多个开发脚本以测试 Java 应用程序的工具。要通过录制生成 Vuser 脚本，请使用 Jacada、 CORBA 或 RMI Vuser。要通过编程创建脚本，请使用自定义 Java Vuser。

EJB 测试 Vuser 与标准的 Java Vuser 不同，因为 VuGen 会自动创建脚本以测试 EJB 功能，无需录制或编程。生成脚本前，需要指定 JNDI 属性和关于应用程序服务器的其他信息。 LoadRunner 的 EJB 检测器将扫描应用程序服务器并确定哪些 EJB 可用。您需要选择要测试的 EJB， LoadRunner 将生成测试每个 EJB 方法的脚本。它将为每个方法创建事务，以便您可以度量其性能并找出问题。另外，每个方法都包装在一个 *try and catch* 块中，以便进行异常处理。

注意，要创建 EJB 测试脚本，必须在应用程序服务器主机上安装 LoadRunner EJB 检测器，而且检测器必须处于活动状态。下列部分对该检测器进行了描述。

VuGen 还具有内置实用程序，用于将方法插入到脚本中。使用此实用程序，可以显示所有可用的程序包，选择所需的方法，然后将这些方法插入到脚本中。详细信息，请参阅第 549 页的 “运行 EJB Vuser 脚本”。

# 使用 EJB 检测器

EJB 检测器是一个独立的代理程序，必须将其安装在要扫描查找 EJB 的每一台计算机上。该代理程序将检测计算机上的 EJB。安装 EJB 检测器前，请验证计算机上是否有有效的 JDK 环境。

## 安装 EJB 检测器

可以在应用程序服务器计算机或者客户端计算机上安装和调用 EJB 检测器。要在客户端计算机上运行 EJB 检测器，应用程序服务器计算机上必须已装入驱动器。

**要安装 EJB 检测器代理程序，请执行下列操作：**

1. 在应用程序服务器计算机或客户端计算机 （装入了提到的文件系统）上创建 EJB 检测器的主目录。
2. 将 *<LR_root>\ejbcomponent\ejbdetector.jar* 文件解压缩到 EJB 检测器目录。

## 运行 EJB 检测器

必须先运行 EJB 检测器，然后才能在 VuGen 中启动 EJB 脚本生成流程。可以在应用程序服务器上运行 EJB 检测器，也可以在客户端计算机上运行 （在这种情况下，请确保从 EJB 检测器 （客户端）计算机装入应用程序服务器，在搜索根目录中指定安装目录，并将生成的脚本更改为连接到已装入的计算机，而不是本地计算机上）。

可以从命令行或批处理文件中运行 EJB 检测器。

**要从命令行运行 EJB 检测器，请执行下列操作：**

1. 从命令行运行 EJB 检测器前，向 CLASSPATH 环境变量中添加 DETECTOR_HOME\classes 和 DETECTOR_HOME\classes\xerces.jar。
2. 如果正在使用 EJB1.0 （Weblogic 4.x、 WebSphere 3.x），请向 CLASSPATH 中添加正在测试的 EJB 类以及下列供应商 EJB 类：

   对于 WebLogic 4.x：<WebLogic 目录 >\lib\weblogicaux.jar

   对于 WebSphere 3.x：<WebSphere 目录 >\lib\ujc.jar

3 如果 EJB 使用其他类目录或 .jar 文件，请将它们添加到 CLASSPATH 中。

4 要从命令行运行 EJB 检测器，请使用以下字符串：

java EJBDetector [ 搜索根目录 ] [ 侦听端口 ]

*搜索根目录*

一个或多个要在其中搜索 EJB 的目录或文件 （用分号分隔）。请遵循下列指南：
BEA WebLogic Servers 4.x 和 5.x：指定应用程序服务器根目录。
BEA WebLogic Servers 6.x：指定域文件夹的完整路径。
WebSphere Servers 3.x：指定已部署的 EJB 文件夹的完整路径。
WebSphere Servers 4.0：指定应用程序服务器根目录。
Oracle OC4J：指定应用程序服务器根目录。
Sun J2EE Server：指定可部署的 *.ear* 文件或包含多个 *.ear* 文件的目录的完整路径。

如果未指定，将搜索类路径。

*侦听端口*

EJB 检测器的侦听端口。默认端口是 2001。如果更改了此端口，则必须在 “生成 EJB 测试”对话框的 “主机名” 框中对其进行指定。

例如，如果主机是 metal，而且正在使用默认端口，则可指定 *metal*。如果使用的是其他端口，如端口 2002，请输入 *metal:2002*。

**要从批处理文件中运行 EJB 检测器，请执行下列操作：**

可使用批处理文件 *EJB_Detector.cmd* 来启动 EJB 检测器。解压缩 *ejbdetector.jar* 后，该文件将位于 EJB 检测器安装的根目录下。

**1** 打开 EJB 检测器根目录中的 env.cmd，然后根据环境修改下列变量：

| | |
|---|---|
| *JAVA_HOME* | JDK 安装的根目录 |
| *DETECTOR_INS_DIR* | 检测器安装的根目录 |
| *APP_SERVER_DRIVE* | 宿主应用程序服务器安装的驱动器 |
| *APP_SERVER_ROOT* | 请遵循下列指南：<br>BEA WebLogic Servers 4.x 和 5.x：指定应用程序服务器根目录。<br>BEA WebLogic Servers 6.x：指定域文件夹的完整路径。<br>WebSphere Servers 3.x：指定已部署的 EJB 文件夹的完整路径。<br>WebSphere Servers 4.0：指定应用程序服务器根目录。<br>Oracle OC4J：指定应用程序服务器根目录。<br>Sun J2EE Server：指定可部署的 *.ear* 文件或包含多个 *.ear* 文件的目录的完整路径。 |
| *EJB_DIR_LIST （可选）* | 用 “;” 分隔的目录 / 文件的列表，包含可部署的 .ear/.jar 文件，以及正在测试的 EJB 使用的任何其他类目录或 .jar 文件。 |

**2** 保存 *env.cmd*。

**3** 如果正在使用 EJB1.0 （Weblogic 4.x、 WebSphere 3.x），请向 env 文件的 CLASSPATH 中添加正在测试的 EJB 类以及下列供应商 EJB 类：

对于 WebLogic 4.x：<WebLogic 目录 >\lib\weblogicaux.jar

对于 WebSphere 3.x：<WebSphere 目录 >\lib\ujc.jar

**4** 运行 EJB_Detector.cmd 或 EJB_Detector.sh （Unix 平台）批处理文件以收集关于包含 EJB 的可部署应用程序的信息，例如：

```
C:\>EJB_Detector [listen_port]
```

其中 listen_port 是可选参数，用于指定 EJB 检测器侦听传入请求的端口号 （默认是 2001）。

## EJB 检测器输出和日志文件

可检查 EJB 检测器的输出以查看它是否已检测到所有活动的 EJB。输出日志将显示检查 EJB 的路径。在扫描结束时，它将显示找到的 EJB 列表、名称和位置。

例如：

```
检查 EJB 项：f:/weblogic/myserver/ejb_basic_beanManaged.jar...
检查 EJB 项：f:/weblogic/myserver/ejb_basic_statefulSession.jar...
检查 EJB 项：f:/weblogic/myserver/ejb_basic_statelessSession.jar...
------------------------ 找到 3 个 EJB --------------------------
 ** 路径 :f:/weblogic/myserver/ejb_basic_beanManaged.jar
       - BEAN:examples.ejb.basic.beanManaged.AccountBean
 ** 路径 :f:/weblogic/myserver/ejb_basic_statefulSession.jar
       - BEAN:examples.ejb.basic.statefulSession.TraderBean
 ** 路径 :f:/weblogic/myserver/ejb_basic_statelessSession.jar
       - BEAN:examples.ejb.basic.statelessSession.TraderBean
```

如果没有检测到任何 EJB （即 “找到 0 个 EJB”），请检查 “**检查 EJB 项 : ...**” 行中是否列出了 EJB jar 文件。如果没有列出，请检查 “搜索根目录”路径是否正确。如果正在检查但仍没有检测到任何 EJB，请检查这些 EJB jar 文件是否可以部署 （可以成功部署到应用程序服务器中）。可部署的 jar 文件应该包含 Home 接口、 Remote 接口、 Bean 实现、部署描述符文件 （xml 文件或 .ser 文件）和其他供应商特定的文件。

如果仍然有问题，则需要设置位于 DETECTOR_HOME\classes 目录下的 *detector.properties* 文件中的调试属性，以检索其他调试信息。

检测到 EJB 后， HTTP 服务器将进行初始化并等待 VuGen EJB 测试 Vuser 的请求。如果在此通信过程中出现问题，请启用位于 DETECTOR_HOME\classes 目录下的 *webserver.properties* 文件中的 webserver.enableLog 属性。

这样可以打印出其他调试信息和 *webserver.log* 文件中的其他可能重要的错误消息。

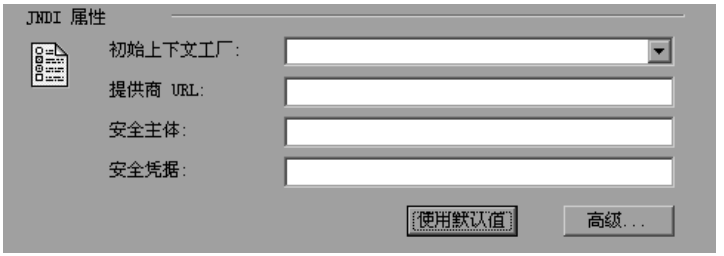
# 创建 EJB 测试 Vuser

要创建 EJB Vuser 脚本，请执行下列操作：

**1** 选择 “文件” > “新建” 或者单击 “新建” 按钮。将打开 “新建虚拟用户” 对话框。

**2** 从 “Enterprise Java Bean” 类别中选择 “EJB 测试”，然后单击 “确定”。VuGen 将打开空白的 Java Vuser 脚本并打开 “生成 EJB 脚本” 对话框。

**3** 指定要安装 LoadRunner EJB 检测器的计算机。注意，检测器必须在运行状态下才能进行连接。单击 “连接”。将启用 “JNDI 属性” 部分。




**4** EJB 检测器自动检测默认的 JNDI 属性。可在相应编辑框中手动修改这些属性。可以修改的属性是 “初始上下文工厂” 和 “提供商 URL” 的字符串。

如果应用程序服务器要求身份验证，请在 “安全主体” 框中输入用户名，在 “安全凭据” 框中输入密码。

下面是两个 JNDI 强制属性的默认值：

| 类型 | 初始上下文工厂 | 提供商 URL |
|---|---|---|
| WebLogic | weblogic.jndi.WLInitialContextFactory | t3://< 应用程序服务器主机 >:7001 |
| WebSphere 3.x | com.ibm.ejs.ns.jndi.CNInitialContextFactory | iiop://< 应用程序服务器主机 >:900 |
| WebSphere 4.x | com.ibm.websphere.naming.WsnInitial ContextFactory | iiop://< 应用程序服务器主机 >:900 |
| Sun J2EE | com.sun.enterprise.naming.SerialInitContextFactory | N/A |
| Oracle | com.evermind.server.AppplicationClientInitialContextFactory | ormi://< 应用程序服务器主机 >/< 应用程序名 >（<oc4j>/config/server.xml 中 EJB 的应用程序名） |

**5** 要设置 JNDI 的高级属性，请单击 “高级”，打开 “JNDI 高级属性”对话框。

指定所需的属性：对象工厂、状态工厂、 URL 包前缀、安全协议和安全身份验证。单击 “确定”。

**6** 在对话框的 “EJB” 部分，单击 “选择”，选择要为其创建测试的 EJB。将打开一个对话框，其中显示当前可从应用程序服务器获得的所有 EJB 的列表。

**7** 突出显示要测试的 EJB，再单击 “选择”。

**8** 在 “生成 EJB 脚本” 对话框中，单击 “生成”。 VuGen 通过 LoadRunner Java Vuser 函数创建脚本。脚本包含连接到应用程序服务器和执行 EJB 方法的代码。

**9** 保存脚本。

注意，无法在现有脚本内生成其他 EJB 的测试代码。要为其他 EJB 创建测试，请打开新的脚本并重复步骤 2-9。

## 设置 EJB 录制选项

EJB Vuser 可以使用的录制选项位于 “类路径” 和 “代码生成” 区域。有关类路径选项的信息，请参阅第 14 章 “设置 Java 录制选项”。

通过 “EJB 代码生成” 选项，可以设置自动事务和值检查区域中的属性。还可以指示存储初始化方法的位置。

**要设置 “EJB 代码生成”录制选项，请执行下列操作：**

1 单击 “开始录制”对话框中的 “选项”。选择 “录制选项”树中的 “EJB 选项：代码生成选项”节点，以编辑代码生成选项。

2 启用 “自动事务”选项，将所有 EJB 方法自动标记为事务。这将用 **lr.start_transaction** 和 **lr.end_transaction** 函数将所有方法括起来。默认情况下，此选项处于启用状态 (true)。

3 启用 “插入值检查”选项，自动将 **lr.value_check** 函数插入到每个 EJB 方法的结尾。该函数检查原始值和字符串的期望返回值。

4 选择 “EJB 初始化方法”。这种方法中将写入 EJB/JNDI 初始化属性。可用的方法为 *init*（默认）和 *action*。

# 了解 EJB Vuser 脚本

VuGen 将根据您在创建 Vuser 脚本时指定的 JNDI（Java 命名和目录接口）属性，生成用来测试 EJB 的脚本。JNDI 是 Sun 的编程接口，用于将 Java 程序连接到诸如 DNS 和 LDAP 的命名和目录服务。

每个 EJB Vuser 脚本都包含三个主要部分：

- 使用 JNDI 查找 EJB Home
- 创建实例
- 调用 EJB 方法

## 使用 JNDI 查找 EJB Home

脚本的第一部分包含用于检索 JNDI 属性的代码。它使用指定的上下文工厂和提供商 URL，来连接到应用程序服务器、查找指定的 EJB 和查找 EJB Home。

在下例中， JNDI 上下文工厂为 weblogic.jndi.WLInitialContextFactory，提供商 URL 为 t3://dod:7001 ，选定 EJB 的 JNDI 名称为 carmel.CarmelHome。

```
public class Actions
{

      public int init() {
           CarmelHome _carmelhome = null;
           try {
                     // 获取 JNDI 初始上下文
                     java.util.Properties p = new java.util.Properties();

p.put(javax.naming.Context.INITIAL_CONTEXT_FACTORY,
"weblogic.jndi.WLInitialContextFactory");
                     p.put(javax.naming.Context.PROVIDER_URL, "t3://dod:7001");
                     javax.naming.InitialContext _context = new
javax.naming.InitialContext(p);

                     // 在 JNDI 上下文中查找 Home 接口并使其变窄
                     Object homeobj = _context.lookup("carmel.CarmelHome");
                     _carmelhome =
(CarmelHome)javax.rmi.PortableRemoteObject.narrow(homeobj,
CarmelHome.class);

           } catch (javax.naming.NamingException e) {
                     e.printStackTrace();
           }
```

**注意**：如果通过在客户端 （而不是应用程序服务器）上运行的 EJB 检测器生成脚本，则必须手动修改提供商的 URL。例如，在下行中，提供商将 dod 指定为 EJB 检测器主机名：

```
p.put(javax.naming.Context.PROVIDER_URL, "t3://dod:7001")
```

使用应用程序服务器名替换录制的主机名，例如：

```
p.put(javax.naming.Context.PROVIDER_URL, "t3://bealogic:7001")
```

可以在录制前指定提供商 URL，这样就不必在 “生成 EJB 脚本”对话框的 “JDNI 属性”部分中对其进行手动修改。

## 创建实例

执行 EJB 方法前，脚本将为 EJB 创建 Bean 实例。可将实例的创建标记为事务，这样即可在执行脚本后对其进行分析。另外，创建实例的过程将包装在一个 *try and catch* 块中，以便进行异常处理。

对于会话 Bean – 使用 EJB Home 的 “create”方法来获得新的 EJB 实例。

在下例中，脚本为 Carmel EJB 创建实例。

```
// 创建 Bean 实例
Carmel _carmel = null;
try {
        lr.start_transaction("create");
        _carmel = _carmelhome.create();
        lr.end_transaction("create", lr.AUTO);
} catch (Throwable t) {
        lr.end_transaction("create", lr.FAIL);
        t.printStackTrace();
}
```

对于实体 Bean – 使用 findByPrimaryKey 方法在现有数据库中查找 EJB 实例，如果找不到，则使用 create 方法在那里创建实例。

在下例中，脚本尝试查找帐户 EJB 的实例，如果失败，则创建实例。

```
// 查找 Bean 实例
try {
      com.ibm.ejs.doc.account.AccountKey _accountkey = new
com.ibm.ejs.doc.account.AccountKey();
      _accountkey.accountId = (long)0;

      lr.start_transaction("findByPrimaryKey");
      _account = _accounthome.findByPrimaryKey(_accountkey);
      lr.end_transaction("findByPrimaryKey", lr.AUTO);
} catch (Throwable thr) {

      lr.end_transaction("findByPrimaryKey", lr.FAIL);
      lr.message("Couldn't locate the EJB object using a primary
key.Attempting to manually create the object...["+thr+"]");

      // 创建 Bean 实例
      try {
           lr.start_transaction("create");
           _account =
_accounthome.create((com.ibm.ejs.doc.account.AccountKey)null);
           lr.end_transaction("create", lr.AUTO);
      } catch (Throwable t) {
           lr.end_transaction("create", lr.FAIL);
           t.printStackTrace();
      }

}
```

可使用由实体 Bean 提供的其他 find... 方法来查找 EJB 实例。例如：

```
// 获得代表数据库中的名称 “John” 的
// 所有电子邮件 EJB 实例的枚举列表。
Enumeration enum = home.findByName("John");
      while (enum.hasMoreElements()) {
         Email addr = (Email)enum.nextElement();
           ...
}
```

### 调用 EJB 方法

脚本的最后部分包含了 EJB 的实际方法。每个方法都标记为事务，这样即可在运行脚本后对其进行分析。另外，每个方法都包装在一个 *try and catch* 块中，以便进行异常处理。出现异常时，事务将被标记为 “失败”，脚本将继续下一个方法。VuGen 将为每个 EJB 方法创建单独的块。

```
// ------- 方法 ------------

        int _int1 = 0;
        try {
                lr.start_transaction("buyTomatoes");
                _int1 = _carmel.buyTomatoes((int)0);
                //lr.value_check(_int1, 0, lr.EQUALS);
                lr.end_transaction("buyTomatoes", lr.AUTO);
        } catch (Throwable t) {
                lr.end_transaction("buyTomatoes", lr.FAIL);
                t.printStackTrace();
        }
```

VuGen 将插入方法的默认值，例如，为整数插入 0，为字符串插入空字符串 (""), 为复杂 Java 对象插入 NULL。如有必要，可以在生成的脚本内修改默认值。

```
                _int1 = _carmel.buyTomatoes((int)0);
```

下例显示如何使用参数化来更改非基元类型的默认值：

```
Detail details = new Details(<city>,<street>,<zip>,<phone>);
JobProfile job = new JobProfile(<department>,<position>,<job_type>);
Employee employee=new Employee(<first>,<last>, details, job, <salary>);
_int1 = _empbook.addEmployee((Employee)employee);
```

对于返回原始的非复杂值或字符串的方法， VuGen 将插入已注释的方法 **lr.value_check**。通过该 LoadRunner 方法可以指定 EJB 方法的期望值。要使用该验证方法，请删除注释标记 (//) 并指定期望值。例如， *carmel.buyTomatoes* 方法将返回整数值。

```
                _int1 = _carmel.buyTomatoes((int)0);
                //lr.value_check(_int1, 0, lr.EQUALS);
```

如果希望该方法返回值 500，请按如下方式修改代码：

```
_int1 = _carmel.buyTomatoes((int)0);
lr.value_check(_int1, 500, lr.EQUALS);
```

如果要检查方法是否返回了某个值，请按如下方式修改代码：

```
_int1 = _carmel.buyTomatoes((int)0);
lr.value_check(_int1, 10, lr.NOT_EQUALS);
```

如果没有检测到期望值，则将出现异常，输出窗口中将记录相关信息。

```
System.err:java.lang.Exception:lr.value_check failed.[Expected:500 Actual:5000]
```

EJB Vuser 脚本 支持所有标准的 Java 约定。例如，可通过在文本前面加上两个正斜杠 “//” 来插入注释。

Java Vuser 脚本作为可伸缩的多线程应用程序运行。如果脚本中包括自定义类，请确保代码是线程安全的。非线程安全的代码可能导致结果不准确。对于非线程安全的代码，请将 Java Vuser 作为进程来运行。（请参阅 “运行时设置”）这将为每个进程创建单独的 Java 虚拟机，导致脚本的可伸缩性降低。

## 运行 EJB Vuser 脚本

为 EJB 测试生成脚本并进行必要的修改后，即可运行脚本。通过 EJB 脚本可以执行两种类型的测试：功能测试和负载测试。功能测试验证 EJB 在环境中是否能正常工作。通过负载测试，可以在一次执行多个用户的情况下评估 EJB 的性能。

**要运行功能测试，请执行下列操作：**

1. 修改自动生成的默认值。
2. 使用 **lr.value_check** 方法插入值检查。这些方法在脚本中以注释的方式生成（请参阅第 548 页的 “调用 EJB 方法”）。
3. 插入其他方法并修改其默认值。（请参阅第 13 章 “录制 Java 语言 Vuser 脚本”中有关插入 Java 函数的部分）。
4. 设置脚本的常规运行时设置。详细信息，请参阅第 9 章 “配置运行时设置”。

**5** 设置 Java 虚拟机运行时设置：指定所有其他的类路径和其他虚拟机参数。确保包括应用程序服务器 EJB 类。测试的 EJB 的实际类保存在 Vuser 目录下，而且将在回放期间自动检索这些类。有关指定其他类路径和设置 Java 虚拟机运行时设置的信息，请参阅第 16 章 “配置 Java 运行时设置”。

**6** 对于 Websphere 3.x 用户：

使用 IBM JDK 1.2 或更高版本：

➤ 向类路径中添加 <WS>\lib\ujc.jar。

使用 Sun JDK 1.2.x：

➤ 删除文件 <JDK>\jre\lib\ext\iiimp.jar

➤ 将以下文件从 <WS>\jdk\jre\lib\ext 文件夹复制到 <JDK>\jre\lib\ext 目录：iioprt.jar, rmiorb.jar。

➤ 将 “ujc.jar” 从 <WS>\lib 文件夹复制到 <JDK>\jre\lib\ext。

➤ 将文件 <WS>\jdk\jre\bin\ioser12.dll 复制到 <JDK>\jre\bin 文件夹。

其中 <*WS*> 是 WebSphere 安装的主文件夹， <*JDK*> 是 JDK 安装的主文件夹。

清除 “使用 -Xbootclasspath VM 参数”复选框，关闭此选项。

**7** 对于 WebSphere 4.0 用户：

请确保 IBM JDK1.3 计算机上具有有效的 Java 环境。打开 “运行时设置”对话框并选择 “Java 虚拟机”节点。向 “其他类路径”部分中添加下列项：

```
<WS>/lib/webshpere.jar;
<WS>/lib/j2ee.jar;
```

其中 <WS> 是 WebSphere 安装的主目录。

清除 “使用 -Xbootclasspath VM 参数”复选框，关闭此选项。

---

**注意**：如果应用程序服务器安装在 UNIX 计算机上，或者如果正在使用 WebSphere 3.0.x，则必须在客户端计算机上安装 IBM JDK 1.2.x，以获得所需的文件。

---

**8** 对于 Oracle OC4J 用户：

请确保 JDK1.2 或更高版本 （最好是 JDK1.3）的计算机上具有有效的 Java 环境。打开 “运行时设置”对话框并选择 “Java 虚拟机”节点。向 “其他类路径”部分中添加下列项：

```
<OC4J>/orion.jar;<OC4J>/ejb.jar;<OC4J>/jndi.jar; ;<OC4J>/xalan.jar;
<OC4J>/crimson.jar
```

其中 <OC4J> 是应用程序服务器安装的主文件夹。

**9** 对于 Sun J2EE 用户：

请确保 JDK1.2 或更高版本的计算机上具有有效的 Java 环境。打开 “运行时设置”对话框并选择 “Java 虚拟机”节点。向 “其他类路径”部分中添加下列项：

```
<J2EE>/j2ee.jar;<AppClientJar>
```

其中 <J2EE> 是应用程序服务器安装的主文件夹， <AppClientJar> 是在部署过程中由 sdk 工具自动创建的应用程序客户端 jar 文件的完整路径。

**10** 对于 WebLogic 4.x - 5.x 用户：

请确保计算机上具有有效的 Java 环境 （路径和类路径）。打开 “运行时设置”对话框并选择 “Java 虚拟机”节点。向 “其他类路径”部分中添加下列两项：

```
<WL>/classes;<WL>/lib/weblogicaux.jar
```

其中 <*WL*> 是 WebLogic 安装的主文件夹。

**11** 对于 WebLogic 6.x 和 7.0 用户：

请确保计算机上具有有效的 Java 环境 （路径和类路径）。 WebLogic 6.1 要求 JDK 1.3。打开 “运行时设置”对话框并选择 “Java 虚拟机”节点。向 “其他类路径”部分中添加以下项：

```
<WL>/lib/weblogic.jar;   // Weblogic 6.x
<WL>/server/lib/weblogic.jar   // Weblogic 7.x
```

其中 <*WL*> 是 WebLogic 安装的主文件夹。

清除 “使用 -Xbootclasspath VM 参数”复选框，关闭此选项。

**12** 运行脚本。单击“运行”按钮或者选择“Vuser”>“运行”。查看“执行日志”节点以查看任何运行时错误。执行日志存储在脚本文件夹的 *mdrv.log* 文件中。Java 编译器（Sun 的 javac）将检查其中的错误并编译脚本。

验证 EJB 运行正常后，可以执行负载测试，方法是将它分配给负载场景中的多个 Vuser。详细信息，请参阅《LoadRunner Controller 用户指南》。

# 第 X 部分

## ERP/CRM 协议

# 47

# 创建 Oracle NCA Vuser 脚本

可以使用 VuGen 来创建模拟 Oracle NCA 用户的脚本。可以使用 VuGen 录制典型的 NCA 业务流程，然后运行脚本来模拟与系统进行交互的用户。

本章描述：

- Oracle NCA Vuser 入门
- 录制指南
- 启用按名称录制对象的功能
- 通过个人主页启动 Oracle 应用程序
- 使用 Oracle NCA Vuser 函数
- 了解 Oracle NCA Vuser
- 在树视图和脚本视图之间切换
- 配置运行时设置
- 测试 Oracle NCA 应用程序
- 关联 Oracle NCA 语句以实现负载平衡
- 以 Pragma 模式录制

***以下信息仅适用于 Oracle NCA 协议。***

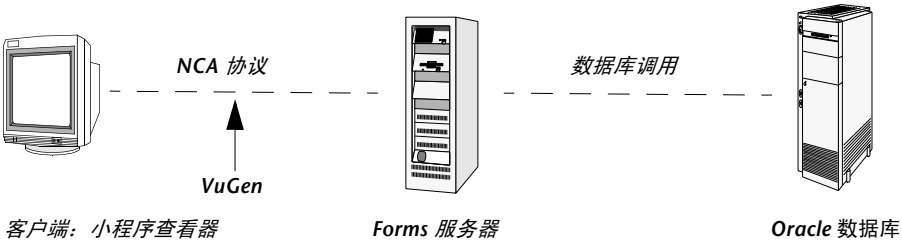
# 关于创建 Oracle NCA Vuser 脚本

Oracle NCA 是一种基于 Java 的数据库协议。使用浏览器，可以启动数据库客户端 （一个小程序查看器）。可以通过 NCA 数据库的小程序查看器对其执行操作，从而无需使用客户端软件，并可以从所有支持小程序查看器的平台执行数据库操作。有一个专门为模拟 Oracle NCA 客户端而设计的 Vuser 类型。

NCA 环境是一个三层环境。用户首先将一个 HTTP 调用从其浏览器发送到 Web 服务器。此调用将访问调用 Oracle 应用程序小程序的启动 HTML 页。该小程序在客户机上本地运行，所有后续调用都通过专用的 NCA 协议在客户端和 Forms 服务器之间传递。

客户端 （小程序查看器）与应用程序服务器 （Oracle Forms 服务器）进行通信，然后该服务器将信息提交到数据库服务器 (Oracle 8.x)。

VuGen 录制并回放客户端和 Forms 服务器 （应用程序服务器）之间的 NCA 通信。




在录制 Oracle NCA 会话时， VuGen 将录制所有 NCA 和 Web 操作 （即使只创建了一个单协议脚本）。如果事先知道 Web 函数对于测试的重要性，请从一开始就为 Oracle NCA 和 Web 协议创建一个多协议脚本。

如果最初为 Oracle NCA 创建了一个单协议脚本，并且后来需要使用 Web 函数来进行测试，则不必重新录制会话即可在 VuGen 中重新生成脚本，以添加 Web 函数。可以通过 “重新生成 Vuser” 对话框中的 “协议” 节点来指明此操作。详细信息，请参阅第 3 章 “使用 VuGen 进行录制”。

# Oracle NCA Vuser 入门

以下过程概述了如何创建 Oracle NCA Vuser 脚本。

**1 确保录制计算机的配置正确。**

在启动 VuGen 之前，确保计算机已配置为运行 Oracle NCA 小程序查看器。还必须确保 LoadRunner 支持您的 Oracle Forms 版本。详细信息，请参阅第 558 页的 “录制指南”和自述文件。

**2 创建骨架 Oracle NCA Vuser 脚本。**

使用 VuGen 来创建骨架 Oracle NCA Vuser 脚本。详细信息，请参阅第 26 页的 “Vuser 脚本部分”。

**3 录制典型的用户操作。**

开始进行录制，并通过小程序查看器执行典型操作和业务流程。 VuGen 将录制您的操作并生成 Vuser 脚本。详细信息，请参阅第 28 页的 “新建虚拟用户脚本”。

**4 增强 Vuser 脚本。**

为了增强 Vuser 脚本，可以使用 “插入”菜单来添加事务、集合点、注释和消息。详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

**5 参数化脚本。**

用参数替换录制的常量。详细信息，请参阅第 7 章 “定义参数”。

**6 设置脚本的运行时属性。**

配置 Vuser 脚本的运行时设置。运行时设置定义了脚本执行的某些特定方面。详细信息，请参阅第 9 章 “配置运行时设置”。

**7 保存和运行 Vuser 脚本。**

通过 VuGen 运行脚本，并查看执行日志，了解运行时信息。详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

## 录制指南

录制 Oracle NCA Vuser 脚本时，请遵循下列指导原则：

➤ 指定在录制 Oracle NCA 会话时 VuGen 要使用的浏览器。在 “开始录制”对话框中的 “要录制的程序”列表内选择所需的浏览器。该列表包含所有可用浏览器。

➤ 在开始录制之前，请关闭所有浏览器。

➤ 在 *vuser_init* 部分录制登录过程。在 *Actions* 部分录制典型的业务流程。运行脚本时，可以为特定业务流程指定多个循环。详细信息，请参阅第 28 页的 “新建虚拟用户脚本”。

```
vuser_init()
{
connect_server("199.203.78.170", "9000"/*version=107*/,"
      module=e:\\appsnca\\fnd\\7.5\\forms\\us\\fndscsgn
      userid=applsyspub/pub@vision fndnam=apps");
edit_set("FNDSCSGN.SIGNON.USERNAME.0","VISION");
edit_set("FNDSCSGN.SIGNON.PASSWORD.0","WELCOME");
button_press("FNDSCSGN.SIGNON.CONNECT_BUTTON.0");
lov_retrieve_items("Responsibilities",1,17);

      return 0;
}
```

➤ 由于 Netscape 的局限性，当计算机上已有另一个 Netscape 浏览器正在运行时，不能在 Netscape 中启动 Oracle NCA 会话。

➤ VuGen 支持使用 Forms 侦听器 Servlet 以多协议模式录制 Oracle Forms 应用程序。在 Oracle Forms 中，应用程序服务器使用 Forms 侦听器 Servlet 来为每个客户端创建运行时进程。运行时进程称为 Forms 服务器运行时进程，它维护与客户端的永久连接，并且与服务器之间相互发送信息。

要在回放过程中支持 Forms 4.5，请设置 *mdrv.dat* 文件中的以下各项：

```
[Oracle_NCA]
ExtPriorityType=protocol
WINNT_EXT_LIBS=ncarp110.dll
WIN95_EXT_LIBS=ncarp110.dll
LINUX_EXT_LIBS=liboranca.so
SOLARIS_EXT_LIBS=liboranca.so
HPUX_EXT_LIBS=liboranca.sl
AIX_EXT_LIBS=liboranca.so
LibCfgFunc=oracle_gui_configure
UtilityExt=lrun_api
```

要还原 Forms 6 或 9 支持，请还原原始设置。

## 启用按名称录制对象的功能

录制 Oracle NCA 脚本时，必须使用对象名（而不是标准对象 ID）来录制会话。如果使用对象 ID 录制脚本，则回放将失败，因为对象 ID 是由服务器动态生成的，并且在录制和回放时不同。通过检查 nca_connect_server 语句，可以验证脚本是否是使用对象名来录制的。

```
nca_connect_server("199.35.107.119","9002"/*version=11i*/,"module=/d1/ora
cle/visappl/fnd/11.5.0/forms/US/FNDSCSGN userid=APPLSYSPUB/PUB@VIS
fndnam=apps record=names ");
```

如果在 **nca_connect_server** 函数中未出现 **record=names** 参数，则录制的是对象 ID。通过修改下列项目之一可以指示 VuGen 录制对象名：

➤ 启动 HTML 文件

➤ 要录制的 URL

➤ 配置文件

注意，将在 Oracle Forms6i Patch 9 （Oracle Forms 版本：6.0.8.18.3）中引入捕获所有对象的开发人员名称的功能。在 Oracle Forms 6i Patch 9 发布之前编写的测试启动程序工具包 (Test Starter Kit) 脚本中，未将开发人员名称作为对象物理描述的一部分 （编辑字段除外）。

### 启动 HTML 文件

如果拥有对启动 HTML 文件的访问权限，则通过在启动文件 （在启动 Oracle NCA 应用程序时加载的文件）中设置 **record=names** 标志，可以指示 VuGen 录制对象名而不是对象 ID。

编辑小程序查看器开始运行时调用的启动文件。修改以下命令行：

<PARAM name="serverArgs ... fndnam=APPS">

并添加 Oracle 关键字 "record=names"：

<PARAM name="serverArgs ... fndnam=APPS **record=names**">

### 要录制的 URL

如果没有对启动 HTML 文件的访问权限，则通过修改要录制的 URL，仍然可以让 Oracle NCA 录制对象名而不是对象 ID。仅当启动 HTML 文件在加载时未引用其他文件时，以下解决方案才适用。

对于此解决方案，需要在 “开始录制”对话框中的 URL 后面 （在要录制的 URL 名称之后）添加 "**?record=names**"。这将使 VuGen 可以录制该会话的对象名。

### Forms 配置文件

如果应用程序具有引用 Forms Web CGI 配置文件 *formsweb.cfg* 的启动 HTML 文件 （一种常见的引用），则在向该启动文件中添加 **record=names** 时可能会遇到问题。

在这种情况下，应该修改配置文件。

**要修改配置文件以录制对象名，请执行下列操作**

1 转至 Forms Web CGI 配置文件。

2 在该文件中定义一个新参数 （请参阅下面的 Web CGI 配置文件示例以了解此更改）。

```
serverApp=forecast
serverPort=9001
serverHost=easgdev1.dats.ml.com
connectMode=socket
archive=f60web.jar
archive_ie=f60all.cab
xrecord=names
```

3 打开启动 HTML 文件，并查找 PARAM NAME="serverArgs"。

4 将变量名作为参数添加到 ServerArgs 参数中，例如 **record=%xrecord%**。

```
<PARAM NAME="serverArgs" VALUE="module=%form% userid=%userid%
%otherParams% record=%xrecord%">
```

5 还可以选择编辑 Oracle Forms 安装目录中的 *basejini.htm* 文件。此文件是用于在 Web 上运行表单的默认 HTML 文件，它使用 JInitiator 样式的标记来包含 Forms 小程序。在 basejinin.hmt 文件中，向参数定义中添加以下命令行：

```
<PARAM NAME="recordFileName" VALUE="%recordFileName%">
```

在 <EMBED> 标记中，添加以下命令行：

```
...
serverApp="%serverApp%"
logo="%logo%"
imageBase="%imageBase%"
formsMessageListener="%formsMessageListener%"
recordFileName="%recordFileName%"
```

编辑该文件（而不是 Servlet 配置文件 *formsweb.cfg*）的缺点是在重新安装 Oracle Forms 时将替换该文件。为了避免出现这种情况，可以创建 *basejini.htm* 文件的副本，并将其存储在其他位置。在 Servlet 配置文件中，编辑 *baseHTMLJinitiator* 参数以指向新文件。

## 通过个人主页启动 Oracle 应用程序

通过登录**个人主页**启动 Oracle Forms 6i 应用程序时，必须在用户级别设置几个系统配置文件选项。在用户级别传递此类变量是可取的方法，如果在站点级别进行传递，将会影响所有用户。

**要配置 “ICX: 表单启动程序”配置文件，请执行下列操作：**

1. 登录到应用程序，并选择 “系统管理员”身份的职责。
2. 从 “导航”菜单中选择 “配置文件 / 系统”。
3. 在 “查找系统配置文件值”窗体中，执行下列操作：

   选择 “显示 : 站点”选项

   “用户” = < *用户登录名* >（例如， operations、 mfg 等）

   “输入配置文件” =%ICX%Launch%

   单击 “查找”。
4. 更新 “ICX: 表单启动程序”配置文件的用户值：

   如果没有将任何参数传递给 URL，则在用户值的 URL 的末尾追加以下字符串：**?**play=&record=names

   如果已将参数传递给 URL，则在用户值的 URL 的末尾追加以下字符串：**&**play=&record=names
5. 保存事务。
6. 注销 Oracle Forms 会话。
7. 注销个人主页会话。
8. 使用您的用户名通过个人主页重新登录。

如果无法在用户级别更新 “ICX: 表单启动程序”配置文件选项，请打开 “应用程序开发人员”身份的职责，然后为 ICX_FORMS_LAUNCHER 配置文件选择 “可更新”选项。

传递给 URL 的第一个参数必须以问号 (?) 开头。用 & 号传递所有后续参数。在大多数情况下， URL 已经包含您可以通过搜索问号加以识别的参数。

# 使用 Oracle NCA Vuser 函数

在执行典型的 NCA 业务流程时， VuGen 将自动录制本部分列出的大部分函数。将使用 **nca** 前缀录制函数。（仍然支持在较低版本的 VuGen 中未使用 **nca** 前缀录制的 NCA 函数。）还可以手动将任一函数写到您的 Vuser 脚本中。在树视图中工作时，请单击相应步骤的图标。在文本视图中，可以手动添加所需的函数。有关 Oracle NCA Vuser 函数的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

### 按钮对象函数

| | |
|---|---|
| **nca_button_double_press** | 在按钮上双击 |
| **nca_button_press** | 激活按钮 |
| **nca_button_set** | 设置指定按钮的状态 |

### 组合框对象函数

| | |
|---|---|
| **nca_combo_select_item** | 在组合框中选择项 |
| **nca_combo_set_item** | 在组合框中设置新项 |

### 连接函数

| | |
|---|---|
| **nca_connect_server** | 连接到 Oracle NCA 服务器 |
| **nca_logon_connect** | 登录到 Oracle NCA 数据库 |
| **nca_logon_cancel** | 与 Oracle NCA 数据库断开连接 |

### 编辑对象函数

| | |
|---|---|
| **nca_edit_box_press** | 单击编辑框消息 |
| **nca_edit_click** | 单击编辑对象 |
| **nca_edit_get_text** | 返回编辑对象中的文本 |
| **nca_edit_press** | 激活编辑字段中的浏览按钮 |
| **nca_edit_set** | 替换编辑对象的全部内容 |

### Flex 对象函数

| | |
|---|---|
| **nca_flex_click_cell** | 单击 Flexfield 窗口中的表单元格 |
| **nca_flex_get_cell_data** | 获取 Flexfield 单元格中的数据 |
| **nca_flex_get_column_name** | 获取 Flexfield 窗口中的列名 |
| **nca_flex_press_clear** | 单击 Flexfield 窗口中的 “清除” |
| **nca_flex_press_find** | 单击 Flexfield 窗口中的 “查找” |
| **nca_flex_press_help** | 单击 Flexfield 窗口中的 “帮助” |
| **nca_flex_press_lov** | 单击 Flexfield 窗口中的 “值列表” 按钮 |
| **nca_flex_press_ok** | 单击 Flexfield 窗口中的 “确定” |
| **nca_flex_set_cell_data** | 在 Flexfield 窗口中插入数据 |
| **nca_flex_set_cell_data _press_ok** | 输入数据之后，单击 Flexfield 窗口中的 “确定” |

### 列出项函数

| | |
|---|---|
| **nca_list_activate_item** | 激活列表中的项 （双击） |
| **nca_list_select_index_item** | 按照索引选择列表项 |
| **nca_list_select_item** | 按照名称选择列表项 |
| **nca_lov_auto_select** | 指定项的第一个字母 |
| **nca_lov_find_value** | 单击 “值列表” 窗口中的 “查找” |
| **nca_lov_get_item_name** | 按照项的索引编号检索值列表中的项的名称 |
| **nca_lov_retrieve_items** | 检索值列表 |
| **nca_lov_select_index_item** | 按照索引编号从值列表中选择项 |
| **nca_lov_select_item** | 从值列表中选择项 |
| **nca_lov_select_random_item** | 从值列表中选择随机项 |

### Java 对象函数

| | |
|---|---|
| **nca_java_action** | 在 Java 对象上执行事件 |
| **nca_java_get_value** | 检索 Java 对象的值 |
| **nca_java_set_reply_property** | 设置 Java 对象的 reply 属性 |

### 菜单对象函数

| | |
|---|---|
| **nca_menu_select_item** | 从菜单中选择项 |

### 消息函数

| | |
|---|---|
| **nca_popup_message_press** | 单击弹出窗口中的按钮 |
| **nca_message_box_press** | 单击消息窗口中的按钮 |

### 对象函数

| | |
|---|---|
| **nca_obj_get_info** | 返回对象属性的值 |
| **nca_obj_mouse_click** | 单击对象 |
| **nca_obj_mouse_dbl_click** | 双击对象 |
| **nca_obj_status** | 返回指定对象的状态 |
| **nca_obj_type** | 在编辑框中键入特殊字符 |

### 响应对象函数

| | |
|---|---|
| **nca_response_press_lov** | 单击 “响应”框中的下拉箭头 |
| **nca_response_press_ok** | 单击 “响应”框中的 “确定” |
| **nca_response_set_cell_data** | 向 “响应”框中的单元格中插入数据 |
| **nca_response_set_data** | 向 “响应”框中插入数据 |

### 滚动对象函数

| | |
|---|---|
| **nca_scroll_drag_from_min** | 将滚动条从最小化位置 (0) 拖到指定距离处 |
| **nca_scroll_line** | 滚动指定的行数 |
| **nca_scroll_page** | 滚动指定的页数 |

### 会话函数

| | |
|---|---|
| **nca_console_get_text** | 检索控制台消息 |
| **nca_set_iteration_offset** | 设置对象 ID 的偏移值 |
| **nca_set_server_response_time** | 设置服务器响应时间 |
| **nca_set_exception** | 指定处理异常的方式 |
| **nca_set_think_time** | 设置思考时间范围 |

### 树对象函数

| | |
|---|---|
| **nca_tree_activate_item** | 激活 NCA 树中的项 |
| **nca_tree_collapse_item** | 折叠树项 |
| **nca_tree_expand_item** | 展开树项 |
| **nca_tree_select_item** | 选择 NCA 树中的项 |

### 窗口对象函数

| | |
|---|---|
| **nca_win_get_info** | 返回窗口属性的值 |
| **nca_win_close** | 关闭窗口 |
| **nca_set_window** | 指明当前窗口的名称 |

可以使用 C Vuser 函数 （如 **lr_output_message** 和 **lr_rendezvous**）进一步增强脚本。有关如何使用这些函数的信息，请参阅第 6 章 “增强 Vuser 脚本”。

# 了解 Oracle NCA Vuser

创建 Oracle NCA Vuser 脚本时，VuGen 将录制客户端和应用程序服务器之间的所有 NCA 通信。在录制的时候，VuGen 将生成上下文相关函数。这些函数利用 GUI 对象 （如窗口、列表和按钮）来描述对数据库执行的操作。在进行录制时，VuGen 会将上下文相关函数插入到 Vuser 脚本中。

录制完成后，可以修改脚本中的函数或添加其他函数来增强脚本。有关增强 Vuser 脚本的信息，请参阅第 6 章 “增强 Vuser 脚本”。有关可用 Oracle NCA Vuser 函数的列表，请参阅第 563 页的 “使用 Oracle NCA Vuser 函数”。有关这些函数的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

在以下程序段中， 用户将执行以下操作 ： 从列表中选择一项 (**nca_list_activate_item**)、 按下一个按钮 (**nca_button_press**)、检索一个列表值 (**nca_lov_retrieve_items**) 以及在编辑字段中执行单击 (**nca_edit_click**)。对象的逻辑名就是这些函数的参数。

```
...
nca_lov_select_item("Responsibilities","General Ledger, Vision
Operations");
nca_list_activate_item("FNDSCSGN.NAVIGATOR.LIST.0","+  Journals");
nca_list_activate_item("FNDSCSGN.NAVIGATOR.LIST.0","       Enter");
nca_button_press("GLXJEENT.TOOLBAR.LIST.0");
nca_lov_find_value("Batches","");
nca_lov_retrieve_items("Batches",1,9);
nca_lov_select_item("Batches","AR 1020 Receivables 2537:A 1020");
nca_edit_click("GLXJEENT.FOLDER_QF.BATCH_NAME.0");
...
```

在某些测试 （如在 Oracle Configurator 应用程序上执行的测试）中，整个会话都需要使用一个函数返回的信息。通过在脚本中插入 **web_reg_save_param** 函数， VuGen 可以将动态信息自动保存到参数中。在下例中，连接信息被保存到名为 *NCAJServSessionID* 的参数中。

```
web_reg_save_param ("NCAJServSessionId", "LB=\r\n\r\n", "RB=\r",
        LAST);
web_url("f60servlet",
        "URL=http://usscifforms05.sfb.na/servlet/f60servlet\?config
        =mult", LAST);
```

在上例中，右边界是 \r。实际的右边界可能因系统而异。

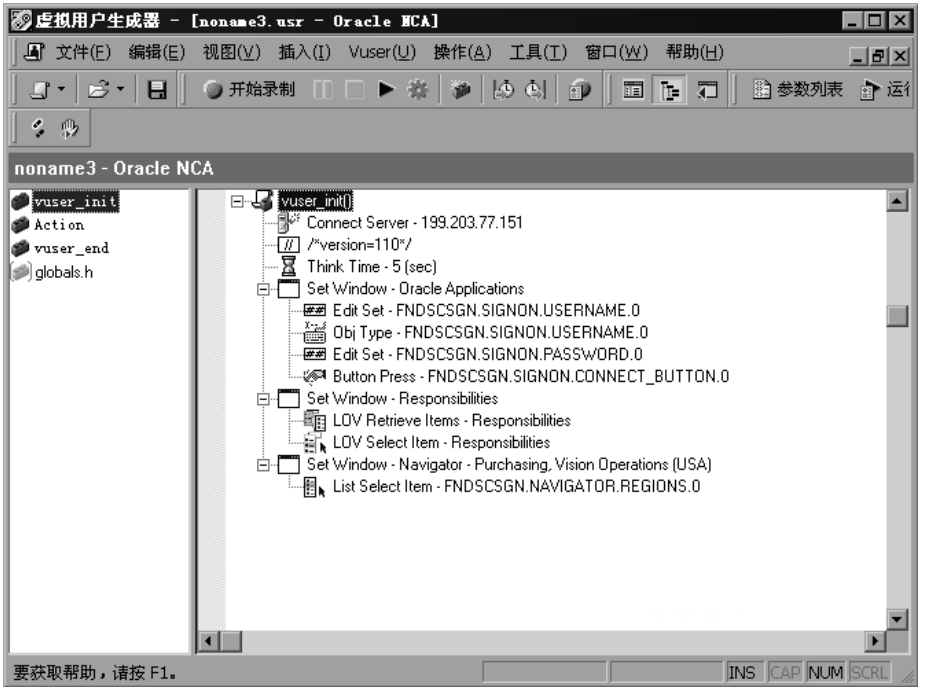
# 在树视图和脚本视图之间切换

在 VuGen 中查看和编辑 Oracle NCA Vuser 脚本时，可以选择在基于图标的树视图中或基于文本的脚本视图中查看脚本。

### 要显示 Oracle NCA Vuser 脚本的树视图，请执行下列操作

从 VuGen 主菜单中，选择 “视图” > “树视图”，或者单击 “按树查看脚本” 图标。 Vuser 脚本将显示在基于图标的树视图中。如果已经处于树视图中，该菜单项将被禁用。

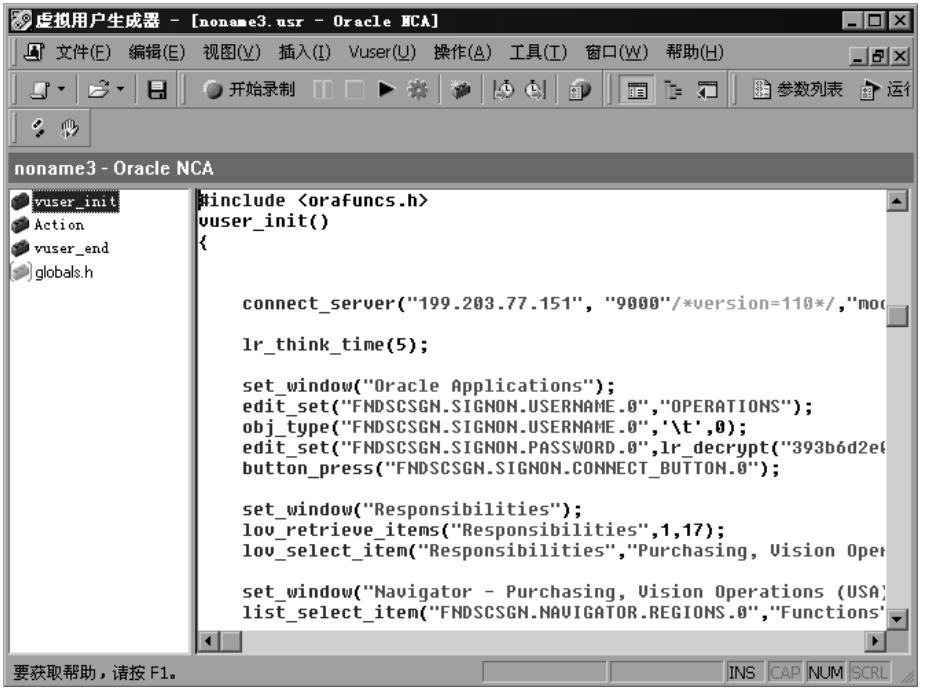
*树视图*



**要显示脚本视图，请执行下列操作**

从 VuGen 主菜单中，选择 “视图” > “脚本视图”，或者单击 “按文本查看脚本”图标。 Vuser 脚本将显示在基于文本的脚本视图中。如果已经处于脚本视图中，该菜单项将被禁用。

*脚本视图*




## 配置运行时设置

在运行脚本之前，可以设置运行时设置，以使脚本能够准确模拟实际用户。有关所有协议的常规运行时设置 （如思考时间、步和日志记录）的信息，请参阅第 9 章 “配置运行时设置”。有关与网络速度相关的设置的信息，请参阅第 10 章 “配置网络运行时设置”。

以下部分描述了特定于 Oracle NCA Vuser 的运行时设置。通过这些运行时设置可以指明通信参数。

### 客户端仿真运行时设置

可以配置多种网络设置，以准确模拟 Oracle NCA 客户端。

可以设置下列选项：

**网络 – 套接字超时**：Oracle NCA Vuser 等待服务器响应的时间。默认值 -1 用于禁用超时，客户端将无限期地等待。

**Pragma 模式**：在 Pragma 模式中，将以 Oracle 定义的 Pragma 模式执行通信。该通信级别位于 HTTP 和 Servlet 级别之上，其特征是定期发送消息。在该模式中，客户端可识别到服务器无法立即使用数据做出响应。服务器将以指定的时间间隔发送消息，直至它能够发送请求的数据。

**最大重试次数**：表示在出现错误之前，客户端将从服务器接受的 *IfError* 消息的最大数目。 IfError 消息是服务器定期发送到客户端的消息，内容指明它将尽可能快地使用数据做出响应。

“重试间隔”定义了出现 *IfError* 消息时的重试时间间隔。

**事务处理中包括重试间隔**：包含重试时间间隔，将其作为事务处理持续时间的一部分。

有关以 Pragma 模式进行录制的信息，请参阅第 577 页的 “以 Pragma 模式录制”。

**启用心跳线**：可以启用或禁用发送到 Oracle 服务器的心跳线。心跳线用于验证与服务器的通信是否正常。如果 Oracle NCA 服务器上有密集的负载，请禁用心跳线。如果启用心跳线，则可以设置向服务器发送心跳线消息的频率。

**表单版本**：在录制过程中检测到的 Oracle Forms 服务器的版本。仅当自录制以来对服务器进行了升级时，才修改此设置。

**要设置 “客户端仿真”设置，请执行下列操作：**

1 打开 “运行时设置”对话框。选择 “Vuser” > “运行时设置”，或者单击 VuGen 工具栏上的 “运行时设置”按钮。要从 LoadRunner Controller 打开 “运行时设置”对话框，请单击 “运行时设置”按钮。

2 从 “运行时设置”树中选择 “Oracle NCA: 客户端仿真”节点。

3 设置网络超时值 （秒）。要指示客户端无限期地等待服务器响应，请使用默认值 -1。

4 以 Pragma 模式执行操作时，应指定在出现错误之前客户端接受的重试次数 “最大重试次数”（即所接受的 *IfError* 消息的数目）。默认值是 20。

5 要将心跳线发送到 Oracle NCA 服务器，请选择 “启用心跳线”选项。在下一行中，指定发送心跳线的频率 （秒）。默认值是 120 秒。

6 单击 “确定”接受设置并运行脚本。

## 测试 Oracle NCA 应用程序

以下部分包含有关测试安全 Oracle NCA 应用程序和 Servlet 的几点提示。

### 测试安全的 Oracle NCA 应用程序

➤ 在选择要录制的协议时，只需要从协议列表中选择 “Oracle NCA”，而不需要选择 “Web 协议”。 VuGen 在内部录制安全信息，因此不需要显式的 Web 函数。

➤ 在 “端口映射”录制选项中，删除端口 443 的所有现有项，并为 Oracle 服务器名创建一个新项：

**服务 ID**：HTTP
**目标服务器**：Oracle Forms Server 的 IP 地址或长主机名
**目标端口**：443
**连接类型**：SSL
**SSL 版本**：活动的 SSL 版本。如果不能肯定，请选择 SSL 2/3

详细信息，请参阅第 5 章 “配置端口映射”。

➤ 在回放 NCA HTTPS 脚本的过程中，如果执行 **nca_connect_server** 命令时遇到问题，请在脚本的开头插入以下函数。

```
web_set_sockets_option("SSL_VERSION","3");
```

### 测试 Servlet 和其他 Oracle NCA 应用程序

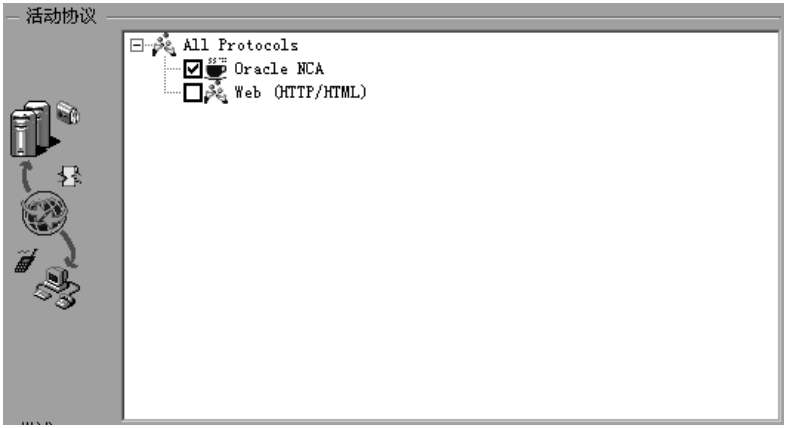
某些 NCA 会话使用 Servlet。

➤ Forms 侦听器 Servlet

➤ 同时使用 NCA 和 HTTP 通信的应用程序或模块，如 Oracle Configurator

➤ NCA 应用程序的初始化 （下载小程序、 jar 和 gif 文件）

在录制 Servlet 时，必须同时录制 Oracle NCA 和 Web 函数。通过在最初创建一个多协议脚本即可做到这一点。如果为 Oracle NCA 创建了单协议脚本，还可以在 “录制选项”中打开 “常规 : 协议”节点，并启用 Web 协议。然后可以开始录制。




如果不能肯定应用程序是否使用 Servlet，请检查脚本目录中的 *default.cfg* 文件。查找相关项

**UseServletMode=**

如果值是 1 或 2，则表明正在使用 Servlet，并且除了启用 Oracle NCA 以外，还必须启用 HTTP 录制。

如果已经录制了脚本，则可以自动重新生成代码以包含 Web 函数，而不必重新录制。选择 “工具” > “重新生成 Vuser”，然后在 “协议”部分中选择 Web 协议。

### 确定录制模式

在录制 Oracle NCA 脚本时，VuGen 将自动确定正确的连接模式：HTTP 或套接字模式。一般情况下，无需修改任何录制设置，因为 VuGen 将自动检测系统配置。在由其他应用程序保留标准端口映射的系统中，可能需要根据录制模式修改“端口映射”设置。

可以采用下列方式之一确定录制模式：

➤ 使用 NCA 应用程序时，打开 Java 控制台。

```
proxyHost=null
proxyPort=0
connectMode=HTTP
Forms Applet 的版本为：60812
```

**connectMode** 项表明连接模式是 HTTP、 HTTPS 还是套接字。

➤ 录制 NCA 会话之后，打开 Vuser 目录中的 *default.cfg* 文件，然后检查 **UseHttpConnectMode** 项的值。

```
[HttpConnectMode]
UseHttpConnectMode= 2
// 0 = socket 1 = http 2 = https
```

在 “服务器项”对话框中定义新的端口映射时，对于 HTTP 或 HTTPS 模式使用 *HTTP* 的服务 ID。对于套接字模式，使用 *NCA* 的服务 ID。

有关 “端口映射”设置的详细信息，请参阅第 5 章 “配置端口映射”。

## 关联 Oracle NCA 语句以实现负载平衡

LoadRunner 支持多个应用程序服务器的负载平衡。可以将 HTTP 返回值与 **nca_connect_server** 参数相关联。然后， LoadRunner 将在执行测试的过程中连接到相关服务器，并应用负载平衡。

**要关联语句以实现负载平衡，请执行下列操作：**

**1 录制一个多协议脚本。**

为 Oracle NCA 和 Web 协议录制一个多协议脚本。执行所需的操作并保存该脚本。

**2 定义主机参数。**

定义两个变量 serverHost 和 serverArgs 以进行参数化：

```
web_set_max_html_param_len("512");
web_reg_save_param("serverHost", "NOTFOUND=ERROR",
        "LB=<PARAM name=\"serverHost\" value=\"","RB=\">" , LAST );
web_reg_save_param("serverArgs", "NOTFOUND=ERROR",
        "LB=<PARAM name=\"serverArgs\" value=\"","RB=\">" , LAST );
```

**3 调用 web_url 函数，以向 serverHost 和 serverArgs 赋值：**

```
web_url("step_name", "URL=http://server1.merc-int.com/test.htm", LAST);
```

**4 将以下形式的 nca_connect_server 语句：**

```
nca_connect_server("199.203.78.170",
        9000"/*version=107*/, "module=e:\\appsnca...ndnam=apps ");
```

修改为：

```
nca_connect_server("< serverHost >", "9000"/*version=107*/, "<
serverArgs >");
```

现在，脚本应如下所示：

```
web_set_max_html_param_len("512");
web_reg_save_param("serverHost", "NOTFOUND=ERROR",
        "LB=<PARAM name=\"serverHost\" value=\"","RB=\">" , LAST );
web_reg_save_param("serverArgs", "NOTFOUND=ERROR",
        "LB=<PARAM name=\"serverArgs\" value=\"","RB=\">" , LAST );
web_url("step_name", "URL=http://server1.merc-int.com/test.htm", LAST);

nca_connect_server("<serverHost>","9000"/*version=107*/,"<serverArgs>");
```

# 其他建议的关联

在录制 Oracle NCA 会话时， VuGen 将录制动态值，即对于每个录制和回放会话都更改的值。两个常见的动态参数是 icx_ticket 和 JServSessionIdroot。

## icx_ticket

*icx_ticket* 变量是在 **web_url** 和 **nca_connect_server** 函数中发送的信息的一部分：

```
web_url("fnd_icx_launch.runforms",
"URL=http://ABC-
123:8002/pls/VIS/fnd_icx_launch.runforms\?ICX_TICKET=5843A55058947ED3
&RESP_APP=AR&RESP_KEY=RECEIVABLES_MANAGER&SECGRP_KEY=STANDAR
D", LAST);
```

对于每次录制， *icx_ticket* 值都不同。它包含客户端发送的 cookie 信息。要关联录制，请在所录制的 icx_ticket 值第一次出现之前添加 **web_reg_save_param**：

```
web_reg_save_param("icx_ticket", "LB=TICKET=", "RB=&RES", LAST);

...

web_url("fnd_icx_launch.runforms",
"URL=http://ABC-
123:8002/pls/VIS/fnd_icx_launch.runforms\?ICX_TICKET=<icx_ticket>&RESP_A
PP=AR&RESP_KEY=RECEIVABLES_MANAGER&SECGRP_KEY=STANDARD", LAST);
```

---

**注意**：根据您的应用程序设置， **web_reg_save_param** 的左边界和右边界可能有所不同。

---

### JServSessionIdroot

*JServSessionIdroot* 值是应用程序设置以存储会话 ID 的 cookie。大多数情况下，VuGen 会自动关联此值并插入 **web_reg_save_param** 函数。如果 VuGen 未自动添加该函数，请手动进行添加，并用参数名替换其所有实例。

要标识需要关联的值，请打开 “执行日志”（“视图” > “输出窗口”），并查找响应正文。

```
vuser_init.c(8):Set-Cookie:JServSessionIdroot=my1sanw2n1.JS4; path=/\r\n
vuser_init.c(8):Content-Length:79\r\n
vuser_init.c(8):Content-Type:text/plain\r\n
vuser_init.c(8):\r\n
vuser_init.c(8):81-byte response body for "http://ABC-
123/servlet/oracle.forms.servlet.ListenerServlet?ifcmd=getinfo&ifhost=mercury
&ifip=123.45.789.12" (RelFrameId=1)
vuser_init.c(8):/servlet/oracle.forms.servlet.ListenerServlet?JServSessionIdr
oot=my1sanw2n1.JS4\r\n
```

要关联该动态值，请在该值第一次出现之前插入 **web_reg_save_param** 函数，然后在整个脚本中用参数名替换变量值。在该示例中，右边界和左边界分别是 \r 和 \n，但是您应该检查具体环境，以确定环境中的实际边界。

```
web_reg_save_param("NCAJServSessionId","LB=\r\n\r\n","RB=\r","ORD=1"
,LAST);

web_url("f60servlet",
    "URL= http://ABC-
"123/servlet/oracle.forms.servlet.ListenerServlet?ifcmd=getinfo&"
"ifhost=mercury&ifip=123.45.789.12", LAST);

web_url("oracle.forms.servlet.ListenerSer",
    "URL=http://ABC-123<NCAJServSessionId>?ifcmd=getinfo&"
"ifhost=mercury&ifip=123.45.789.12", LAST);
```

# 以 Pragma 模式录制

可以将 Oracle NCA Vuser 的客户端配置为向名为 *Pragma* 的服务器发送额外的标头。标头是一个计数器，其行为方式如下：NCA 握手的初始消息的值为 1。对该握手后面的消息进行计数，从 3 开始。客户端每发送一条消息，计数器将增加 1。

如果从服务器收到的消息的类型是 *plain\text*，并且消息的正文以 ifError:#/#00 打头，那么客户端将向服务器发送一条 0 字节的消息，并且 Pragma 值的符号将更改为负号。客户端成功地从服务器接收信息之后，该符号将恢复原样。

只有在多协议模式 （Oracle NCA 和 Web）中才支持 Pragma 标头的录制。可以在脚本的 *default.cfg* 文件中标识 Pragma 模式。以 Pragma 模式进行操作时，UseServletMode 被设置为 2。

```
[HttpConnectMode]
UseHttpConnectMode=1
RelativeURL=<NCAJServSessionId>
UseServletMode=2
```

有关与 Pragma 相关的运行时设置的信息，请参阅第 570 页的 “客户端仿真运行时设置”。

要标识 Pragma 模式，可以执行 WinSocket 级别录制，并检查缓冲区的内容。在第一个示例中，缓冲区包含 Pragma 值以作为计数器：

```
send buf108
     "POST
/ss2servlet/oracle.forms.servlet.ListenerServlet?JServSessionIdss2ser"
     "vlet=gk5q79uqy1 HTTP/1.1\r\n"
     "Pragma:1\r\n"
     ...
send buf110
     "POST
/ss2servlet/oracle.forms.servlet.ListenerServlet?JServSessionIdss2ser"
     "vlet=gk5q79uqy1 HTTP/1.1\r\n"
     "Pragma:3\r\n"
     ...
```

在下例中，缓冲区包含 Pragma 值作为错误指示器：

```
recv buf129 281
     "HTTP/1.1 200 OK\r\n"
     "Date:Tue, 21 May 2002 00:03:48 GMT\r\n"
     "Server:Oracle HTTP Server Powered by Apache/1.3.19 (Unix)
mod_fastcgi/2.2"
     ".10 mod_perl/1.25 mod_oprocmgr/1.0\r\n"
     "Content-Length:13\r\n"
     "Content-Type:text/plain\r\n"
     "\r\n"
     "ifError:8/100"

send  buf130
     "POST
/ss2servlet/oracle.forms.servlet.ListenerServlet?JServSessionIdss2ser"
     "vlet=gk5q79uqy1 HTTP/1.1\r\n"
     "Pragma:-12\r\n"
     ...
```

# 48

# 开发 SAPGUI Vuser 脚本

在不断成长的 ERP（企业资源规划）领域中， SAP 提供了一些软件解决方案，通过这些方案，可以使各公司管理其所有业务流程。 Mercury 在功能及负载测试两个级别上提供了测试 SAP 解决方案模块的工具。

本章讨论用于测试 Windows 客户端的 SAPGUI 的 LoadRunner 解决方案。有关 mySAP Workplace 和 Portal 客户端的测试解决方案的信息，请参阅第 48 章 “开发 SAPGUI Vuser 脚本”。

本章描述：

- 检查 SAPGUI Vuser 的环境
- 创建 SAPGUI Vuser 脚本
- 设置 SAPGUI 录制选项
- 了解 SAPGUI Vuser 脚本
- 增强 SAPGUI Vuser 脚本
- 回放 SAPGUI Vuser 脚本
- 设置 SAPGUI 运行时设置
- SAPGUI 函数
- SAPGUI Vuser 脚本提示
- SAPGUI Vuser 脚本疑难解答
- 其他资源

***下列信息仅适用于 SAPGUI 协议。***

# 关于开发 SAPGUI Vuser 脚本

录制会话之前，请验证 VuGen 是否支持您的模块和客户端接口。下列部分将描述 SAP 业务应用程序的 SAP 客户端模块。

- SAP Web 客户端或 mySAP.com：使用 SAP-Web Vuser 类型。
- Windows 的 SAPGUI – 基于 Windows 的客户端：
- Java 的 SAPGUI：不支持该客户端

**6.10 版及更低版本：**使用 QuickTest Professional for R/3。要运行负载测试，请在 LoadRunner Controller 中将该脚本作为 SAP Vuser 运行。

**6.20 版及更高版本：**

**对于功能测试：**使用 mySAP.com 客户端的 QuickTest Professional 加载项。

**对于负载测试：**使用 LoadRunner SAPGUI 协议在 VuGen 中创建脚本，并在 Controller 中运行场景。

可以使用 VuGen 录制器录制典型的业务流程。 VuGen 将录制 SAP 业务流程中 Windows 客户端活动的 SAPGUI，并生成 Vuser 脚本。在 Windows 客户端的 SAPGUI 中执行操作时， VuGen 会生成描述该活动的函数。每个函数都以 **sapgui** 前缀开头。回放时，这些函数将模拟 SAPGUI 对象上的用户活动。例如，**sapgui_select_radio_button** 会选择 “蓝色”单选按钮。

```
sapgui_select_radio_button("Blue",
        "usr/radRB7",
        BEGIN_OPTIONAL,
            "AdditionalInfo=sapgui1027",
        END_OPTIONAL);
```

# 检查 SAPGUI Vuser 的环境

在录制 SAPGUI Vuser 之前检查和设置系统的基本步骤为检查修补级别和启用脚本。正确配置环境之后，便可在 VuGen 中录制和回放典型的 SAP 会话。

## 检查修补级别

可以从“关于”框中检查 Windows 客户端 SAPGUI 的修补级别。支持的最低修补级别为 32。

**要检查修补级别，请执行下列操作：**

**1** 调用 SAPGUI 登录窗口。单击“SAP 登录”对话框的左上角并从菜单中选择“关于 SAP 登录”。

**2** 将打开 “SAP 版本信息”对话框。验证修补级别项是否为 32 或更高。

## 启用脚本

Mercury Interactive 可支持 Windows 客户端的 SAPGUI，该支持功能基于 SAP 的脚本 API。通过该 API， Vuser 便可与 SAPGUI 客户端交互操作，并能够接收通知和执行操作。

只有 SAP 内核的最新版本才有脚本 API。在支持脚本的内核版本中，该选项在默认情况下是禁用的。要使用 Mercury Interactive 工具，首先请确保 SAP 服务器支持脚本 API，并在服务器和客户端上启用该 API。要查看详细信息并下载修补程序，请参阅 SAP OSS 注释 #480149。

LoadRunner 提供了一个用于检查系统是否支持脚本的实用程序。该实用程序 *VerifyScript.exe* 位于 CD 上的 Patches and Tools 目录中。详细信息，请参阅随该实用程序提供的自述文件。

下列部分将显示启用脚本所需的步骤。

- 检查配置
- 在 SAP 应用程序服务器上启用脚本
- 在 SAPGUI 6.20 客户端上启用脚本

### 检查配置

启用脚本的第一步是确保安装了正确的内核版本，如有需要，还要对该内核版本进行更新。

检查下表，以了解您的 SAP 应用程序服务器版本所需的最小内核修补级别。如果需要，请下载并安装最新的修补程序。

| 软件组件 | 版本 | 程序包名 | 内核修补级别 |
|---|---|---|---|
| SAP_APPL | 31I | SAPKH31I96 | 内核 3.1I 级别 650 |
| SAP_APPL | 40B | SAPKH40B71 | 内核 4.0B 级别 903 |
| SAP_APPL | 45B | SAPKH45B49 | 内核 4.5B 级别 753 |
| SAP_BASIS | 46B | SAPKB46B37 | 内核 4.6D 级别 948 |
| SAP_BASIS | 46C | SAPKB46C29 | 内核 4.6D 级别 948 |
| SAP_BASIS | 46D | SAPKB46D17 | 内核 4.6D 级别 948 |
| SAP_BASIS | 610 | SAPKB61012 | 内核 6.10 级别 360 |

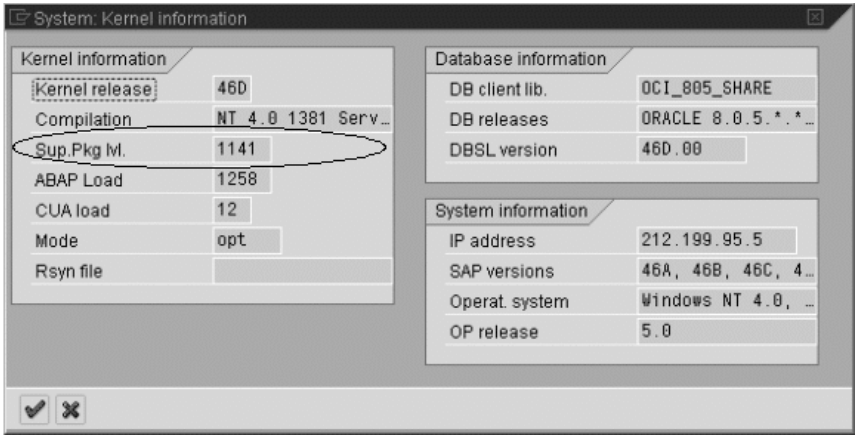
要检查内核修补级别，请执行下列操作：

1. 登录到 SAP 系统。
2. 选择 “系统” > “状态”。

**3** 单击 “其他内核信息” 按钮 （带有黄色箭头）。

**4** 在 “内核信息”部分中，检查 **Sup. Pkg. lvl.** 的值。

如果级别低于 948，则必须下载最新的内核版本并升级现有版本。有关如何升级的详细说明，请参阅 SAP OSS 注释 #480149。

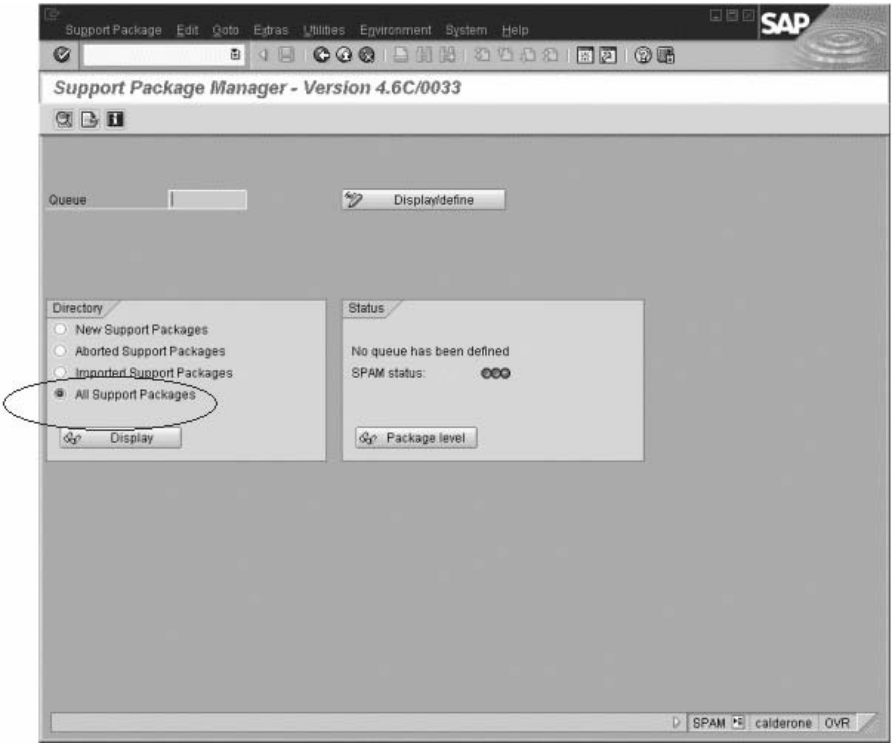


要检查 R/3 支持包，请执行下列操作：

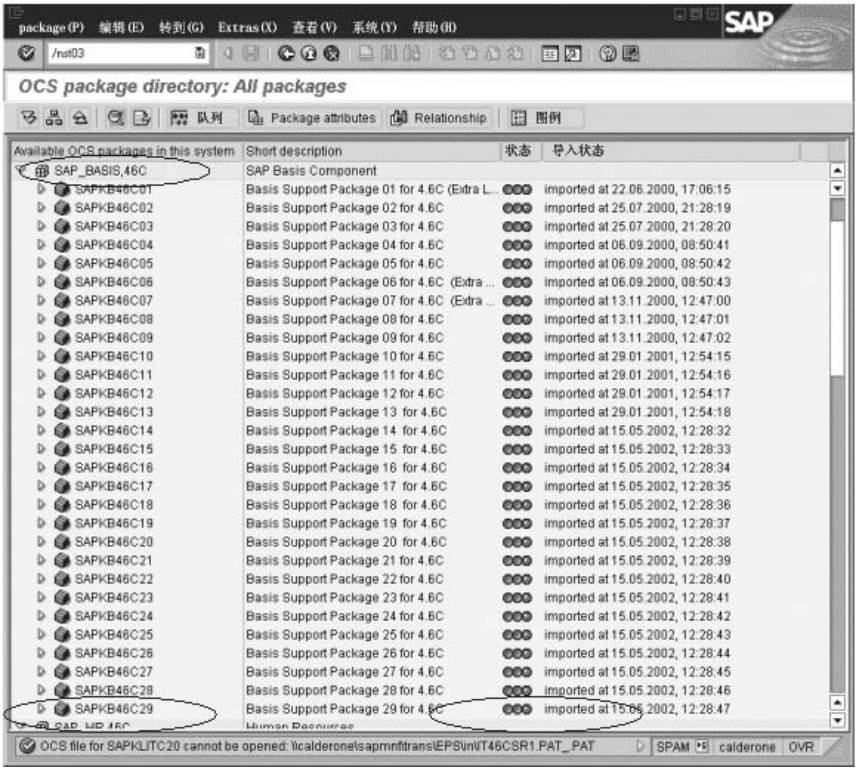
1 登录到 SAP 系统。

2 运行 SPAM 事务。

3 在 “目录”部分中，选择 “所有支持包”，并单击 “显示”按钮。



**4** 验证是否为 SAP_BASIS, 4.6C 安装了 SAPKB46C29。如果已安装，将在 “状态”列中显示绿色圆圈。



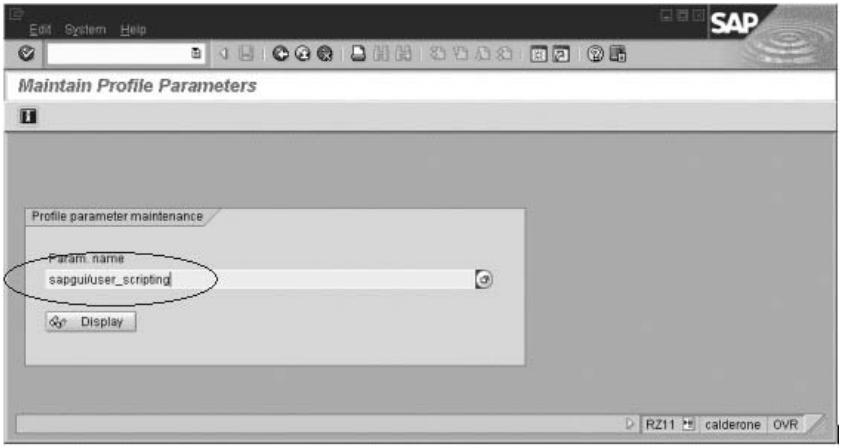

如果未安装 OCS 包，请从 www.sap.com 网站下载并安装它。详细信息，请参阅 SAP OSS 注释 #480149。

### 在 SAP 应用程序服务器上启用脚本

具有管理权限的用户可以通过在应用程序服务器上将 *sapgui/user_scripting* 配置文件参数设置为 **TRUE** 来启用脚本。要为所有用户启用脚本，请在所有应用程序服务器上设置该参数。要为特定的用户组启用脚本，请只在带有所需访问限制的应用程序服务器上设置该参数。

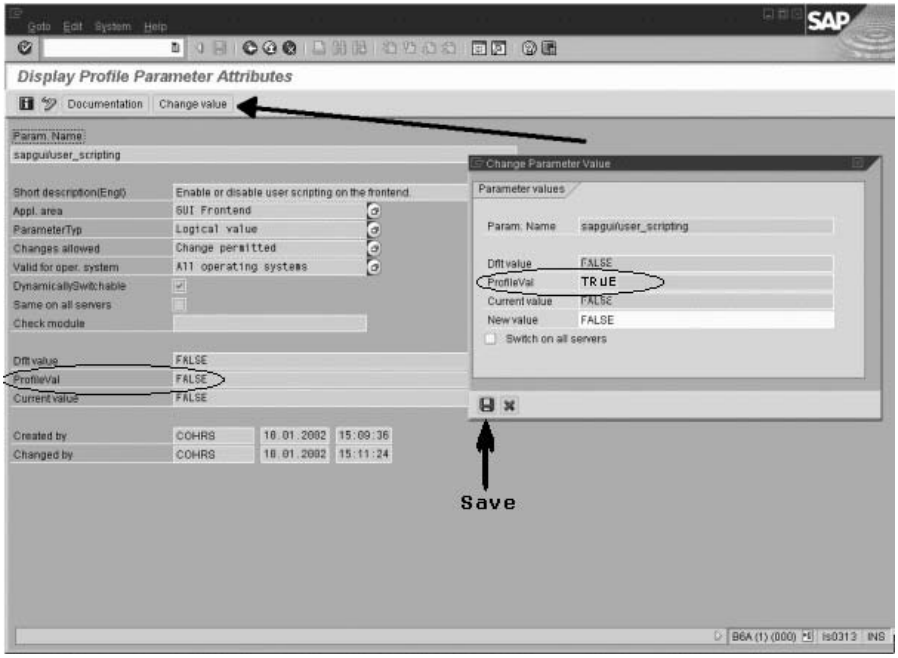
**要更改配置文件参数，请执行下列操作：**

**1** 打开事务 rz11。将该参数名指定为 *sapgui/user_scripting* 并单击 “显示”。将打开 “显示配置文件参数属性”窗口。




如果状态栏显示 “参数名未知”，则表明缺少当前支持包。如第 583 页的 “检查配置”中所述，导入与 SAP BASIS 及应用程序服务器内核版本相对应的支持包。

**2** 如果 “配置文件 Val” 为 FALSE，则需要修改其值。单击工具栏中的 “更改值” 按钮。将打开 “更改参数值” 窗口。在 “ProfileVal” 框中输入 *TRUE*，并单击 “保存” 按钮。




保存更改后，该窗口将关闭， **ProfileVal** 即被设置为 TRUE。

**3** 重新启动应用程序服务器，因为只有在登录到该系统时所做的更改才会生效。

如果即使重新启动该服务器后更新的 **ProfileVal** 也未发生变化，则该应用程序服务器的内核已过期。按照第 583 页的 “检查配置” 部分中的说明，导入所需的内核修补程序。

注意，在下列内核版本中，配置文件值可通过事务 rz11 动态激活，而无需重新启动应用程序服务器。

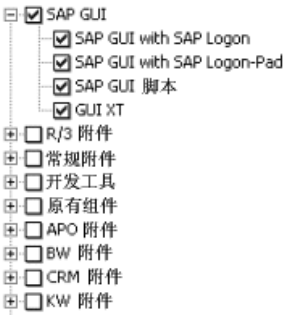
| 版本 | 内核版本 | 修补级别 |
|---|---|---|
| 4.6B、4.6C、4.6D | 4.6D | 972 |
| 6.10 | 6.10 | 391 |
| 6.20 | 所有版本 | 所有级别 |

### 在 SAPGUI 6.20 客户端上启用脚本

要使 VuGen 能够运行脚本，还必须在 SAPGUI 客户端上启用脚本。同时，还应对该客户端进行配置，使其不显示某些消息，例如，建立连接时或将脚本附加到 GUI 进程时显示的消息。

**要配置 SAPGUI 客户端以使其与 VuGen 配合使用，请执行下列操作：**

➤ **在安装过程中：**安装 SAPGUI 客户端时，请启用 “SAP GUI 脚本”选项。




➤ **安装完成后：**取消警告消息。打开 SAPGUI 客户端的 “选项”对话框。选择 “脚本”选项卡并清除下列选项：

1. 在脚本附加到正在运行的 GUI 时通知。
2. 在脚本打开连接时通知。

也可以通过将以下注册表项中 *WarnOnAttach* 和 *WarnOnConnection* 值设置为 0 来防止弹出这些消息：

HKCU\SOFTWARE\SAP\SAPGUI Front\SAP Frontend Server\Security.

## 创建 SAPGUI Vuser 脚本

创建 SAPGUI Vuser 脚本的第一步是选择 Vuser 和脚本类型。 SAP Vuser 类型 **SAPGUI** 属于 **ERP/CRM** 类别。可以创建单协议或多协议的 Vuser 脚本。

**要创建 SAPGUI Vuser 脚本，请执行下列操作：**

1 调用 VuGen，并选择 “文件” > “新建”。

2 要录制简单的 SAPGUI 客户端会话 （不使用浏览器控件），请使用 **SAPGUI** 类型的 Vuser 创建单协议的 Vuser 脚本。

3 要录制使用浏览器控件的 SAPGUI 会话，请创建多协议的 Vuser 脚本。指定 **SAPGUI** 和 **SAP-Web** Vuser 类型。这样，当遇到浏览器控件时， VuGen 将录制特定于 Web 的函数。

4 单击 “确定”打开该 Vuser 脚本。

# 设置 SAPGUI 录制选项

可以使用录制选项为该录制会话设置与 SAP 相关的首选项。要打开 “录制选项” 对话框，请选择 “工具” > “录制选项”，或者在 “开始录制” 对话框中单击 “选项”。键盘快捷键为 CTRL+F7。

可以在下列区域中设置录制选项：

- SAPGUI：常规录制选项
- SAPGUI：代码生成录制选项

如果正在使用 SAP-Web Vuser 类型录制多协议 Vuser 脚本，则可参阅第 35 章 “为 Internet 协议设置录制选项” 来设置更多录制选项。

## SAPGUI：常规录制选项

可以使用这些录制选项设置录制会话期间的常规首选项。

**要设置常规录制选项，请执行下列操作：**

1. 打开 “录制选项” 对话框并选择 “SAPGUI：常规” 节点。
2. 选择 “自动转至主窗口” 指示 VuGen 在登录后导航到主屏幕。录制期间在登录屏幕中输入信息后，VuGen 将自动确认后面显示的所有屏幕，直到主窗口打开为止。
3. 选择 “在打开的会话上启用持续录制” 可使 VuGen 从以前打开的 SAPGUI 窗口中录制事件。
4. 要在录制期间拍摄 SAPGUI 屏幕快照，请选择 “捕捉屏幕快照”。
5. 单击 “确定” 接受设置并关闭对话框。

## SAPGUI：代码生成录制选项

可以使用这些录制选项设置代码生成首选项。

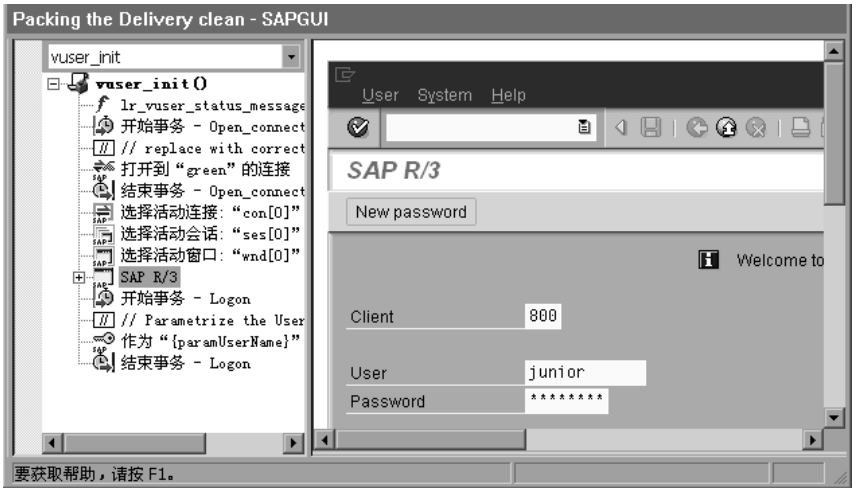
要设置代码生成录制选项，请执行下列操作：

1. 打开 “录制选项”对话框并选择 “SAPGUI：代码生成”节点。
2. 选择“生成单步登录操作”指示 VuGen 为所有登录操作生成一个 **sapgui_logon** 方法。这有助于简化代码。如果遇到登录问题，请禁用该选项。
3. 为了提高 SAPGUI 脚本的可读性，默认情况下，VuGen 将针对特定对象生成相应的函数。例如，在 SAPGUI 网格中录制的所有函数都以 **sapgui_grid** 前缀开头。要指示 VuGen 使用低级函数 （如 **sapgui_set_property** 和 **sapgui_call_method**）生成脚本，请选择 “生成低级脚本”。注意，如果禁用该选项以便生成高级函数，则仅会增强可读性，这两个录制级别的开销并没有差别。
4. 单击 “确定”接受设置并关闭对话框。

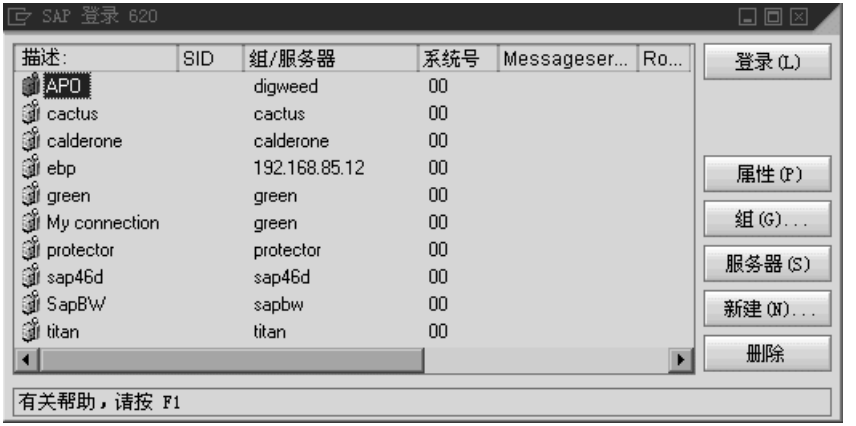
# 了解 SAPGUI Vuser 脚本

SAPGUI Vuser 脚本通常包含若干个 SAP 事务，这些事务构成一个业务流程。业务流程由模拟用户操作的函数组成。打开树视图可看到每个用户操作即是一个 Vuser 脚本步骤。

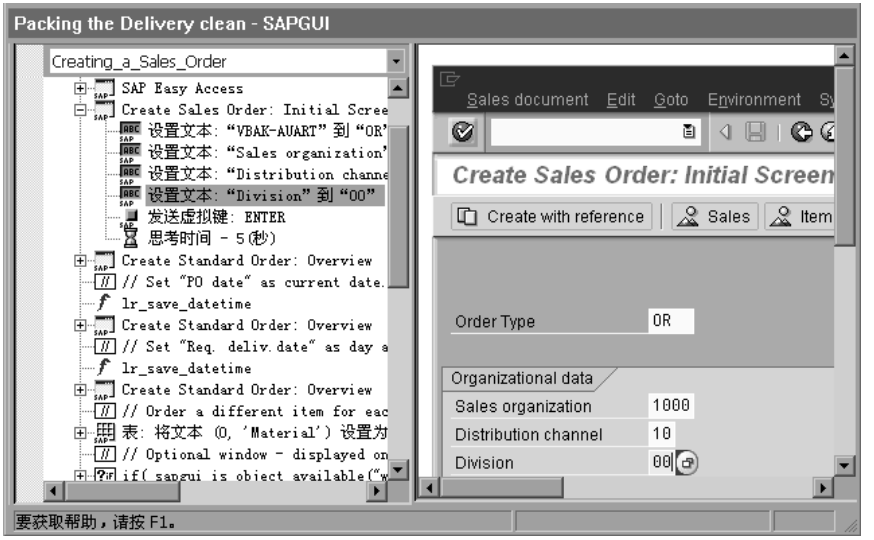
下例显示了 SAPGUI 客户端的典型录制。第一部分 *vuser_init* 包含打开连接和登录。



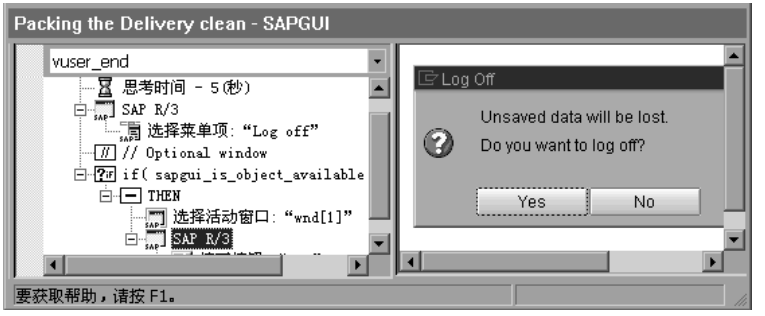

注意，“打开连接”步骤将使用 “SAP 登录”中的 “描述”列表下的一个连接名。如果指定的名称不在该列表中， Vuser 将使用该名称查找服务器。



在以下部分中，函数将模拟典型的用户操作 （例如菜单选择和复选框设置）。




最后一部分 *vuser_end* 显示了注销过程。



在为 SAPGUI 和 Web 录制多协议脚本时，VuGen 将针对这两种协议分别生成相应的步骤。在脚本视图中，可以查看 **sapgui** 和 **web** 函数。下例说明了一个多协议录制，其中 SAPGUI 客户端打开一个 Web 控件。注意从 **sapgui** 到 **web** 函数的切换。

```
sapgui_tree_double_click_item("Use as general WWW browser, REPTITLE",
      "shellcont/shell",
      "000732",
      "REPTITLE",
      BEGIN_OPTIONAL,
          "AdditionalInfo=sapgui1020",
      END_OPTIONAL);

...
sapgui_set_text("",
      "http:\\\\yahoo.com",
      "usr/txtEDURL",
      BEGIN_OPTIONAL,
          "AdditionalInfo=sapgui1021",
      END_OPTIONAL);

...
web_add_cookie("B=7pt5cisv1p3m2&b=2; DOMAIN=www.yahoo.com");

web_url("yahoo.com",
          "URL=http://yahoo.com/",
          "Resource=0",
          "RecContentType=text/html",
          "Referer=",
          "Snapshot=t1.inf",
          "Mode=HTML",
          EXTRARES,
"URL=http://srd.yahoo.com/hpt1/ni=17/ct=lan/sss=1043752588/t1=1043
752575385/d1=1251/d2=1312/d3=1642/d4=4757/0.409700948728773
9/*1", "Referer=http://www.yahoo.com/", ENDITEM,
          LAST);
```

# 增强 SAPGUI Vuser 脚本

审查了录制的 Vuser 脚本后，可以通过下列方式增强该脚本：

- **事务：**将事务、集合点和控制流结构插入到该脚本中。详细信息，请参阅第 6 章“增强 Vuser 脚本”
- **验证：**插入 SAPGUI 验证函数验证 SAPGUI 对象的当前状态。详细信息，请参阅添加验证函数。
- **检索信息：**插入 SAPGUI 函数验证 SAPGUI 对象的当前值。可以使用 **sapgui_get_xxx** 函数检索信息。详细信息，请参阅第 598 页的 “检索信息”。
- **定义参数 （可选）**。为录制到 Vuser 脚本中的固定值定义参数。通过将固定值替换为参数，可以使用不同的值多次重复同一业务流程。详细信息，请参阅第 7 章“定义参数”

### 添加验证函数

使用可选或动态窗口或帧时，可以使用验证函数确定该窗口或对象是否已打开。这样，即使打开了可选窗口或异常，该 Vuser 脚本仍可继续运行。

第一个示例将检查窗口是否已打开。如果该窗口已打开， Vuser 会在继续运行前将其关闭。

```
if (!sapgui_is_object_available("wnd[1]"))
        sapgui_call_method("{ButtonID}",
                "press",
                LAST,
                AdditionalInfo=info1011");
sapgui_press_button(.....)
```

下一个示例显示了 ME51N 事务中的动态对象。文档概览帧是可选的，可以通过“文档概览开 / 关”按钮将其打开或关闭。

以下代码将检查 “文档概览”按钮上的文本。如果该按钮上的文本显示 “文档概览开”，则可以单击该按钮关闭文档概览帧。

```
if(sapgui_is_object_available("tbar[1]/btn[9]"))
{
    sapgui_get_text("Document overview on/off button",
            "tbar[1]/btn[9]",
            "paramButtonText",
            LAST);

    if(0 == strcmp("Document overview off",
lr_eval_string("{paramButtonText}")))
            sapgui_press_button("Document overview off",
                "tbar[1]/btn[9]",
                BEGIN_OPTIONAL,
                    "AdditionalInfo=sapgui1013",
                END_OPTIONAL);
}
```

## 检索信息

使用 SAGUI Vuser 时，可以使用 **sapgui_get_<xxx>** 函数检索 SAPGUI 对象的当前值。可以使用该值作为另一个业务流程的输入，或者将其显示在输出日志中。

### 检索状态栏信息

下例说明了如何保存状态栏消息中的部分内容以检索顺序号。

**要从状态栏中检索顺序号，请执行下列操作：**

1. 导航到要检查状态栏文本的位置，并选择 “插入” > “新建步骤”。选择 **sapgui_status_bar_get_type** 函数。这将验证 Vuser 是否能成功地从状态栏中检索文本。
2. 插入 *if* 语句，检查前面的语句是否执行成功。如果成功，请使用 **sapgui_status_bar_get_param** 保存该参数的值。

该 **sapgui_status_bar_get_param** 函数会将该顺序号保存在用户定义的参数中。此时，该顺序号是状态栏字符串的第二个索引。

```
sapgui_press_button("Save (Ctrl+S)",
      "tbar[0]/btn[11]",
      BEGIN_OPTIONAL,
           "AdditionalInfo=sapgui1038",
      END_OPTIONAL);

sapgui_status_bar_get_type("Status");
if(0==strcmp(lr_eval_string("{Status}"),"Success"))
  sapgui_status_bar_get_param("2", " Order_Number ");
```

测试执行期间，执行日志会指明值和参数名：

```
Action.c(240)：按下按钮 “保存 (Ctrl+S)”
Action.c(248)：状态栏类型为 “成功”
Action.c(251)：状态栏中参数 2 的值为 “33232”
```

### 保存日期信息

在创建使用日期的脚本时，该脚本可能会运行不正常。例如，如果在 6 月 2 日录制脚本，而在 6 月 3 日回放，则日期字段将会不正确。因此，需要在测试执行期间将该日期保存在参数中，并将存储的值作为其他日期字段的输入。要在脚本执行期间保存当前日期或时间，请使用 **lr_save_datetime** 函数。将该函数插入到需要日期信息的函数之前。注意，该日期的格式是针对本地区域设置的。请在 **lr_save_datetime** 函数中使用相关的格式。例如，将 “月 . 日 . 年” 格式指定为 “%m.%d.%Y”。

在下例中，**lr_save_datetime** 将保存当前日期。**sapgui_set_text** 函数使用该值将传送日期设置为两天后。

```
lr_save_datetime("%d.%m.%Y", DATE_NOW + (2 * ONE_DAY),
    "paramDateTodayPlus2");

sapgui_set_text("Req. deliv.date",
        "{paramDateTodayPlus2}",
        "usr/ctxtRV45A-KETDAT",
        BEGIN_OPTIONAL,
                "AdditionalInfo=sapgui1025",
        END_OPTIONAL);
```

## 回放 SAPGUI Vuser 脚本

创建并增强 SAPGUI Vuser 脚本后，可以为其配置运行时设置并从 VuGen 运行它来检查其功能。

通过运行时设置，可以控制 Vuser 在回放期间的行为。可以在运行该 Vuser 脚本之前配置这些设置。可以设置常规运行时配置，还可以设置特定于 SAPGUI 的运行时设置。

常规设置包括运行逻辑、步、日志记录、思考时间和性能首选项。有关常规运行时设置的详细信息，请参阅第 9 章 “配置运行时设置”。有关特定于 SAPGUI 的设置，请参阅下列部分。

配置好运行时设置后，可以保存该 Vuser 脚本并在 VuGen 中运行它来验证其是否运行正常。有关作为独立测试运行 Vuser 脚本的详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。确认 Vuser 脚本可以运行之后，可以将其集成到场景中。详细信息，请参阅 《LoadRunner Controller 用户指南》。

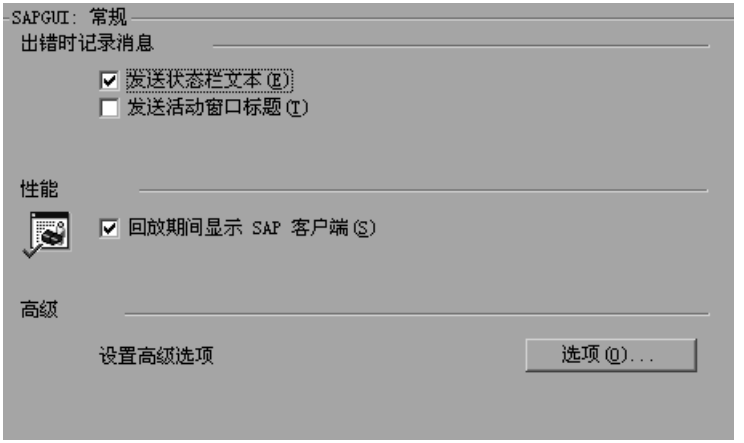
# 设置 SAPGUI 运行时设置

可以在下列区域中配置特定于 SAPGUI 的运行时设置：

- SAPGUI：常规运行时设置
- SAPGUI：高级运行时设置

## SAPGUI：常规运行时设置

通过常规运行时设置，可以为 SAPGUI Vuser 脚本设置常规设置。 VuGen 将在运行该脚本时使用这些设置。




“日志”运行时设置用于指定在发生错误时 Vuser 发送到执行日志中的信息。

**发送状态栏文本**：将状态栏中的文本发送到日志文件中。

**发送活动窗口标题**：将活动窗口标题文本发送到日志文件中。

通过 “性能”运行时设置，可以指明回放期间是否显示 SAP 客户端。

**回放期间显示 SAP 客户端**：回放期间在 SAP 客户端中显示操作动画。显示用户界面 (UI) 的优点在于，可以查看填写表单的过程并密切跟踪 Vuser 的操作。但是，该选项需要其他资源并可能影响负载测试的性能。

通过高级选项，可以为 *SAPfewgsvr.exe* 进程设置超时。详细信息，请参阅第 602 页的 “SAPGUI：高级运行时设置”。

**要设置 SAPGUI 运行时设置，请执行下列操作：**

1 打开 “运行时设置”对话框。单击 VuGen 工具栏上的 “运行时设置”按钮，或者选择 “Vuser” > “运行时设置”。

2 选择 “SAPGUI：常规”节点。

3 在 “出错时记录消息”部分中，选择一个或多个消息源：“发送状态栏文本”或“发送活动窗口标题”。

4 在 “性能”部分中，选中 “回放期间显示 SAP 客户端”复选框以便在回放期间显示 SAPGUI 用户界面。

5 单击 “选项”为 *SAPfewgsvr.exe* 进程设置超时。

### SAPGUI：高级运行时设置

每个 Vuser 都在测试执行期间调用一个单独的 *SAPfewgsvr.exe* 进程。在某些实例中，即使回放会话已结束，该进程仍处于打开状态。可以检查 Windows 任务管理器查看该进程是否仍处于活动状态。

通过高级 SAPGUI 设置，可以为该应用程序设置超时。到达超时时间后，VuGen 将关闭先前未终止的所有 *SAPfewgsvr* 进程。

通过高级运行时设置，可以为 *SAPfewgsvr.exe* 进程设置值。

**设置 SAPfewgsvr 应用程序超时**：通过该设置，可以修改 *SAPfewgsvr.exe* 进程的超时。

**SAPfewgsvr 超时**：*SAPfewgsvr.exe* 进程的超时以秒为单位。默认值是 300 秒。

本章中的提示适用于场景中的录制、回放和运行。此外，还可以直接从 SAP 支持站点获取信息。

# SAPGUI 函数

SAPGUI 录制会话期间， VuGen 将生成模拟用户与 SAPGUI 客户端交互操作的函数。录制 Windows 客户端的 SAPGUI 时，VuGen 将生成带有 **sapgui** 前缀的函数。本部分将列出所有 **sapgui** 函数。

当使用 Web 界面 （如 SAP Workplace 或 Portal）录制 SAP 会话时，或者在 SAPGUI 客户端打开 Web 控件， VuGen 都会生成带有 **web** 前缀的函数。

有关 **sapgui** 和 **web** 函数的详细信息，请使用 “编辑” 菜单中的 “显示函数语法” 功能，或者请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

虽然大多数函数都是录制的，但也可以手动在脚本中插入任何函数。未录制的函数包括以 **sapgui_get** 开头的数据检索函数以及以 **sapgui_is** 开头的验证函数。

**sapgui** 函数有多种类别：连接和会话函数、方法和属性函数、验证和数据检索函数和对象函数。对象函数是指在 SAPGUI 对象内执行操作的函数，例如，日历函数、网格函数、状态栏函数、表函数、树函数、窗口函数和常规对象函数。

### 连接和会话函数

| | |
|---|---|
| **sapgui_create_session** | 新建 SAPGUI 会话 |
| **sapgui_logon** | 登录到 SAP 服务器 |
| **sapgui_open_connection** | 打开与 SAP 服务器的连接 |
| **sapgui_open_connection_ex** | 打开与连接字符串指定的 SAP 服务器的连接 |
| **sapgui_select_active_connection** | 将指定的连接设置为活动连接 |
| **sapgui_select_active_session** | 设置活动 SAPGUI 会话 |

### 方法和属性函数

| | |
|---|---|
| **sapgui_get_property_of_active_object** | 检索活动对象的属性 |
| **sapgui_active_object_from_parent_method** | 通过调用父对象的方法在父对象中选择对象 |
| **sapgui_active_object_from_parent_property** | 选择用作父对象属性的对象 |

| | |
|---|---|
| **sapgui_call_method** | 调用 SAPGUI 对象的方法 |
| **sapgui_call_method_of_active_object** | 调用活动对象的方法 |
| **sapgui_get_property** | 获取 SAPGUI 对象的属性 |
| **sapgui_set_collection_property** | 设置类型为 *SAP GuiCollection* 的对象的属性 |
| **sapgui_set_property** | 设置 SAPGUI 对象的属性 |
| **sapgui_set_property_of_active_object** | 设置活动对象的属性 |

### 日历函数

| | |
|---|---|
| **sapgui_calendar_focus_date** | 将焦点设置在特定日期上 |
| **sapgui_calendar_scroll_to_date** | 滚动到日历中的特定日期 |
| **sapgui_calendar_select_interval** | 在日历中选择日期范围 |

### 网格函数

| | |
|---|---|
| **sapgui_grid_clear_selection** | 清除网格中的选定内容 |
| **sapgui_grid_click** | 在网格中单击 |
| **sapgui_grid_click_current_cell** | 在网格的活动单元格中单击 |
| **sapgui_grid_double_click** | 在网格中双击 |
| **sapgui_grid_double_click_current_cell** | 在网格的活动单元格中双击 |
| **sapgui_grid_get_cell_data** | 从网格单元格中检索文本 |
| **sapgui_grid_get_current_cell_column** | 在网格中检索当前单元格所在列的键（内部 ID） |
| **sapgui_grid_get_current_cell_row** | 检索网格的当前单元格的行号 |
| **sapgui_grid_is_checkbox_selected** | 检查网格中复选框的状态 |
| **sapgui_grid_open_context_menu** | 在网格中右键单击打开上下文菜单 |
| **sapgui_grid_press_button** | 在网格单元格中单击按钮 |
| **sapgui_grid_press_button_current_cell** | 在活动的网格单元格中单击按钮 |
| **sapgui_grid_press_column_header** | 按下网格的列标头 |

| | |
|---|---|
| **sapgui_grid_press_ENTER** | 在网格中按 ENTER 键 |
| **sapgui_grid_press_F1** | 在网格中按 F1 键 |
| **sapgui_grid_press_F4** | 在网格中按 F4 键 |
| **sapgui_grid_press_toolbar_button** | 在网格中单击工具栏按钮 |
| **sapgui_grid_press_toolbar_context_button** | 在网格中单击工具栏上下文按钮 |
| **sapgui_grid_press_total_row** | 在网格中单击汇总行区域 |
| **sapgui_grid_press_total_row_current_cell** | 在当前活动的网格单元格中单击汇总行按钮 |
| **sapgui_grid_scroll_to_row** | 滚动到网格中的一行 |
| **sapgui_grid_select_cell** | 选择网格中的一个单元格 |
| **sapgui_grid_select_cell_column** | 在当前行的指定列中选择单元格 |
| **sapgui_grid_select_cell_row** | 在当前列的指定行中选择单元格 |
| **sapgui_grid_select_cells** | 选择网格中的多个单元格 |
| **sapgui_grid_select_columns** | 选择网格中的列 |
| **sapgui_grid_select_context_menu** | 选择网格中的上下文菜单 |
| **sapgui_grid_select_rows** | 选择网格中的多行 |
| **sapgui_grid_select_toolbar_menu** | 选择网格中的工具栏菜单 |
| **sapgui_grid_set_cell_data** | 在网格单元格中插入文本 |
| **sapgui_grid_set_checkbox** | 选中或清除网格复选框 |
| **sapgui_grid_set_column_order** | 设置网格中的列顺序 |

### 状态栏函数

| | |
|---|---|
| **sapgui_status_bar_get_param** | 从状态栏中获取参数 |
| **sapgui_status_bar_get_text** | 从状态栏中获取文本 |
| **sapgui_status_bar_get_type** | 检索状态栏信息：**成功**、**警告**或**错误** |

### 表函数

| | |
|---|---|
| **sapgui_table_is_checkbox_selected** | 检查表中复选框的状态 |
| **sapgui_table_is_row_selected** | 检查是否选中表中的某行 |
| **sapgui_table_get_text** | 检索表单元格中的文本 |
| **sapgui_table_is_radio_button_selected** | 检查表中单选按钮的状态 |
| **sapgui_table_press_button** | 按下表中的按钮 |
| **sapgui_table_select_combobox_entry** | 在表中选择列表项 |
| **sapgui_table_select_radio_button** | 在表中选择单选按钮 |
| **sapgui_table_set_checkbox** | 在表中选中或清除复选框 |
| **sapgui_table_set_focus** | 将焦点设置在表上 |
| **sapgui_table_set_password** | 在表中设置密码 |
| **sapgui_table_set_row_selected** | 选择或取消选择表中的行 |
| **sapgui_table_set_text** | 将文本插入到表单元格中 |

### 树函数

| | |
|---|---|
| **sapgui_tree_click_link** | 单击树中的链接 |
| **sapgui_tree_collapse_node** | 折叠树中的节点 |
| **sapgui_tree_double_click_item** | 双击树中的项 |
| **sapgui_tree_double_click_node** | 双击树中的节点 |
| **sapgui_tree_expand_node** | 展开树中的节点 |
| **sapgui_tree_get_item_text** | 获取树中某项的文本 |
| **sapgui_tree_get_node_text** | 获取树中某节点的文本 |
| **sapgui_tree_is_checkbox_selected** | 检查是否选中树中的复选框 |
| **sapgui_tree_open_default_context_menu** | 打开树的默认上下文相关菜单 |
| **sapgui_tree_open_header_context_menu** | 打开树标头的上下文相关菜单 |
| **sapgui_tree_open_item_context_menu** | 打开树中某项的上下文相关菜单 |
| **sapgui_tree_open_node_context_menu** | 打开树中某节点的上下文相关菜单 |

| | |
|---|---|
| **sapgui_tree_press_button** | 单击树中的按钮 |
| **sapgui_tree_press_header** | 单击树中的列标头 |
| **sapgui_tree_press_key** | 从树中按某键 |
| **sapgui_tree_scroll_to_item** | 滚动到树中的某项 |
| **sapgui_tree_scroll_to_node** | 滚动到树中的某节点 |
| **sapgui_tree_select_column** | 选择树中的某列 |
| **sapgui_tree_select_item** | 选择树中的某项 |
| **sapgui_tree_select_node** | 选择树中的某节点 |
| **sapgui_tree_set_checkbox** | 选中或清除树中的复选框 |
| **sapgui_tree_set_column_width** | 设置树的列宽 |
| **sapgui_tree_set_hierarchy_header_width** | 设置树层次结构的宽度 |
| **sapgui_tree_set_selected_node** | 选择树中的节点 |
| **sapgui_tree_unselect_all** | 取消树中的所有选定内容 |
| **sapgui_tree_unselect_column** | 取消选择树中的列 |
| **sapgui_tree_unselect_node** | 取消选择树中的节点 |

### 窗口函数

| | |
|---|---|
| **sapgui_window_close** | 关闭 SAPGUI 客户端窗口 |
| **sapgui_window_maximize** | 将窗口设置为全屏大小 |
| **sapgui_window_resize** | 将窗口调整为指定大小 |
| **sapgui_window_restore** | 将窗口还原为非最大化状态 |
| **sapgui_window_scroll_to_row** | 滚动到窗口中的某行 |

### 验证和数据检索函数

| | |
|---|---|
| **sapgui_get_active_window_title** | 检索活动窗口的标题 |
| **sapgui_get_ok_code** | 检索命令字段文本 |
| **sapgui_get_text** | 获取对象中的文本 |

| | |
|---|---|
| **sapgui_is_checkbox_selected** | 检查是否选中某复选框 |
| **sapgui_is_object_available** | 检查某对象是否可用 |
| **sapgui_is_object_changeable** | 检查是否可以更改某对象 |
| **sapgui_is_radio_button_selected** | 检查是否选中某单选按钮 |
| **sapgui_is_tab_selected** | 检查是否选中某选项卡 |

### 常规对象函数

| | |
|---|---|
| **sapgui_htmlviewer_send_event** | 将事件发送到 HTML 查看器 |
| **sapgui_select_combobox_entry** | 选择某列表项 |
| **sapgui_press_button** | 按下某按钮 |
| **sapgui_select_active_window** | 将指定窗口设置为活动窗口 |
| **sapgui_select_radio_button** | 选择某单选按钮 |
| **sapgui_select_tab** | 选择某选项卡 |
| **sapgui_send_vkey** | 发送虚拟键 |
| **sapgui_set_checkbox** | 选中或清除某复选框 |
| **sapgui_set_focus** | 将焦点设置在指定对象上 |
| **sapgui_select_menu** | 选择指定菜单 |
| **sapgui_set_password** | 设置密码字段文本 |
| **sapgui_set_text** | 将文本插入到文本框中 |
| **sapgui_set_ok_code** | 设置命令字段文本 |

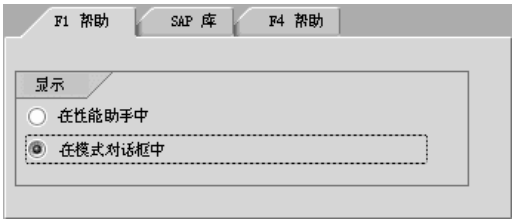
# SAPGUI Vuser 脚本提示

下列部分提供了 SAPGUI Vuser 的录制提示、回放提示和在场景中回放的提示。

## 录制提示

本部分提供了 SAPGUI Vuser 脚本的录制提示。

- 确保将操作录制到适当的部分中：将登录过程录制到 **vuser_init** 部分，将要重复的操作录制到 **Actions** 部分，将注销过程录制到 **vuser_end** 部分。
- 在录制多协议脚本（其中 SAPGUI 客户端包含 Web 控件）时，请在录制前关闭 *SAPLogon* 应用程序。
- 使用模式对话框获取 F1 帮助。指示 SAPGUI 客户端在模式对话框中打开 F1 帮助。选择“帮助”>“设置”。单击“F1 帮助”选项卡，并在“显示”部分中选择“在模式对话框中”选项。

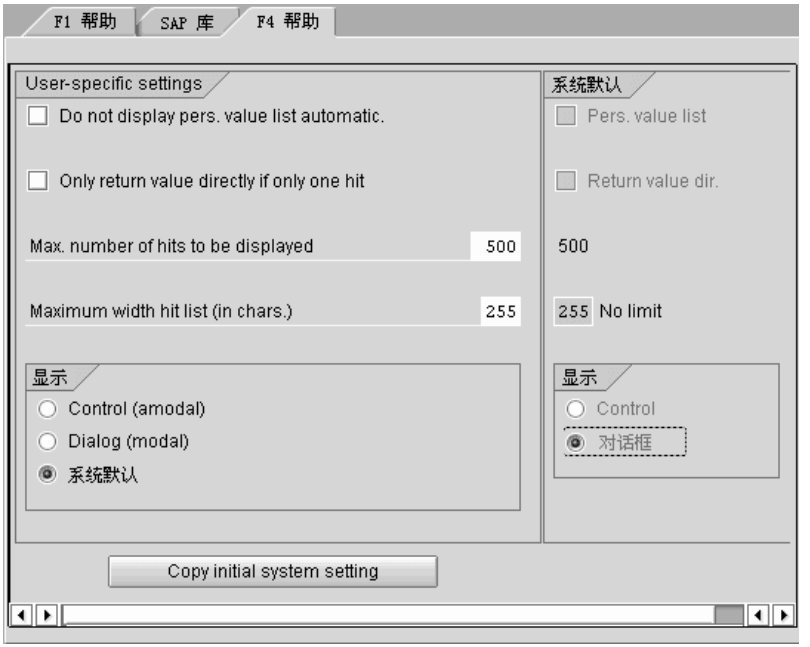



- 使用模式对话框获取 F4 帮助。指示 SAPGUI 客户端在模式对话框中打开 F4 帮助。以下过程必须由 SAP 管理员执行：

**要在模式对话框中打开 F4 帮助，请执行下列操作：**

**1** 确保所有用户都已从服务器上注销。

**2** 选择 “帮助” > “设置”。单击 “F4 帮助” 选项卡。




**3** 在 “显示” 部分 （左下方）中，选择 “系统默认值”。

**4** 在 “系统默认值” 的 “显示” 部分 （右下方）中，选择 “对话框”。

**5** 保存更改 – 单击 “复制初始系统设置” 或 CTRL+S。

**6** 验证状态栏是否显示 “数据已保存” 消息。

**7** 关闭会话。

**8** 通过 SAP 管理控制台重新启动服务。

### 回放提示

在独立模式下回放脚本之前，请遵循这些规则：

- 用实际密码替换录制期间生成的 **sapgui_logon** 函数中的加密密码。这是该函数的第二个参数，位于用户名之后：sapgui_logon("user", "pswd", "800", "EN"); 要增强安全性，可以对代码中的密码进行加密。选择密码文本（实际文本，而不是*****），并从右键单击菜单中选择 “加密字符串”。 VuGen 将在密码位置插入 **lr_decrypt** 函数：sapgui_logon("user", lr_decrypt("3ea037b758"), "800", "EN");。
- 第一次运行脚本时，请对 VuGen 进行配置，使其在回放期间显示 SAPGUI 用户界面，从而可以通过 UI 查看正在执行的操作。要在回放期间显示用户界面，请打开运行时设置 (F4) 并在 “SAPGUI：常规”节点中选择 “回放期间显示 SAP 客户端”选项。执行负载场景期间，请禁用该选项，因为显示多个 Vuser 的 UI 要占用大量的系统资源。

### 在场景中回放的提示

下列部分为在 Controller 或负载生成器计算机上运行脚本提供了配置提示。

#### Controller 设置

在场景中使用 LoadRunner Controller 运行脚本时，请在负载场景配置中设置下列值：

**加压**：在计划程序中依次进行，确保正确登录。

**思考时间**：运行时设置中的随机思考时间。

**每个负载生成器上的用户数**：对于 512 MB 内存的计算机，在 “负载生成器”对话框中有 50 个 Vuser。

注意，只有 SAP 服务器的 4.6 或更低版本才支持 SAP 监视器。

#### 负载生成器设置

在场景中运行脚本时，请检查代理模式并在负载生成器计算机上配置终端会话。

**代理模式**：确保 LoadRunner 远程代理以进程模式运行。不支持服务模式。

要检查该模式，请将鼠标移到 Windows 任务栏区域中的代理图标上，阅读其中的描述。如果该描述显示 “LoadRunner 代理服务”，则表明该代理正在作为服务运行。

**要将代理作为进程重新启动，请执行下列操作：**

1 停止该代理。右键单击 LoadRunner 代理图标并选择 “关闭”。

2 在 LoadRunner 安装下，运行 *launch_service\bin* 目录中的 *magentproc.exe*。

3 为确保下次启动计算机时能够启动正确的代理，请将代理服务的启动类型由自动更改为手动。然后向 Windows 启动文件夹中添加 *magentproc.exe* 的快捷方式。

**终端会话：**由于可用图形资源有限，运行 SAPGUI Vuser 的计算机可能只能运行有限数量的 Vuser。要增加每台计算机的 Vuser 数，请在负载生成器计算机上打开更多的终端服务器会话。从 “开始” > “程序” > “LoadRunner” > “高级设置”中选择 “代理配置”，并选择 “启用终端服务”选项。在负载生成器计算机属性中指定终端会话数。详细信息，请参阅 《LoadRunner Controller 用户指南》中的 “配置终端服务”。

## SAPGUI Vuser 脚本疑难解答

**问题 1：**能够录制脚本，但为什么在回放时失败？

**解答：**确保 LoadRunner 远程代理正以进程模式运行。不支持服务模式。详细信息，请参阅第 611 页的 “回放提示”。

**问题 2：**为什么某些 SAPGUI 控件没有被录制？

**解答：**某些 SAPGUI 控件只能出现在其菜单或工具栏上下文中。请尝试使用不同的方法 （如通过菜单选项、上下文菜单、工具栏等）执行有问题的任务。

**问题 3：**为什么不能在 VuGen 中录制或回放任何脚本？

**解答：**

1 请验证是否安装了最新的 SAPGUI 6.20 修补程序。允许的最低修补级别为修补程序 32。

2 确保已启用脚本功能。请参阅第 583 页的 “检查配置”。

3 请验证是否在 Windows 客户端的 SAPGUI 中禁用了通知。单击 “自定义本地布局”按钮或按 ALT+F12。单击 “选项”并选择 “脚本”选项卡。清除两个 “通知”选项。

**问题 4：**尝试运行脚本时，弹出的错误消息是什么意思？

**解答：**某些 SAP 应用程序将存储每个用户的最终布局 （例如，哪些帧是可见的，哪些帧是隐藏的）。如果在录制脚本后存储的布局发生了变化，则可能会发生回放问题。例如，在 ME52N 事务中，使用 “文档概览开 / 关”按钮可更改可见帧的数目。

**如果发生这种情况：**

1 请在开始回放之前将该事务导航到其录制期间所在的相同位置上。可以使用快照查看器查看录制时的布局。

2 向脚本中添加语句以确保回放期间该事务处于所需的布局中。例如，如果可选帧有碍于回放，可插入验证函数检查该帧是否已打开。如果已打开，则可以单击按钮关闭它。要查看验证示例，请参阅第 597 页的 “添加验证函数”。

**问题 5：**在远程计算机上运行脚本时，能否使用单一登录机制？

**解答：**不能。 VuGen 不支持单一登录连接机制。在 SAPGUI 客户端中，打开 “高级选项”并清除 “启用安全网络通信”功能。注意，修改连接首选项后必须重新录制脚本。

**问题 6：**VuGen 能录制所有 SAP 对象吗？

**解答：**录制功能不适用于 SAPGUI 脚本不支持的对象。有关那些对象类型的信息，请参阅录制日志。

**问题 7：**是否支持所有业务流程？

**解答：**VuGen 既不支持调用 APO 或 Microsoft Office 模块控件的业务流程，也不支持需要使用 GuiXT 的业务流程。可以通过 Windows 客户端 SAPGUI 的 “选项”菜单禁用 **GuiXT**。

# 其他资源

### LoadRunner

要获得有关对话框的联机帮助，请在对话框中按 F1 键。也可以选择 “帮助” > “目录和索引”手动打开该帮助。在 “索引”选项卡中，找到 “SAPGUI Vuser 脚本”项并单击适当的子项。

要获得某个函数的联机帮助，请在该函数或步骤中单击，然后按 F1 键打开《LoadRunner Function Reference》。

### SAP

详细信息，请参阅下列资源：

- **SAP 网站** – www.sap.com
- **SAP 注释** – https://websmp103.sap-ag.de/notes

  注释 #480149：为前端的用户脚本新建配置文件参数

  注释 #587202：拖放是 SAPGUI 界面的已知限制条件
- **SAP 修补程序** – https://websmp104.sap-ag.de/patches

  Windows 的 SAP GUI - SAPGUI 6.20 修补程序 （允许的最低级别为 32）

# 49

# 开发 SAP-Web Vuser 脚本

使用 VuGen 的 SAP-Web Vuser 类型，可以录制 SAP Workplace 或 SAP Portal 客户端中的活动。

本章描述：

- 创建 SAP-Web Vuser 脚本
- 设置 SAP-Web 录制选项
- 了解 SAP-Web Vuser 脚本
- 回放 SAP-Web Vuser 脚本

***以下信息仅适用于 SAP-Web 协议。***

# 关于开发 SAP-Web Vuser 脚本

可以使用 VuGen 录制典型的 SAP 业务流程。在业务流程期间， VuGen 将录制 SAP Workplace 或 Portal 的活动，并生成 Vuser 脚本。在浏览器中执行操作时，VuGen 将生成描述此活动的函数。每个函数都以 **web** 前缀开头。回放期间，这些函数模拟 SAP Workplace 或 Portal 客户端上的用户活动。例如， **web_url** 将导航到 PageBuilder。

```
web_url("PageBuilder[myPage]",
"URL=http://sonata.mercury.co.il/hrnp$30001/sonata.mercury.co.il:80/Acti
on/PageBuilder[myPage]?pageName=com.sapportals.pct.home.mynews",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://sonata.mercury.co.il/sapportal",
        "Snapshot=t2.inf",
        "Mode=HTML",
        EXTRARES,
        "Url=/irj/services/laf/themes/portal/sap_mango_polarwind/..,
ENDITEM,
        LAST);
```

# 创建 SAP-Web Vuser 脚本

创建 SAP-Web Vuser 脚本的第一步是选择 Vuser 和脚本类型。 **SAP-Web** Vuser 在 **ERP/CRM** 类别下。可以创建单协议或多协议 Vuser 脚本。

**要创建 SAP-Web Vuser，请执行下列操作：**

1. 调用 VuGen 并选择 “文件” > “新建”。
2. 要录制未在 Workplace 或 Portal 客户端内合并任何 SAPGUI 控件的会话，请使用 **SAP-Web** Vuser 类型创建单协议 Vuser 脚本。

3 要录制使用 SAPGUI 控件的会话，请创建多协议 Vuser 脚本。同时指定 **SAP-Web** 和 **SAPGUI** Vuser 类型。

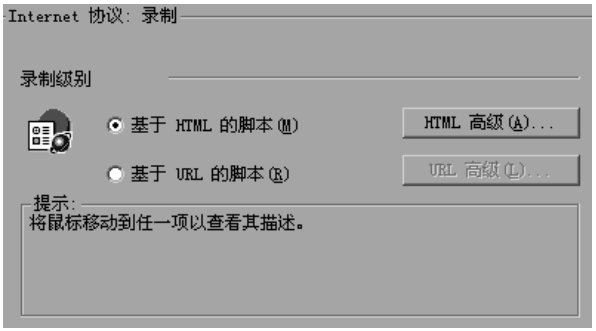
## 设置 SAP-Web 录制选项

使用录制选项，可以设置 VuGen 生成 Vuser 脚本的首选项。

建议将 “Internet 协议：录制”节点设置为：

**对于 SAP Workplace 录制**：基于 URL 的脚本

**对于 SAP Portal 录制**：基于 HTML 的脚本 （默认）




有关其他与 Web 相关的录制选项的信息，请参阅第 35 章 “为 Internet 协议设置录制选项”。

# 了解 SAP-Web Vuser 脚本

SAP-Web Vuser 脚本通常包含数个构成业务流程的 SAP 事务。业务流程由模拟用户操作的函数构成。有关这些函数的信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）中的 Web 函数。

下例显示典型的 SAP Portal 客户端录制：

```
vuser_init()
{
	web_reg_find("Text=SAP Portals Enterprise Portal 5.0",
		LAST);

	web_set_user("junior{UserNumber}",
		lr_decrypt("3ed4cfe457afe04e"),
		"sonata.mercury.co.il:80");

	web_url("sapportal",
		"URL=http://sonata.mercury.co.il/sapportal",
		"Resource=0",
		"RecContentType=text/html",
		"Snapshot=t1.inf",
		"Mode=HTML",
		EXTRARES,

"Url=/SAPPortal/IE/Media/sap_mango_polarwind/images/header/brandin
g_image.jpg",
"Referer=http://sonata.mercury.co.il/hrnp$30001/sonata.mercury.co.il:80/
Action/26011[header]", ENDITEM,

"Url=/SAPPortal/IE/Media/sap_mango_polarwind/images/header/logo.gif",
"Referer=http://sonata.mercury.co.il/hrnp$30001/sonata.mercury.co.il:80/
Action/26011[header]", ENDITEM,
...
		LAST);
```

下面的部分说明多协议录制，其中 Portal 客户端打开了一个 SAP 控件。请注意从 **web_xxx** 到 **sapgui_xxx** 函数的切换。

```
web_url("dummy",

"URL=http://sonata.mercury.co.il:1000/hrnp$30000/sonata.mercury.co.il:1
000/Action/dummy?PASS_PARAMS=YES&dummyComp=dummy&Tcode=V
A01&draggable=0&CompFName=VA01&Style=sap_mango_polarwind",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://sonata.mercury.co.il/sapportal",
        "Snapshot=t9.inf",
        "Mode=HTML",
        LAST);

    sapgui_open_connection_ex(" /H/Protector/S/3200 /WP",
        "",
        "con[0]");

    sapgui_select_active_connection("con[0]");

    sapgui_select_active_session("ses[0]");

    /* 运行脚本之前，在 logon 函数中输入密码替换星号 */

    sapgui_logon("JUNIOR{UserNumber}",
        "ides",
        "800",
        "EN",
        BEGIN_OPTIONAL,
            "AdditionalInfo=sapgui102",
        END_OPTIONAL);
```

# 回放 SAP-Web Vuser 脚本

创建并增强 SAP-Web Vuser 脚本后，请配置它的运行时设置，并从 VuGen 中运行它以检查其功能。

通过运行时设置可以控制回放期间 Vuser 的行为。请在运行 Vuser 脚本之前配置这些设置。您可以同时设置常规运行时设置和与 Web 相关的运行时设置。

常规设置包括运行逻辑、步、日志记录、思考时间和性能首选项。有关常规运行时设置的信息，请参阅第 9 章 “配置运行时设置”。对于特定于 SAP-Web 的设置，请参阅第 37 章 “配置 Internet 运行时设置”。

配置运行时设置后，请保存该 Vuser 脚本，并将其作为独立测试在 VuGen 中运行，以验证它是否运行正常。详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

验证 Vuser 脚本能够正常工作后，请将其集成到场景中。详细信息，请参阅《LoadRunner Controller 用户指南》。

# 50

# 开发 Siebel-Web Vuser 脚本

使用 VuGen 录制 Siebel Web 环境中的活动，并生成 Vuser 脚本。运行脚本时，Vuser 将模拟 Siebel 环境中的操作。

本章描述：

- 录制 Siebel-Web 会话
- 关联 Siebel-Web 脚本
- Siebel-Web Vuser 脚本疑难解答

***以下信息仅适用于 Siebel-Web Vuser 脚本。***

## 关于开发 Siebel-Web Vuser 脚本

Siebel-Web 协议与标准 Web Vuser 类似，通过在默认设置中进行一些更改，可以使该协议使用 “Siebel 客户关系管理 (CRM)” 应用程序。

录制 Siebel 会话中的典型活动。VuGen 录制这些操作并生成带有 **web_** 前缀的函数，这些函数模拟相应的操作。

以下部分提供有关如何使用录制 Siebel-Web 的 Vuser 脚本的提示，并提供需要进行关联的参数的示例。

## 录制 Siebel-Web 会话

录制 Siebel-Web 会话时，请使用下列规则：

**要录制 Siebel-Web Vuser 脚本，请执行下列操作：**

1. 从 ERP 类别中创建 Siebel-Web 类型 Vuser 脚本。
2. 设置下列录制选项：
   - 录制节点：**基于 HTML 的脚本**

     高级 HTML – 脚本选项：**仅包含明确 URL 的脚本**

     高级 HTML – 非 HTML 生成的元素：**不录制**
   - 高级节点：清除 “重置每个操作的上下文”选项
3. 在 *vuser_init* 部分录制登录。
4. 将 “业务流程”录制到 *Action1*。
5. 在 *vuser_end* 部分录制注销。
6. 在 “运行时设置”中，清除 “浏览器仿真”节点中的 “每次循环模拟一个新用户”选项。

有关如何配置 “录制选项”和 “运行时设置”的详细信息，请参阅第 35 章 “为 Internet 协议设置录制选项”和第 37 章 “配置 Internet 运行时设置”。

## 关联 Siebel-Web 脚本

在 Siebel-Web 会话中，大多数变量都可以使用该协议的内置关联规则进行关联。详细信息，请参阅第 41 章 “配置 Web Vuser 脚本的关联规则”。

本节讨论：

- 回调
- 关联 SWECount
- 关联行 ID 长度
- 处理时间戳 (SWETS)

### 回调

当 VuGen 使用边界作为条件检测匹配时，它将对参数名及其相应的值执行回调。回调将分析数组并将数组的每个参数保存到单独的参数中。在 Siebel 的关联规则中定义了这些回调。

由这些规则添加的 **web_reg_save_param** 函数具有附加参数：

AutoCorrelationFunction=<callback_name>

回调将为对 **web_reg_save_param** 进行的每个调用保存数个参数。由这些回调保存的参数有：

<callback_name>_1 = arg1
<callback_name>_2 = arg2
...
<callback_name>_rowid = <rowid>

```
web_reg_save_param("Siebel_Star_Array118",
        "LB/IC=`ValueArray`",
        "RB/IC=`",
        "Ord=2",
        "Search=Body",
        "RelFrameId=1",
        "AutoCorrelationFunction=flCorrelationCallbackParseStarArray",
        LAST);
```

然后可以在后面所跟的 **web_submit_data** 函数中替换这些参数

### 关联 SWECount

SWECount 参数值通常是由一位或两位构成的小数字。通常很难确定用参数替换所录制的值的位置。

在 **web_submit_data** 函数中， VuGen 仅替换 SWEC 字段中的值。

在 URL 中， VuGen 仅替换显示在字符串 “SWEC=”或 “SWEC”后面的值。

所有 SWECount 关联的参数名都是相同的。

### 关联行 ID 长度

在某些特定的情况下， *rowid* 的前面是其长度，该长度使用十六进制格式编码。由于该长度可以更改，因此该值必须进行关联。

例如，字符串 xxx6_1-4ABCyyy 是长度值和 RowID 构成，其中 6 是长度，而 1-4ABC 是 RowID。

如果将用于关联字符串的参数定义为

xxx{rowid_Length}_{rowid}yyy

则 VuGen 将使用该增强的关联在字符串之前生成以下函数：

web_save_param_length("rowid", LAST);

该函数获取 rowid 的值，并以十六进制格式将其长度保存到参数 *rowid_length* 中。

### 处理时间戳 (SWETS)

脚本中的 SWETS 值是自 1970 年 1 月 1 日午夜以来的毫秒数。

VuGen 将脚本中的所有非空时间戳替换为参数 {SiebelTimeStamp}。在将值保存到该参数之前， VuGen 将生成以下函数：

web_save_timestamp_param("SiebelTimeStamp", LAST);

该函数将当前时间戳保存到 *SiebelTimeStamp* 参数中。

## Siebel-Web Vuser 脚本疑难解答

本部分讨论创建 Siebel-Web Vuser 脚本时可能遇到的典型错误：

- 后退或刷新错误
- 相同值
- 无内容 HTTP 响应
- 恢复上下文
- 找不到记录
- 文件结束
- 无法检索搜索类别

### 后退或刷新错误

与 “后退”或 “刷新”相关的错误消息通常包含以下文本：

我们无法处理您的请求。很可能是因为使用了浏览器 “后退”或 “刷新”按钮而导致出现该错误。

**原因**：该问题可能的原因是：

- 未为当前请求正确关联 SWEC。
- 未为当前请求正确关联 SWETS。
- 在未更新 SWEC 的情况下向 Siebel 服务器提交了两次请求。
- 上一请求应已打开帧以便浏览器进行下载。未在服务器上创建该帧，可能是因为录制后 SWEMethod 已更改。

### 相同值

对 “相同值”错误的典型网页响应是：

**@0`0`3`3``0`UC`1`Status`Error`SWEC`10`0`1`Errors`0`2`0`Level0`0`ErrMsg`已存在 “Name”的相同值。如果想要输入新的记录，请确保字段值唯一。`ErrCode`28591`**

**原因**：该问题可能的原因是：请求中的某个值 （在上述示例中，为 Name 字段的值）与数据库表其他行中的值重复。该值需要替换为唯一的值，以便用于每个用户的每次循环。推荐的解决方案是将行 ID 替换为其参数，以确保该值唯一。

### 无内容 HTTP 响应

“无内容 HTTP 响应”类型错误的典型 HTTP 响应是：

HTTP/1.1 204 无内容
服务器：Microsoft-IIS/5.0
日期：2003 年 1 月 31 日星期五 （GMT 时间 21:52:30）
内容语言：en
高速缓存控制：no-cache

**原因**：该问题可能的原因是：行 ID 根本未进行关联或者被错误关联。

### 恢复上下文

对 “恢复上下文”类型错误的典型网页响应是：

@0`0`3`3``0`UC`1`Status`Error`SWEC`9`0`1`Errors`0`2`0`Level0`0`ErrMsg` 恢复请求位置的上下文期间发生错误 `ErrCode`27631`

**原因**：该问题可能的原因是：行 ID 未进行关联或者被错误关联。

### 找不到记录

对 “找不到记录”类型错误的典型网页响应是：

@0`0`3`3``0`UC`1`Status`Error`SWEC`23`0`2`Errors`0`2`0`Level0`0`ErrMsg` 在视图中找不到记录：联系详细信息 – 机会视图小程序：机会列表小程序。 `ErrCode`27573`

**原因**：该问题可能的原因是：输入名 SWERowId 不包含网页中记录的行 ID。该输入名应该已被参数化。参数的源值可能已经更改其位置。

### 文件结束

对 “文件结束”类型错误的典型网页响应是：

@0`0`3`3``0`UC`1`Status`Error`SWEC`28`0`1`Errors`0`2`0`Level0`0`ErrMsg` 已发生了文件结束错误。请继续操作；如果问题仍存在，请让系统管理员检查应用程序配置。 `ErrCode`28601`

**原因**：该问题可能的原因是：输入名 SWERowId 不包含网页中记录的行 ID。该输入名应该已被参数化。参数的源值可能已经更改其位置。

### 无法检索搜索类别

对 “无法检索搜索类别”类型错误的典型网页响应是：

**原因**：该问题可能的原因是：未从服务器中下载搜索帧。当上一个请求未请求服务器正确创建搜索帧时，会出现这种错误。

# 51

# 创建 Baan Vuser 脚本

您可以使用 VuGen 开发 Baan Vuser 脚本。使用 VuGen 录制典型 Baan 会话，并使用 Baan Vuser 函数增强脚本。

本章描述：

- Baan Vuser 脚本入门
- Baan Vuser 函数
- 创建 Baan Vuser 脚本
- 了解 Baan Vuser 脚本
- 自定义 Baan Vuser 脚本

***下列信息仅适用于 Baan Vuser 脚本。***

## 关于开发 Baan Vuser 脚本

使用 Baan 类型的 Vuser 脚本可以测试 Baan 应用程序以及负载情况下的系统性能。 VuGen 可以录制整个 Baan 会话，包括登录到 Baan 服务器的信息。

录制操作时， VuGen 会使用上下文相关函数创建一个脚本。上下文相关函数将根据 GUI 对象 （例如窗口、列表和按钮）描述被测试应用程序中的操作。每录制一个操作，就会生成一个函数描述选定的对象和执行的操作。

# Baan Vuser 脚本入门

在 VuGen 中录制 Baan Vuser 脚本之前，应确保您的计算机可以打开 Baan 会话。

创建 Baan Vuser 脚本时，请按下列步骤进行操作。

1 **在 VuGen 中创建 Baan 脚本**。

新建 Baan 模板。

2 **录制用户操作**。

录制典型用户操作。

3 **添加事务、集合、注释和消息**。

使用 “插入”菜单添加事务、集合、注释和消息，以便增强脚本。

4 **添加异常处理和设置运行时属性**。

添加异常处理函数，设置思考时间和超时时间。配置日志和循环的运行时设置。

5 **执行参数化**。

用参数替换录制的常量。

6 **保存并运行 Vuser 脚本**。

从 VuGen 中运行 Baan 脚本，并查看 “执行日志”选项卡获得运行时信息。

# Baan Vuser 函数

Baan 用户会话期间， VuGen 将自动录制本部分中列出的大多数函数。也可以手动将任一函数编写到脚本中。有关 Baan Vuser 函数的详细信息，请参阅《LoadRunner Function Reference》（“帮助” > “函数参考”）。

### 会话函数

| | |
|---|---|
| **init_session** | 初始化 Baan 会话 |
| **close_session** | 关闭所有 Baan 会话和窗口 |
| **start_session** | 开始特定的 Baan 会话 |
| **set_exception** | 指定如何处理异常 |
| **set_think_time** | 设置思考时间范围 |
| **set_default_timeout** | 设置默认超时 |

### 按钮对象函数

| | |
|---|---|
| **button_press** | 激活按钮 |
| **button_set** | 设置指定单选按钮或复选框的状态 |

### 编辑对象函数

| | |
|---|---|
| **edit_get_text** | 返回编辑对象中的文本 |
| **edit_set** | 替换编辑对象的全部内容 |
| **edit_set_insert_pos** | 将光标放置到指定点 |
| **edit_set_selection** | 选择编辑对象中的文本 |
| **edit_type** | 在编辑对象中键入字符串 |

### 列表对象函数

| | |
|---|---|
| **list_activate_item** | 激活列表中的项 |
| **list_select_item** | 选择列表项 |
| **list_get_selected** | 返回列表中当前选择的项 |
| **list_expand_item** | 显示列表中的隐藏项 |
| **list_collapse_item** | 隐藏列表中的项 |

### 菜单对象函数

| | |
|---|---|
| **menu_select_item** | 选择菜单中的项 |

### 对象函数

| | |
|---|---|
| **obj_get_info** | 返回对象属性的值 |
| **obj_get_text** | 读取对象中的文本 |
| **obj_mouse_click** | 在对象内单击 |
| **obj_mouse_dbl_click** | 在对象内双击 |
| **obj_mouse_drag** | 在对象内拖动鼠标 |
| **obj_type** | 将键盘输入发送到对象 |

### 滚动对象函数

| | |
|---|---|
| **scroll_drag_from_min** | 将滚动对象拖动到距最小位置指定的距离处 |
| **scroll_line** | 滚动指定数目的行 |
| **scroll_page** | 将滚动对象移动指定数目的页 |

### 选项卡和工具栏对象函数

| | |
|---|---|
| **tab_select_item** | 选择活动窗口中的选项卡 |
| **toolbar_button_press** | 单击工具栏按钮 |

### 静态对象函数

| | |
|---|---|
| **static_get_text** | 返回静态文本对象的内容 |

### 同步函数

| | |
|---|---|
| **obj_wait_info** | 等待对象属性的值 |
| **tbl_wait_selected_cell** | 等待指定的单元格出现在焦点位置 |
| **win_wait_info** | 等待窗口属性的值 |

### 表函数

| | |
|---|---|
| **tbl_activate_cell** | 在指定的单元格中单击“输入” |
| **tbl_get_cell_data** | 检索表中指定单元格的内容 |
| **tbl_get_selected_cell** | 返回表中当前处于焦点的单元格 |
| **tbl_press_zoom_button** | 单击表的缩放按钮 |
| **tbl_set_cell_data** | 将表中单元格的内容设置为指定的文本 |
| **tbl_set_selected_cell** | 选择表单元格 |
| **tbl_set_selected_ rows** | 选择表中指定的行 |

### 窗口对象函数

| | |
|---|---|
| **set_window** | 指定接收后续输入的窗口 |
| **win_activate** | 激活窗口 |
| **win_close** | 关闭窗口 |
| **win_get_text** | 读取窗口中的文本 |
| **win_get_info** | 返回窗口属性的值 |
| **win_max** | 最大化窗口填充整个屏幕 |
| **win_min** | 将窗口最小化为图标 |
| **win_mouse_click** | 在窗口内单击 |
| **win_mouse_dbl_click** | 在窗口内双击 |
| **win_mouse_drag** | 在窗口内拖动鼠标 |
| **win_move** | 将窗口移动到新的绝对位置 |
| **win_resize** | 调整窗口大小 |
| **win_restore** | 将窗口从图标化或最大化状态恢复到原始大小 |
| **win_type** | 将键盘输入发送到窗口 |

### 其他函数

| | |
|---|---|
| **wait** | 使测试执行暂停指定的时间长度 |

可以使用通用 Vuser 函数 （如 **lr_output_message** 和 **lr_rendezvous**）进一步增强脚本。有关 Vuser 函数的信息，请参阅 《LoadRunner Function Reference》。

# 创建 Baan Vuser 脚本

创建 Baan 模板后，开始录制用户操作。

**要新建一个 Baan Vuser 脚本，请执行下列操作：**

1 选择 *vuser_init* 部分，以便将登录过程录制到该部分。

2 单击 “录制”按钮并在 “开始录制”对话框中指定 Baan 应用程序的位置。

3 切换到 *Actions* 部分并录制典型用户操作。

4 插入思考时间、处理异常和设置超时的 Baan Vuser 函数。

```
set_think_time(MINTHINK,MAXTHINK);
set_window ("Menu browser [User:bsp] [812]", 10);
menu_select_item ("File;Run Program...");
...
```

5 向脚本中添加事务。选择 “插入” > “开始事务”指定事务的开始，选择 “插入” > “结束事务”指定事务的结束。

```
lr_start_transaction("all_str_ses");
  button_press0 ("F1_OK");
  set_window ("tdpur4101m000:Maintain Purchase Orders [812]", 300);
lr_end_transaction("all_str_ses", LR_PASS);
```

6 使用 “插入”菜单向脚本中添加集合点、注释或消息。

7 参数化脚本。单击要用参数替换的字符串 （在引号中），右键单击并选择 “使用参数替换”。详细信息，请参阅第 7 章 “定义参数”。

8 配置适当的循环和日志运行时设置

9 保存并在 VuGen 中运行脚本。

# 了解 Baan Vuser 脚本

录制的脚本将显示录制期间用户执行的所有操作。上下文相关函数将显示对应用程序对象执行的所有操作。在下例中， VuGen 录制将焦点置于窗口、选择菜单项以及单击按钮等操作。此外，还对事务作了标记以便分析对象 *Form1* 成为焦点所需的时间。

```
set_window ("tccom1501m000:Display Customers [550]", 30);
menu_select_item ("Edit;Find...Ctrl+F");
set_window ("Display Customers - Find", 300);
type ("100004");
lr_start_transaction("rses_find");
button_press0 ("F1_OK");
set_window ("tccom1501m000:Display Customers [550]", 30);
obj_wait_info("Form 1","focused","1",100);
lr_end_transaction("rses_find", LR_PASS);
```

可以添加控制流逻辑，在脚本内创建循环，而不需要对整个脚本执行循环。

```
for (loop = 0; loop < READLOOP; loop++)
{
 set_window ("tccom1501m000:Display Customers [550]", 30);
 menu_select_item ("Edit;Find...Ctrl+F");
 set_window ("Display Customers - Find", 300);
 type ("100004");
 lr_start_transaction("rses_find");
 button_press0 ("F1_OK");
 set_window ("tccom1501m000:Display Customers [550]", 30);
 obj_wait_info("Form 1","focused","1",100);
 lr_end_transaction("rses_find", LR_PASS);
...
```

注意，可能需要参数化语句，以便避免将数据复制到数据库中。详细信息，请参阅第 7 章 “定义参数”。

# 自定义 Baan Vuser 脚本

在 VuGen 中，可以随时查看和编辑脚本。可以使用 Baan 特定的函数在下列区域自定义脚本执行：

- 思考时间
- 处理异常
- 设置超时

## 思考时间

可以设置脚本执行时的思考时间范围。思考时间模拟实际用户的工作模式，即用户在操作之间暂停的时间。使用 set_think_time 函数设置思考时间范围的开始和结束。每一语句之后 Vuser 会暂停一段思考时间，时间长度为指定范围内的随机值。

如果希望在整个脚本中思考时间保持不变，则可以将开始范围和结束范围定义为常量，如下例所示。

在下例中，思考时间范围被设置为 500 至 1000 毫秒：

```
#define MINTHINK 500
#define MAXTHINK 1000

int LR_FUNC Actions(LR_PARAM p)
{
     set_think_time(MINTHINK,MAXTHINK);
     set_window ("Menu browser [User:bsp] [812]", 10);
...
```

## 处理异常

可以指示 Baan Vuser 如何处理回放期间出现的异常，例如消息或错误窗口。

使用 **set_exception** 函数，可以指定出现异常时执行的函数。

在下例中，**set_exception** 函数指示 Vuser 当 Print Sales Invoices 窗口打开时执行 *close* 函数。 *close* 函数是脚本中已定义的函数。

```
int close(char title[])
{
	win_close(title);
}
Actions()
{
	set_exception("ttstps0014:Print Sales Invoices",close);
	set_window ("Menu browser [User:bsp] [812]", 10);
	menu_select_item ("File;Run Program...");
	set_window ("ttdsk2080m000:Run Program [812]", 10);
	type ("tdsls4101m000");
	...;
```

### 设置超时

可以设置函数的默认超时时间。该超时适用于使用同步的所有函数，例如 **obj_wait_info** 和 **win_wait_info** 等。

在包含超时时间参数的函数 （如 **set_window**）中，指定的超时时间将覆盖默认的超时时间。

```
button_press ("F3_Continue");
win_wait_info("ttstpsplopen:Select Device [000]","displayed","0",10);
```

# 第 XI 部分

## 传统协议

# 52

# 介绍 RTE Vuser 脚本

RTE Vuser 在 Windows 环境中操作终端仿真器。本章描述基于 Windows 的 RTE Vuser 脚本的开发。

本章描述：

- 介绍 RTE Vuser
- 了解 RTE Vuser 技术
- RTE Vuser 脚本入门
- 使用 TE 函数
- 将终端键映射到 PC 键盘键

***以下信息仅应用于 RTE (Windows) Vuser 脚本。***

## 关于开发 RTE Vuser 脚本

RTE Vuser 操作终端仿真器，以便加载测试客户端 / 服务器系统。

您可以使用 VuGen 来录制终端仿真器会话以表示实际用户的操作。然后，可以使用事务和同步函数来增强您录制的脚本。

本章描述基于 Windows 的 RTE Vuser 脚本的开发。

# 介绍 RTE Vuser

RTE Vuser 将字符输入键入终端仿真器，然后将数据提交到服务器，并等待服务器响应。例如，假设您有一台用于维护某个维修公司的客户信息的服务器。现场服务代表每次修改数据时，都通过调制解调器使用终端仿真器来访问服务器数据库。服务代表访问有关客户的信息，然后录制他执行的修理操作的详细信息。

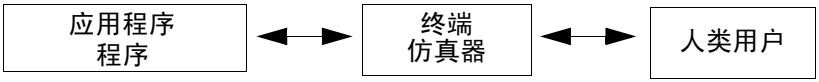
您可以使用 RTE Vuser 来模拟这一案例。 RTE Vuser 将执行下列操作：

1. 在命令行键入 “**60**”以打开应用程序。
2. 键入 “**F296**”– 现场服务代表的编号。
3. 键入 “**NY270**”– 客户编号。
4. 等待屏幕上显示文字 “Details”。显示 “Details”表明客户的所有详细信息都在屏幕中显示。
5. 键入 “**Changed gasket P249, and performed Major Service**”– 当前修理操作的详细信息。
6. 键入 “**Q**”以关闭应用程序。

您可使用 VuGen 来创建 RTE Vuser 脚本。脚本生成器会录制人类用户在终端仿真器中的操作。它会录制在终端窗口中的键盘输入，生成相应的语句，然后将它们插入 Vuser 脚本中。进行录制时，脚本生成器自动将同步函数插入到该脚本中。详细信息，请参阅第 55 章 “同步 RTE Vuser 脚本”。

# 了解 RTE Vuser 技术

RTE Vuser 可以模拟实际用户的操作。人类用户使用终端或终端仿真器来操作应用程序。




在 RTE Vuser 环境中，由 Vuser 来替代人类。由 Vuser 操作终端仿真器 RTE-Gate。

RTE-Gate 的作用如同一个标准的终端仿真器，它支持常用协议，例如 IBM 3270、IBM 5250、VT100 和 VT220。由于 RTE-Gate 专门针对负载测试进行了增强，因此您可以使用它完成下列任务：

- 使用利用 LoadRunner 函数操作应用程序的 Vuser 脚本来替代人类用户。
- 录制您对应用程序执行的操作，从而快速而容易地生成 Vuser 脚本。
- 在一台工作站上运行大量的 RTE Vuser。

使用 Controller 控制终端 （查看、隐藏和终止）。

## RTE Vuser 脚本入门

本部分概述使用 VuGen 开发 RTE Vuser 脚本的过程。

**要开发 RTE Vuser 脚本，请执行下列操作：**

1 **使用 VuGen 录制基本脚本。**

使用虚拟用户生成器 (VuGen) 录制您在终端仿真器中执行的操作。 VuGen 会录制终端窗口中的键盘输入，生成相应的语句，然后将这些语句插入 Vuser 脚本中。

详细信息，请参阅第 53 章 “录制 RTE Vuser 脚本”。

2 **增强脚本。**

通过向脚本中插入事务、集合点、同步函数和控制流结构来增强 Vuser 脚本。

详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

3 **定义参数 （可选）。**

为录制到脚本中的固定值定义参数。通过将这些固定值替换为参数，可以使用不同的值多次重复相同的业务流程。

详细信息，请参阅第 7 章 “定义参数”。

4 **配置运行时设置。**

运行时设置用于控制脚本执行过程中 Vuser 的行为。这些设置包括循环、日志和计时信息。

详细信息，请参阅第 9 章 “配置运行时设置”。

**5 通过 VuGen 运行脚本。**

从 VuGen 运行脚本以验证该脚本是否能够正确运行。查看标准输出以验证程序是否能够与服务器正常通信。

详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

成功创建 RTE 脚本后，可将该脚本集成到场景中。有关将脚本集成到场景中的详细信息，请参阅您的 《LoadRunner Controller 用户指南》。

## 使用 TE 函数

为模拟与服务器进行通信的终端而开发的函数称为 TE Vuser 函数。每个 TE Vuser 函数都有一个 **TE** 前缀。在 RTE 会话期间， VuGen 自动录制本节所列的大多数 TE 函数。你还可以手动将其中任一函数编写到您的脚本中。有关 TE 函数的语法和示例，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

### 终端仿真器连接函数

| | |
|---|---|
| **TE_connect** | 将终端仿真器连接到指定的主机 |

### 文本检索函数

| | |
|---|---|
| **TE_find_text** | 在屏幕的指定区域中搜索文本 |
| **TE_get_text_line** | 读取屏幕上指定行中的文本 |
| **TE_get_cursor_pos** | 返回光标的当前位置 |
| **TE_get_line_attribute** | 返回文本格式的有关信息 |

### 系统变量函数

| | |
|---|---|
| **TE_getvar** | 返回 RTE 系统变量的值 |
| **TE_setvar** | 设置 RTE 系统变量的值 |

### 错误代码函数

| | |
|---|---|
| **TE_perror** | 将错误代码打印到 “输出” 窗口 |
| **TE_sperror** | 将错误代码转换为字符串 |

### 键入函数

| | |
|---|---|
| **TE_type** | 将带格式的字符串发送到客户端应用程序 |
| **TE_typing_style** | 确定将文本键入终端仿真器的方式 |

### 同步函数

| | |
|---|---|
| **TE_wait_cursor** | 等待光标在终端窗口的指定位置出现 |
| **TE_wait_silent** | 等待客户端应用程序静默指定的秒数 |
| **TE_wait_sync** | 等待系统从 X-SYSTEM 或 “禁止输入” 模式返回 |
| **TE_wait_text** | 等待字符串在指定位置显示 |
| **TE_wait_sync_transaction** | 记录系统保持最近的 X SYSTEM 模式的时间 |

以下 TE 函数可以参数化：**TE_connect**、**TE_find_text**、**TE_get_text_line** 和 **TE_type**。有关在 Vuser 脚本中对函数进行参数化的详细信息，请参阅第 7 章 “定义参数”。

## 将终端键映射到 PC 键盘键

由于您将使用终端仿真器，因此应该能够使用 PC 键盘键来替代终端键盘。终端键盘中的许多键在 PC 键盘中未提供。例如， IBM 5250 键盘中的 HELP、AUTOR 和 PUSH 键等。要成功地操作终端仿真器以及任何关联的应用程序，您可能需要将某些终端键映射到 PC 键盘上的键。

**要将终端键映射到 PC 键盘键上的键，请执行下列操作：**

**1** 在终端仿真器中，选择 “选项” > “键盘映射”，或者单击 “键盘映射”按钮。将显示 “键盘映射”对话框。

**2** 单击工具栏上的 “键盘映射”按钮。要将终端键映射到 PC 键，请从上方的终端键中将键拖至下方的键盘中的 PC 键上。

您可以单击上方的键盘中的 Shift 和 / 或 Control 键，以显示只有首先选择这两个键中的某一个后，才能看到的其他键的功能。然后，可以将需要的键从上方的终端键盘拖至下方的 PC 键盘上的某个键。

要取消定义，请将 PC 键定义拖至垃圾桶中。此操作将还原 PC 键的默认功能。

要还原默认映射，请单击 “默认值”。

# 53

# 录制 RTE Vuser 脚本

您可以使用 VuGen 录制基于 Windows 的远程终端仿真 (RTE) Vuser 脚本。

本章描述：

- 新建 RTE Vuser 脚本
- 录制终端设置和连接过程
- 录制典型的用户操作
- 录制注销过程
- 设置 RTE 录制选项
- 在终端仿真器中键入输入
- 生成唯一设备名
- 设置字段分界字符

***下列信息仅适用于终端仿真 (RTE) Vuser 脚本。***

## 关于录制 RTE Vuser 脚本

您可以使用 VuGen 录制基于 Windows 的 RTE Vuser 脚本。 VuGen 使用 PowerTerm 终端仿真器模拟各种类型的终端。您可以使用 PowerTerm 执行典型的终端连接，然后执行典型的业务处理。之后，可以注销过程 . 在终端仿真器中执行典型的用户操作时， VuGen 会生成相应的语句，并将它们插入到 Vuser 脚本中。您可以在录制的同时查看并编辑脚本。

录制 RTE Vuser 脚本之前，请确保正确设置录制选项。录制 Vuser 脚本时，可以使用录制选项控制 VuGen 生成某些函数的方式。 VuGen 将在所有后续的录制会话中应用这些录制选项。

# 新建 RTE Vuser 脚本

将用户操作录制到 Vuser 脚本之前，必须打开一个脚本。您可以打开现有的脚本，也可以新建一个脚本。您可以使用 VuGen 新建 Vuser 脚本。

**要新建 RTE Vuser 脚本，请执行下列操作：**

**1** 选择 “开始” > “程序” > “LoadRunner” > “虚拟用户生成器”，启动 VuGen。将打开 VuGen 窗口。

**2** 单击 “新建”按钮。将打开 “新建虚拟用户”对话框：

**3** 选择 “原有”协议类型，然后选择 “终端仿真器 (RTE)”。单击 “确定”。VuGen 将生成并显示空的 RTE 脚本，并会定位光标以便在 *vuser_init* 部分中开始录制。

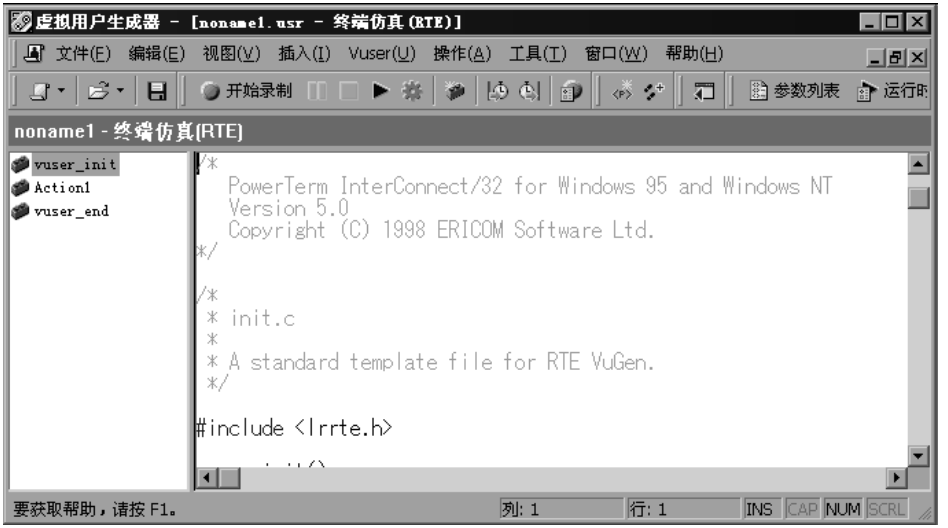
## 录制终端设置和连接过程

创建主干 Vuser 脚本后，您可以将终端设置和连接过程录制到该脚本中。当您录制 RTE Vuser 脚本时， VuGen 将使用 PowerTerm 终端仿真器。

**要录制终端设置和连接过程，请执行下列操作：**

**1** 打开现有的 RTE Vuser 脚本，或新建一个脚本。

**2** 在 “部分”框中，选择让 VuGen 将录制的语句插入到其中的部分。可用部分为 *vuser_init*、 *Actions* 和 *vuser_end*。




---

**注意：**请始终将终端设置和连接过程录制到 *vuser_init* 部分中。当您运行 Vuser 脚本的多个循环时，不会重复执行 *vuser_init* 部分，而只会重复执行 *Actions* 部分。有关循环设置的详细信息，请参阅第 9 章 “配置运行时设置”。

---

**3** 在 Vuser 脚本中，将光标置于要开始录制的位置。

**4** 单击 “录制”按钮 。将打开 PowerTerm 主窗口。

**5** 从 PowerTerm 菜单栏中，选择 “终端” > “设置”，以显示 “终端设置” 对话框。

**6** 从 “VT 终端” 和 “IBM 终端” 类型中选择仿真类型，然后单击 “确定”。

---

**注意**：选择 IBM 终端类型可连接至 AS400 计算机或 IBM 大型机；选择 VT 终端类型可连接至 UNIX 工作站。

---

**7** 选择 “通信” > “连接”，以显示 “连接” 对话框。

**8** 在 “会话类型” 下，选择要使用的通信类型。

**9** 在 “参数” 下，指定所需的选项。根据所选的会话类型，可用的参数会有所不同。有关参数的详细信息，请单击 “帮助”。

---

**注意：**您可以保存所定义的参数以备将来重复使用。要保存参数，请单击 “另存为”。保存的参数集将显示在 “会话列表” 框中。

---

**10** 单击 “连接”。 PowerTerm 将连接至指定的系统，并且 VuGen 会在插入点将 **TE_connect** 函数插入到脚本中。

插入的 **TE_connect** 语句如下所示：

```
/* *** 终端类型为 VT220-7。 */
    TE_connect("comm-type = telnet;host-name = frodo;
                    terminal-type = vt220-7;terminal-model = vt320;", 60000);
    if (TE_errno != TE_SUCCESS)
        return -1;
```

插入的 **TE_connect** 语句后面始终跟有 *if* 语句，该 if 语句检查 **TE_connect** 函数在回放期间是否成功。

---

**注意：**请不要在 Vuser 脚本中录制多于一个的与服务器的连接 **(TE_connect)**。

---

终端设置和连接过程已完成。如下所述，您现在可以开始将典型的用户操作录制到 Vuser 脚本中。

## 录制典型的用户操作

录制设置过程后，可以执行典型的用户操作或业务进程。您可以将以上进程录制到 Vuser 脚本的 *Actions* 部分中。运行脚本的多个循环时，只有 Vuser 脚本的*操作*部分会重复执行。有关设置循环的详细信息，请参阅第 9 章 “配置运行时设置”。

录制会话时， VuGen 录制的是文字笔划，而不是文字。因此，建议您不要将命令复制并粘贴到 PowerTerm 窗口中，而应直接键入它们。

**要录制用户操作，请执行下列操作：**

1. 打开现有的 RTE Vuser 脚本，然后单击 “部分”框中的*操作*。
2. 接下来在终端仿真器中执行典型的用户操作。当您键入时， VuGen 将生成相应的语句，并将它们插入到 Vuser 脚本中。如有必要，您可以在录制脚本时编辑录制的语句。

---

**注意：**默认情况下， VuGen 在生成相应的 **TE_type** 函数之前，最多会在连续击键操作之间等待 5 秒钟。要更改等待时间，请参阅第 653 页的 “设置 RTE 录制选项”。

---

录制完典型的用户操作后，请接着按照下一节中的描述录制注销过程。

## 录制注销过程

您可以将 Vuser 注销过程录制到 Vuser 脚本的 *vuser_end* 部分。当您运行脚本的多个循环时， *vuser_end* 部分不会重复执行。有关设置循环的详细信息，请参阅第 9 章 “配置运行时设置”。

**要录制注销过程，请执行下列操作：**

1. 确保已按照上一节中的描述执行并录制了典型的用户操作。
2. 在 VuGen 主窗口中，单击 “部分”框中的 *vuser_end*。
3. 执行注销过程。 VuGen 会将该过程录制到脚本的 *vuser_end* 部分中。

4. 单击 “录制”工具栏上的 “停止录制”。主 VuGen 窗口将显示所有录制的语句。

5. 单击 “保存”以保存录制的会话。将打开 “另存为”对话框 （仅限于新的 Vuser 脚本）。指定脚本名。录制脚本后，您可以在 VuGen 的主窗口中手动编辑该脚本。

## 设置 RTE 录制选项

通过设置录制选项，可以自定义 VuGen 为 RTE 函数生成的代码。您可以使用 “录制选项”对话框设置录制选项。要打开 “录制选项”对话框，请单击工具栏上的 “录制选项”按钮，或者选择 “工具” > “录制选项”。选择 “RTE:RTE” 节点。

您可以设置下列录制选项：

- 自动同步命令
- 自动屏幕标题注释 （仅限于 IBM 终端）
- 自动 X-System 事务 （仅限于 IBM 终端）
- 键盘录制超时

### 自动同步命令

VuGen 会在录制时自动生成一些 TE 同步函数，并将它们插入到脚本中。

1 您可以指定在录制过程中每次显示新屏幕时， VuGen 都生成 **TE_wait_sync** 函数。为此，请选中 “录制选项”对话框中的 “X-System”复选框。

默认情况下，在录制过程中每次显示新屏幕时， VuGen 都会自动生成 **TE_wait_sync** 函数。

---

**注意**：VuGen 仅在录制 IBM 块模式终端时，才会生成 **TE_wait_sync** 函数。

---

2 您可以指定 VuGen 在每个 **TE_type** 函数之前生成 **TE_wait_cursor** 函数。为此，请选中 “录制选项”对话框中的 “光标”复选框。

默认情况下， VuGen 不会自动生成 **TE_wait_cursor** 函数。

3 您可以指定 VuGen 在每个 **TE_type** 函数之前生成 **TE_wait_text** 函数 （如果合适的话）。为此，请选中 “录制选项”对话框中的 “提示符”复选框。

默认情况下， VuGen 不会在每个 **TE_type** 函数之前自动生成 **TE_wait_text** 函数。

---

**注意**：VuGen 仅在录制 VT 类型终端时，才会生成有意义的 **TE_wait_text** 函数。在录制块模式 (IBM) 终端时，请不要使用 **TE_wait_text** 函数自动生成功能。

---

### 自动屏幕标题注释 （仅限于 IBM 终端）

您可以指示 VuGen 在录制 Vuser 脚本时自动生成屏幕标题注释，并将这些注释插入到该脚本中。生成的注释通过标识每个新屏幕 （当它显示在终端仿真器中时），使录制的脚本更易阅读。生成的注释包含显示在终端仿真器窗口第一行上的文本。以下生成的注释说明终端仿真器中显示了 “Office 任务”屏幕：

```
/* OFCTSK                Office 任务                */
```

要确保 VuGen 在您录制脚本时自动生成注释，请选中 “录制选项”对话框中的 “生成屏幕标题注释”复选框。

默认情况下， VuGen 不会自动生成屏幕注释。

---

**注意**：仅当使用块模式终端仿真器 （例如 IBM 5250）时，才能自动生成注释。

---

### 自动 X-System 事务 （仅限于 IBM 终端）

您可以指定 VuGen 录制系统在场景运行期间处于 X SYSTEM 模式下的时间。为此， VuGen 会在每个 **TE_wait_sync** 函数后插入 **TE_wait_sync_transaction** 函数。每个 **TE_wait_sync_transaction** 函数都将创建一个名为 “default”的事务。每个 **TE_wait_sync_transaction** 函数都将录制系统在上一个 X SYSTEM 状态下所花费的时间。

要指示 VuGen 在录制的同时插入 **TE_wait_sync_transaction** 语句，请选中 “录制选项”对话框中的 “生成自动 X SYSTEM 事务”复选框。

默认情况下， VuGen 不会自动生成事务。

### 键盘录制超时

如果您在录制的同时在终端仿真器中键入文本，则 VuGen 将监控文本输入操作。每次击键后， VuGen 都会为下一次击键等待指定的时间长度。如果在指定的时间内没有后续击键操作， VuGen 会认为命令已完成。然后， VuGen 将生成相应的 **TE_type** 函数并将其插入到脚本中。

要设置 VuGen 在连续击键之间等待的最大时间量，请在 “键盘录制超时”框中输入时间量。

默认情况下， VuGen 在生成相应的 **TE_type** 函数之前，最多会在连续击键之间等待 5 秒钟。

# 在终端仿真器中键入输入

通过两个 TE Vuser 函数， Vuser 可以将字符输入 “键入” 到 RTE-GATE 终端仿真器中：

- **TE_type** 会将字符发送至终端仿真器。录制时，VuGen 会为终端窗口中的键盘输入自动生成 **TE_type** 函数。详细信息，请参阅下面的 “使用 TE_type 函数”。
- **TE_typing_style** 用于确定 Vuser 的键入速度。通过在 Vuser 脚本中插入 **TE_typing_style** 函数，您可以手动定义键入样式。详细信息，请参阅下面的 “设置键入样式”。或者，您还可以通过使用运行时设置来设置键入样式。详细信息，请参阅第 661 页的 “配置 RTE 运行时设置”。

---

**注意**：录制 RTE Vuser 脚本时，请不要使用鼠标在终端仿真器窗口中重定位光标。 VuGen 不录制这些光标运动。

---

## 使用 TE_type 函数

录制脚本时， VuGen 将录制所有的键盘输入并生成相应的 **TE_type** 函数。执行过程中， **TE_type** 函数会将带格式的字符串发送至终端仿真器。

键盘输入将被定义为常规文本字符串 （包括空格）。例如：

```
TE_type ("hello, world");
```

超过一个字符的输入键的名称用以字母 *k* 开头的标识符表示，并且括在大于号和小于号 (< >) 之间。例如，函数：

```
TE_type("<kReturn><kControl-y>");
```

描述了输入 Return 键以后，接着输入 Control 键和 y 键。其他一些示例包括：<kF1>、 <kUp>、 <kF10>、 <kHelp>、 <kTab>。

要确定键名，请录制该键上的操作，然后在录制的语句中检查其名称。

---

**注意：**当您编写 **TE_type** 语句（而非录制该语句）时，请使用 《LoadRunner Function Reference》（“帮助” > “函数参考”）中提供的键定义。

---

### 设置 TE_type 超时值

如果 Vuser 试图在系统处于 X SYSTEM （或禁止输入）模式时提交 TE_type 语句，则 Vuser 会等到 X SYSTEM 模式结束后再键入内容。如果系统在 X SYSTEM 模式中停留的时间超过 TE_XSYSTEM_TIMEOUT 毫秒，则 TE_type 函数将返回 TE_TIMEOUT 错误。您可以使用 TE_setvar 设置 TE_XSYSTEM_TIMEOUT 的值。 TE_XSYSTEM_TIMEOUT 的默认值为 30 秒。

### 允许 Vuser 提前键入

在某些情况下，您可能希望 Vuser 提交击键操作，即使系统正处于 X SYSTEM （或禁止输入）模式下。例如，您可能希望 Vuser 按 Break 键。此时可以使用 TE_ALLOW_TYPEAHEAD 变量，使 Vuser 能够提交击键操作（即使系统正处于 X SYSTEM 模式）。

将 TE_ALLOW_TYPEAHEAD 设置为零会禁用提前键入功能，而将其设置为任何非零数字都会允许提前键入。您可以使用 **TE_setvar** 来设置 TE_ALLOW_TYPEAHEAD 的值。默认情况下， TE_ALLOW_TYPEAHEAD 被设置为零，以防在 X SYSTEM 模式下发送击键操作。

有关 **TE_type** 函数及其约定的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

## 设置键入样式

您可以为 RTE Vuser 设置两种键入样式：FAST 和 HUMAN。在 FAST 样式中，Vuser 会尽快将输入内容键入到终端仿真器中。在 HUMAN 样式中， Vuser 将在键入每个字符之后停顿一下。这样， Vuser 便可以更逼真地模拟在键盘上进行键入操作的人类用户。

可以使用 **TE_typing_style** 函数来设置键入样式。 **TE_typing_style** 函数的语法为：

int **TE_typing_style (**char ******style***);**

其中， *style* 可以为 FAST 或 HUMAN。默认键入样式为 HUMAN。如果您选择 HUMAN 键入样式，则格式为：

HUMAN, *delay* [,*first_delay*]

*delay* 表示击键之间的时间间隔 （毫秒）。可选参数 *first_delay* 表示在键入字符串中第一个字符之前等待的时间 （毫秒）。例如：

```
TE_typing_style ("HUMAN, 100, 500");
TE_type ("ABC");
```

意味着：Vuser 在键入字母 A 之前将等待 0.5 秒，然后在键入 B 之前将等待 0.1 秒，最后在键入 C 之前再等待 0.1 秒。

有关 **TE_typing_style** 函数及其约定的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

除了使用 **TE_typing_style** 函数设置键入样式以外，还可以使用运行时设置。详细信息，请参阅第 661 页的 “配置 RTE 运行时设置”。

## 生成唯一设备名

某些协议 （如 APPC）要求登录到系统的每个终端都具有唯一的设备名。使用运行时设置，可以指定 **TE_connect** 函数为每个 Vuser 生成唯一的 8 字符设备名，并使用该名称进行连接。尽管这解决了唯一性要求，但某些系统还有其他要求：设备名必须遵循特定格式。有关运行时设置的详细信息，请参阅第 9 章 “配置运行时设置”

要定义 **TE_connect** 函数将 Vuser 连接至系统时所使用的设备名格式，请向 Vuser 脚本中添加 **RteGenerateDeviceName** 函数。该函数具有以下原型：

void RteGenerateDeviceName(char buf[32])

应将该设备名写入 *buf* 中。

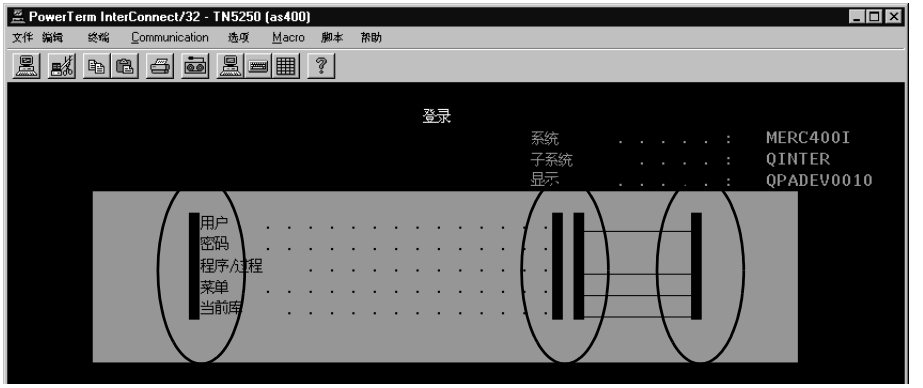
如果 **RteGenerateDeviceName** 函数存在于 Vuser 脚本中，则 Vuser 会在每次需要新设备名时调用该函数。如果未在脚本中定义 **RteGenerateDeviceName** 函数，并且需要唯一设备名，则 **TE_connect** 函数将生成所需的名称。

在以下示例中， **RteGenerateDeviceName** 函数将采用 “TERMx”格式生成唯一的设备名。第一个名称为 TERM0，后面依次是 TERM1、 TERM2 等。

```
RteGenerateDeviceName(char buf[32])
{
      static int n=0;
      sprintf(buf, "TERM%d", n);
      n=n+1;
}
```

## 设置字段分界字符

某些终端仿真器使用分界字符来标记每个字段的开始和结束。这些分界字符不可见 （在屏幕上显示为空格）。在下面显示的终端仿真器中，反转了屏幕中间区域的颜色，以便显示字段分界字符。这些字符包含在椭圆中。




**TE_wait_text**、 **TE_get_text** 和 **TE_find_text** 函数通过识别屏幕指定部分中的字符来执行操作。如果字段分界字符位于指定部分中，则可以选择将该字符识别为空格或 ASCII 字符。可以使用 TE_FIELD_CHARS 系统变量来指定识别方法。您可以将 TE_FIELD_CHARS 设置为 0 或 1：

- 0 指定字段分界字符位置的字符作为空格返回。
- 1 指定字段分界字符位置的字符作为 ASCII 码 （ASCII 0 或 ASCII 1）返回。

默认情况下， TE_FIELD_CHARS 被设置为 0。

您可以使用 **TE_getvar** 和 **TE_setvar** 函数来检索和设置 TE_FIELD_CHARS 的值。

# 54

# 配置 RTE 运行时设置

录制终端仿真器脚本之后，需要配置其运行时设置。本章描述下列终端仿真器 Vuser 运行时设置：

- 修改连接尝试
- 指定原始设备名
- 设置键入延迟
- 配置 X-System 同步

***下列信息仅适用于 “终端仿真器 (TE)” 类型 Vuser。***

## 关于终端仿真器运行时设置

开发终端仿真器 Vuser 脚本之后，需要设置运行时设置。通过这些设置可以控制运行脚本时 Vuser 的行为。通过终端仿真器运行时设置可以配置 TE Vuser，以便 Vuser 可以准确模拟执行远程终端仿真的实际用户。可以配置下列区域中的设置：

- 修改连接尝试
- 指定原始设备名
- 设置键入延迟
- 配置 X-System 同步

通过 “运行时设置”对话框中的 “RTE”节点可设置与终端仿真器相关的运行时设置。

要显示 “运行时设置”对话框，请单击 VuGen 工具栏上的 “运行时设置”按钮。还可以从 LoadRunner Controller 中修改运行时设置。在 “Controller”窗口中，单击 “运行时设置”按钮。

本章只讨论适用于终端仿真器 Vuser 的运行时设置。有关适用于所有 Vuser 的运行时设置的信息，请参阅第 9 章 “配置运行时设置”。

## 修改连接尝试

录制与主机的连接时， VuGen 将生成 **TE_connect** 函数。回放 RTE Vuser 脚本时， **TE_connect** 函数会将终端仿真器连接到指定的主机。如果第一个连接尝试不成功， Vuser 将重试若干次直到连接成功。每个连接的详细信息都将记录到报告文件 *output.txt* 中。

要设置 Vuser 尝试连接的最大次数，请在 RTE “运行时设置”中的 “最大连接尝试数”框中输入一个数字。

默认情况下， Vuser 将尝试连接 5 次。

有关 **TE_connect** 函数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

## 指定原始设备名

在某些环境中，每个会话 （Vuser）都要求唯一的设备名。 **TE_connect** 函数为每个 Vuser 生成唯一的 8 字符设备名，并且使用该设备名进行连接。要使用设备名 （包含在 **TE_connect** 函数的 **com_string** 参数中）进行连接，请选择 RTE “运行时设置”中的 “使用原始设备名”选项。

---

**注意**：该原始设备名设置仅适用于 IBM 块模式终端。

---

默认情况下， Vuser 使用原始设备名进行连接。

有关 **TE_connect** 函数的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

## 设置键入延迟

延迟设置确定 Vuser 执行 **TE_type** 函数的方式：

要指定 Vuser 在输入字符串的第一个字符之前需要等待的时间，请在 “主关键字”框中输入一个值，单位为毫秒。

要指定 Vuser 在提交连续字符之间需要等待的时间，请在 “后继关键字”框中输入一个值，单位为毫秒。

如果为主关键字和后继关键字延迟都输入零，则 Vuser 会将所有字符作为一个字符串发送，并且字符之间没有延迟。

可以使用 **TE_typing_style** 函数覆盖 Vuser 脚本的某个部分的延迟设置。

有关 **TE_type** 和 **TE_typing_style** 函数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

# 配置 X-System 同步

RTE Vuser 脚本使用 **TE_wait_sync** 函数进行同步。可以设置供 VuGen 应用于所有 **TE_wait_sync** 函数的超时值和稳定时间值。有关 **TE_wait_sync** 函数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

### 超时

回放 **TE_wait_sync** 函数时，如果同步超时到期之前系统不稳定，则 **TE_wait_sync** 函数将返回错误代码。要设置同步超时，请在 RTE “运行时设置”中的 “超时”部分输入值 （秒）。

默认超时值为 60 秒。

### 稳定时间

Vuser 执行 **TE_wait_sync** 函数之后，该 Vuser 将等待，直到终端不再处于 X-SYSTEM 模式下。当终端从 X-SYSTEM 模式下返回之后， Vuser 仍会监视系统一段时间。这样可以确保终端已处于稳定状态，即系统未返回到 X-SYSTEM 模式。然后 **TE_wait_sync** 函数才会终止。

要设置在系统从 X-SYSTEM 模式返回之后 Vuser 继续监视系统的时间，请在 RTE “运行时设置”的 “稳定时间”框中输入值 （毫秒）。

默认稳定时间为 1000 毫秒。

# 55

# 同步 RTE Vuser 脚本

RTE Vuser 脚本中的同步函数可以确保 Vuser 向终端仿真器提交的输入与服务器的响应同步。

本章描述：

- 同步块模式 (IBM) 终端
- 同步字符模式 (TV) 终端

***下列信息仅适用于 RTE (Windows) Vuser 脚本。***

## 关于同步 Vuser 脚本

根据要测试的系统，您可能需要将 Vuser 发送至终端仿真器的输入与服务器随后发出的提示同步。同步输入时，您可以指示 Vuser 挂起脚本执行并等待系统发出信号，然后再让 Vuser 执行下一步操作。例如，假设实际用户需要将以下击键序列提交给银行应用程序：

1. 键入 1 可从银行应用程序菜单中选择 “财务信息”。
2. 当显示消息 “您需要什么信息”时，键入 3 从菜单中选择 “道琼斯工业指数”。
3. 在屏幕中写入了完整的报告后，键入 5 退出该银行应用程序。

在本示例中，输入银行应用程序的信息将被同步，因为在每一步中，实际用户都要等到显示提示后才可键入。该提示可以是屏幕上显示的特定消息，也可以是屏幕上所有信息的稳定性。

您可以使用 TE 同步函数 **TE_wait_sync**、**TE_wait_text**、**TE_wait_silent** 和 **TE_wait_cursor** 以同样的方式同步 Vuser 的输入。这些函数可以有效地模拟在终端窗口键入内容、然后等到服务器发出响应后再键入下一个命令的实际用户。

**TE_wait_sync** 函数仅用于同步块模式 (IBM) 终端。其他 TE 同步函数用于同步字符模式 (VT) 终端。

录制 RTE Vuser 脚本时，VuGen 会自动生成 **TE_wait_sync**、**TE_wait_text** 和 **TE_wait_cursor** 语句，并将它们插入到脚本中。您可以使用 VuGen 的录制选项来指定 VuGen 应插入的同步函数。

---

**注意**：请不要在 Vuser 脚本的 *Vuser_end* 部分中包含任何同步语句。因为 Vuser 会随时中止，所以您无法预测 *Vuser_end* 部分何时执行。

---

## 同步块模式 (IBM) 终端

**TE_wait_sync** 函数用于同步操作块模式 (IBM) 终端的 RTE Vuser。块模式终端显示 “X SYSTEM” 消息，表示系统处于 “禁止输入” 模式。当系统处于 “禁止输入” 模式时，将无法执行任何键入操作，因为终端仿真器正在等待服务器传送数据。

当您在块模式终端上录制脚本时，默认情况下，每次显示 “X SYSTEM” 消息时， VuGen 都会生成 **TE_wait_sync** 函数并将其插入到该脚本中。您可以使用 VuGen 的录制选项指定 VuGen 是否应该自动插入 **TE_wait_sync** 函数。

当运行 Vuser 脚本时， **TE_wait_sync** 函数将检查系统是否处于 X SYSTEM 模式。如果系统处于 X SYSTEM 模式，则 **TE_wait_sync** 函数将挂起脚本执行。当 “X SYSTEM” 消息从屏幕上消失时，脚本将继续执行。

---

**注意**：只能对 IBM 块模式终端仿真器 （5250 和 3270）使用 **TE_wait_sync** 函数。

---

通常情况下， **TE_wait_sync** 函数可为所有块模式终端仿真器提供足够的同步。但是，如果 **TE_wait_sync** 函数在特定情况下无效，可以通过包含 **TE_wait_text** 函数来增强同步功能。有关 **TE_wait_text** 函数的详细信息，请参阅第 670 页的 “等待文本出现在屏幕上” 及 《Online Function Reference》（“帮助” > “函数参考”）。

**TE_wait_sync** 函数的语法为：

```
TE_wait_sync ();
```

在以下脚本片段中， Vuser 使用用户名 “QUSER”和密码 “MERCURY”进行登录。然后， Vuser 按下 Enter 键将登录详细信息提交给服务器。系统等待服务器进行响应的同时，终端仿真器会显示 “X SYSTEM”消息。 **TE_wait_sync** 语句将使 Vuser 一直等到服务器对登录请求做出响应 （即，等到 X SYSTEM 消息消失）后，才执行脚本的下一行。

```
TE_type("QUSER");
lr_think_time(2);
TE_type("<kTab>MERCURY");
lr_think_time(3);
TE_type("<kEnter>");
TE_wait_sync();
....
```

如果 **TE_wait_sync** 函数在显示 X SYSTEM 消息的同时挂起脚本执行，则 Vuser 将继续监控系统，并等待 X SYSTEM 消息消失。如果在同步超时到期之前， X SYSTEM 消息没有消失， **TE_wait_sync** 函数将返回错误代码。默认超时为 30 秒。

**要设置 TE_wait_sync 同步超时，请执行下列操作：**

1. 选择 “Vuser” > “运行时设置”。将显示 “运行时设置”对话框。
2. 在 “运行时设置”树中选择 “RTE:RTE”节点。
3. 在 “X SYSTEM 同步”下的 “超时”框中输入值 （秒）。
4. 单击 “确定”关闭 “运行时设置”对话框。

Vuser 执行 **TE_wait_sync** 函数后， Vuser 将一直等待，直到终端不再处于 X SYSTEM 模式。当终端从 X SYSTEM 模式返回时， Vuser 将继续监控系统一小段时间，以确保终端完全稳定，即系统不会再返回到 X SYSTEM 模式。只有这时， **TE_wait_sync** 函数才会终止，并允许 Vuser 继续执行其脚本。系统从 X SYSTEM 模式返回之后， Vuser 继续监控系统的这段时间叫做稳定时间。默认稳定时间为 1000 毫秒。

如果系统出现以下情况，则可能需要增加稳定时间：

当系统从 X SYSTEM 模式返回时，在短时间内会反复进入和脱离 X SYSTEM 模式，直到系统稳定为止。如果系统脱离 X SYSTEM 模式的时间超过一秒钟后再返回到 X SYSTEM 模式，**TE_wait_sync** 函数将认为该系统是稳定的。然后，如果 Vuser 尝试在系统中键入信息，则系统将切换至键盘锁定模式。

另外，如果系统从 X SYSTEM 模式返回时从未发生反复，则您可以缩短稳定时间使其小于默认值一秒钟。

**要更改 TE_wait_sync 函数的稳定时间，请执行下列操作：**

1. 选择 “Vuser” > “运行时设置”。将显示 “运行时设置”对话框。
2. 选择 “RTE:RTE”节点。
3. 在 “X SYSTEM 同步”下的 “稳定时间”框中输入值 （毫秒）。
4. 单击 “确定”关闭 “运行时设置”对话框。

有关 **TE_wait_sync** 函数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

您可以指示 VuGen 记录每次进入 X SYSTEM 模式时系统在该模式中停留的时间。为此， VuGen 会在每个 **TE_wait_sync** 函数后插入 **TE_wait_sync_transaction** 函数，如以下脚本片段所示：

```
TE_wait_sync();
TE_wait_sync_transaction("default");
```

每个 **TE_wait_sync_transaction** 函数都将创建一个名为 “default”的事务。这样，便可以分析终端仿真器在场景运行期间等待服务器做出响应的时间。您可以使用录制选项指定 VuGen 是否应该生成并插入 **TE_wait_sync_transaction** 语句。

**要指示 VuGen 插入 TE_wait_sync_transaction 语句，请执行下列操作：**

1. 选择 “Vuser” > “录制选项”。将显示 “录制设置”对话框。
2. 选择 “生成自动 X SYSTEM 事务”选项，然后单击 “确定”。

# 同步字符模式 (TV) 终端

有三种同步操作可用于字符模式 (VT) 终端。您选择的同步类型取决于：

- 在终端仿真器中运行的应用程序的设计
- 要同步的特定操作

## 等待光标出现在特定位置

VT 类型终端的首选同步方法是光标同步。光标同步对全屏或窗体类型的应用程序而言特别有用，对于滚动式或 TTY 类型的应用程序则正好相反。

光标同步使用 **TE_wait_cursor** 函数。当运行 RTE Vuser 脚本时，**TE_wait_cursor** 函数将指示 Vuser 挂起脚本执行，直到光标出现在屏幕上的指定位置。光标出现在指定位置表示该应用程序准备好接收来自终端仿真器的下一个输入。

**TE_wait_cursor** 函数的语法为：

int **TE_wait_cursor (**int *col*, int *row*, int *stable*, int *timeout***);**

在脚本执行期间， **TE_wait_cursor** 函数将等待光标到达由 *col* 和 *row* 指定的位置。

*stable* 参数指定光标必须在指定位置停留的时间 （毫秒）。如果您使用 VuGen 录制脚本，默认情况下， *stable* 将被设置为 100 毫秒。如果客户端应用程序在 *timeout* 参数指定的时间内未达到稳定状态，则该函数将返回 TIMEOUT。如果您使用 VuGen 录制脚本，则默认情况下， *timeout* 将被设置为 TIMEOUT，该值为 90 秒。您可以直接编辑所录制的脚本更改 *stable* 和 *timeout* 参数的值。

以下语句将等待光标保持稳定状态三秒钟。如果光标未能在 10 秒内达到稳定状态，则该函数将返回 TIMEOUT。

```
TE_wait_cursor (10, 24, 3000, 10);
```

有关 **TE_wait_cursor** 函数的详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

您可以指示 VuGen 在录制脚本时自动生成 **TE_wait_cursor** 语句并将它们插入到脚本中。以下是由 VuGen 自动生成的 **TE_wait_cursor** 语句的示例：

```
TE_wait_cursor(7, 20, 100, 90);
```

**要指示 VuGen 在录制的同时自动生成 TE_wait_cursor 语句并将它们插入到脚本中，请执行下列操作：**

1. 选择 “Vuser” > “录制选项”。将显示 “录制设置”对话框。
2. 在 “生成自动同步命令”下，选中 “光标”复选框，然后单击 “确定”。

### 等待文本出现在屏幕上

您可以使用文本同步来同步在 VT 终端仿真器上运行的 RTE Vuser。文本同步使用 **TE_wait_text** 函数。在脚本执行期间，**TE_wait_text** 函数将挂起脚本执行，然后等到特定字符串出现在终端窗口中以后，继续执行脚本。对于那些光标不能始终出现在预定义的屏幕区域中的应用程序而言，文本同步非常有用。

---

**注意**：尽管文本同步是为字符模式 (VT) 终端设计的，但它也可用于 IBM 块模式终端。但是，请不要对块模式终端使用自动文本同步。

---

**TE_wait_text** 函数的语法为：

int **TE_wait_text (**char *_pattern_, int _timeout_, int _col1_, int _row1_, int _col2_, int _row2_, int *_retcol_, int *_retrow_, char *_match_**);**

该函数会等待与 *pattern* 匹配的文本出现在由 *col1*、*row1*、*col2*、*row2* 定义的矩形中。与 pattern 匹配的文本将被返回给 *match*，实际的行和列位置将被返回给 *retcol* 和 *retrow*。如果在 *timeout* 到期之前 *pattern* 没有出现，该函数将返回错误代码。 *pattern* 可以包括常规表达式。有关使用常规表达式的详细信息，请参阅《Online Function Reference》。除了 *pattern* 和 *timeout* 参数以外，所有其他参数都是可选的。

当 *pattern* 作为空字符串传递时，如果该函数在矩形内发现文本，则它将等待超时。如果矩形内没有文本，它将立即返回。

如果 *pattern* 的确出现，则该函数将等待，直到仿真器在由 TE_SILENT_SEC 和 TE_SILENT_MILLI 系统变量定义的时间间隔内保持稳定状态 （完成重绘且不显示任何新字符）。实际上，这可以使终端达到稳定状态并模拟实际用户。

如果终端在由 TE_SILENT_TIMEOUT 定义的时间间隔内未达到稳定状态，则将继续执行脚本。该函数会返回 0 以表示成功，但是将设置 TE_errno 变量，以表明文本出现后终端未处于静止状态。

要修改或检索任何 TE_SILENT 系统变量的值，请使用 **TE_getvar** 和 **TE_setvar** 函数。详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

在以下示例中， Vuser 键入其名称，然后等待应用程序做出响应。

```
/* 声明 TE_wait_text 的变量 */
int ret_row;
int ret_col;
char ret_text [80];

/* 键入用户名。 */
TE_type ("John");

/* 等待通知程序做出响应。 */
TE_wait_text ("Enter secret code:", 30, 29, 13, 1, 13, &ret_col, &ret_row,
                    ret_text);
```

您可以指示 VuGen 在您录制脚本时自动生成 **TE_wait_text** 语句并将它们插入到脚本中。

**要指示 VuGen 在录制的同时自动生成 TE_wait_text 语句并将它们插入到脚本中，请执行下列操作：**

1. 选择 “Vuser” > “录制选项”。将显示 “录制设置”对话框。
2. 在 “生成自动同步命令”下，选中 “提示”复选框，然后单击 “确定”。

以下是由 VuGen 自动生成的 **TE_wait_text** 语句的示例：该函数最多等待 20 秒，以确定字符串 “keys”是否显示在屏幕上的任意位置。注意， VuGen 在生成 **TE_wait_text** 函数时将忽略所有可选参数。

```
TE_wait_text("keys", 20);
```

### 等待终端处于静止状态

当光标同步和文本同步都不起作用时，您可以使用 “静止状态同步”来同步脚本。使用 “静止状态同步”时， Vuser 将等待终端仿真器在指定的时间段后达到静止状态。如果仿真器在指定的时间段内没有从服务器收到任何输入，则认为它处于静止状态。

---

**注意**：仅当光标同步和文本同步都不起作用时，才使用静态同步。

---

您可以使用 **TE_wait_silent** 函数来指示脚本等待终端达到静止状态。您可以指定终端必须保持静止状态的期限。如果终端在指定期限内保持静止状态，则 **TE_wait_silent** 函数将认为应用程序已停止在终端屏幕上输出文本，并且该屏幕已达到稳定状态。

该函数的语法为：

int **TE_wait_silent (**int *sec*, int *milli*, int *timeout***);**

**TE_wait_silent** 函数将等待终端仿真器在 *sec* （秒）和 *milli* （毫秒）指定的时间达到静止状态。如果仿真器没有从服务器收到任何输入，则认为它达到静止状态。如果仿真器在由 *timeout* 时间变量指定的时间内未得到静止状态 （例如，停止接收字符），则该函数将返回错误。

例如，以下语句将等待屏幕在三秒后达到稳定状态。如果十秒后屏幕仍未达到稳定状态，该函数将返回错误。

```
TE_wait_silent (3, 0, 10);
```

详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）

# 56

# 读取终端屏幕中的文本

RTE Vuser 可以读取终端仿真器用户界面中的文本，然后使用该文本执行各种任务。

本章描述：

- 搜索屏幕中的文本
- 读取屏幕中的文本

***下列信息仅应用于 RTE (Windows) Vuser 脚本。***

## 关于读取终端屏幕中的文本

RTE Vuser 可以使用几种 Vuser 函数来读取终端屏幕中的文本。可以使用函数 **TE_find_text** 和 **TE_get_text_line** 来检查终端仿真器是否能够正确响应，或者增强脚本中的逻辑。

录制后，您可以手动将 **TE_find_text** 和 **TE_get_text_line** 语句直接插入到 RTE Vuser 脚本中。

## 搜索屏幕中的文本

**TE_find_text** 函数可以搜索屏幕中的文本行。该函数的语法为：

int **TE_find_text (**char **pattern*, int *col1*, int *row1*, int *col2*, int *row2*,
int **retcol*, int **retrow*, char **match*);

此函数可以搜索与由 *col1*、*row1*、*col2*、*row2* 所定义的矩形中的模式相匹配的文本。与该模式匹配的文本将被返回到 *match*，实际的行位置和列位置则被返回到

*retcol* 和 *retrow*。搜索操作从左上角开始。如果有多个字符串与所定义的模式相匹配，则返回距离左上角最近的那个字符串。

模式可以包含常规表达式。有关使用常规表达式的详细信息，请参阅 《Online Function Reference》。

您必须手动将 **TE_find_text** 语句键入到 Vuser 脚本中。有关 **TE_find_text** 函数的语法的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

## 读取屏幕中的文本

**TE_get_text_line** 函数可以读取指定的屏幕区域中的文本行。该函数的语法为：

char ***TE_get_text_line (***int *col*, int *row*, int *width*, char **text***);*

该函数可以将终端屏幕中的文本行复制到缓冲区文本。该行中的第一个字符由 *col*、 *row* 定义。该行中最后一个字符的列坐标由 *width* 表示。屏幕中的文本将被返回到缓冲区文本。如果该行包含制表符或空格，将返回相等数量的空格。

此外， **TE_get_cursor_position** 函数可以用于检索终端屏幕上的光标的当前位置。 **TE_get_line_attribute** 函数可返回文本行的字符格式 （例如粗体或下划线）。

您必须手动将 **TE_get_text_line** 语句键入到 Vuser 脚本中。有关 **TE_get_text_line** 函数的语法的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

# 第 XII 部分

## 邮件服务协议

# 57

# 开发用于邮件服务的 Vuser 脚本

通过 VuGen，可以在协议级别上测试若干个邮件服务。它可以模拟发送邮件的操作，以及针对邮件服务器执行的大多数标准操作。

本章描述：

- 邮件服务 Vuser 脚本入门
- 使用 IMAP 函数
- 使用 MAPI 函数
- 使用 POP3 函数
- 使用 SMTP 函数

***下列信息仅适用于 IMAP、 MAPI、 POP3 和 SMTP 虚拟用户脚本。***

## 关于开发用于邮件服务的 Vuser 脚本

邮件服务协议模拟用户使用电子邮件客户端、查看和发送电子邮件的操作。支持下列邮件服务：

- Internet 消息传输 (IMAP)
- MS Exchange (MAPI)
- 邮局协议 (POP3)
- 简单邮件传输协议 (SMTP)

邮件协议既支持录制也支持回放，但 MAPI 除外，它只支持回放。

当使用邮件协议之一录制应用程序时， VuGen 将生成模拟邮件客户端操作的函数。您可以指定创建 Vuser 脚本所采用的编程语言 – C 或 Visual Basic 脚本。详细信息，请参阅第 4 章 “设置脚本生成首选项”。如果通过多个协议执行通信，则可以同时对这些协议进行录制。可以录制多个邮件协议，也可以将一个邮件协议同 HTTP 或 WinSock 一起录制。有关指定多个协议的说明，请参阅第 3 章 “使用 VuGen 进行录制”。

所有邮件服务函数都是成对出现的，即一个函数用于全局会话，另一个函数可指示特定的邮件会话。例如，使用 **imap_logon** 全局登录到 IMAP 服务器，而使用 **imap_logon_ex** 则针对特定会话登录到 IMAP 服务器。

## 邮件服务 Vuser 脚本入门

本节概述了使用 VuGen 开发用于邮件服务的 Vuser 脚本的过程。

**要开发邮件服务 Vuser 脚本，请执行下列操作：**

**1 使用 VuGen 创建一个基本脚本。**

调用 VuGen，并为一个邮件协议或多个协议创建新的 Vuser 脚本。

**2 使用 VuGen 录制一个基本脚本。（MAPI 除外）**

选择要录制的应用程序。在应用程序中执行常见的操作。详细信息，请参阅第 3 章 “使用 VuGen 进行录制”。

对于 MAPI，不支持录制。而是创建空的 MAPI 脚本，并在其中手动插入 **mapi** 函数。例如，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

**3 增强该脚本。**

通过在脚本中插入事务、集合点和控制流结构来增强该脚本。

详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

**4 定义参数 （可选）。**

为录制在脚本中的固定值定义参数。通过将这些固定值替换为参数，可以使用不同的值多次重复相同的业务流程。

详细信息，请参阅第 7 章 “定义参数”。

**5 关联语句 （可选）。**

通过关联语句，可以在后面的业务流程中使用前一个流程的结果。

详细信息，请参阅第 8 章 “关联语句”。

**6 配置运行时设置。**

运行时设置控制脚本执行过程中 Vuser 的行为。这些设置包括循环、日志和计时信息。

详细信息，请参阅第 9 章 “配置运行时设置”。

**7 从 VuGen 运行该脚本。**

在 VuGen 中保存并运行该脚本，以验证其是否能正确运行。

详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

创建虚拟用户脚本后，可以将该脚本集成到 Windows 或 UNIX 平台上的场景中。有关在场景中集成虚拟用户脚本的详细信息，请参阅 《LoadRunner Controller 用户指南》。

## 使用 IMAP 函数

IMAP Vuser 脚本函数录制 Internet 邮件应用协议。每个 IMAP 函数都以 **imap** 前缀开头。有关这些函数的详细语法信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

| 函数名 | 描述 |
|---|---|
| **imap_append[_ex]** | 向邮箱末尾追加消息 |
| **imap_check[_ex]** | 请求当前邮箱的检查点 |
| **imap_close[_ex]** | 关闭当前邮箱 |
| **imap_copy[_ex]** | 将邮件复制到另一个邮箱 |
| **imap_create[_ex]** | 创建邮箱 |
| **imap_custom_request[_ex]** | 执行自定义 IMAP 请求 |
| **imap_delete[_ex]** | 删除指定的邮箱 |
| **imap_examine[_ex]** | 检查邮箱 |
| **imap_expunge[_ex]** | 删除标记为删除的所有邮件 |
| **imap_fetch[_ex]** | 检索与邮箱邮件相关的数据 |
| **imap_free_ex** | 释放 IMAP 会话描述符 |

| | |
|---|---|
| **imap_get_attribute_int[_ex]** | 返回邮箱属性 |
| **imap_get_attribute_sz[_ex]** | 以字符串的形式返回邮箱属性 |
| **imap_get_result[_ex]** | 获取 IMAP 服务器返回代码 |
| **imap_list_mailboxes[_ex]** | 列出可用邮箱 |
| **imap_list_subscriptions[_ex]** | 列出订阅的或活动的邮箱 |
| **imap_logon[_ex]** | 登录到 IMAP 服务器 |
| **imap_logout[_ex]** | 从 IMAP 服务器注销 |
| **imap_noop[_ex]** | 执行空操作 |
| **imap_search[_ex]** | 按关键字搜索邮箱 |
| **imap_select[_ex]** | 选择邮箱 |
| **imap_status[_ex]** | 请求邮箱的状态 |
| **imap_store[_ex]** | 更改与邮箱邮件相关的数据 |
| **imap_subscribe[_ex]** | 订阅或激活邮箱 |
| **imap_unsubscribe[_ex]** | 取消订阅或停用邮箱 |

在以下示例中， **imap_create** 函数将创建几个新邮箱：*Products*、 *Solutions* 和 *FAQs*。

```
Actions()
{
    imap_logon("ImapLogon",

"URL=imap://johnd:letmein@exchange.mycompany.com",
                LAST);

    imap_create("CreateMailboxes",
                "Mailbox=Products",
                "Mailbox=Solutions",
                "Mailbox=FAQs",
                LAST);

    imap_logout();

    return 1;
}
```

## 使用 MAPI 函数

MAPI Vuser 脚本函数录制与 MS Exchange 服务器之间进行的活动。每个 MAPI 函数都以 **mapi** 前缀开头。有关这些函数的详细语法信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

| 函数名 | 描述 |
|---|---|
| **mapi_delete_mail[_ex]** | 删除当前或选定的电子邮件项 |
| **mapi_get_property_sz[_ex]** | 从 MAPI 会话获取属性值 |
| **mapi_logon[_ex]** | 登录到 MS Exchange |
| **mapi_logout[_ex]** | 从 MS Exchange 注销 |
| **mapi_read_next_mail[_ex]** | 阅读邮箱中的下一封邮件 |
| **mapi_send_mail[_ex]** | 将电子邮件发送到收件人 |
| **mapi_set_property_sz[_ex]** | 设置 MAPI 属性 |

在以下示例中， **mapi_send_mail** 函数将通过 MS Exchange 服务器发送便笺。

```
Actions()
{
    mapi_logon("Logon",
                "ProfileName=John Smith",
                "ProfilePass=Tiger",
                LAST);

    // 发送便签消息
    mapi_send_mail("SendMail",
                "To=user1@techno.merc-int.com",
                "Cc=user0002t@techno.merc-int.com",
                "Subject=<GROUP>:<VUID> @ <DATE>",
                "Type=Ipm.StickyNote",
                "Body=Please update your profile today.",
                LAST);

    mapi_logout();

    return 1;
}
```

## 使用 POP3 函数

POP3 Vuser 脚本函数模拟使用邮局协议 POP3 执行的操作。每个函数都以 **POP3** 前缀开头。有关这些函数的详细语法信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

| 函数名 | 描述 |
|---|---|
| **pop3_command[_ex]** | 向 POP3 服务器发送命令 |
| **pop3_delete[_ex]** | 删除服务器上的邮件 |
| **pop3_free[_ex]** | 释放 POP3 服务器，使其不处理命令 |
| **pop3_list[_ex]** | 列出 POP3 服务器上的邮件 |
| **pop3_logoff[_ex]** | 从 POP3 服务器注销 |
| **pop3_logon[_ex]** | 登录到 POP3 服务器 |
| **pop3_retrieve[_ex]** | 检索 POP3 服务器上的邮件 |

在以下示例中， **pop3_retrieve** 函数将从 POP3 服务器上检索五封邮件。

```
Actions()
{
pop3_logon("Login", "
        URL=pop3://user0004t:my_pwd@techno.merc-int.com",
        LAST);

// 列出服务器上的所有邮件并接收该值
totalMessages = pop3_list("POP3", LAST);

// 显示接收到的值 （pop3_list 函数也会显示该值）
lr_log_message("There are %d messages.\r\n\r\n", totalMessages);

// 检索服务器上的五封邮件，而不删除它们
pop3_retrieve("POP3", "RetrieveList=1:5", "DeleteMail=false", LAST);
pop3_logoff();
    return 1;
}
```

## 使用 SMTP 函数

SMTP Vuser 脚本函数模拟简单邮件传输协议 (SMTP) 流量。每个 SMTP 函数都以 **smtp** 前缀开头。有关这些函数的详细语法信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

| 函数名 | 描述 |
|---|---|
| **smtp_abort_mail[_ex]** | 中止 SMTP 邮件的传输 |
| **smtp_free[_ex]** | 释放 SMTP 服务器，使其不处理命令 |
| **smtp_logon[_ex]** | 登录到 SMTP 服务器 |
| **smtp_logout[_ex]** | 从 SMTP 服务器注销 |
| **smtp_send_mail[_ex]** | 发送 SMTP 消息 |
| **smtp_translate[_ex]** | 转换 SMTP 消息 |

在以下示例中， **smtp_send_mail** 函数将通过 SMTP 邮件服务器 *techno* 发送一封邮件。

```
Actions()
{
      smtp_logon("Logon",
            "URL=smtp://user0001t@techno.merc-int.com",
                  "CommonName=Smtp Test User 0001",
            NULL);

      smtp_send_mail("SendMail",
                        "To=user0002t@merc-int.com",
                        "Subject=MIC Smtp:Sample Test",
                        "MAILOPTIONS",
                        "X-Priority: 3",
                        "X-MSMail-Priority:Medium",
                        "X-Mailer:Microsoft Outlook Express 5.50.400\r\n",
                        "X-MimeOLE:By Microsoft MimeOLE V5.50.00\r\n",
                        "MAILDATA",
                        "MessageText="
                            "Content-Type:text/plain;\r\n"
                            "\tcharset=\"iso-8859-1\"\r\n"
                            "Test,\r\n"
                            "MessageBlob=16384",
                     NULL);

      smtp_logout();

      return 1;
}
```

# 第 XIII 部分

## 中间件协议

# 58

# 开发 Jacada Vuser 脚本

通过 VuGen 可以录制与 Jacada Interface Server 的通信。可以运行录制的脚本，也可以使用标准的 Java 库函数以及 LoadRunner 特有的 Java 函数增强该脚本。

本章描述：

- Jacada Vuser 入门
- 录制 Jacada Vuser
- 回放 Jacada Vuser
- 了解 Jacada Vuser 脚本
- 使用 Jacada Vuser 脚本

***下列信息仅适用于 Jacada Vuser 脚本。***

## 关于 Jacada Vuser 脚本

Jacada Interface Server 为大型机应用程序提供了一个接口层。此层将用户接口从应用程序逻辑中分离出来，以便使组织免受标准和技术变化的影响。 Jacada 服务器将环境转换为友好的用户界面，而不是使用绿屏应用程序。

VuGen 可以录制 Jacada 的 Java 瘦客户端。要通过 HTML 瘦客户端录制与 Jacada 服务器的通信，请使用 Web HTTP/HTML 类型的 Vuser。详细信息，请参阅第 32 章 “创建 Web Vuser 脚本”。

要创建脚本，请调用 VuGen 并录制典型的操作和业务流程。 VuGen 将生成一个代表您所有操作的脚本。此脚本兼容 java。

准备好脚本后，可以在 VuGen 中以独立模式运行它。 Sun 的标准 Java 编译器 *javac.exe* 将检查脚本中是否有错误，并对其进行编译。确认脚本可正常运行后，可将该脚本合并到 LoadRunner 场景中。

通过录制和手动增强创建脚本时，与 Java Vuser 脚本相关的所有规则和限制都适用。有关函数语法和系统配置的重要信息，请参阅第 24 章 “编写 Java 脚本”。

下面几部分讨论录制选项、运行时设置以及关联功能。

## Jacada Vuser 入门

下列过程概述如何创建 Jacada Vuser 脚本。

**1 确保录制计算机配置正确。**

开始录制之前，确保计算机针对 Java 配置正确。详细信息，请参阅第 24 章 “编写 Java 脚本”和自述文件。

**2 新建 Jacada Vuser 脚本。**

从中间件组选择 *Jacada* 类型的 Vuser。

**3 为 Vuser 脚本设置录制参数和选项。**

为小程序或应用程序指定参数，例如工作目录和路径。还可以设置 JVM、关联、录制器和调试录制选项。详细信息，请参阅第 14 章 “设置 Java 录制选项”。

**4 录制典型的用户操作。**

开始录制脚本。对 Jacada 服务器执行典型操作。 VuGen 将录制所执行的操作并生成 Vuser 脚本。

**5 增强 Vuser 脚本。**

向 Vuser 脚本添加 LoadRunner 特定的函数。详细信息，请参阅第 24 章“编写 Java 脚本”使用 VuGen 的 Navigator 函数添加类或方法。（请参阅第 46 章 “执行 EJB 测试”。）

**6 参数化 Vuser 脚本。**

将录制的常量替换为参数。可以参数化整个字符串，也可以参数化字符串的一部分。详细信息，请参阅第 7 章 “定义参数”。

**7 为脚本配置运行时设置。**

配置 Vuser 脚本的运行时设置。运行时设置用于定义在执行脚本时的运行时特征。有关 Java 的特定运行时设置，请参阅第 16 章 “配置 Java 运行时设置”。

**8 保存并运行 Vuser 脚本。**

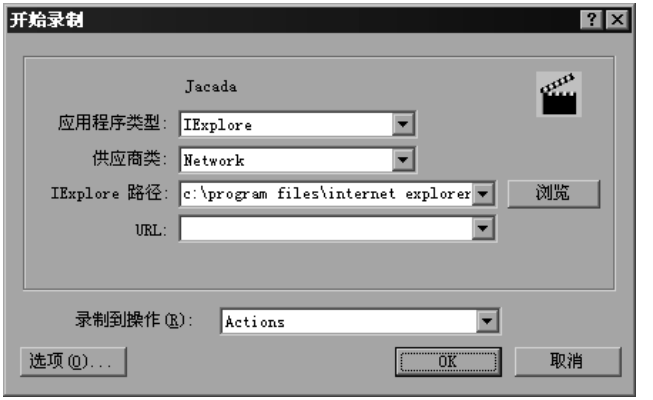
从 VuGen 运行脚本并查看 “执行日志”选项卡中的消息。详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

# 录制 Jacada Vuser

可以录制 Jacada 脚本创建完全兼容的 Java 程序。

**要录制 Jacada 脚本，请执行下列操作：**

**1** 要开始录制，请选择 “文件” > “新建”，然后从中间件 Vuser 类型中选择 “Jacada”。将打开 “开始录制”对话框。




**2** 选择一种应用程序类型：Internet Explorer 或 Netscape。

**3** 在 “供应商类”框中，默认值为 “网络”类。如果 *clbase.jar* 在您的类路径中，请选择 “本地”供应商类。

**4** 指定浏览器路径和 Jacada 服务器开始页的 URL。

注意，工作目录仅对于访问工作目录的应用程序而言才是必需的 （例如，读属性文件或写日志文件）。

**5** 要设置录制选项 （例如 JVM 的命令行参数），请单击 “选项”。有关设置录制选项的信息，请参阅 第 14 章 “设置 Java 录制选项”。

**6** 在“录制到操作”框中，选择要录制到的方法的相应部分。 Actions 类包含三个方法：*init*、 *action* 和 *end*，分别对应于 vuser_init、 Actions 和 vuser_end 部分。下表显示了要包括在每个方法中的内容以及执行每个方法的时间。

| Actions 类中的方法 | 录制到操作 | 模拟的情形 ... | 执行期间 ... |
|---|---|---|---|
| *init* | *vuser_init* | 登录到服务器 | 初始化 |
| *action* | *Actions* | 客户端活动 | 正在运行 |
| *end* | *vuser_end* | 注销过程 | 已完成或已停止 |

**7** 单击“确定”开始录制。 VuGen 将启动应用程序，并将自身最小化，同时打开进度栏和浮动录制工具栏。该进度工具栏显示类在加载时所用的名称。这指明已激活了 Java 录制支持。

**8** 在应用程序内执行典型操作。使用浮动工具栏在录制过程中切换方法。

**9** 录制典型的用户操作后，从浮动工具栏中选择 *vuser_end* 方法。

执行注销过程。 VuGen 会将该过程录制到脚本的 *vuser_end* 方法中。

**10** 单击“录制”工具栏上的“停止录制”。 VuGen 编辑器将显示所有录制的语句。

**11** 单击“保存”保存该脚本。将打开“保存测试”对话框（仅限于新的 Vuser 脚本）。指定脚本名称。

## 回放 Jacada Vuser

请确保已在运行 LoadRunner 的计算机上正确安装了 Sun 的一个 JDK 版本 – 只有 JRE 是不够的。并验证 *classpath* 和 *path* 环境变量是否是根据 JDK 安装说明设置的。回放 Vuser 脚本之前，请验证是否已针对 JDK 和相关 Java 类恰当地配置了您的环境。

回放之前，还必须从 Jacada 服务器下载 *clbase.jar* 文件。 Java Vuser 使用的所有类必须位于类路径中 （在计算机的 CLASSPATH 环境变量中设置，或在 “运行时设置”的 “类路径”节点的 “类路径项”列表中设置）。

Jacada 服务器从旧系统返回屏幕的顺序可能与显示在已录制脚本中的不同。这可能会导致回放中发生异常。有关如何处理这些异常的信息，请与 Mercury Interactive 支持部门联系。

有关所需环境设置的详细信息，请参阅第 24 章 “编写 Java 脚本”。

## 了解 Jacada Vuser 脚本

录制会话时， VuGen 将记录对服务器的所有调用，并使用 LoadRunner 增强功能生成脚本。这些函数描述了应用程序或小程序中的所有操作。该脚本还包含为正确播放而进行的异常处理。

录制的脚本由下列三部分组成：

- **Imports**

  **Imports** 部分位于脚本的开头。它包含对编译该脚本所需的所有数据包的引用。

- **Code**

  **Code** 部分包含 Actions 类以及在 *init*、*actions* 和 *end* 方法中录制的代码。code 部分还包含发送到服务器的每个命令的异常处理程序 “try-catch”块。

- **Variables**

  **Variables** 部分位于 *end* 方法之后，包含代码中使用的变量的所有类型声明。

录制完成以后，可以修改脚本中的函数，或添加其他 Java 或 LoadRunner 函数增强脚本。注意，如果希望将 Java Vuser 作为线程运行，则添加到脚本中的 Java 代码必须是线程安全的。有关函数语法的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

# 使用 Jacada Vuser 脚本

Jacada 脚本的 Actions 方法包括两个主要部分：“属性”和 “主体”。属性部分用于获取服务器属性。然后， VuGen 将设置系统属性并连接 Jacada 服务器。

```
// 设置系统属性 ...
      _properties = new Properties(System.getProperties());
      _properties.put("com.ms.applet.enable.logging", "true");
      System.setProperties(_properties);

      _jacadavirtualuser = new cst.client.manager.JacadaVirtualUser();

      lr.think_time(4);
      _jacadavirtualuser.connectUsingPorts("localhost", 1100,
"LOADTEST", "", "", "");
      ...
```

脚本的主体包含用户操作以及 **checkFieldValue** 和 **checkTableCell** 方法的异常处理块。

```
      l...
/*
try {
      _jacadavirtualuser.checkFieldValue(23, "S44452BA");
      }catch(java.lang.Exception e) {
       lr.log_message(e.getMessage());
      }
*/    l...
/*
try {
     _jacadavirtualuser.checkTableCell(41, 0, 0, "");
      }catch(java.lang.Exception e) {
       lr.log_message(e.getMessage());
      }
*/    l...
```

**checkField** 方法包含两个参数：字段 ID 号和期望值。 **checkTableCell** 方法包含四个参数：表 ID、行、列和期望值。如果期望值和收到的值不匹配，则生成异常。

默认情况下，“try-catch”包装块将注释掉。要在脚本中使用它们，请删除注释标记。

除了录制的脚本，还可以添加任何 LoadRunner Java 函数。有关这些函数的列表以及如何将它们添加到脚本中的信息，请参阅第 24 章 “编写 Java 脚本”。

# 59

# 开发 Tuxedo Vuser 脚本

使用 VuGen 可以记录 Tuxedo 客户端应用程序与 Tuxedo 应用程序服务器之间的通信。得到的脚本称为 Tuxedo Vuser 脚本。

本章描述：

- Tuxedo Vuser 脚本入门
- 使用 LRT 函数
- 了解 Tuxedo Vuser 脚本
- 查看 Tuxedo 缓冲区数据
- 定义 Tuxedo Vuser 的环境设置
- 调试 Tuxedo 应用程序
- 关联 Tuxedo 脚本

***下列信息仅适用于 PeopleSoft-Tuxedo、 Tuxedo 6 和 Tuxedo 7 Vuser 脚本。***

## 关于 Tuxedo Vuser 脚本

录制 Tuxedo 应用程序时，VuGen 将生成描述所录制的操作的 LRT 函数。这些函数模拟 Tuxedo 客户端与服务器之间的通信。每个 LRT 函数都以 **lrt** 前缀开头。

除 **lrt** 前缀以外，某些特定函数还使用附加前缀 **tp**、 **tx** 或 **F**。这些子前缀指明函数类型，这与实际的 Tuxedo 函数类似。 **tp** 子前缀表示 Tuxedo 客户端 **tp** 会话。例如，**lrt_tpcall** 发送服务请求并等待其回复。**tx** 子前缀表示全局 tx 会话。例如， **lrt_tx_begin** 用于开始全局事务。 **F** 子前缀表示 FML 缓冲区相关函数。例如， **lrt_Finitialize** 用于初始化现有缓冲区。

不带附加前缀的函数用于模拟标准的 C 函数。例如， **lrt_strcpy** 用于复制字符串，与 C 函数 *strcpy* 类似。

您可以从 VuGen 的主窗口中查看并编辑所录制的脚本。会话期间录制的 LRT 函数将显示在 VuGen 窗口中，以便您直观地跟踪网络活动。

### 录制之前

录制之前，请验证 Tuxedo 目录 %TUXDIR%\bin 是否包含在路径中。

如果自上次重新启动 VuGen 后环境变量有所更改，VuGen 可能会录制原始变量值而不是当前值。

要避免出现任何不一致的情况，应在录制 Tuxedo 应用程序之前重新启动 VuGen。

## Tuxedo Vuser 脚本入门

本部分概述了使用 VuGen 开发 Tuxedo Vuser 脚本的过程。

**要开发 Tuxedo Vuser 脚本，请执行下列操作：**

**1 使用 VuGen 录制基本脚本。**

调用 VuGen 并新建 Vuser 脚本。指定 Tuxedo6（用于录制 Tuxedo 版本 6.x）或 Tuxedo7（用于录制 Tuxedo 版本 7.x）作为 Vuser 的类型。选择要录制的应用程序。录制应用程序的典型操作。

详细信息，请参阅第 3 章 “使用 VuGen 进行录制”。

**2 增强脚本。**

通过将事务、集合点和控制流结构插入到脚本中，可以增强脚本。

详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

**3 定义参数（可选）。**

为录制到脚本中的固定值定义参数。通过用参数替换固定值，可以使用不同的值多次重复执行相同的业务流程。

详细信息，请参阅第 7 章 “定义参数”。

**4 关联语句（可选）。**

通过关联语句，可以将一个业务流程的结果用于后续业务流程中。

详细信息，请参阅第 8 章 “关联语句”。

**5 配置运行时设置。**

运行时设置可控制脚本执行期间 Vuser 的行为。这些设置包括循环、日志和计时信息。

详细信息，请参阅第 9 章 “配置运行时设置”。

**6 从 VuGen 中运行脚本。**

在 VuGen 中保存并运行该脚本以验证该脚本是否可以正常运行。

详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

创建 Tuxedo Vuser 脚本后，请在 Windows 或 UNIX 平台上将其集成到场景中。有关将 Vuser 脚本集成到场景中的详细信息，请参阅 《LoadRunner Controller 用户指南》。

## 使用 LRT 函数

所开发的用于模拟 Tuxedo 客户端与服务器之间通信的函数称为 LRT 函数。每一个 LRT Vuser 函数都有 **lrt** 前缀。Tuxedo 会话期间，VuGen 将自动录制本部分中列出的大多数 LRT 函数。您还可以手动将任意函数编写到脚本中。有关 LRT 函数的语法和示例的信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

---

**注意：**一些 FML 缓冲区函数在函数名中显示可选的 “32”。这些是 FML32 版本的函数。

---

### 缓冲区操作函数

| | |
|---|---|
| **lrt_Fadd[32]_fld** | 向 FML 缓冲区中添加新字段 |
| **lrt_Finitialize[32]** | 初始化现有的 FML 缓冲区 fbfr |
| **lrt_Fldid[32]** | 将字段名映射到字段标识符 |
| **lrt_Fname[32]** | 为字段名提供映射字段标识符 |
| **lrt_memcpy** | 将源中指定数量的字节复制到目标 |
| **lrt_strcpy** | 复制字符串，与 C 函数 strcpy 类似 |
| **lrt_tpalloc** | 返回 *type* 类型缓冲区的指针 |
| **lrt_tprealloc** | 更改类型化缓冲区的大小 |
| **lrt_tpfree** | 释放类型化缓冲区 |
| **lrt_tptypes** | 确定类型化缓冲区的相关信息 |

### 客户端 / 服务器会话函数

| | |
|---|---|
| **lrt_tpchkauth** | 检查应用程序是否要求身份验证 |
| **lrt_tpinitialize** | 启用客户端以加入 System/T 应用程序 |
| **lrt_tpterm** | 从 System/T 应用程序中删除客户端 |

### 通信函数

| | |
|---|---|
| **lrt_tpacall** | 发送服务请求 |
| **lrt_tpbroadcast** | 按名称广播通知 |
| **lrt_tpcall** | 发送服务请求并等待其回复 |
| **lrt_tpcancel** | 取消调用描述符 |
| **lrt_tpchkunsol** | 检查未经请求的消息 |
| **lrt_tpconnect** | 建立会话式服务连接 |
| **lrt_tpdequeue** | 使队列中的消息出列 |
| **lrt_tpdiscon** | 终止会话式服务连接 |
| **lrt_tpenqueue** | 在队列中存储消息 |

| | |
|---|---|
| **lrt_tpgetrply** | 返回上次发送的请求的回复 |
| **lrt_tpgprio** | 返回上次发送或接收的请求的优先级 |
| **lrt_tpnotify** | 向客户端发送通知 |
| **lrt_tppost** | 发布事件 |
| **lrt_tprecv** | 在会话式连接中接收消息 |
| **lrt_tpsend** | 在会话式连接中发送消息 |
| **lrt_tpsetunsol** | 设置处理未经请求的消息的方法 |
| **lrt_tpsprio** | 设置下次发送或转发的请求的优先级 |
| **lrt_tpsubscribe** | 订阅事件 |
| **lrt_tpunsubscribe** | 取消订阅事件 |

### 环境变量函数

| | |
|---|---|
| **lrt_set_env_list** | 设置环境变量列表 |
| **lrt_tuxgetenv** | 返回与环境名相应的值 |
| **lrt_tuxputenv** | 修改现有环境值或向环境中添加值 |
| **lrt_tuxreadenv** | 从文件向环境添加变量 |

### 错误处理函数

| | |
|---|---|
| **lrt_abort_on_error** | 如果上一个 Tuxedo 函数调用产生了错误，则中止当前事务 |
| **lrt_Fstrerror[32]** | 检索 FML 错误的错误消息字符串 |
| **lrt_getFerror[32]** | 检索上一个失败的 FML 操作的错误状态代码 |
| **lrt_gettperrno** | 检索上一个 Tuxedo 事务监视器函数的错误状态代码 |
| **lrt_gettpurcode** | 检索应用程序返回代码 |
| **lrt_tpstrerror** | 检索 System/T 错误的错误消息字符串 |

### 事务处理函数

| | |
|---|---|
| **lrt_tpabort** | 中止当前事务 |
| **lrt_tpbegin** | 开始一个事务 |
| **lrt_tpcommit** | 提交当前事务 |
| **lrt_tpgetlev** | 检查事务是否正在进行 |
| **lrt_tpresume** | 继续执行全局事务 |
| **lrt_tpscmt** | 设置 lrt_tpcommit 应返回的时间 |
| **lrt_tpsuspend** | 挂起全局事务 |
| **lrt_tx_begin** | 开始全局事务 |
| **lrt_tx_close** | 关闭一组资源管理器 |
| **lrt_tx_commit** | 提交全局事务 |
| **lrt_tx_info** | 返回全局事务信息 |
| **lrt_tx_open** | 打开一组资源管理器 |
| **lrt_tx_rollback** | 回滚全局事务 |
| **lrt_tx_set_commit_return** | 将 commit_return 特征设置为 when_return 中指定的值 |
| **lrt_tx_set_transaction_control** | 将 transaction_control 特征设置为 control 中指定的值 |
| **lrt_tx_set_transaction_timeout** | 将 transaction_timeout 特征设置为 timeout 中指定的值 |

### 关联语句函数

| | |
|---|---|
| **lrt_display_buffer** | 在文件中存储缓冲区信息 |
| **lrt_save[32]_fld_val** | 将 FML 缓冲区的当前值保存到参数中 |
| **lrt_save_parm** | 将字符数组 （例如 STRING 或 CARRAY 缓冲区）的一部分保存到参数中 |
| **lrt_save_searched_string** | 在缓冲区中搜索字符串实例，并将缓冲区的一部分 （相对于字符串实例）保存到参数中 |

**注意**：通常，建议使用 **lrt_save_parm** 将字符数组的一部分保存到参数中。当您要相对于字符数组中特定字符串的位置保存信息时，请使用 **lrt_save_searched_string**。对于 PeopleSoft Vuser，建议使用 lrt_save_searched_string，因为从 PeopleSoft 服务器返回的回复缓冲区在回放期间的大小与在录制期间看到的大小通常会有所不同。

## 了解 Tuxedo Vuser 脚本

录制会话之后，可以在 VuGen 的内置编辑器中查看录制的代码。您可以在脚本中滚动，查看应用程序生成的 Tuxedo 语句，并检查服务器返回的数据。VuGen 窗口提供有关所录制的 Tuxedo 会话的重要信息。在主窗口中查看脚本时，可以看到 VuGen 录制活动的顺序。

以下示例中，VuGen 录制了客户端在 Tuxedo 银行应用程序中的操作。客户端执行了开设银行帐户的操作，并指定了所有必需的详细信息。当客户端指定期初余额为零时，中止该会话。

```
lrt_abort_on_error();
lr_think_time(65);
tpresult_int = lrt_tpbegin(30, 0);
data_0 = lrt_tpalloc("FML", "", 512);
lrt_Finitialize((FBFR*)data_0);

/* 用新帐户信息填充数据缓冲区 data_0*/
lrt_Fadd_fld((FBFR*)data_0, "name=BRANCH_ID", "value=8",
LRT_END_OF_PARMS);
lrt_Fadd_fld((FBFR*)data_0, "name=ACCT_TYPE", "value=C",
LRT_END_OF_PARMS);
lrt_Fadd_fld((FBFR*)data_0, "name=MID_INIT", "value=Q",
LRT_END_OF_PARMS);
lrt_Fadd_fld((FBFR*)data_0,"name=PHONE","value=123-456-7890",
            LRT_END_OF_PARMS);

lrt_Fadd_fld((FBFR*)data_0, "name=ADDRESS", "value=1 Broadway
            New York, NY 10000", LRT_END_OF_PARMS);
lrt_Fadd_fld((FBFR*)data_0,"name=SSN","value=111111111",
LRT_END_OF_PARMS);
```

```
lrt_Fadd_fld((FBFR*)data_0,"name=LAST_NAME",
"value=Doe",LRT_END_OF_PARMS);

lrt_Fadd_fld((FBFR*)data_0,"name=FIRST_NAME",
"value=BJ",LRT_END_OF_PARMS);

lrt_Fadd_fld((FBFR*)data_0, "name=SAMOUNT",
"value=0.00",LRT_END_OF_PARMS);

/* 开设新帐户 */
tpresult_int = lrt_tpcall("OPEN_ACCT", data_0, 0, &data_0, &olen_2, 0);
lrt_tpabort(0);
lrt_tpcommit(0);
lrt_tpfree(data_0);
lrt_tpterm();
```

### 在 Tuxedo 脚本中使用参数

可以在 Tuxedo 脚本中定义参数 （如第 7 章 “定义参数”中所述）。注意 Tuxedo 脚本包含类型为 “name=...”或 “value=...”的字符串。可以只为等号 (=) 后面的字符串部分定义参数。例如：

```
lrt_Fadd_fld((FBFR*)data_0,"name=PHONE","value=<parameter_1>",
            LRT_END_OF_PARMS);
```

### 运行 Tuxedo 脚本

如果在录制或运行 Tuxedo 应用程序时遇到了错误，请检查在没有 VuGen 的情况下是否可以运行 Tuxedo 应用程序，以及是否已正确定义了环境变量。详细信息，请参阅第 703 页的 “查看 Tuxedo 缓冲区数据”。注意，设置或修改 Tuxedo 变量之后，应重新启动 VuGen 和应用程序，使更改生效。如果应用程序为 16 位程序，则还须终止 NTVDM 进程。

如果在执行期间遇到了问题，请检查服务器端的 Tuxedo 日志文件以获取错误消息。默认情况下，可在由环境变量 APPDIR 指明的目录中找到该文件。文件名的形式为 ULOG.mmddyy，其中 mmddyy 表示当前的年、月和日。1999 年 3 月 12 日的文件为 ULOG.031299。可以通过在服务器上设置环境变量 ULOGPFX 来更改该文件的默认位置。还可以在客户端的当前目录下找到日志文件，除非 ULOGPFX 变量更改了其位置。

# 查看 Tuxedo 缓冲区数据

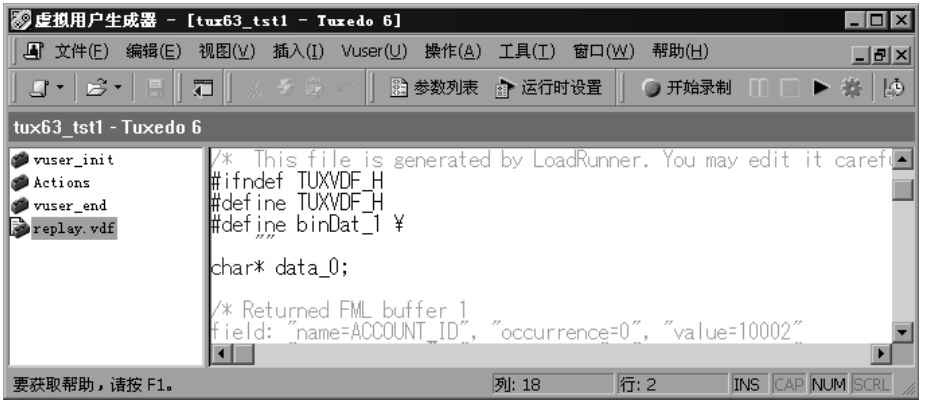
使用 VuGen 创建 Tuxedo Vuser 脚本时，操作将被录制到脚本的三个部分中：*vuser_init*、*Actions* 和 *vuser_end*。

所接收或传输的数据将被存储在数据缓冲区中，缓冲区可以很大。为了简化脚本的外观，实际数据将被存储在外部文件而非 C 文件中。发生数据传输时，会将外部文件中的数据复制到临时缓冲区中。

外部文件称为 *replay.vdf*，该文件包含所有临时缓冲区的内容。缓冲区的内容将存储为连续的记录。这些记录由表示数据被发送还是被接收的标识符以及缓冲区描述符进行标记。 LRT 函数使用缓冲区描述符访问数据。

可以使用 VuGen 通过在左窗格的树视图中选择 *replay.vdf* 文件来查看数据文件的内容。

默认情况下，查看数据文件的选项可以用于 Tuxedo 脚本。



# 定义 Tuxedo Vuser 的环境设置

下面的部分描述在 Windows 和 UNIX 平台上运行的 Tuxedo Vuser 的系统变量设置。可在“控制面板”/“系统”对话框 (NT) 或 .cshrc、.login 文件 (UNIX) 中定义系统变量。

| | |
|---|---|
| TUXDIR | Tuxedo 源的根目录 |
| FLDTBLDIR | 包含 FML 缓冲区信息的目录的列表。在 Windows 中，请用分号分隔目录名。在 UNIX 平台上，请用冒号分隔目录名 |
| FIELDTBLS | 包含 FML 缓冲区信息的文件的列表。在 Windows 和 UNIX 平台上，都请使用逗号分隔文件名 |

例如：

```
SET FLDTBLDIR=%TUXDIR%\udataobj;%TUXDIR%\APPS\WS (PC)
SET FIELDTBLS=bankflds,usysflds (PC)
setenv FLDTBLDIR $TUXDIR/udataobj:$TUXDIR/apps/bankapp (Unix)
setenv FIELDTBLS bank.flds,Usysflds (Unix)
```

必须为执行期间使用 Tuxedo/WS 工作站扩展的 Tuxedo 客户端定义下列系统变量：

| | |
|---|---|
| WSNADDR | 指定工作站侦听器进程的网络地址。这样可使客户端应用程序能够访问 Tuxedo。注意，要在一个 WSNADDR 语句中定义多个地址，每个地址都必须用逗号分隔 |
| WSDEVICE | 指定访问网络的设备。注意，不需要为某些网络协议定义此变量 |

例如：

```
SET WSNADDR=0x0002ffffc7cb4e4a (PC)
setenv WSNADDR 0x0002ffffc7cb4e4a (Unix)
setenv WSDEVICE /dev/tcp (Unix)
```

# 调试 Tuxedo 应用程序

通常，使用 **Tuxedo 6** 录制那些利用 Tuxedo 6.x 或更低版本的应用程序，而使用 **Tuxedo 7** 录制那些利用 Tuxedo 7.1 的应用程序。

如果在录制或回放 Tuxedo 应用程序时遇到了问题，或者脚本缺少对 lrt_tpinitialize 的调用，请与客户支持联系以检查应用程序所使用的 DLL。

如果应用程序使用 *wtuxws32.dll* 而非 *libwsc.dll*，请与客户支持联系以获得修补程序，使录制能够成功。

# 关联 Tuxedo 脚本

VuGen 支持对用 Tuxedo 应用程序录制的 Vuser 脚本进行关联。使用关联语句，可以通过保存缓冲区的一部分并将其用在后续语句中来链接语句。

要关联语句，请使用下列 LRT 函数之一在 VuGen 编辑器中修改录制的脚本。

- **lrt_save[32]_fld_val** 将 FML 或 FML32 缓冲区 （形式为 “name=<NAME>”或 “id=<ID>”的字符串）的当前值保存到参数中。
- **lrt_save_parm** 将字符数组 （例如 STRING 或 CARRAY 缓冲区）的一部分保存到参数中。
- **lrt_save_searched_string** 在缓冲区中搜索字符串实例，并将缓冲区的一部分 （相对于该字符串实例）保存到参数中。

有关这些函数的语法的其他信息，请参阅 《Online Function Reference》。

### 关联 FML 和 FML32 缓冲区

使用 **lrt_save_fld_val** 或 **lrt_save32_fld_val** 保存 FML 或 FML32 缓冲区的内容。

**要使用 lrt_save_fld_val 关联语句，请执行下列操作：**

1 在要保存当前 FML （或 FML32）缓冲区的内容的位置，将 **lrt_save_fld_val** 语句插入到脚本中。

lrt_save_fld_val (fbfr, "name", occurrence, "**param_name**");

2 引用参数。

用要替换为已保存缓冲区的内容的录制值查找 **lrt** 语句。将录制值的所有实例替换为尖括号中的参数名。

以下示例中，开设了一个银行帐户，并且将帐号存储到参数 *account_id* 中。

```
/* 用新帐户信息填充 data_0 缓冲区 */
data_0 = lrt_tpalloc("FML", "", 512);
lrt_Finitialize((FBFR*)data_0);
lrt_Fadd_fld((FBFR*)data_0, "name=BRANCH_ID", "value=1",
LRT_END_OF_PARMS);
lrt_Fadd_fld((FBFR*)data_0, "name=ACCT_TYPE", "value=S",
LRT_END_OF_PARMS);
...

LRT_END_OF_PARMS);
lrt_Fadd_fld((FBFR*)data_0, "name=LAST_NAME", "value=Doe", ...);
lrt_Fadd_fld((FBFR*)data_0, "name=FIRST_NAME", "value=John", ...);
lrt_Fadd_fld((FBFR*)data_0, "name=SAMOUNT", "value=234.12", ...);

/* 开设新帐户并保存新帐号 */
tpresult_int = lrt_tpcall("OPEN_ACCT", data_0, 0,&data_0, &olen_2, 0);
lrt_abort_on_error();
lrt_save_fld_val((FBFR*)data_0, "name=ACCOUNT_ID", 0, "account_id");

/* 使用第一个查询的结果填充存款缓冲区 */
lrt_Finitialize((FBFR*)data_0);
lrt_Fadd_fld((FBFR*)data_0, "name=ACCOUNT_ID",
"value=<account_id>", LRT_END_OF_PARMS);
lrt_Fadd_fld((FBFR*)data_0, "name=SAMOUNT", "value=200.11", ...);
```

上述示例中，帐户 ID 由字段名 ACCOUNT_ID 表示。录制期间，一些系统用 ID 号而不是字段名来表示字段。

可以通过字段 ID 进行关联，如下所示：

```
lrt_save_fld_val((FBFR*)data_0, "id=8302", 0, "account_id");
```

### 关联字符串

使用 **lrt_save_parm** 或 **lrt_save_searched_string** 关联字符串。

- 通常，建议使用 **lrt_save_parm** 将字符数组的一部分保存到参数中。
- 当您要相对于字符数组中特定字符串的位置保存信息时，请使用 **lrt_save_searched_string**。如果将 Vuser 用于 PeopleSoft，则建议使用 **lrt_save_searched_string**，因为从 PeopleSoft 服务器返回的回复缓冲区在回放期间的大小与在录制期间看到的大小通常会有所不同。

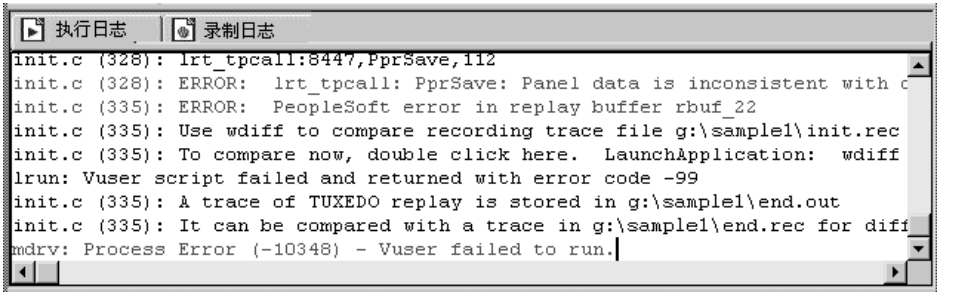
### 确定要关联的值

使用 CARRAY 缓冲区时， VuGen 会在录制期间和回放期间生成日志文件（在录制期间，文件扩展名为 *.rec*；在回放期间，文件扩展名为 *.out*），可以使用 *wdiff* 实用程序比较这两个日志文件。可以查看录制日志与回放日志之间的差异，以确定 CARRAY 缓冲区的哪些部分需要进行关联。

**要比较日志文件，请执行下列操作：**

1. 选择 “视图” > “输出” 以显示脚本的执行日志和录制日志。
2. 检查执行日志。

   错误消息的后面应跟有以下面的短语开头的语句：*Use wdiff to compare*。




3. 在执行日志中双击该语句以启动 *wdiff* 实用程序。

   将打开 WDiff，同时以黄色高亮显示录制文件与回放文件之间的差异。有关 Wdiff 实用程序的详细信息，请参阅第 8 章 “关联语句”。

**要使用 lrt_save_parm 关联语句，请执行下列操作：**

决定对哪个值进行关联后，可以使用 **lrt_save_parm** 将字符数组（例如 STRING 或 CARRAY 缓冲区）的一部分保存到参数中。

**1** 在要保存当前缓冲区内容的位置，将 **lrt_save_parm** 语句插入到脚本中。

lrt_save_parm (buffer, offset, length, "**param_name**");

**2** 在 *replay.vdf* 文件中，查找要替换为已保存的缓冲区内容的缓冲区数据。

在 VuGen 主窗口的 “数据文件”框中，通过选择 *replay.vdf* 文件来查看缓冲区内容。

**3** 将该值的所有实例都替换为尖括号中的参数名。

以下示例中，必须保存 CARRAY 缓冲区中的员工 ID 供以后使用。录制的值为 “G001”，如输出中所示。

```
lrt_tpcall:227, PprLoad, 1782
Reply Buffer received.
...
123 "G001"
126 "..."
134 "Claudia"
```

使用偏移量 123，在发送 “PprLoad”和 227 字节的请求缓冲区后的紧邻位置插入 **lrt_save_parm**。

```
/* 请求 CARRAY 缓冲区 57 */
      lrt_memcpy(data_0, buf_143, 227);
      tpresult_int = lrt_tpcall("PprLoad",
           data_0, 227, &data_1, &olen, TPSIGRSTRT);
      lrt_save_parm(data_1, 123, 9, "empid");
```

在 *replay.vdf* 文件中，将录制的值 “G001” 替换为参数 *empid*。

```
char buf_143[] =
"\xf5\x0\x0\x0\x4\x3\x2\x1\x1\x0\x0\x0\xbc\x2\x0\x0\x0\x0\x0
\x0"
   "X"
"\x89\x0\x0\x0\xb\x0"
   "SPprLoadReq"
   "\xff\x0\x10\x0\x0\x4\x3\x6"
   "<empid>" // G001
   "\x7"
   "Claudia"
   "\xe"
   "LAST_NAME_SRCH"
...
```

该函数还可以用于将字符数组的一部分保存到 FML 缓冲区中。以下示例中，电话号码是字符数组，而区号是前三个字符。首先， **lrt_save_fld_val** 语句将电话号码保存到参数 *phone_num* 中。 **lrt_save_parm** 语句使用 **lr_eval_string** 将电话号码变为字符数组，然后将区号保存到名为 *area_code* 的参数中。

```
lrt_save_fld_val((FBFR*)data_0, "name=PHONE", 0, "phone_num");
lrt_save_parm(lr_eval_string("<phone_num>"), 0, 3, "area_code");
lr_log_message("The area code is %s\n", lr_eval_string("<area_code>"));
```

**要使用 lrt_save_searched_string 关联语句，请执行下列操作：**

使用 **lrt_save_searched_string** 在缓冲区中搜索字符串，然后将缓冲区的一部分（相对于字符串实例）保存到参数中。

**1** 在要保存当前缓冲区内容的位置，将 **lrt_save_searched_string** 语句插入到脚本中。

```
lrt_save_searched_string (buffer, buf_size, occurrence, string, offset,
                                    length, "param_name");
```

注意，此处的偏移量是相对于字符串开头的偏移量。

**2** 在 *replay.vdf* 文件中，查找要替换为已保存的缓冲区内容的缓冲区数据。

在 VuGen 主窗口的 “数据文件”框中，通过选择 *replay.vdf* 文件来查看缓冲区内容。

**3** 将该值的所有实例都替换为尖括号中的参数名。

在以下示例中， 将 Certificate 保存到参数， 供以后使用。 **lrt_save_searched_string** 函数将指定的 olen 缓冲区中的 16 个字节保存到参数 *cert1* 中。缓冲区中已保存的字符串的位置是字符串 “ScertRep”的第一个实例之后的 9 个字节处。

当缓冲区的标头信息随录制环境的不同而不同时，该应用程序非常有用。

证书总是出现在 “ScertRep”的第一个实例之后的 9 个字节处，但该字符串前面的信息的长度会有所不同。

```
/* 请求 CARRAY 缓冲区 1 */
lrt_memcpy(data_0, sbuf_1, 41);
lrt_display_buffer("sbuf_1", data_0, 41, 41);
data_1 = lrt_tpalloc("CARRAY", "", 8192);
tpresult_int = lrt_tpcall("GetCertificate",
      data_0,
      41,
      &data_1,
      &olen,
      TPSIGRSTRT);

/* 回复 CARRAY 缓冲区 1 */
lrt_display_buffer("rbuf_1", data_1, olen, 51);
lrt_abort_on_error();

lrt_save_searched_string(data_1, olen, 0, "SCertRep", 9, 16, "cert1");
```

# 第 XIV 部分

## 流数据协议

# 60

# 开发流式数据 Vuser 脚本

流媒体是一个增长迅速的市场，使用流媒体可以在 Internet 上传送音频 / 视频内容。流媒体的含义是：可以将音频 / 视频内容传输给最终用户，而不需要首先下载整个文件。流式传输的工作原理是：让服务器在客户端显示内容的同时将内容源源不断地传输到客户端。

RealPlayer 和 Media Player 都是显示流式内容的应用程序。

使用 VuGen 可记录使用 RealPlayer 或 Media Player 协议进行通信的客户端应用程序与服务器之间的通信。得到的脚本称为 Real 或 Media Player Vuser 脚本。

本章描述：

- 流式数据 Vuser 脚本入门
- 使用 RealPlayer LREAL 函数
- 使用 Media Player MMS 函数

***下列信息仅适用于 Real 和 Media Player 协议。***

## 关于录制流式数据虚拟用户脚本

通过流式数据协议可以模拟播放媒体或流式数据文件的用户。

使用流式数据协议录制应用程序时， VuGen 将生成描述操作的函数。对于 RealPlayer 会话， VuGen 将生成带有 **lreal** 前缀的函数。对于 Media Player 会话，VuGen 将使用带有 **mms** 前缀的函数。注意，Media Player mms 函数不支持录制 – 仅支持回放。

# 流式数据 Vuser 脚本入门

本部分概述了使用 VuGen 开发 RealPlayer 和 Media Player 流式数据 Vuser 脚本的过程。

**要开发 RealPlayer 或 Media Player Vuser 脚本，请执行下列操作：**

**1 使用 VuGen 录制基本脚本。（仅限于 RealPlayer）**

对于 RealPlayer，请调用 VuGen 并新建一个虚拟播放器脚本。选择要录制的应用程序。录制应用程序中的典型操作。详细信息，请参阅第 3 章 “使用 VuGen 进行录制”。

对于 Media Player，不支持录制操作。您可以改而创建一个空白的 Media Player 脚本，然后将 **mms** 函数手动插入到该脚本中。有关示例，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

**2 增强脚本。**

通过将事务、集合点和控制流结构插入到脚本中，可以增强脚本。

详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

**3 定义参数 （可选）。**

为录制到脚本中的固定值定义参数。通过用参数替换固定值，可以使用不同的值多次重复执行相同的业务流程。

详细信息，请参阅第 7 章 “定义参数”。

**4 关联语句 （可选）。**

通过关联语句，可以将一个业务流程的结果用于后续业务流程中。

详细信息，请参阅第 8 章 “关联语句”。

**5 配置运行时设置。**

运行时设置可控制脚本执行期间虚拟播放器 / 用户的行为。这些设置包括循环、日志和计时信息。

详细信息，请参阅第 9 章 “配置运行时设置”。

**6 从 VuGen 中运行脚本。**

在 VuGen 中保存并运行该脚本以验证该脚本是否可以正常运行。

详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

创建 Vuser 脚本后，请在 Windows 或 UNIX 平台上将其集成到场景中。有关将 Vuser 脚本集成到场景中的详细信息，请参阅 《LoadRunner Controller 用户指南》。

## 使用 RealPlayer LREAL 函数

所开发的用于模拟使用 RealPlayer 协议在客户端与服务器之间进行的通信的函数称为 Real Player 函数。每个 Real Player 函数都具有 **lreal** 前缀。Real Player 会话期间， VuGen 将自动录制本部分中列出的大多数 LREAL 函数。您还可以手动将任一函数编写到脚本中。有关 LREAL 函数的详细信息，请参阅 《Online Function Reference》（“帮助” > “函数参考”）。

| | |
|---|---|
| **lreal_clip_size** | 返回当前片段的大小 |
| **lreal_close_player** | 关闭 RealPlayer 实例 |
| **lreal_open_player** | 新建 RealPlayer 实例 |
| **lreal_open_url** | 打开 URL |
| **lreal_pause** | 暂停播放 RealPlayer 片段 |
| **lreal_play** | 播放 RealPlayer 片段 |
| **lreal_seek** | 查找 RealPlayer 片段中的位置 |
| **lreal_stop** | 停止播放 RealPlayer 片段 |

例如， **lreal_play** 函数采用以下形式：

int **lreal_play** (int miplayerID, long *mulTimeToPlay*);

要播放该片段直到结束，请为 *mulTimeToPlay* 使用任意负值。要播放该片段一段特定的时间 （毫秒），请指定毫秒数。 *miplayerID* 代表 RealPlayer 实例的唯一 ID。

# 使用 Media Player MMS 函数

所开发的用于模拟 Media Player 使用 MMS 协议进行的客户端 / 服务器通信的函数称为 MMS 虚拟用户函数 – 每个函数都具有 **mms** 前缀。

所有 MMS 函数都成对出现 – 一个用于全局会话，另一个用于特定会话。例如，**mms_close** 将全局性地关闭 Media Player，而 **mms_close_ex** 将关闭 Media Player 的特定会话。

有关这些函数的详细语法信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

| 函数名 | 描述 |
|---|---|
| **mms_close[_ex]** | 关闭 Media Player |
| **mms_get_property[_ex]** | 检索 Media Player 片段的属性 |
| **mms_isactive[_ex]** | 验证 Media Player 是否处于活动状态 |
| **mms_pause[_ex]** | 暂停播放 Media Player 片段 |
| **mms_play[_ex]** | 播放 Media Player 片段 |
| **mms_resume[_ex]** | 继续播放 Media Player 片段 |
| **mms_sampling[_ex]** | Media Player 片段取样 |
| **mms_set_property[_ex]** | 设置 Media Player 片段属性 |
| **mms_set_timeout[_ex]** | 设置 Media Player 片段的超时值 |
| **mms_stop[_ex]** | 停止播放 Media Player 片段 |

例如， **mms_play** 函数采用以下形式：

**int mms_play (char message, < 属性列表 >, LAST);**

以下示例中， **mms_play** 函数按不同的持续时间播放 *asf* 文件：

```
// 播放持续时间为 10 秒。
mms_play("Welcome","URL=mms://server/welcome.asf","
duration=10",
LAST);

// 等待 5 秒钟后，播放该片段直到结束。
mms_play ("Welcome","URL=mms://server/welcome.asf",
return -1;
"starttime=5",
LAST);
```

# 第 XV 部分

## 无线协议

# 61

# 介绍无线 Vuser

通过 VuGen 可以为使用 WAP、VoiceXML 或 i-mode 协议的无线应用程序开发脚本。 VuGen 通过录制无线网络上的操作来创建 Vuser 脚本。

本章描述：

- 了解 WAP 协议
- 了解 i-mode 系统
- i-mode 与 WAP 之比较
- 了解 VoiceXML

## 关于无线 Vuser

VuGen 支持三种无线协议：

- WAP （无线应用协议）
- i-mode
- VoiceXML

每个协议都有特定的特征，在用户内容的实现和开发方面都有所不同。

开发人员可以使用工具包作为使用无线协议创建内容和应用程序时的开发环境。

# 了解 WAP 协议

无线应用协议 (WAP) 是一种开放式全局规范，它使具有无线设备的移动用户能够即时访问信息和服务并与其交互。

WAP 协议使用称为 WML 的新标准来指定微型浏览器瘦客户端，该标准已针对无线手持式移动终端进行了优化。 WML 是 XML 的精简版本。

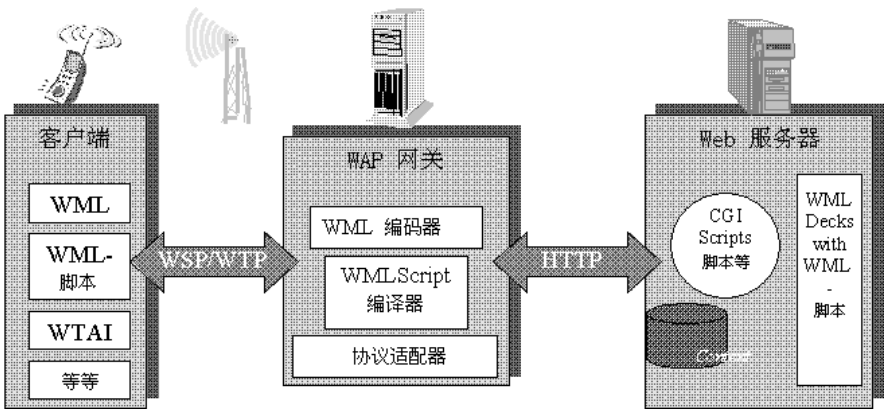
WAP 还指定代理服务器，该代理服务器将：

- 充当无线网络和有线 Internet 之间的网关
- 提供协议转换
- 针对无线手机优化数据传输

WAP 体系结构和 WWW 模型很相似。所有内容都被指定使用类似于标准 Internet 格式的格式。在 WWW 域中，使用标准协议传输内容，而在无线域（无线会话协议）中，则使用优化的类似于 HTTP 的协议传输内容。您可以使用 WWW 标准 URL 来查找所有 WAP 内容。

WAP 使用很多 WWW 标准，包括创作和发布方法。 WAP 以能够反映设备和网络特征的方式增强了某些 WWW 标准。增加了 WAP 扩展以支持移动网络服务，如 “调用控制”和 “消息传送”。它还顾及到了移动终端中存在的内存容量和 CPU 处理能力等方面的限制。 WAP 还支持低带宽和高滞后网络。

WAP 假定存在负责对移动客户端发送和接收的数据进行编码和解码的网关。对传送到客户端的内容进行编码的目的是，尽可能减小在空中发送到客户端的数据的大小，同时尽可能减小客户端为处理该数据所需的计算能力。可以将网关功能添加到源服务器，或者放在专用网关中，如下所述。



### WAP 工具包

为了帮助开发人员生成 WAP 应用程序和服务，许多一流的公司 （如 Nokia、Ericsson 和 Phone.com 等）已经开发了“工具包”。 WAP 工具包还为希望为移动终端提供 Internet 服务和内容的开发人员提供了环境。它使开发人员可以在基于 PC 的模拟电话上编写、测试、调试和运行应用程序。该工具包使用户可以通过 HTTP 连接或 WAP 网关来浏览 WAP 站点。

移动电话使用 WSP 协议与网关通信；工具包可以与网关通信，也可直接与服务器通信。 VuGen 使您可以用下列两种模式进行录制：WSP 和 HTTP。如果对发往网关的流量感兴趣，请以 WSP 模式进行录制。如果您希望检查服务器和内容提供商，则可以用 HTTP 模式录制工具包会话并回避网关。

VuGen 使用自定义 API 函数来模拟用户会话。大多数函数都是使用 HTTP 协议的标准 Web 协议函数。还有几个 WAP 函数模拟特定于 WAP Vuser 的操作。有关支持的函数的列表，请参阅第 729 页的 “使用无线 Vuser 函数”。

## 了解 i-mode 系统

I-mode 协议是 NTT DoCoMo 的移动 Internet 访问系统。从技术上讲， i-mode 与普通移动语音系统重叠。语音系统是电路交换系统 （即需要拨号），而 i-mode 是数据包交换系统。这意味着从原则上说，只要您可以获得 i-mode 信号， i-mode 将始终处于连接状态。当您从手机菜单上选择 i-mode 项目时，通常会立即下载数据，没有常见的拨号延迟。不过，接收数据时可能会有延迟，具体取决于数据大小和网络带宽。

I-mode 的使用方式类似于通过浏览器访问 Internet。例如，它们可以发送电子邮件、查看天气预报和体育比赛结果、玩游戏、执行联机股票交易、购买机票和查找饭店。

I-mode 协议使用 *cHTML* （压缩 HTML），这是普通 HTML 的子集。除了标准 HTML 标记以外，还有几个特定于 i-mode 的标记。例如，有一个 i-mode 标记用于设置可拨通某个电话号码的链接。另外有一个特定于 i-mode 的标记可通知搜索引擎某个特定的网页是 i-mode 页面。

另外，还有很多充当符号的 DoCoMo 特殊字符。例如，有一些特殊字符表示欢乐、爱、悲伤、电话、火车、圈中的数字等等。

既然 cHTML 是 HTML 的子集，因此可以使用 Netscape 或 IE 浏览器查看 i-mode 页面，如 http://www.eurotechnology.com/i/ 或 http://www.eu-japan.com/i/。不过，由于几乎所有 i-mode 用户都是日本人，因此几乎所有 i-mode 内容都是日语内容。于是，浏览器中需要有日语文本显示支持。在常规浏览器中查看 i-mode 内容时，将无法查看特定于 i-mode 的标记。另外，还无法显示特殊的 DoCoMo i-mode 符号。

### i-mode 工具包

要帮助开发人员生成 i-mode 服务，须提供多个工具包，而且由 VuGen 对其进行支持。 I-mode 工具包还为希望为移动终端提供 Internet 服务和内容的开发人员提供了环境。工具包使开发人员可以在基于 PC 的模拟电话上编写、测试、调试和运行应用程序。工具包使用户可以通过标准 HTTP 连接来浏览 i-mode 站点。部分支持的工具包包括 CompactViewer 和 Pixo 版本 2.0、 2.1。

## i-mode 与 WAP 之比较

当前，基于 i-mode 的服务和基于 WAP 的服务在实现方式上有几项重大差异。I-mode 使用 cHTML，它是 HTML 的子集，掌握起来要比 WAP 的标记语言 WML 容易些。当前， i-mode 通过数据包交换系统来实现，该系统原则上 “始终处于连接状态”，而 WAP 系统使用电路交换模型，即需要拨号。注意，数据包交换或电路交换是指服务所基于的电信系统存在的技术差异。原则上，可以通过数据包交换系统或电路交换系统传送使用 i-mode 和 WAP 编码的网页。

另外一个差异存在于定价方法方面：I-mode 用户按照下载的信息量和各种增值服务来收费。 WAP 用户则按照连接时间进行收费。

## 了解 VoiceXML

VoiceXML （即 VXML）是一种使用语音识别技术通过语音浏览器或电话来与 Internet 进行交互的技术。使用 VoiceXML 时，通过收听预先录制的或由计算机合成的音频，并通过自然发音或小键盘 （如电话）提交输入，来与语音浏览器进行交互。

VoiceXML 包含可访问 Web 上的静态或动态 VoiceXML 内容的 VoiceXML 网关。该网关具有 VoiceXML 浏览器、文字语音转换、自动语音识别 (ASR) 和连接公用交换电话网络 (PSTN) 的电话硬件。它通过下列线路之一连接到电话网络：T1、POTS 或 ISDN。普通旧式电话服务器 (POTS) 线路类似于在住宅中使用的线路，它只处理单一连接；而 T1 线路具有 24 根独立的电话线。

典型语音对话包括：

1. 通过电话 （无线或固定电话）拨通系统。电话硬件将拾取语音并将呼叫传递到 VoiceXML 浏览器。
2. VoiceXML 网关将从指定的 Web 浏览器检索扩展名为 *vxml* 的 VoiceXML 文档并播放提示音。
3. 您可对着电话说话或者在电话键盘上按相应键。
4. 电话设备使用 VoiceXML 文档中包含的预定义词典，将记录的声音传递到语音识别引擎 （如果是语音的话）。
5. VoiceXML 浏览器将根据语音分析结果执行文档中的命令，并再次播放预先录制的或合成的提示音。

VuGen 支持录制 VoiceXML 会话。录制的脚本包含可模拟操作的 **web_url** 函数。在以下示例中，用户请求包含股票信息的页面。

```
Action1()
{
    web_add_auto_header("Accept",
        "text/x-vxml, */*");

    web_add_auto_header("Content-Type",
        "application/x-www-form-urlencoded");

    web_add_auto_header("User-Agent",
        "Motorola VoxGateway/2.0");

    web_url("top.vxml",
        "URL=http://testserver1/Vxmlexample/top.vxml?DNIS=-",
        "Resource=0",
        "RecContentType=application/octet-stream",
        "Referer=",
        "Mode=HTTP",
        LAST);

    web_url("stock.vxml",
        "URL=http://testserver1/Vxmlexample/stock.vxml",
        "Resource=0",
        "RecContentType=application/octet-stream",
        "Referer=",
        "Mode=HTTP",
        LAST);
return 0;
}
```

# 62

# 录制无线 Vuser 脚本

使用 VuGen 可以通过录制典型的无线会话来生成无线 Vuser 脚本。在运行脚本时，生成的 Vuser 将模拟工具包或电话与 Web 服务器 （或 WAP 的网关）之间的活动。

本章描述：

- 无线 Vuser 脚本入门
- 使用无线 Vuser 函数
- 无线 Vuser 脚本疑难解答

***下列信息仅适用于所有无线协议、 WAP、 i-mode 和 VoiceXML。***

## 关于录制无线 Vuser 脚本

假设您拥有一个按客户显示购买请求状态的网站。您希望确保任何客户查询的响应时间小于某个指定值 （例如， 20 秒），即使在大量用户 （例如 200 个）同时访问该网站时也是如此。您可以使用 Vuser 来模拟这一场景，其中由 Web 或 WAP 服务器负责处理并发的信息请求。每个 Vuser 可以：

- 加载打开的页面
- 提交请求
- 等待服务器的响应

您可以将数百个 Vuser 分布在可用的测试计算机上，每个 Vuser 都使用服务器的 API 来访问服务器。这样，便可以度量在许多用户的负载下服务器的性能。

# 无线 Vuser 脚本入门

本部分概述了使用 VuGen 开发无线 Vuser 脚本的过程。

**要开发无线脚本，请执行下列操作：**

**1 使用 VuGen 新建一个脚本。**

选择“文件”>“新建”或者单击“新建”按钮，以单协议模式或多协议模式新建一个脚本。

有关新建脚本的详细信息，请参阅第 3 章“使用 VuGen 进行录制”。

**2 设置录制选项。**

设置录制选项。有关设置常见 Internet 录制选项的信息，请参阅第 35 章“为 Internet 协议设置录制选项”。有关特定于无线的录制选项的信息，请参阅第 64 章“设置无线 Vuser 录制选项”。

**3 使用 VuGen 录制操作。**

通过工具包会话录制操作。

有关录制的信息，请参阅第 3 章“使用 VuGen 进行录制”。

**4 增强 Vuser 脚本。**

通过将事务、集合点和控制流结构插入到脚本中，增强 Vuser 脚本。

详细信息，请参阅第 6 章“增强 Vuser 脚本”。

**5 定义参数（可选）。**

为录制到 Vuser 脚本中的固定值定义参数。通过使用参数替换固定值，可以使用不同的值多次重复执行同一业务流程。

详细信息，请参阅第 7 章“定义参数”。

**6 配置运行时设置。**

运行时设置用于控制脚本执行过程中 Vuser 的行为。这些设置包括运行逻辑、步、日志记录、思考时间和性能首选项。

有关常规运行时设置的信息，请参阅第 9 章“配置运行时设置”。

有关常见 Internet 协议运行时设置的信息，请参阅第 37 章“配置 Internet 运行时设置”。

有关特定于 WAP 的运行时设置的信息，请参阅第 65 章“配置 WAP 运行时设置”。

**7 执行关联。**

检查脚本以确定是否有需要关联的动态值。对于无线协议，可通过添加 **web_reg_save_param** 函数执行手动关联。

详细信息，请参阅第 502 页 “执行手动关联”。

**8 通过 VuGen 保存并运行 Vuser 脚本。**

通过 VuGen 保存并运行 Vuser 脚本，以验证该脚本是否运行正常。录制时，VuGen 会创建一系列配置、数据和源代码文件。这些文件包含 Vuser 运行时信息和设置信息。 VuGen 将这些文件与脚本一起保存。

有关将 Vuser 脚本作为独立测试运行的详细信息，请参阅第 11 章 “在独立模式下运行 Vuser 脚本”。

创建无线 Vuser 脚本后，请将其集成到场景中。详细信息，请参阅《LoadRunner Controller 用户指南》。

## 使用无线 Vuser 函数

所开发的用来模拟无线设备和 Web 服务器 （或 WAP 的网关）之间通信的函数称为 Vuser 函数。一些函数在录制脚本时生成；其他函数必须手动插入到脚本中。录制后，还可以向 Vuser 脚本中添加 LoadRunner 消息函数和自定义 C 函数。

表示标准 HTTP 操作的函数具有 web 前缀。有关这些函数的信息，请参阅第 33 章 “使用 Web Vuser 函数”。

常规 Vuser 函数以 **lr** 前缀开头。详细信息，请参阅第 18 页“使用 C Vuser 函数”。

以下部分描述表示特定于 WAP 的操作函数，这些函数具有 wap 前缀。

有关所有 Web 相关函数的完整列表，请参阅第 33 章 “使用 Web Vuser 函数”或者参阅 《Online Function Reference》（“帮助” > “函数参考”）。

对于以无线会话协议 (WSP) 模式运行脚本的 WAP Vuser，仅支持下列函数：

| | |
|---|---|
| **操作函数：** | **web_custom_request**、 **web_submit_data** 和 **web_url** |
| **身份验证函数：** | 全部 – **web_set_user**、 **web_set_certificate**[_ex] |
| **Cookie 函数：** | 全部 – **web_add_cookie**、 **web_cleanup_cookie**、 **web_remove_cookie** |
| **标头函数：** | 全部 – **web_add_auto_header**、 **web_add_header**、 **web_cleanup_auto_headers**、 **web_save_header** |
| **关联函数：** | 全部 – **web_create_html_param**[_ex]、 **web_reg_save_param**、 **web_set_max_html_param_len** |

以下部分列出了特定于 WAP 的函数：

### 特定于 WAP 的函数

| | |
|---|---|
| **wap_add_const_header** | 指定要传递给 WAP 网关的固定标头 |
| **wap_connect** | 连接到 WAP 网关 |
| **wap_disconnect** | 从 WAP 网关断开连接 |
| **wap_format_si_message** | 设置 SI 类型消息的格式 |
| **wap_format_sl_message** | 设置 SL 类型消息的格式 |
| **wap_mms_msg_add_field** | 向 MMSC 消息中添加字段 |
| **wap_mms_msg_add_mulitpart_entry** | 向 MMSC 消息中添加多部分项 |
| **wap_mms_msg_create** | 为 MMSC 创建消息 |
| **wap_mms_msg_destroy** | 销毁 MMSC 消息 |
| **wap_mms_msg_submit** | 将消息发送到 MMSC |
| **wap_pi_push_cancel** | 取消发送到 PPG 的消息 |
| **wap_pi_push_submit** | 提交 Push 消息 |

| | |
|---|---|
| **wap_radius_connection** | 连接 RADIUS 服务器或从 RADIUS 服务器断开连接 |
| **wap_send_sms** | 发送 SMS 类型消息 |
| **wap_set_bearer** | 设置基础载体 – UDP 或 CIMD2 (SMS) |
| **wap_set_capability** | 为 WAP 网关连接设置客户端功能 |
| **wap_set_connection_mode** | 设置连接模式和安全级别 |
| **wap_set_gateway** | 设置网关 IP 地址和端口 |
| **wap_set_sms_user** | 设置 SMSC 的登录信息 |
| **wap_wait_for_push** | 等待推消息到达 |

详细信息，请在 VuGen 编辑器中选择相应的函数并按 F1 键，或者请参阅《LoadRunner Function Reference》（“帮助” > “函数参考”）。

# 无线 Vuser 脚本疑难解答

## Nokia 工具包

对于 Nokia 工具包 （3.0 和 3.1），需要通过将正确的 IP 地址分配给该工具包来手动启动它。

**要重新启动该工具包，请执行下列操作：**

1. 确保在 VuGen 计算机上没有网关正在运行。如果存在其他网关，可能会阻塞伪网关的端口。
2. 启动 VuGen。
3. 调用该工具包。（必须在启动 VuGen 之后调用该工具包）
4. 在 “录制选项”中，选择 “Internet 协议 :WAP 工具包”节点，并选择 “手动启动 WAP 工具包”。
5. 单击 “确定”。将打开一个消息框，其中包含由 VuGen 分配的网关 IP 地址。
6. 复制该 IP 地址并粘贴到该工具包的连接设置中。

**7** 在该工具包中，输入需要的 URL。忽略消息 “网关未连接”。再次输入 URL。VuGen 此时可能已经录制了几个 **web_add_const_header** 事件。

每个新的录制会话都会要求关闭该工具包并重复执行步骤 2 - 7。

---

**注意：**上面的过程也可用于其他遇到录制问题的工具包。

---

# 63

# 使用 WAP Vuser 脚本

可以使用 VuGen 开发 WAP （无线应用协议） Vuser 脚本。 VuGen 通过录制您在使用 WAP 设备时执行的操作来创建 Vuser 脚本。

本章描述：

- 通过电话录制
- 载体支持
- RADIUS 支持
- 推支持
- LoadRunner 推支持
- MMS 支持

## 关于 WAP Vuser

无线应用协议 (WAP) 是一种开放性规范，它使具有无线设备的移动用户能够即时访问信息和服务并与其交互。有关 WAP 技术的概述，请参阅第 61 章 “介绍无线 Vuser”。

VuGen 使用自定义 API 函数来模拟用户会话。大多数函数是使用 HTTP 协议的标准 Web 协议函数。其中几个 WAP 函数模拟特定于 WAP Vuser 的操作。有关支持的函数的列表，请参阅第 729 页的 “使用无线 Vuser 函数”。

您可以使用工具包或电话来录制 WAP 会话。有关通过工具包进行录制的信息，请参阅第 64 章 “设置无线 Vuser 录制选项”。有关通过电话进行录制的信息，请参阅 “通过电话录制”。

您可以对脚本进行编程，使用 **wap** Vuser 函数来模拟 WAP 会话。详细信息和示例，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

通过 VuGen 对 WAP 的支持，您可以选择载体、标识 RADIUS 服务器以及模拟“推”机制。本章中描述了此项支持。

## 通过电话录制

您可以录制电话或工具包和 WAP 网关之间的 WSP 会话。要录制 WSP 会话，请确保工具包或电话网关的设置是可配置的。

在录制过程中， VuGen 会启动 “伪”网关。 VuGen 将捕获该网关上的 WSP 流量并创建脚本。

要为 WSP 录制会话配置 VuGen，您必须在录制选项的 “录制模式”选项卡中启用 WSP （请参阅第 64 章 “设置无线 Vuser 录制选项”）。

输入原始网关 IP 地址并将录制模式设置为 CO 或 CL。确保工具包或电话支持您选择的录制模式。

要使用电话通过无线连接进行录制，必须首先拨通 ISP 以接入 Internet。将电话配置为 VuGen 计算机的 IP 地址，并将电话设置为需要的录制模式 （CO 或 CL）。

VuGen 计算机可以具有下列配置之一：

- 如果您通过第三方 ISP 连接，则带有 “伪”网关的 VuGen 计算机应该可以通过 Internet 访问 – 它必须位于防火墙之外。
- 如果您通过远程访问服务器 (RAS) 拨入，则可以将 VuGen 计算机作为网络的一部分进行访问。

## 载体支持

WAP 体系结构中的传输层协议包括无线事务协议 (WTP) 和无线数据报协议 (WDP)。

基础载体是一个用于在两个设备之间携带 WDP 协议的数据传输机制。基础载体包括 SMS-CIMD2、UDP、CSD、GSM GPRS、GSM CSD 和 Packet Data 等。

LoadRunner WAP Vuser 当前支持 UDP （用户数据报协议）和 SMS-CIMD2 （短消息服务）载体。

UDP 载体不需要单独的连接 - 它们通过 IP 网络运行。但是，要使用 SMS-CIMD2，您必须连接到 SMS 中心 (SMSC) 并提供适当的信息：

- **IP 和端口信息**：对于 UDP 载体，可在 “运行时设置” 的 “载体” 选项卡中定义端口和登录信息 （请参阅第 750 页的 “配置载荷信息”）。
- **SMS 中心的登录信息**：可在 “运行时设置” 的 “载体” 选项卡中定义 SMS 登录信息。还可以通过 **wap_set_sms_user** 函数来设置此信息。在需要使用参数化为许多 Vuser 设置登录信息时，其有助于负载测试。
- **CIMD2 的登录信息**：可在 “运行时设置” 的 “载体” 选项卡中设置 CIMD2 载体信息 （请参阅第 65 章 “配置 WAP 运行时设置”。）。

在某些情况下，您可能需要使用多种类型的载体。例如，某人在您的电话关机时通过 UDP 协议向您发送消息。您打开电话后，通过 SMS 协议来检索消息。在脚本执行期间，您可以使用 **wap_set_bearer** 函数在不同的载体类型之间进行切换。

## RADIUS 支持

*RADIUS* （远程身份验证拨入用户服务）是一个客户端 / 服务器的协议和软件，远程访问服务器可以通过它与中央服务器进行通信，以便对拨入的用户进行身份验证，并授予这些用户访问所要求的系统或服务的权限。

RADIUS 使公司可以在所有远程服务器都能共享的中央数据库中维护用户配置文件。它提供了更好的安全性，使公司能够建立可在单个托管网络位置应用的策略。使用中央服务，可以更容易地跟踪使用情况以便进行记帐以及存储网络统计信息。

RADIUS 有两个子协议：

- **Authentication**：授权并控制用户访问。
- **Accounting**：跟踪使用情况以便进行记帐以及保存网络统计信息。

在 VuGen 中，对 RADIUS 协议的支持仅限于两种 Radius 子协议（authentication 和 accounting）的 WSP 回放。

在 “运行时设置”的 “Radius”选项卡中提供拨入信息。详细信息，请参阅第 65 章 “配置 WAP 运行时设置”。

## 推支持

在普通的客户端 / 服务器模型中，客户端从服务器请求信息或服务。服务器通过向客户端传输信息或执行服务进行响应。这称为 “拉”技术 - 客户端从服务器拉信息。

与此对应，还存在 “推”技术。 WAP 推框架可以将信息传输到设备，而不需要用户提前操作。该技术也基于客户端 / 服务器模型，但在服务器传输其内容之前，没有来自客户端的明确请求。

要使用 WAP 执行推操作，“推发起者 (PI)”需要将内容传输到客户端。但是，推发起者协议与 WAP 客户端并不完全兼容 – 推发起者位于 Internet 中，而 WAP 客户端位于 WAP 域中。因此，我们需要插入一个转换网关，作为推发起者和 WAP 客户端之间的媒介。转换网关也称为 “推代理网关 (PPG)”。

Internet 端的访问协议称为 “推访问协议 (PAP)”。

WAP 端的协议称为 “空中推送 (OTA)”协议。

推发起者使用 PAP Internet 协议通过 Internet 与推代理网关 (PPG) 进行联系。PAP 使用可以通过各种常见的 Internet 协议 （例如 HTTP）传送的 XML 消息。PPG 将推送的内容转发到 WAP 域。这些内容随后被利用 OTA 协议通过移动网络传输到目标客户端。 OTA 协议是基于 WSP 服务的。

除了提供简单的代理网关服务以外， PPG 还能够将推操作的最终状态通知给推发起者。在双向移动网络中， PRG 还能够等待客户端接受或拒绝内容。

### 推服务类型

推服务可以是 SL 类型或 SI 类型的：

- **SL** – 服务加载 (SL) 内容类型提供了一个功能，可以使移动客户端上的用户代理加载和执行服务，例如 WML deck。 SL 包含一个 URI，指明在适当的时候将由用户代理在没有用户干预的情况下加载的服务。
- **SI** – 服务通知 (SI) 内容类型提供了以异步方式向最终用户发送通知的功能。例如，通知可能与新的电子邮件、股价变化、新闻标题和广告有关。

  大多数基本形式的 SI 都包含一个短消息和一个指明服务的 URI。消息将在收到时呈现给最终用户，用户可以选择立即启动由 URI 指明的服务，或者推迟 SI 待以后处理。如果推迟 SI，客户端将存储该 SI，并且最终用户将能够在以后某个时间来处理它。

## LoadRunner 推支持

VuGen 的推支持分为 3 个部分：

- 客户端的推支持 – 能够接受推消息。
- 对 WAP HTTP Vuser 的推支持 – 模拟推发起者。
- 推消息 （SI 和 SL）格式服务 – 设置推消息的格式。

### 客户端推支持

在客户端， VuGen 对于所有回放模式 （CO 和 CL）都支持两种推服务 （SL 和 SI）。 **wap_wait_for_push** 函数指示 Vuser 等待推消息到达。可以在 “运行时设置”中设置该函数的超时。

推消息到达时， LoadRunner 对其进行分析以确定其类型并检索其属性。如果分析成功，则生成并执行一个推事务，以检索相关数据。您可以禁用推事件，通过配置 “运行时设置”来指示 LoadRunner 不要检索消息数据。详细信息，请参阅第 65 章 “配置 WAP 运行时设置”。

### 模拟推发起者

通过 LoadRunner 对 WAP HTTP Vuser 的推支持，您可以执行 PPG 负载测试。推支持使 Vuser 可以充当支持 “推访问协议 (PAP)” 的推发起者。PAP 定义了 PI 和 PPG 之间的下列操作集：

1. 提交推请求。
2. 取消推请求。

3 提交对推请求状态的查询。

4 提交对无线设备功能的状态的查询。

5 发起从 PPG 到 PI 的结果通知消息传输。

所有操作都是请求 / 响应 – 对于发起的每个消息，都将向 PI 发回响应。 PI 操作基于 LoadRunner 所支持的常规 HTTP POST 方法。当前，仅通过 **wap_push_submit** 和 **wap_push_cancel** 支持前两种操作。

您可以使用 **web_submit_data** 函数向 Web 服务器提交数据。但是，使用该函数发送复杂的长数据结构很困难。为了克服此困难并提供更直观的 API 函数，将增加几个新的 API 函数以正确设置 XML 消息数据的格式：**wap_format_si_msg** 和 **wap_format_sl_msg**。有关这些函数的详细信息，请参阅 《LoadRunner Function Reference》。

## MMS 支持

多媒体消息服务 (MMS) 是一个系统应用程序， WAP 客户端可以通过该应用程序提供各种媒体类型的消息传送操作。通过下列 Vuser 函数来实现 MMS 客户端模拟。

详细信息，请参阅 《LoadRunner Function Reference》（“帮助” > “函数参考”）。

| | |
|---|---|
| **wap_mms_msg_add_field** | 向 MMS 消息添加字段 |
| **wap_mms_msg_add_mulitpart_entry** | 向 MMS 消息添加多部分项 |
| **wap_mms_msg_create** | 创建 MMS 消息 |
| **wap_mms_msg_destroy** | 销毁 MMS 消息 |
| **wap_mms_msg_submit** | 向 MMS 发送消息 |

# 64

# 设置无线 Vuser 录制选项

录制无线会话前，可以自定义录制选项。

本章描述：

- 指定录制模式 （仅限于 WAP）
- 指定要录制的信息 （i-mode 和 VoiceXML）
- 指定工具包

## 关于设置录制选项

通过 VuGen，可以录制用户使用其无线接口在网站上执行的常见进程，从而生成无线 Vuser 脚本。

录制前，可以配置录制选项并指定要录制的信息、录制时要使用的工具包以及全局代理服务器设置。

可以设置常见的 Internet 协议录制选项，如代理服务器设置和其他高级设置。详细信息，请参阅第 35 章 “为 Internet 协议设置录制选项”。

# 指定录制模式 （仅限于 WAP）

可以通过 “录制选项”对话框 （“工具” > “录制选项”）中的 “录制模式”设置来定义 WAP Vuser 在录制会话期间由 VuGen 录制的信息。

**要定义 WAP Vuser 的录制信息，请执行下列操作：**

在 “录制模式”部分中选择下列选项之一：

- **WSP**：指示 VuGen 录制工具包或电话与网关之间的所有 WSP 流量。这些操作都被录制为 URL 步骤。输入网关的 IP 地址，并从 WSP 模式框中选择 “无连接”或 “面向连接”。要使用安全 WAP 进行录制，请选中 “启用安全性”复选框。

  为在 WSP 模式下进行录制， VuGen 包含对 Phone.com UP Simulator 4.1 工具包的本机支持。它会检测是否安装了该工具包，并会自动设置其配置参数，然后启动该工具包。对于 Nokia 工具包 （1.3 和 2.0），需要为其分配正确的 IP 地址来手动启动该工具包。详细信息，请参阅第 731 页的 “无线 Vuser 脚本疑难解答”。

- **HTTP**：指示 VuGen 将工具包与 Web 服务器之间的 HTTP 流量录制为 URL 步骤。选中 “将所有 HTTP 请求都录制为自定义请求”复选框，即可通过生成 **web_custom_request** 函数，将所有 HTTP 请求录制为无上下文的自定义 HTTP 请求。

# 指定要录制的信息 （i-mode 和 VoiceXML）

使用 “录制选项”树中的 “录制”节点，可以定义录制会话期间由 VuGen 录制的信息。可以选择 i-mode 录制模式和 VoiceXML 录制模式。

唯一可用的录制模式为 HTTP。该模式指示 VuGen 将工具包与 Web 服务器之间的 HTTP 流量录制为 URL 步骤。

**要定义 i-mode 或 VoiceXML Vuser 的录制信息，请执行下列操作：**

1. 打开 “录制选项”（“工具” > “录制选项”），然后在 “录制选项”树中选择 “Internet 协议 : 录制”节点。
2. 在 “要录制的信息”部分中，选择 “将所有 HTTP 请求都录制为自定义请求”选项，从而通过生成 **web_custom_request** 函数来将所有的 HTTP 请求录制为无上下文的自定义 HTTP 请求。

### i-mode 录制模式

通过该节点，可以指定 i-mode Vuser 的录制模式。

选择哪种录制模式取决于您的需要和环境要求。可用的录制模式有 HTTP 或自定义请求。

**HTTP**：捕获由于用户操作而发送到服务器的所有 HTTP 请求，并为每个请求创建 **web_url** 语句。该录制级别甚至会捕获非 HTML 应用程序（如小程序）和非浏览器应用程序。

**将所有 HTTP 请求都录制为自定义请求**：将所有的 HTTP 请求都录制为自定义 HTTP 请求，从而忽略其上下文。自定义请求不依赖于特定的结构或 HTTP 请求语句。 VuGen 将为每个页面及其所录制的资源生成 **web_custom_request** 函数，而不是 **web_url**、 **web_image** 或 **web_submit_form** 函数。

### VoiceXML 录制模式

通过该节点，可以指定 VoiceXML Vuser 的录制模式。

选择哪种录制模式取决于您的需要和环境要求。可用的录制模式有 HTTP 或自定义请求。

**HTTP**：捕获由于用户操作而发送到服务器的所有 HTTP 请求，并为每个请求创建 **web_url** 语句。该录制级别甚至会捕获非 HTML 应用程序（如小程序）和非浏览器应用程序。

**将所有 HTTP 请求都录制为自定义请求**：将所有的 HTTP 请求都录制为自定义 HTTP 请求，从而忽略其上下文。自定义请求不依赖于特定的结构或 HTTP 请求语句。 VuGen 将为每个页面及其所录制的资源生成 **web_custom_request** 函数，而不是 **web_url**、 **web_image** 或 **web_submit_form** 函数。

# 指定工具包

可以指定在录制无线 Vuser 脚本时 VuGen 应使用的工具包。可以使用 “录制选项”树中的“工具包”节点来指定所需的 VoiceXML 工具包、i-mode 工具包或 VoiceXML 工具包。

可以使用下列选项：

- **使用默认 ... 工具包** – 使用默认工具包进行录制。
- **手动启动 ... 工具包** – 手动启动工具包。
- **使用支持的 ... 工具包** – 使用支持的工具包列表中的工具包。

## WAP 工具包

在该节点中，可以指定录制期间要使用的 WAP 工具包。下面列出了所支持的已安装工具包。

**要指定录制 WAP Vuser 脚本时所使用的工具包，请执行下列操作：**

1 选择“工具”>“录制选项”，然后选择“WAP 工具包（或 i-mode VoiceXML 工具包）”节点。

2 在 “工具包位置”部分中选择下列选项之一：

- **使用默认 WAP（i-mode 或 VoiceXML）工具包**：指示 VuGen 使用录制计算机上的默认工具包 （当前处于禁用状态）。

- **手动启动 WAP（i-mode 或 VoiceXML）工具包**：指示 VuGen 在开始录制时不启动工具包。必须在启动录制会话后手动启动 WAP 工具包。
- **使用支持的 WAP（i-mode 或 VoiceXML）工具包**：指示 VuGen 使用该计算机上安装的特定工具包。选择对话框中列出的某个可用工具包。

### i-mode 工具包

在该节点中，可以指定录制期间要使用的 i-mode 工具包。下面列出了所支持的已安装工具包。

**要指定录制 i-mode Vuser 脚本时所使用的工具包，请执行下列操作：**

1 选择“工具”>“录制选项”，然后选择“i-mode 工具包”节点。

2 在“工具包位置”部分中选择下列选项之一：

- **使用默认 i-mode 工具包**：指示 VuGen 使用录制计算机上的默认工具包（当前处于禁用状态）。
- **手动启动 i-mode 工具包**：指示 VuGen 在开始录制时不启动工具包。必须在启动录制会话后手动启动 WAP 工具包。
- **使用默认 i-mode 工具包**：指示 VuGen 使用该计算机上安装的特定工具包。选择对话框中列出的某个可用工具包。

### VoiceXML 工具包

在该节点中，可以指定录制期间要使用的 VoiceXML 工具包。下面列出了所支持的已安装工具包。

**要指定录制 VoiceXML Vuser 脚本时所使用的工具包，请执行下列操作：**

1 选择“工具”>“录制选项”，然后选择“VoiceXML 工具包”节点。

2 在“工具包位置”部分中选择下列选项之一：

- **使用默认 VoiceXML 工具包**：指示 VuGen 使用录制计算机上的默认工具包（当前处于禁用状态）。
- **手动启动 VoiceXML 工具包**：指示 VuGen 在开始录制时不启动工具包。必须在启动录制会话后手动启动 WAP 工具包。
- **使用支持的 VoiceXML 工具包**：指示 VuGen 使用该计算机上安装的特定工具包。选择对话框中列出的某个可用工具包。

# 65

# 配置 WAP 运行时设置

录制 WAP Vuser 脚本之后，可以配置 WAP 特定的运行时设置。

本章描述：

- 配置网关选项
- 配置载荷信息
- 配置 RADIUS 连接数据

有关设置 WAP 通用 Internet 协议运行时设置和所有无线协议的信息，请参阅第 37 章 “配置 Internet 运行时设置”。

## 关于 WAP 运行时设置

开发 WAP Vuser 脚本之后，可以设置 WAP 特定的运行时设置。通过这些设置，可以控制 WAP Vuser 的行为，以便它们能够在 WAP 设备上准确模拟实际用户。可以在网关、 Radius 和载荷设置区域中配置 WAP 运行时设置。

可以在 “运行时设置”对话框中设置 WAP 运行时设置。单击适当的选项卡，查看并指定所需的设置。

要显示 “运行时设置”对话框，请单击 VuGen 工具栏上的 “运行时设置”按钮。还可以通过单击 “设计”选项卡中的 “运行时设置”按钮，从 LoadRunner Controller 中修改运行时设置。

本章讨论 WAP Vuser 的网关运行时设置。有关适用于所有无线 Vuser 的常规运行时设置信息，请参阅第 37 章 “配置 Internet 运行时设置”。

# 配置网关选项

可以使用 “运行时设置”树中的 “WAP: 网关”节点来设置网关设置：

### 通信协议

仅当您希望使用 WSP 协议运行 Vuser，并通过 WAP 网关访问 Web 服务器 （通过选中 “通过 WAP 网关回放”选项）时，该网关设置才有意义。如果通过网关运行脚本，则必须指定 IP 和端口地址。

如果正在 HTTP 模式中运行 Vuser，并直接访问 Web 服务器 （通过清除 “通过 WAP 网关回放”选项），那么该网关设置无意义。

### 设置

**IP**：指定网关的 IP 地址。

**端口**：指定网关的端口。通过 WAP 网关运行 Vuser 时， VuGen 将根据选定的模式自动设置默认的端口号。但也可以自定义该设置并为网关指定自定义的 IP 地址和端口。

**高级**：打开 “高级网关选项”对话框，可以在其中设置客户端功能及其他高级网关选项。

### 连接设置

在本部分中，可以指定所需的回放连接模式。

- **面向连接**：将 WSP 会话的连接模式设置为 “面向连接”。
- **无连接**：将 WSP 会话的连接模式设置为 “无连接”。
- **启用安全性**：启用与 WAP 网关的安全连接。

### 实际电话模拟

通过 VuGen，可以指定用于回放 Vuser 的电话设备类型。可以从常见供应商的电话列表中进行选择。 VuGen 将为选定的电话确定正确的客户端标头，并相应地进行模拟。

- **模拟实际电话**：指示 VuGen 模拟实际电话。
- **电话型号**：从该菜单中选择要模拟的电话型号。

注意，在启用实际电话模拟时，所有高级网关选项将被忽略。相反，它从 Vugen 配置文件中检索标头和客户端功能信息，该配置文件定义了所支持的每一种电话。

如果需要对几种不同的电话进行测试，实际电话模拟会特别有用。例如，可以为 *Motorola Timeport* 录制脚本，然后在 *Nokia 6110* 上进行回放。使用实际电话模拟回放脚本时，脚本中的所有 **wap_set_capability** 和 **wap_add_const_header** 函数将被忽略。它从配置文件中检索所有必要的信息，该配置文件定义了每个电话的标头。

如果要模拟的电话不在列表中，则可以将其手动添加到配置文件 （LoadRunner 的 *dat* 目录中的 *LrwWapPhoneDB.dat* 文件）中，从而将其添加到 “运行时设置”界面上。详细信息，请参阅配置文件开头的注释。

### 高级网关选项

在“网关”节点中单击“高级”时， VuGen 将打开 “功能”对话框。在该对话框中，可以配置 WAP 功能及其他高级网关选项：

- **服务器 SDU 大小** – 会话期间可发送到服务器的最大事务服务数据单元 （默认值为 4000）。
- **客户端 SDU 大小** – 会话期间可发送到客户端的最大事务服务数据单元 （默认值为 4000）。
- **确认标头** – 返回为网关提供信息的标准标头 （默认情况下为禁用）。
- **推操作支持** – 启用通过网关的推操作类型信息 （默认情况下为禁用）。
- **确认推操作支持** – 在 CO 模式中，如果接收到推操作消息，该选项将指示 Vuser 确认是否收到该消息 （默认情况下为禁用）。
- **检索消息** – 接收到推操作消息时，该选项将指示 Vuser 从该推操作消息指定的 URL 中检索消息数据 （默认情况下为禁用）。
- **支持 Cookie** – 为保存和检索 cookie 提供支持 （默认情况下为禁用）。

### 设置网关选项

以下部分描述设置 WAP 网关选项的过程。

**要设置 WAP 网关选项，请执行下列操作：**

1 单击 “运行时设置”按钮或选择 “Vuser” > “运行时设置”以显示 “运行时设置”对话框。选择 “WAP: 网关”节点。

2 要在 WSP （而不是 HTTP）模式下回放脚本，请选择 “通过 WAP 网关回放”。

3 指定网关的 IP 地址和端口。也可以使用由 VuGen 指定的默认端口。

4 选择连接模式 – “面向连接”或 “无连接”。要指定安全连接模式，请选择 “使用安全连接”选项。

5 要模拟常见的电话，请选择 “模拟实际电话”，并从下拉列表中选择所需的电话。

6 如果不模拟常见电话，请单击 “高级”，设置客户端功能及其他高级网关选项。

- 输入 “服务器 SDU”和 “客户端 SDU”的值。
- 要指示 Vuser 检索确认标头，请选择 “确认标头”选项。
- 要支持推操作消息，请在 “推操作支持”旁边的列中选择 “True”。
- 要确认推操作消息，请在 “确认推操作支持”旁边的列中选择 “True”。
- 要从推操作消息 URL 中检索数据，请在 “检索消息”旁边的列中选择 “True”。
- 要启用 cookie，请在 “支持 Cookie”旁边的列中选择 “True”。

# 配置载荷信息

基本载荷是一种数据传输机制，用于在两台设备之间传输 WDP 协议。基本载荷的示例包括 SMS、 UDP、 CSD、 GSM、 GPRS 和包数据。

LoadRunner 支持 UDP 和 SMS 载荷。可以在 “运行时设置”中指定初始载荷。回放期间，可以使用 **wap_set_bearer** 函数在载荷之间切换。注意，如果要同时使用这两种载荷，则必须在回放前先在运行时设置中启用它们。

要使用 SMS-CIMD2 载荷，必须连接到短消息系统中心 (SMSC) 并提供登录信息。可以在 “运行时设置”内的 “WAP: 载荷”节点中定义端口信息。

可以使用 **wap_set_sms_user** API 函数或通过 “运行时设置”来设置 SMS 登录信息。通过函数设置登录信息的优点在于，可以使用参数来运行带有多个值的脚本。 API 函数中的值将覆盖运行时设置。可以在 “运行时设置”内的 “载荷”节点中设置载荷属性：

| 载荷设置 | 描述 |
| --- | --- |
| **启用 UDP 载荷** | 打开与 UDP 载荷的连接 |
| **启用 SMS-CIMD2 载荷** | 打开与 CIMD2 载荷的连接 |
| **载荷类型** | 默认载荷类型 （UDP 或 CIMD2） |
| **CIMD2：IP 地址** | SMSC 服务器的 IP 地址 |
| **CIMD2：端口号** | SMSC 服务器的端口号 |
| **CIMD2：网关 ID** | SMSC 中定义的 WAP 网关应用程序 ID |
| **CIMD2：用户名** | 用于登录到服务器的用户名 |
| **CIMD2：用户密码** | 该用户的密码 |
| **CIMD2：起始地址** | 用户起始地址 |

**要设置 WAP 载荷选项，请执行下列操作：**

**1** 单击 “运行时设置”按钮或选择 “Vuser” > “运行时设置”以显示 “运行时设置”对话框。选择 “载荷”节点。

**2** 选择 “True”以启用所需的载荷：**UDP** 或 **SMS-CIMD2**。

**3** 从右列中选择起始 “载荷类型” - UDP 或 SMS-CIMD2。

**4** 以点分形式输入 SMSC IP 地址。

**5** 输入 SMSC 端口号。

**6** 输入 SMSC 网关 ID，而非 SMS 网关 ID。

**7** 输入 SMSC 用户名。

**8** 输入 SMSC 用户密码。

**9** 输入 SMSC 起始地址。

**10** 单击 “确定”接受设置并关闭对话框。

# 配置 RADIUS 连接数据

RADIUS（远程身份验证拨入用户服务）是一个客户端 / 服务器的协议和软件，远程访问服务器可以通过它与中央服务器进行通信，以便对拨入的用户进行身份验证，并授予这些用户访问所要求的系统或服务的权限。

可以在 “运行时设置”内的 “Radius”节点中提供拨入信息：

| 属性 | 值 |
|---|---|
| 网络类型 | 记帐网络类型：GPRS （通用分组无线业务）或 CSD （电路交换数据） |
| IP 地址 | Radius 服务器的 IP 地址 |
| 身份验证端口号 | Radius 服务器的身份验证端口 |
| 记帐端口号 | Radius 服务器的记帐端口 |
| 密钥 | Radius 服务器的密钥 |

**要设置 WAP Radius 选项，请执行下列操作：**

**1** 单击 “运行时设置”按钮或选择 “Vuser” > “运行时设置”以显示 “运行时设置”对话框。单击 “Radius”节点。

**2** 在 “Radius 身份验证”部分中，选择 “True”可启用身份验证，或选择 “False”可禁用身份验证。

3 以点分形式输入 Radius 服务器的 IP 地址。

4 输入 Radius 服务器的身份验证端口号和记帐端口号。

5 键入 Radius 或记帐身份验证的密钥。

6 输入用于登录到 Radius 服务器的用户名。

7 输入该用户名的用户密码。

8 输入 **MSISDN** 标识号。

9 选择记帐网络类型：GPRS （通用分组无线业务）或 CSD （电路交换数据）。

10 单击 “确定”接受设置并关闭对话框。

# 第 XVI 部分

## GUI Vuser 脚本

# 66

# 开发 GUI Vuser 脚本

GUI Vuser 用于在基于 Windows 的环境中操作图形用户界面 (GUI) 应用程序。使用 WinRunner 可以创建 GUI Vuser 脚本， WinRunner 是 Mercury Interactive 用于 GUI 应用程序的自动化测试工具。

本章描述：

- GUI Vuser 简介
- 了解 GUI Vuser 技术
- GUI Vuser 入门
- 使用 WinRunner 创建 GUI Vuser 脚本
- 度量服务器性能：事务
- 生成较重的用户负载：集合点
- 了解 GUI Vuser 脚本
- 在 GUI Vuser 脚本中使用 Vuser 函数
- 向 Controller 发送消息
- 获取关于 Vuser 和负载生成器的信息

***下列信息仅适用于 GUI Vuser 脚本。***

# 关于开发 GUI Vuser 脚本

通过 GUI Vuser 可以在客户端 / 服务器系统处于负载下时，度量和监视端到端用户响应时间。 GUI Vuser 模拟实际用户的整个环境。例如，实际用户坐在计算机旁，使用键盘和鼠标操作应用程序，读取计算机显示器上的信息。同样地， GUI Vuser 在自己的计算机上运行并操作应用程序。可以编写程序使 Vuser 阅读和操作在其计算机显示器上显示的信息。

*GUI Vuser 脚本*

在 GUI Vuser 脚本中描述了每个 GUI Vuser 的操作。使用 WinRunner 可以创建 GUI Vuser 脚本。 WinRunner 是一种自动化 GUI 测试工具，通过它可以创建、编辑和调试 GUI Vuser 脚本。

GUI Vuser 脚本是使用 Mercury Interactive 的测试脚本语言 (TSL) 创建的。TSL 是一种易于使用的高级编程语言，与 C 语言类似。它将传统编程语言的功能和灵活性，与设计用于测试客户端 / 服务器系统的功能结合起来。

录制基本的 Vuser 脚本以后，可以向该脚本中插入用于度量服务器性能 （事务）的语句，并确保模拟特定用户负载 （集合点）。有关 GUI Vuser 的详细信息，请参阅 《LoadRunner Controller 用户指南》。

# GUI Vuser 简介

假设您有一台为许多自动取款机 (ATM) 提供服务的银行服务器。可以创建一个实现下列功能的 GUI Vuser 脚本：

- 打开 ATM 应用程序
- 输入帐号
- 输入要提取的金额
- 从帐户提取现金
- 核对帐户余额
- 关闭 ATM 应用程序
- 重复此过程

仅用 Controller 监视和管理 GUI Vuser。例如，可以使用 Controller 运行、暂停或查看 Vuser，以及监视场景状态。

# 了解 GUI Vuser 技术

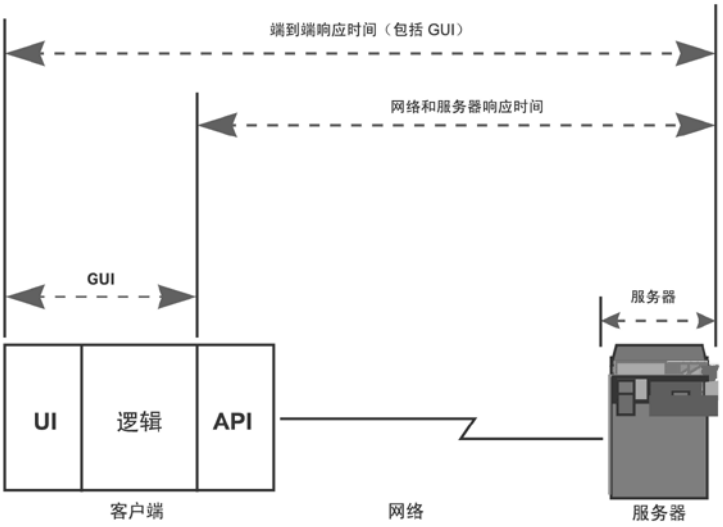
本部分概述如何创建 Vuser 脚本。有关创建 Vuser 脚本的详细信息，请参阅《WinRunner User's Guide》。

---

**注意**：不能使用 VuGen 运行 GUI Vuser 脚本。使用 Controller 可以将 GUI Vuser 脚本作为场景的一部分运行；使用 WinRunner 可以在独立模式中运行 GUI Vuser 脚本。

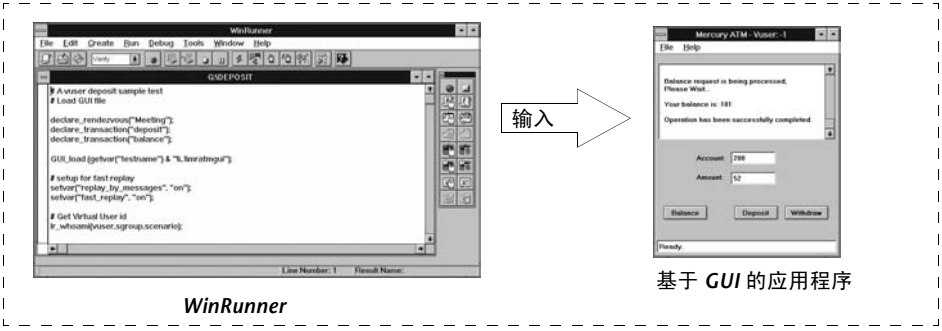
---

GUI Vuser 度量实际的端到端响应时间。端到端响应时间表示用户提交请求以后等待响应的总时间。端到端响应时间包括 GUI 响应时间以及网络和服务器响应时间。



### GUI Vuser

GUI Vuser 由一个 WinRunner 的副本 （WinRunner 是 Mercury Interactive 用于基于 Windows 的应用程序的 GUI 测试工具）和一个基于 GUI 的应用程序组成。




*GUI Vuser*

WinRunner 可以代替实际用户并操作应用程序。例如，WinRunner 可以从菜单选择命令，单击图标或键入文本。 WinRunner 可以操作任何访问服务器的应用程序。

## GUI Vuser 入门

本部分概述如何创建 GUI Vuser 脚本并将其集成到场景中。

**1 使用 WinRunner 创建 GUI Vuser 脚本。**

使用 WinRunner 在基于 GUI 的应用程序中录制键盘和鼠标操作。详细信息，请参阅 《WinRunner User's Guide》。

**2 编写其他 TSL 语句并将其添加到脚本。**

使用循环和其他控制流结构增强 Vuser 脚本。有关编辑脚本的详细信息，请参阅《WinRunner User's Guide》。有关 TSL 的详细信息，请参阅 《TSL Online Reference》。

**插入事务以度量系统性能。**

通过将事务插入 Vuser 脚本来度量服务器性能。

**插入集合点以在服务器上生成密集的用户负载。**

集合点指示多个 Vuser 在同一时刻执行任务。

3 **运行 Vuser 脚本。**

运行脚本以确保它能正常运行。如有必要，请调试该脚本。详细信息，请参阅《WinRunner User's Guide》。

创建 GUI Vuser 脚本后，需要将其集成到场景中。有关将 Vuser 脚本集成到场景中的详细信息，请参阅 《LoadRunner Controller 用户指南》。

# 使用 WinRunner 创建 GUI Vuser 脚本

使用 WinRunner 可以创建 GUI Vuser 脚本。使用 WinRunner 录制基本的 Vuser 脚本后，可以手动进行下列操作：

- 将事务语句插入脚本以度量服务器的性能。详细信息，请参阅第 762 页的“度量服务器性能：事务”。
- 将集合语句插入脚本以确保模拟特定用户负载。详细信息，请参阅第 762 页的 “生成较重的用户负载：集合点”。

## 创建 GUI Vuser 脚本

WinRunner 是一个用于创建、编辑和调试基于 Windows 的 GUI Vuser 脚本的完整开发环境。使用 WinRunner 可以录制实际用户在应用程序上的操作。例如，可以录制用户将帐号输入 ATM，然后提取五十美元。这些操作被自动转录到使用 Mercury Interactive 的测试脚本语言 (TSL) 编写的脚本。

## 度量服务器性能：事务

事务

可以定义事务以度量服务器的性能。每个事务度量服务器响应指定 Vuser 请求所用的时间。这些请求可以是简单任务 （如等待对单个查询的响应），也可以是复杂任务 （如提交多个查询和生成报告）。

要度量事务，需要插入 Vuser 函数以标记任务的开始和结束位置。在脚本内，可以标记的事务不受数量限制，每个事务的名称都不同，开始和结束位置也不同。

在场景执行期间， Controller 将度量执行每个事务所用的时间。例如，可以定义一个事务，用于度量银行服务器处理 Vuser 查看帐户余额的请求的时间。场景运行后，可使用 LoadRunner 的图和报告来分析每个事务的服务器性能。

定义事务后，即可使用该事务来度量服务器在不同负载下的性能情况。例如，可以度量服务器在单个用户、一百个用户或者一千个用户的负载下的性能情况。在场景运行的同时， LoadRunner 积累事务的性能数据。之后，您可以使用该信息生成性能分析报告和性能分析图。

**要标记事务的开始，请执行下列操作：**

将 **start_transaction** 语句插入 Vuser 脚本。

**要标记事务的结束，请执行下列操作：**

将 **end_transaction** 语句插入 Vuser 脚本。

有关 **start_transaction** 和 **end_transaction** 函数的语法，请参阅《TSL Online Reference》（可以从 WinRunner 帮助菜单获得）。

## 生成较重的用户负载：集合点

集合点

为了模拟较重的用户负载并度量服务器性能，需要同步各个 Vuser 以便在同一时刻执行查询。通过创建会合位置 （称为集合点），可以确保多个 Vuser 同步操作。当某个 Vuser 到达该集合点时， Controller 会将其保留，直到参与该集合的全部 Vuser 都到达。当满足集合条件时， Controller 将释放 Vuser。

可通过将集合点插入到 Vuser 脚本来指定会合位置。在 Vuser 执行脚本并遇到集合点时，脚本将暂停执行， Vuser 将等待 Controller 的允许以继续执行。 Vuser 从集合被释放后，将执行脚本中的下一个任务。

例如，要生成银行服务器的峰值负载，可以将集合点插入 Vuser 脚本，指示所有 Vuser 同时向他们的帐户存入现金。

**要插入集合点，请执行下列操作：**

将 **rendezvous** 语句插入 Vuser 脚本。

有关 **rendezvous** 函数的语法，请参阅 《TSL Online Reference》（可以从 WinRunner 帮助菜单获得）。

## 了解 GUI Vuser 脚本

GUI Vuser 脚本是使用 Mercury Interactive 的测试脚本语言 TSL 编写的。 TSL 是一种易于使用的高级编程语言，与 C 语言类似。它将传统编程语言的功能和灵活性，与专门设计用于测试的功能结合起来。有关 TSL 的详细信息，请参阅 《TSL Online Reference》。

本部分介绍在 WinRunner 中创建的一个简单 Vuser 脚本。注意，该脚本不能在 Unix 计算机上运行。该脚本启动 ATM 应用程序 (mratm.exe)，输入帐号，存入五十美元，然后关闭应用程序。

该脚本的第一部分启动一个应用程序，并将其移到屏幕上的新位置。 **system** 函数启动 ATM 应用程序。 **win_move** 函数将 ATM 应用程序移到屏幕上的指定位置。

```
# 初始化并调用 ATM 客户端应用程序。
system ("mratm.exe");
win_move ("Mercury ATM", 325, 0);
```

然后， Vuser 将帐号输入 ATM。 **set_window** 函数激活 ATM 窗口。 **edit_set** 函数指示 Vuser 将帐号输入 ATM 的 “帐户” 字段。

```
# 在 “帐户” 字段键入帐号。
account = 100;
set_window ("Mercury ATM");
edit_set ("Account", account);
```

输入帐号以后， Vuser 输入要存入的数量并按 “存入”按钮。 **edit_set** 函数在 “数量”字段输入要存入的数量。 **button_press** 函数告知 Vuser 按 ATM 的 “存入”按钮。

```
# 在 “数量”字段输入要存入的数量。
amount = 50;
set_window ("Mercury ATM");
edit_set ("Account", account);
# 按 “存入”按钮。
button_press ("Deposit");
```

该测试的最后一部分告知 Vuser 关闭 ATM 应用程序。 **menu_select_item** 函数从 “文件”菜单选择 “退出”命令。

```
# 关闭客户端应用程序。
menu_select_item ("File; Exit");
```

## 在 GUI Vuser 脚本中使用 Vuser 函数

本部分列举 Vuser 函数，可以使用这些函数增强您的 GUI Vuser 脚本。有关这些函数的语法和示例，请参阅下一部分或 《TSL Online Reference》（可以从 WinRunner 帮助菜单获得）。

| | |
|---|---|
| **declare_rendezvous** | 声明一个集合 |
| **declare_transaction** | 声明一个事务 |
| **end_transaction** | 标记事务的结束以用于性能分析 |
| **error_message** | 向 Controller 发送错误消息 |
| **get_host_name** | 返回负载生成器的名称 |
| **get_master_host_name** | 返回 Controller 负载生成器的名称 |
| **lr_whoami** | 返回关于 Vuser 执行脚本的信息 |
| **output_message** | 向 Controller 发送消息 |
| **rendezvous** | 在 Vuser 脚本中设置集合点 |

| | |
|---|---|
| **start_transaction** | 标记事务的开始以用于性能分析 |
| **user_data_point** | 录制用户定义的数据示例 |

## 向 Controller 发送消息

运行场景时，Controller 的输出窗口将显示有关脚本执行的有价值的信息。除了 WinRunner 自动发送的消息，还可以将语句包括进向 Controller 发送错误和通知消息的每个脚本。例如，可以插入显示应用程序当前状态的消息。场景执行以后，可以将这些消息保存到文件。

**error_message** 函数向 Controller 的输出窗口发送错误消息。该函数的语法是：

**error_message (***message***);**

其中 *message* 是一个文本字符串。在下例中，脚本执行期间出现致命错误时 Vuser 脚本将发送一条消息。

```
if (fatal_error < 0){
      mess = sprintf ("fatal error - Exiting.");
      error_message (mess);
      texit (1);
}
```

**output_message** 函数用于发送非错误消息的特别通知。该函数的语法是：

**output_message (***message***);**

其中 *message* 是一个文本字符串。

有关 **error_message** 和 **output_message** 函数的详细信息，请参阅 《TSL Online Reference》（可以从 WinRunner 帮助菜单获得）。

# 获取关于 Vuser 和负载生成器的信息

场景执行期间，可以获取下列项的标识：

- 在场景中的特定时刻执行任务的 Vuser
- 执行脚本的负载生成器
- 运行 Controller 的计算机

例如，可以编写语句并将其添加到 Vuser 脚本，以便返回当前使用应用程序的每个活动 Vuser 的 ID，并将该信息打印到文件。

下列函数包含有关 Vuser 和负载生成器的信息：

| | |
|---|---|
| **lr_whoami** | 返回 Vuser 及其所属 Vuser 组的名称 |
| **get_host_name** | 返回执行脚本的计算机的名称 |
| **get_master_host_name** | 返回运行 Controller 的计算机名 |

在下例中， **get_host_name** 函数返回当前运行脚本的负载生成器的名称。**print** 语句将该信息保存到文件。

```
my_host_name = get_host_name();
print("my local load generator name is:"& my_host_name) > vuser_file;
```

有关这些函数的详细信息，请参阅 《TSL Online Reference》（可以从 WinRunner 帮助菜单获得）。

# 第 XVII 部分

## 高级用户信息

# 67

# 在 Visual Studio 中创建 Vuser 脚本

可以使用 Visual C 或 Visual Basic 在 Visual Studio 中创建 Vuser 脚本模板。可以像编译常规 C 或 Visual Basic 程序一样对其进行编译。

本章描述：

- 用 Visual C 创建 Vuser 脚本
- 用 Visual Basic 创建 Vuser 脚本
- 配置运行时设置和参数

## 关于在 Visual Studio 中创建 Vuser 脚本

可以使用多种方法创建 Vuser 脚本：通过 VuGen，或通过开发环境，如 Visual Studio。

| | |
|---|---|
| *VuGen* | 可以通过录制或者在 VuGen 编辑器中手动编程，使用 VuGen 来创建在 Windows 或 UNIX 平台上运行的 Vuser 脚本。在 Windows 环境中创建脚本，并在 Windows 或 UNIX 中运行脚本 – UNIX 中不支持录制 |
| *Visual Studio* | 对于使用 Visual Studio 的用户，可以在 Visual Basic、C 或 C++ 中编写程序。必须将这些程序编译到动态链接库 (dll) 中 |

本附录描述如何通过在 Visual C 和 Visual Basic 环境中编程来开发 Vuser 脚本。在这些环境中，可以在开发应用程序内开发 Vuser 脚本，同时导入 LoadRunner 库。

还可以从 VuGen 编辑器中编写 Vuser 脚本，同时集成应用程序的库或类。可以在 VuGen 中使用 C、Java、Visual Basic、VBScript 和 JavaScript 编程。详细信息，请参阅第 23 章 “创建自定义 Vuser 脚本”。

要通过编程创建 Vuser 脚本，可以使用 LoadRunner 模板作为较大型 Vuser 脚本的基础。该模板提供：

➤ 正确的程序结构

➤ LoadRunner API 调用

➤ 创建动态库的源代码和生成文件

从模板中创建基本 Vuser 脚本后，可以增强脚本以提供运行时信息和统计信息。详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

Vuser 脚本中使用的公用函数的联机 C 引用包含在 《LoadRunner Function Reference》中 （“帮助” > “函数参考”）。

## 用 Visual C 创建 Vuser 脚本

请注意，您可以使用 Visual C 版本 6.0 或更高版本创建 Vuser 脚本。

**要用 Visual C 创建 Vuser 脚本，请执行下列操作：**

1 在 Visual C 中，创建新项目 – 动态链接库 (dll)。选择 “文件” > “新建” 并单击 “项目” 选项卡。

2 在向导中选择 *empty dll*。

3 向项目中添加下列文件：

➤ 带有 3 个导出函数的新 *cpp* 文件：*init*、 *run*、 *end* （名称可以自定义）。

➤ 库：lrun50.lib （位于 <lr installation dir>/lib 中）。

4 在项目设置中更改下列内容：

➤ 选择 “C/C++” 选项卡，然后选择 “代码生成”（类别） > “使用运行时库”（列表）。将其更改为：**多线程 dll**。

➤ 选择 “C/C++” 选项卡，然后选择 “预处理器”（类别） > “预处理器定义”（编辑字段） Remove _DEBUG。

5 从客户端应用程序中添加代码，或按正常方式编程。

6 用常规 LoadRunner 函数增强脚本。例如，使用 **lr_output_message** 发布消息，使用 **lr_start_transaction** 标记事务，等等。详细信息，请参阅 《LoadRunner Function Reference》中的常规函数 （“帮助” > “函数参考”）。

7 构建项目。输出将为 DLL。

8 用与 DLL 相同的名称创建一个目录，然后将 DLL 复制到该目录中。

9 在 *Template* 目录下的 **lrvuser.usr** 文件中，用 DLL 名更新 USR 文件密钥 *BinVuser*：BinVuser=<*DLL_name*>。

以下示例中， **lr_output_messsage** 函数发出消息，指出正在执行哪个部分。**lr_eval_string** 函数用于检索用户名。要使用以下示例，请验证 LoadRunner 包含文件 *lrun.h* 的路径是否正确。

```
#include "c:\mercury\lrun_5\include\lrun.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
```

# 用 Visual Basic 创建 Vuser 脚本

**要在 Visual Basic 中创建 Vuser，请执行下列操作：**

1. 在 Microsoft Visual Basic 中，新建项目。选择 “文件” > “新建项目”。
2. 选择 “LoadRunner 虚拟用户”。将用一个类和模板为 Vuser 新建一个项目。
3. 请在继续编程之前保存该项目。选择 “文件” > “保存项目”。
4. 打开 “对象浏览器”（“视图”菜单）。选择 *LoadRunner Vuser* 库，然后双击 Vuser 类模块以打开模板。模板包含三个部分：Vuser_Init、 Vuser_Run 和 Vuser_End。

```
Option Explicit

Implements Vuser

Private Sub Vuser_Init()
'Implement the Vuser initialization code here
End Sub

Private Sub Vuser_Run()
'Implement the Vuser main Action code here
End Sub

Private Sub Vuser_End()
'Implement the Vuser termination code here
End Sub
```

5. 从客户端应用程序中添加代码，或按正常方式编程。
6. 使用 “对象浏览器”向代码中添加所需的 VuGen 元素，例如事务、思考时间、集合点和消息。
7. 用运行时设置和参数增强程序。详细信息，请参阅第 773 页的 “配置运行时设置和参数”。
8. 构建 Vuser 脚本：选择 “文件” > “生成 *project_name*.dll”。

   该项目将以 LoadRunner Vuser 脚本 (.usr) 的形式保存。脚本将驻留在与项目相同的目录中。

## 配置运行时设置和参数

创建脚本的 DLL 之后，需要创建脚本 (*.usr*) 并配置其设置。通过随 VuGen 一起提供的 *lrbin.bat* 实用程序，可以为使用 Visual C 和 Visual Basic 创建的脚本定义参数以及配置运行时设置。该实用程序位于 LoadRunner 安装的 *bin* 目录中。

**要配置运行时设置并参数化脚本，请执行下列操作：**

**1** 在 LoadRunner *bin* 目录中，双击 *lrbin.bat*。将打开 “独立 Vuser 配置”对话框。

**2** 选择 “文件” > “新建”。指定 *usr* 文件的脚本名。脚本名必须与用于保存 DLL 的目录名相同。

**3** 选择 “Vuser” > “高级”，然后在 “高级”对话框中输入 DLL 名。

**4** 选择 “Vuser” > “运行时设置”以定义运行时设置。“运行时设置”对话框与 VuGen 界面中显示的对话框相同。详细信息，请参阅第 9 章 “配置运行时设置”。

**5** 选择 “Vuser” > “参数列表”以定义脚本的参数。“参数”对话框与 VuGen 中的相应对话框相同。详细信息，请参阅第 7 章 “定义参数”。

在独立模式下运行脚本来对其进行测试。选择 “Vuser” > “运行 Vuser”。运行脚本时，将显示 Vuser 执行窗口。

**6** 选择 “文件” > “退出”以关闭配置实用程序。

# 68

# 通过 XML API 编程

可以创建支持完整 XML 结构的 Vuser 脚本。 LoadRunner 提供了可用来查询和操纵 XML 数据的函数。

本章描述：

- 了解 XML 文档
- 使用 XML 函数
- 指定 XML 函数参数
- 使用 XML 属性
- 构建 XML 脚本
- 增强已录制的会话

***下列信息主要适用于 Web、 SOAP 和 无线 Vuser 脚本。***

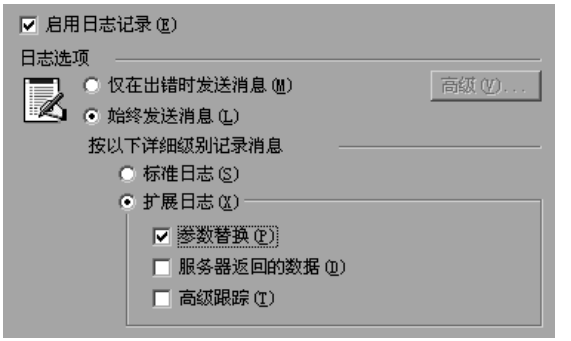
## 关于通过 XML API 编程

通过 VuGen 对 XML 的支持可以在测试执行期间动态地使用 XML 代码和检索值。请按照下列步骤创建有效的 XML 脚本：

- 使用所需的协议 （通常是 Web、 SOAP 或无线）录制脚本。
- 将 XML 结构复制到脚本中。
- 从 LR API 中添加 XML 函数，以便检索动态数据和 XML 元素值。

  LR API 使用 Xpath （XML Path 语言）来操纵 XML 文档中的文本。

可以指示 VuGen 使用运行时设置将 XML 元素的输出值显示在 “执行日志”窗口中。 VuGen 将显示行数、匹配项数和值。要显示值，需要启用参数替换。在“运行时设置”中，打开 “常规 : 日志”节点，选择 “扩展日志”，然后选择“参数替换”。详细信息，请参阅第 9 章 “配置运行时设置”。




所有 LoadRunner XML 函数都将返回成功找到的匹配项数，如果失败则返回 0。

# 了解 XML 文档

XML （可扩展标记语言）是可用于创建自己的自定义标记的标记语言。使用这些标记时，可对标记之间的文本赋予意义。这与标准 HTML 标记 （如 H1、 P、 DIV 等等）相反，后者无法进行自定义，且不能表明文本的内容。

XML 文档由具有很多节点和分支的树组成。有三个描述 XML 文档组成部分的常用术语：标记、元素和属性。以下示例阐述了这些术语：

```
<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>
```

“标记”是左右尖括号之间的文本。 <acme_org>、 <employee> 和 <name> 都是标记示例。存在起始标记 （如 <name>）和结束标记 （如 </name>）。上面的 XML 片段描述了 Acme 公司和两个员工 John Smith 和 Sue Jones。

“元素”是指起始标记、结束标记和这两个标记之间的所有内容。在上面的示例中， <employee> 元素包含三个子元素：<name>、 <cubicle> 和 <extension>。

“属性”是元素起始标记内的 “名称 – 值”对。在本示例中， **type='PT'** 是 <employee> 元素的属性；

在上面的示例中，标记 *name* 是 *employee* 的元素。每个元素都有一个值。 *name* 元素值的一个示例是字符串 “John Smith”。

## 使用 XML 函数

下面的部分将提供有关如何使用 XML 树中数据的示例。通过某些函数可以检索信息，而通过另一些函数可以将信息写入 XML 树。这些示例使用了以下 XML 树，该树中包含 *Acme* 公司中多个员工的姓名和分机号。

```
<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>
```

### 读取 XML 树中的信息

从 XML 树中读取信息的函数是：

| | |
|---|---|
| **lr_xml_extract** | 从 XML 字符串中提取 XML 字符串片段 |
| **lr_xml_find** | 对 XML 字符串执行查询 |
| **lr_xml_get_values** | 检索通过查询找到的 XML 元素的值 |

要通过查询检索特定值，需要以路径格式指定父节点和子节点的标记。

例如，要在财务部门中检索员工姓名，请使用以下字符串：

```
lr_xml_get_values("XML={XML_Input_Param}",
"ValueParam=OutputParam",
   "Query=/acme_org/accounting_dept/employee/name",
LAST);
```

“执行日志”窗口 （“扩展日志记录”已启用）显示了该函数的输出：

```
输出：
Action.c(20):"lr_xml_get_values" was successful, 1 match processed
Action.c(25):Query result = John Smith
```

### 写入到 XML 结构

将值写入到 XML 树的函数是：

| | |
|---|---|
| **lr_xml_delete** | 删除 XML 字符串中的片段 |
| **lr_xml_insert** | 将新的 XML 片段插入到 XML 字符串中 |
| **lr_xml_replace** | 替换 XML 字符串的片段 |
| **lr_xml_set_values** | 设置通过查询找到的 XML 元素的值 |
| **lr_xml_transform** | 将可扩展样式表语言 (XSL) 转换应用到 XML 数据 |

最常用的写入函数是 **lr_xml_set_values**，该函数可设置 XML 字符串中指定元素的值。以下示例使用 **lr_xml_set_values** 来更改 XML 字符串中两个 *employee* 元素的电话分机号。

首先，将 XML 字符串保存到名为 *XML_Input_Param* 的参数中。我们希望对两个值进行匹配和替换，因此准备了两个新的参数 *ExtensionParam_1* 和 *ExtensionParam_2*，并将它们的值设置为两个新的电话分机号 1111 和 2222。

**lr_xml_set_values** 包含参数 “ValueName=ExtensionParam”，它将获取 *ExtensionParam_1* 和 *ExtensionParam_2* 的值。这些参数的值 1111 和 2222 将替换两个员工的当前分机号。然后将求 *OutputParam* 的值，以证明确实替换成了新的电话分机号。

```
Action() {

    int i, NumOfValues;
    char buf[64];

    lr_save_string(xml_input, "XML_Input_Param"); // 将输入保存为参数
    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++) {/* 打印 MultiParam 的多个值 */

         sprintf(buf, "Retrieved value %d :{OutputParam_%d}", i+1, i+1);
         lr_output_message(lr_eval_string(buf));
    }

    return 0;
}
```

输出：

```
Action.c(40):Retrieved value 1: 1111
Action.c(40):Retrieved value 2: 2222
```

# 指定 XML 函数参数

大多数 XML API 函数都要求指定 **XML 元素**和**查询**。您还可以指明您希望检索所有结果还是单个结果。

### 定义 XML 元素

为了定义要查询的 XML 元素，可以指定 XML 元素的字面字符串或包含 XML 数据的参数。以下示例显示了定义为字面字符串的 XML 输入字符串：

“XML=<employee>JohnSmith</employee>”

或者，“XML”字符串可以是包含 XML 数据的参数。例如：

“XML={EmployeeNameParam}”

### 查询 XML 树

假定要查找 XML 标记内的值，例如，某个员工的分机号。需要针对所需的值制定查询。查询将指明元素的位置和要检索或设置的元素。指定的路径会将搜索范围限定到特定的标记。还可以在根节点的所有节点下搜索所有特定类型的元素。

对于特定路径，请使用 ”Query=/*full_xml_path_name*/*element_name*”

对于所有节点下的同一元素名，请使用 ”Query=//*element_name*”

在 XML 函数的 LoadRunner 实现中，查询范围是整个 XML 树。树信息将被作为 *xml* 参数的值发送到 LoadRunner API 函数。

### 多查询匹配

当您对 XML 元素执行查询时，默认情况下 VuGen 只返回第一个匹配项。要从查询中检索多个值，需要在函数内指定 “SelectAll=yes”属性。 VuGen 将添加后缀 *_index* 以表明有多个参数。例如，如果通过名称 *EmployeeName* 定义了参数，则 VuGen 将创建 *EmployeeName_1*、 *EmployeeName_2*、 *EmployeeName_3* 等等。

```
lr_xml_set_values("XML={XML_Input_Param}",
"ResultParam=NewXmlParam", "ValueParam=ExtensionParam",
"SelectAll=yes", "Query=//extension", LAST);
```

如果使用写入到参数的函数，此时可以求得写入到该参数的值。例如，以下代码检索并打印查询的多个匹配项：

```
NumOfValues = lr_xml_get_values("Xml={XmlParam}", "Query=//name",
        "SelectAll=yes", "ValueParam=EmployeeName", LAST};
```

对于从参数中读取值的函数，必须预定义参数的值。参数还必须使用约定 *ParamName_IndexNumber*，例如 *Param*_1、*Param*_2、*Param*_3 等。该参数集合也称为参数集。

在以下示例中， **lr_xml_set_values** 将从参数集中读取值，然后在 Xpath 查询中使用那些值。表示员工分机号的参数集称为 *ExtensionParam*。它有两个成员：*ExtensionParam_1* 和 *ExtensionParam_2*。 **lr_xml_set_values** 函数将查询 XML 输入字符串，并将第一个匹配项的值设置为 1111，将第二个匹配项的值设置为 2222。

```
lr_save_string("1111", "ExtensionParam_1");
lr_save_string("2222", "ExtensionParam_2");

lr_xml_set_values("XML={XML_Input_Param}",
    "ResultParam=NewXmlParam", "ValueParam=ExtensionParam",
    "SelectAll=yes", "Query=//extension", LAST);
```

## 使用 XML 属性

VuGen 包含对属性的支持。可以使用简单表达式操纵 XML 元素和节点的属性，就像操纵元素本身一样。

可以使用该表达式操纵所需的属性，也可只操纵具有特定值的属性。在以下示例中， **lr_xml_delete** 将删除第一个具有 *name* 属性的 *cubicle* 元素。

```
    lr_xml_delete("Xml={ParamXml}",
                                "Query=//cubicle/@name",
                                    "ResultParam=Result",
                                LAST
        );
```

在下一个示例中， **lr_xml_delete** 将删除其 *name* 属性等于 *Paul* 的第一个 cubicle 元素。

```
lr_xml_delete("Xml={ParamXml}",
                        "Query=//cubicle/@name="Paul",
                            "ResultParam=Result",
                        LAST
    );
```

## 构建 XML 脚本

开始时，可使用所需的协议 （HTTP、 SOAP 等）新建脚本。可以录制使用该协议的会话，还可以在不录制的情况下对整个脚本进行编程。按如下方式构建脚本的 Action 部分：

- XML 输入声明
- Action 部分

**XML 输入部分**包含要用作输入变量的 XML 树。可将该 XML 树定义为 **char** 类型变量。例如：

```
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 部分**包含求变量的值和对元素值进行查询。在以下示例中，使用 **lr_save_string** 来求 xml 输入字符串的值。将查询输入变量以获取员工姓名和分机号。

```
Action() {

    /* 将输入保存为参数。 */
    lr_save_string(xml_input, "XML_Input_Param");

    /* 查询 1 – 从指定的元素中检索员工姓名。 */
    lr_xml_get_values("XML={XML_Input_Param}",
        "ValueParam=OutputParam",
        "Query=/acme_org/employee/name", LAST);

    /* 查询 2 – 在根目录下的任意路径中检索分机号。 */
    lr_xml_get_values("XML={XML_Input_Param}",
        "ValueParam=OutputParam",
        "Query=//extension", LAST);

    return 0;
}
```

## 增强已录制的会话

您可以通过录制会话然后手动添加相关的 XML 和 LoadRunner API 函数，来准备 XML 脚本。

以下 SOAP 协议示例阐释了如何通过 LoadRunner API 函数来增强已录制的会话。注意，所录制的唯一函数是 **web_submit_data**，它显示为粗体格式。

第一部分包含 SOAP 消息的 SoapTemplate 变量的 XML 输入声明。

```
#include "as_web.h"

// SOAP 消息
const char*pSoapTemplate=
        "<soap:Envelope
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">"
            "<soap:Body>"
```

```
            "<SendMail xmlns=\"urn:EmailIPortTypeInft-IEmailService\"/>"
        "</soap:Body>"
    "</soap:Envelope>";
```

以下部分表示用户的操作。

```
Action1()
{
    // 获得响应正文
    web_reg_save_param("ParamXml", "LB=", "RB=", "Search=body", LAST);

    // 通过 HTTP GET 获取天气情况
    web_submit_data("GetWeather",
                    "Action=http://glkev.net.innerhost.com/glkev_ws/
                          WeatherFetcher.asmx/GetWeather",
                      "Method=GET",
                      "EncType=",
                      "RecContentType=text/xml",
                      "Referer=http://glkev.net.innerhost.com
                           /glkev_ws/WeatherFetcher.asmx?op=GetWeather",
        "Snapshot=t2.inf",
        "Mode=HTTP",
        ITEMDATA,
        "Name=zipCode", "Value=10010", ENDITEM,
        LAST);

    // 获取 City 值
    lr_xml_get_values("Xml={ParamXml}",
                      "Query=City",
                      "ValueParam=ParamCity",
                      LAST
    );

    lr_output_message(lr_eval_string("***** City = {ParamCity} *****"));

    // 获取 State 值
    lr_xml_get_values("Xml={ParamXml}",
                      "Query=State",
                      "ValueParam=ParamState",
```

```
                LAST
);

lr_output_message(lr_eval_string("***** State = {ParamState} *****"));

// 通过使用模板一次获取多个值
lr_xml_get_values_ex("Xml={ParamXml}",
                     "Template="
                          "<Weather>"
                              "<Time>{ParamTime}</Time>"
                              "<Temperature>{ParamTemp}</Temperature>"
                              "<Humidity>{ParamHumid}</Humidity>"
                              "<Conditions>{ParamCond}</Conditions>"
                          "</Weather>",
                     LAST
);

lr_output_message(lr_eval_string("***** Time = {ParamTime}, Temperature =
                                                {ParamTemp}, "
                                 "Humidity = {ParamHumid}, Conditions =
                                            {ParamCond} *****"));

// 生成可读的天气预报
lr_save_string(lr_eval_string("\r\n\r\n*** Weather Forecast for {ParamCity}, {ParamState} ***\r\n"
                              "\tTime:{ParamTime}\r\n"
                              "\tTemperature:{ParamTemp} deg. Fahrenheit\r\n"
                              "\tHumidity:{ParamHumid}\r\n"
                              "\t{ParamCond} conditions expected\r\n"
                              "\r\n"),
                              "ParamForecast"
);

// 将 soap 模板保存到参数中
lr_save_string(pSoapTemplate, "ParamSoap");

// 将请求正文插入到 SOAP 模板中
lr_xml_insert("Xml={ParamSoap}",
              "ResultParam=ParamRequest",
              "Query=Body/SendMail",
```

```
                "position=child",
                "XmlFragment="
                        "<FromAddress>taurus@merc-int.com</FromAddress>"
                        "<ToAddress>support@merc-int.com</ToAddress>"
                        "<ASubject>Weather Forecast</ASubject>"
                        "<MsgBody/>",
                LAST
    );

//
//      "<soap:Envelope
xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">"
//          "<soap:Body>"
//              "<SendMail xmlns=\"urn:EmailIPortTypeInft-IEmailService\"/>"
//                      "<FromAddress>taurus@merc-int.com</FromAddress>"
//                      "<ToAddress>support@merc-int.com</ToAddress>"
//                      "<ASubject>Weather Forecast</ASubject>"
//                      "<MsgBody/>"
//              "</SendMail>"
//          "</soap:Body>"
//      "</soap:Envelope>";
//

    // 插入实际的天气预报文本
    lr_xml_set_values("Xml={ParamRequest}",
                    "ResultParam=ParamRequest",
                    "Query=Body/SendMail/MsgBody",
                    "ValueParam=ParamForecast",
                    LAST);

    // 添加 SOAP 的标头
    web_add_header("SOAPAction", "urn:EmailIPortTypeInft-IEmailService");

    // 获取响应正文
    web_reg_save_param("ParamXml", "LB=", "RB=", "Search=body", LAST);

    // 使用 SOAP 请求将天气预报发送给收件人
    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);

    // 验证邮件是否已发送
    lr_xml_find("Xml={ParamXml}",
                "Query=Body/SendMailResponse/return",
                "Value=0",
                LAST
    );

    return 0;
}
```

# 69

# VuGen 调试提示

本章包含获取更详细的调试信息的几种方法，帮助您产生无错误的 Vuser 脚本。

- 常规调试提示
- 使用 C 函数进行跟踪
- 检查回放输出
- 调试数据库应用程序
- 使用 Oracle 应用程序
- 解决 Oracle 应用程序的常见问题
- 两层数据库脚本提示
- 运行 PeopleSoft-Tuxedo 脚本

## 常规调试提示

VuGen 可用作常规文本编辑器。您可以在其中打开任何文本文件并进行编辑。当回放期间在输出窗口中显示错误消息时，您可以双击该错误消息， VuGen 将使光标跳到导致问题的测试行。您还可以将光标置于错误代码上并按 F1 键，查看该错误代码的联机帮助解释。

## 使用 C 函数进行跟踪

您可以使用 C 解释器跟踪选项 （在 230 或更高版本中）调试 Vuser 脚本。通过 ci_set_debug 语句可以在脚本的特定点打开和关闭跟踪和调试。

**ci_set_debug**(ci_this_context, int debug, int trace);

例如，您可以在脚本中添加下列语句：

```
ci_set_debug(ci_this_context, 1, 1) /* 打开跟踪和调试 */
ci_set_debug(ci_this_context, 0, 0) /* 关闭跟踪和调试 */
```

要查看跟踪信息的输出，请使用打印回调函数 ci_set_print_CB()。

## 添加附加 C 语言关键字

在 VuGen 中运行 C 脚本时，其分析程序将使用内置的 C 解释器分析脚本中的函数。您可以添加标准分析程序库之外的关键字。默认情况下，有些常用的 C++ 关键字会在安装期间添加，例如 *size_t* 和 *DWORD*。您可以编辑列表并为您的环境添加附加的关键字。

**要添加附加关键字，请执行下列操作：**

1. 打开位于计算机的 <Windows> 或 <Windows>/System 目录下的 vugen_extra_keywords.ini 文件。
2. 在 EXTRA_KEYWORDS_C 部分中，添加 C 解释器所需的关键字。

该文件包含以下格式：

```
[EXTRA_KEYWORDS_C]
FILE=
size_t=
WORD=
DWORD=
LPCSTR=
```

## 检查回放输出

请查看回放输出 （在 VuGen 或表示 LoadRunner 驱动程序运行输出结果的 **output.txt** 文件中）。为了获取回放的测试的详细日志输出，您还可以更改 VuGen 中的运行时设置选项，以便选择更广泛的日志记录。

## 调试数据库应用程序

下列提示仅适用于数据库应用程序 （Oracle、 ODBC、 Ctlib）：

- 生成调试信息
- 检查编译器信息

### 生成调试信息

---

**注意**：您现在可以设置选项，使用 VuGen 的用户界面查看本部分描述的大多数信息。

---

VuGen 中包含检查器 “引擎”。按照以下方式编辑 *\WINDOWS_DIR\vugen.ini*，您可以强制 VuGen 录制器创建 “检查器”输出：

```
[LogMode]
EnableAscii=ASCII_LOG_ON
```

启用此选项后，VuGen 将在录制快结束时在 Data 目录中创建文件 **vuser.asc**。注意，此选项应仅用于调试目的。此输出文件可能会变得非常大 （数 MB），并对计算机性能和磁盘空间产生严重影响。

对于类似基于 ODBC 的应用程序而言，可以将 ODBC 管理器 （位于 Windows 的 “控制面板”中）配置为提供类似的跟踪输出。打开 ODBC 选件并选择将 “跟踪 ODBC 调用”打开。同样，“ODBC 开发人员工具包”提供了用来跟踪调用的 Spy 实用程序。

要启用更详细的调试信息，请在 \WINDOWS_DIR\vugen.ini 文件中添加以下部分。

```
[INSPECTOR]
TRACE_LEVEL=3
TRACE_FILENAME=c:\tmp\sqltrace.txt
```

sqltrace.txt 文件包含与录制期间进行的挂钩调用有关的有用内部信息。trace_level 介于 1 和 3 之间，其中 3 表示最详细的调试级别。注意，在 VuGen 5.02 和更高版本中，您可以在用户界面中设置跟踪级别。

### 检查编译器信息

您可以查看关于代码生成、预处理和编译等每个阶段的信息，确定所有错误的根源。

### 代码生成信息

查看 Data 目录下的 **vuser.log** 文件。此文件包含代码生成阶段的日志，它是在每次 lrd 录制（即所有的数据库协议）的结尾自动创建的。以下是日志文件的示例：

```
lrd_init：OK
lrd_option：OK
lrd_option：OK
lrd_option：OK
代码生成成功
lrd_option：OK
lrd_end：OK
```

如果有任何消息不正常或不成功，则代码生成过程中出现了问题。

### 预处理和编译信息

运行期间， VuGen 将显示关于预处理和编译过程的信息。

# 使用 Oracle 应用程序

Oracle 应用程序是两层 （“胖”客户端）的打包应用程序，由 35 个不同的模块（Oracle Human Resources、 Oracle Financials 等）组成。

为 Oracle 应用程序录制和回放 Vuser 时，有很多问题要注意：

- 典型的脚本包含数以千计的事件、绑定和分配。
- 典型的脚本中包含每个用户会话的许多数据库连接。
- 脚本几乎始终需要关联的查询。
- Oracle 应用程序的客户端仅为 16 位 （使用 Oracle Developer 2000 开发）。这表示对于调试而言，如果您没有 Oracle 32 位客户端，则需要使用 VuGen 的“强制 16 位”选项。

  新建窗口时，该应用程序将检索文件系统中要显示的 .xpf 文件。当前， VuGen 并没有将这个问题考虑在内，因为它是在客户端 / 服务器级别上进行录制的。因此性能度量的准确性明显不够，因为在大多数情况下，性能问题与客户端和文件服务器之间的网络瓶颈有关。当前，我们正在考虑这个问题，以及如何或是否能够彻底解决。

# 解决 Oracle 应用程序的常见问题

本部分包含在使用 Oracle 应用程序时可能遇到的常见问题及建议的解决方案的列表。

### ORA-20001 和 ORA-06512

如果 lrd_stmt 中包含 pl/sql 块，那么在回放期间将出现错误 ORA-20001 和 ORA-06512：fnd_signon.audit_responsibility(...)

由于每个新连接的登录编号都是唯一的，所以此语句在回放期间将失败。

#### 解决方案

为了解决此问题，需要使用新的关联工具来处理登录编号。这是此语句中分配的第二个值。

扫描要关联的可能值后，请突出显示失败的语句的第二个 lrd_assign_bind() 的值。注意，“关联查询”窗口中的值的显示顺序可能与实际录制语句的顺序不相同。

包含替换值的网格应显示在包含 pl/sql 块的 lrd_stmt 之后：
fnd_signon.audit_user(...).

---

**注意**：由于每个连接的登录编号都是唯一的，因此需要对录制的每个新连接都使用关联。

---

### 解决方案示例

由于第二个值 “1498224”是每个新连接的唯一登录编号，因此以下语句在回放时失败。

```
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);
```

可以使用 “fnd_signon.audit_user”在 lrd_stmt 中找到登录编号。第一个占位符 “a”的值应该保存。“a”的输入始终为 “0”，但是输出则为请求的值。

### 修改后的代码：

```
 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);
```

### 使用大数

有时，大数 （NUMBER 数据类型）在网格和 ASCII 文件中以不同的格式显示。此差异使得在搜索要保存的关联值时更难识别数字。

例如，在网格中显示为 1000003 的值，在录制日志 （ASCII 文件）中显示为 1e+0006。

#### 解决方法

如果在回放期间出现错误，而关联工具在以前的结果中找不到该值，则可以在网格中以其他格式查找该值。

## ORA-00960

如果录制的脚本中的列名不唯一，则会出现这种错误。例如：

```
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);
```

在这种情况下，您将收到以下错误：

“lrdo.c/fjParse：“oparse”错误，返回代码 =960， oerhms=ORA-00960：选择列表中的列名不明确”。

### 解决方法

在至少一个不唯一的列中添加别名，将其映射为新的唯一名称，从而更改该语句。例如：

```
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);
```

备用解决方法：从 lrd 语句中删除 ORDER BY

## ORA-2002

试图使用未打开的光标时，将出现错误 2002。当您以多次循环回放用户并将其录制到脚本的多个部分中时，会出现这种错误。

尤其是光标在 vuser_init 部分打开，而在 Actions 部分关闭时，如果试图使用光标，将在第二次循环时遇到这种错误。这是由于光标已关闭，而没有重新打开所致。

例如：vuser_init 部分中包含 *lrd_open_cursor*，而 Actions 部分中包含 *lrd_close_cursor*。如果以多次循环的方式回放该用户，则将在第二次循环时遇到错误，因为您试图使用未打开的光标 （它在第一次循环中已关闭，而在第二次循环中未重新打开）。

### 解决方法

解决此问题的最简单的方法是将出现问题的光标的 **lrd_close_cursor** 或 / 和 **lrd_close_connection** 移到 *vuser_end* 部分。

### 数据库协议 (lrd)

不支持对录制的异步操作进行回放。

### 客户端版本错误

如果运行了错误的 Oracle 客户端版本，可能会收到错误消息：

“错误：lrdo_open_connection：“olog” LDA/CDA 返回代码 _019：无法在用户端分配内存”

### 解决方法

您需要修改位于 LoadRunner bin 目录下的 *lrd.ini* 文件中的库信息。此文件包含的设置指明在录制或回放期间加载哪个版本的 Mercury 数据库支持。每种主机在此文件中都有对应的部分。例如， *lrd.ini* 文件的以下部分针对 HP/UX 上的 Oracle：

```
[ORACLE_HPUX]
;816=liblrdhpo816.sl
;81=liblrdhpo81.sl
;80=liblrdhpo80.sl
73=liblrdhpo73.sl
72=liblrdhpo72.sl
```

这些设置指明：如果客户端使用 Oracle 8.1.6、 liblrdhpo81.sl for Oracle 8.1.5 等，则 LoadRunner 应使用 Mercury 库 liblrdhpo816.sl。

在 UNIX 上的回放期间， lrd *ini* 文件中的设置必须指明要使用的数据库的正确版本。假设必须使用 Oracle 8.1.5 回放 HP/UX 的 Vuser，那么此时其他 Oracle 版本的前几行开头应使用 “;”加以注释。 lrd.ini 文件的对应部分将如下所示：

```
[ORACLE_HPUX]
;816=liblrdhpo816.sl
81=liblrdhpo81.sl
;80=liblrdhpo80.sl
73=liblrdhpo73.sl
72=liblrdhpo72.sl
```

如果应用程序不使用 lrd.ini 文件中提到的 DLL，则可能还需对 Win32 进行更改。例如，PowerBuilder 6.5 使用 Oracle 8.0.5，但是它使用 ora803.dll 而不是 ora805.dll。在这种情况下，请注释掉 ORACLE_WINNT 部分中的 805 和 804 部分，或将 805 部分从以下形式：

```
805=lrdo32.dll+ora805.dll
```

改为

```
805=lrdo32.dll+ora803.dll
```

# 两层数据库脚本提示

本部分提供两层数据库脚本的解决方案。对于特定于 Siebel 的解决方案，请参阅下一部分。

**问题 1：**当相同的值使用应用程序本身时，由数据驱动的脚本为何会失败？

**解答：**失败可能是由于数据值结尾的空格所致。即使直接键入 GUI 中的数据值有可能会截断，您也应该从数据文件中手动消除这些值。制表符分隔的文件可以隐藏结尾的空格，因此可以掩盖此问题。一般而言，建议使用以逗号分隔的文件。您可以在 Excel 中查看这些文件，了解内容是否正确。

**问题 2：**为什么第二次循环中会出现光标状态无效的 SQL 错误？

**解答：****lrd_close_cursor** 函数可能尚未生成，或者它可能存在于 *end* 部分，而不是 *action* 部分。要使脚本循环成功，需要添加光标闭合函数，或将其从 *end* 部分中移出。

打开新的光标可能会耗费很多资源。因此，建议您仅在第一次循环期间在 *actions* 部分打开一次光标。然后，可以使用 “循环次数”类型将包含循环次数的新参数作为字符串添加。调用参数 *IterationNum*。然后在 *actions* 部分中，将打开新光标的调用

```
lrd_open_cursor(&Csr1, Con1, 0);
```

替换为

```
if (!strcmp(lr_eval_string("<IterationNum>"), "1"))
   lrd_open_cursor(&Csr1, Con1, 0);
```

**问题 3：**如何修复由于 *vdf.h* 文件中的数据声明而导致 VuGen 生成的不编译代码？

**解答：**问题很可能在于 VuGen 不支持 SQL 数据类型。对于 Microsoft SQL，通常可以通过将 *vdf.h* 中的未定义错误消息替换为 “DT_SZ”（以 null 结尾的字符串）来解决此问题。尽管这不是实际的数据类型，但 VuGen 可以正确地编辑脚本。请报告问题，并将原始的脚本发送给客户支持。

**问题 4：**LRD 错误 2048 表示什么意思？

**解答：**VuGen 失败，因为它试图绑定的变量长度超过在录制期间分配的长度。通过扩大 *vdf.h* 中的变量定义，可以更正此问题，从而接收从数据库返回的更长的字符串。在此文件中搜索唯一的数值标识符。您将看到其定义和长度 . 长度是结构中的第三个元素。按要求增加此长度，脚本将成功地回放。例如，在以下脚本中将看到：

```
lrd_assign(&_2_D354, "<ROW_ID>", 0, 0, 0);
```

在 *vdf.h* 中，我们将搜索 *_2_D354* 并找到

```
static LRD_VAR_DESC _2_D354 = {
            LRD_VAR_DESC_EYECAT, 1, 10, LRD_BYTYPE_ODBC,
            {0 ,0, 0}, DT_SZ, 0, 0, 15, 12};
```

我们将其更改为：

```
static LRD_VAR_DESC _2_D354 = {
            LRD_VAR_DESC_EYECAT, 1, 12, LRD_BYTYPE_ODBC,
            {0,0, 0}, DT_SZ, 0, 0, 15, 12};
```

LRD_VAR_DESC 的完整定义显示在 *lrd.h* 中。通过搜索 typedef struct LRD_VAR_DESC 可以找到该定义。

**问题 5：**在使用 ODBC 和 Oracle 时如何才能获取受 UPDATE、 INSERT 或 DELETE 影响的行数？

**解答：**您可以使用 **lrd** 函数获取此信息。对于 ODBC，请使用 **lrd_row_count**。语法为：

```
 int rowcount;
    .
    .
    .
lrd_row_count(Csr33, &rowcount, 0);
```

注意， **lrd_row_count** 必须紧跟有关的语句执行。

对于 Oracle，您可以使用 **lrd_exec** 的第四个参数。

```
lrd_exec(Csr19, 1, 0, &rowcount, 0, 0);
```

如果您使用的是 Oracle 的 OCI 8，则可以使用 **lrd_ora8_exec** 的第五个参数。

```
lrd_ora8_exec(OraSvc1, OraStm3, 1, 0, &uliRowsProcessed, 0, 0, 0, 0, 0);
```

**问题 6：**如何避免重复键冲突？

**解答：**有时在执行 “插入”操作时，会看到重复键冲突。通过比较两次录制应该可以找到主键，从而确定问题所在。请检查此语句或更早的 UPDATE 或 INSERT 语句是否应该使用关联查询。您可以使用数据字典查找已违反的唯一约束中使用的列。

在 Oracle 中，当违反了唯一约束时，您将看到以下消息：

ORA-00001：违反了唯一约束 (SCOTT.PK_EMP)

在本例中， SCOTT 是相关的唯一索引的所有者， PK_EMP 是此索引的名称。使用 SQL*Plus 可以查询数据字典找到对应的列。此查询的模式为：

```
select column_name from all_ind_columns where index_name = '<索引名>' and
index_owner = '< 索引所有者 >';

select column_name from all_ind_columns where index_name = 'PK_EMP' and
index_owner = 'SCOTT';
```

由于插入数据库中的值是新的，因此它们不可能出现在更早的查询中，但是它们可能与更早查询的结果相关，例如多个从更早查询中返回的值。

对于 Microsoft SQL Server，您将看到下列消息之一：

不能在具有唯一索引 “IX_newtab”的对象 “newtab”中插入重复的键行。

违反唯一键约束 “IX_Mark_Table”。不能在对象 “Mark_Table”中插入重复的键。

违反主键约束 “PK_NewTab”。不能在对象 “NewTab”中插入重复的键。

您可以使用查询分析器找出键或索引使用的列。此查询的模式为：

```
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')
```

对于 DB2，您可能会看到以下消息：

SQL0803N INSERT 语句、UPDATE 语句中的一个或多个值，或者由 DELETE 语句导致的外键更新无效，因为它们将在包含主键、唯一约束或唯一索引的表中产生重复的行。 SQLSTATE=23505

如果仍然有问题，请务必检查为 Update 和 Insert 更改的行数，以进行录制和回放。通常 UPDATE 在回放期间无法更改任何行，因为未满足 WHERE 子句。这不会直接产生错误，但是会使表无法正确更新，而且会导致随后的 SELECT 在关联查询时选择错误的值。

此外，还请验证多用户回放期间是否没有问题。在某些情况下，只有一个用户可以成功执行 UPDATE。这种情况发生在 Siebel 中，此时必须手动编写循环克服该问题。

**问题 7：**回放本应修改数据库的脚本后，数据库似乎没有被修改。

**解答：**请通过用户应用程序的 UI，检查尝试查看可供应用程序访问的当前数据时是否显示已更新的值。如果这些值尚未更新，则需要确定它们是否未被更改。很可能在录制应用程序时， UPDATE 语句已更改了一行或多行，但不会在回放期间进行任何更改。

检查下列项目：

- **验证语句：**如果 UPDATE 语句中存在 WHERE 子句，请验证其正确性。
- **检查关联：**录制应用程序两遍，并比较每遍录制中的 UPDATE 语句，确保执行必要的关联。
- **检查总行数：**检查 UPDATE 之后更改的行数。对于 Oracle，此信息存储于 **lrd_exec** 的第四个参数中。对于 ODBC，请使用 **lrd_row_count** 确定已更新的行数。您还可以在用来打印已更新行数的脚本中添加代码。如果此值为 0， UPDATE 将无法修改数据库。
- **检查 SET 子句：**检查 UPDATE 语句的 SET 子句。请确保将所有必要的值都关联起来，而不是对它们进行硬编码。通过比较两次录制的 UPDATE，可以看到这些内容。

在某些情况下， UPDATE 将在回放一个 Vuser 时发挥作用，而在回放多个 Vuser 时不起作用。 Vuser 之间的 UPDATE 可能会彼此干扰。除非您希望每个 vuser 使用相同的值更新，否则请参数化每个 Vuser，以便其在 UPDATE 期间使用不同的值。在这种情况下，请尝试添加重试逻辑第二次执行 UPDATE。

**问题 8：**回放使用 Oracle 应用程序录制的语句时，如何避免唯一列名错误。例如：

```
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);
```

发出了以下错误消息：

“lrdo.c/fjParse：“oparse”错误，返回代码 =960， oerhms=ORA-00960：选择列表中的列名不明确”。

**解答：**在至少一个不唯一的列中添加别名，将其映射为新的唯一名称，从而更改该语句。例如：

```
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 特定脚本提示

本部分为 Siebel 数据库用户提供解决方案。您还应该参考上一部分，该部分中讨论了一些常规数据库脚本提示。

**问题 9：**虚拟用户在 VuGen 中运行良好，但在具有重复键冲突的 Controller 中却会失败。

**解答：**Siebel 客户端把键存储在 S_SSA_ID 表的 NEXT_SUFFIX 列中。此客户端具有用来检测某些情况 （即无法成功获取后缀值块）并从中恢复的代码。

LoadRunner 会自动将 S_SSA_ID 表中的 NEXT_SUFFIX 和 MODIFICATION_NUM 字段关联起来。在 UPDATE 期间， MODIFICATION_NUM 字段将以 1 为增量递增， NEXT_SUFFIX 字段将以 100 为增量递增（以 36 为基数）。但是，在客户端无法获取新的后缀值块时， LoadRunner 不会添加代码。因此，如果尝试在数据库中插入新值，回放将由于唯一约束错误而失败。

如果要在第一次尝试失败后执行重试，必须在脚本中每个获得后缀块的位置手动添加代码。通过在脚本中搜索 SiebelPreSave 可以查找这些位置。此外，还必须添加 *while* 循环，其代码与下例类似。本示例仅适用于 Oracle。对于 ODBC，请使用 **lrd_row_count**，而不要使用 **lrd_exec** 的第四个函数。

```
unsigned long lRowUpdated;
int nAttempt;
```

```
...

// 这种情况将持续循环，直到成功获得 “next_suffix”
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();
   lrd_fetch(Csr13, -1, 4, 0, PrintRow26, 0);
   GRID(26);
   SiebelPostSave_1();

   if (nAttempt > 1)
   {
    lrd_open_cursor(&Csr14, Con1, 0);
    lrd_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);
   }
   lrd_assign_bind(Csr14, "6", "<modification_num>", &_6_D376, 0,
                LRD_BIND_BY_NUMBER, 0);
```

```
    lrd_assign_bind(Csr14,"5", "0-11",&_5_D377,0,LRD_BIND_BY_NUMBER, 0);
    strcpy (szTimeAtNewButton, lr_eval_string("<Now>"));
    sprintf (szTimeStamp, "%s %s", lr_eval_string("<Today>"),
            szTimeAtNewButton);
    lr_save_string (szTimeStamp, "DateTimeStamp");
lrd_assign_bind(Csr14, "4", "<DateTimeStamp>", &_4_D378, 0,
       LRD_BIND_BY_NUMBER, 0);
lrd_assign_bind(Csr14, "3", "<next_modnum>", &_3_D379, 0,
       LRD_BIND_BY_NUMBER, 0);
lrd_assign_bind(Csr14, "2", "<next_suffix_x100>", &_2_D380, 0,
       LRD_BIND_BY_NUMBER, 0);
lrd_assign_bind(Csr14, "1", "1-1E1",&_1_D381,0,LRD_BIND_BY_NUMBER, 0);

// 此 update 不会更新任何行，除非成功获得后缀
lrd_exec(Csr14, 1, 0, &lRowUpdated, 0, 0);
lrd_commit(0, Con1, 0);

}//while
     lr_output_message ("…Rows updated %ld", lRowUpdated);
```

**问题 10：**如何找到要与主键关联的正确值？

**解答：**Siebel 将根据以 36 为基数的数学运算处理 *<next_suffix>*，从而生成键值。请尝试比较几次录制并确定相互之间的关系。关联 Siebel 时，可以忽略日期字段，因为它们似乎并不影响脚本回放。

**问题 11：**如何解决具有重复键冲突的 INSERT into S_SRV_REQ 故障？

**解答：**主键为 SR_NUM。 LoadRunner 的较新版本应该可以使用函数 lrd_siebel_str2num 自动关联此表的插入操作，该函数可将 S_SSA_ID 表中的 NEXT_SUFFIX 从以 36 为基数的值转换为以 10 为基数的等效值。LoadRunner 的早期版本可能无法正确处理此类关联。

**问题 12：**LoadRunner 不会自动执行要正确回放脚本时所需的所有关联。如何添加缺少的关联？

**解答：**当前， LoadRunner 仅保存 S_SSA_ID 表中的 NEXT_SUFFIX 和 MODIFICATION_NUM 列的值，并于随后在脚本中使用时将它们替换为参数。您可能需要手动添加某些附加关联。 *print.inl* 文件的 **SiebelPreSave** 和 **SiebelPostSave** 函数中的关联代码可以作为示例，说明确定需要关联的内容后，如何对特定的值进行关联。

- 有时，会从 S_SSA_ID 表中选择 NEXT_FILE_SUFFIX 和 MODIFICATION_NUM 列。在这种情况下， UPDATE 语句通过以 36 为基数、向该字符串加一并向 MODIFICATION_NUM 加一更新 NEXT_FILE_SUFFIX。 NEXT_FILE_SUFFIX 的值常常会插入表的 FILE_REV_NUM 字段中。此表的名称通常以 *_ATT* 后缀结尾，指明它是附件。
- 每次 Siebel 执行 UPDATE 语句时，都会存在以 1 为增量递增的 MODIFICATION_NUM 列。LoadRunner 仅会为 S_SSA_ID 表自动生成此类关联。其他情况下，您必须手动生成关联。
- Siebel 将根据 ID 号引用记录。 Siebel 通常会找到特定类型（例如协议）的所有记录，然后在尝试更新或删除此类型的现有记录时使用对应的记录 ID 号。为生成有意义的负载测试，回放时需要将该 ID 号替换为参数。 ID 号的形式为一个或多个数字、连字符，后接一个或多个字母数字字符，例如 1-QPF9。LoadRunner 不会自动执行此参数化操作，因此您必须手动执行此操作。
- 如果发现缺少其他任何关联或参数化内容，请通知客户支持部门，以便 Mercury Interactive 可以改进 LoadRunner 对 Siebel 的支持。

## 运行 PeopleSoft-Tuxedo 脚本

要使用 TUXEDO 7.x 运行 PeopleSoft-Tuxedo Vuser，必须更改 *mdrv.dat* 文件中的库扩展名：

```
[PeopleSoft-Tuxedo]
WINNT_EXT_LIBS=lrt7.dll
```

# 70

# 高级主题

本章包含适用于高级 LoadRunner 用户的附加信息。

- 录制期间生成的文件
- 回放期间生成的文件
- 从 Unix 命令行运行 Vuser
- 指定 Vuser 行为
- 命令行参数
- 录制 OLE 服务器
- 检查 .dat 文件
- 添加新的 Vuser 类型

## 录制期间生成的文件

假设已为所录制的测试指定了名称 “vuser”，并将其存储在 c:\tmp 下。以下是录制后生成的重要文件的列表：

| | |
|---|---|
| **vuser.usr** | 包含关于虚拟用户的信息：类型、 AUT、操作文件等 |
| **vuser.bak** | 上一次保存操作之前的 Vuser.usr 副本 |
| **default.cfg** | 包含 VuGen 应用程序中定义的所有运行时设置 （思考时间、循环、日志、 Web）的列表 |
| **vuser.asc** | 最初录制的 API 调用 |
| **vuser.grd** | 包含数据库脚本中的网格的列标题 |
| **default.usp** | 包含脚本的运行逻辑 （包括 actions 部分的运行方式） |

| | |
|---|---|
| **init.c** | 在 VuGen 主窗口中显示的 Vuser_init 函数的精确副本 |
| **run.c** | 在 VuGen 主窗口中显示的 Action 函数的精确副本 |
| **end.c** | 在 VuGen 主窗口中显示的 Vuser_end 函数的精确副本 |
| **vdf.h** | 脚本中使用的 C 变量定义的头文件 |
| **\Data** | Data 目录存储主要用作备份的所有已录制数据。数据保存到此目录中后，将无法访问或使用。例如， **Vuser.c** 是 **run.c** 的副本 |

### Vuser.usr 文件示例

```
[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 文件示例

```
[General]
XlBridgeTimeout=120

[ThinkTime]
Options=NOTHINK
Factor=1
```

```
LimitFlag=0
Limit=1

[Iterations]
NumOfIterations=1
IterationPace=IterationASAP
StartEvery=60
RandomMin=60
RandomMax=90

[Log]
LogOptions=LogBrief
MsgClassData=0
MsgClassParameters=0
MsgClassFull=0
```

## 回放期间生成的文件

本部分描述回放 Vuser 时出现的情况。

1. 将创建 **options.txt** 文件，其中包含预处理器需要的命令行参数。
2. 还将创建 **Vuser.c**，其中包含对所有相关 .c 和 .h 文件的 “包含”。
3. 调用 c 预处理器 **cpp.exe**，以便在开发文件中 “填写”所有宏定义、预编译器指令等。

---

**注意**：beta2 修补程序 **cpp.exe** 是一个共享可执行程序，与早期有问题的版本完全不同。

---

它使用了以下命令行：

```
cpp  -foptions.txt
```

4 将创建文件 **pre_cci.c**，它也是 C 文件 （**pre_cci.c** 是在 **options.txt** 文件中定义的）。将创建文件 **logfile.log** （也是在 **options.txt** 中定义的），其中包含此进程的所有输出。如果预处理阶段没有问题，则此文件应为空。如果此文件不为空，则几乎可以肯定由于致命错误，下一阶段的编译将失败。

5 现在将调用 **cci.exe** C 编译器，创建与平台相关的伪二进制文件 (.ci)，该文件将由在运行时解释它的虚拟用户驱动程序所使用。 cci 会将 **pre_cci.c** 文件作为输入。

6 文件 **pre_cci.ci** 是按以下方式创建的：

```
cci  -errout  c:\tmp\Vuser\logfile.log  -c  pre_cci.c
```

7 文件 **logfile.log** 是包含编译输出的日志文件。

8 文件 **pre_cci.ci** 此时将重命名为 **Vuser.ci**。

由于编译可能包含警告和错误，而且由于驱动程序不了解此进程的结果，因此驱动程序首先检查 **logfile.log** 文件中是否存在任何条目。如果存在，它将接着检查文件 **Vuser.ci** 是否已构建。如果该文件的大小不为零，则表示 cci 已成功编译（如果为零，则表示编辑已失败，并将显示错误消息）。

9 现在，相关的驱动程序运行时将把 **.usr** 文件和 **Vuser.ci** 文件作为输入。例如：

```
mdrv.exe  -usr c:\tmp\Vuser\Vuser.usr  -out c:\tmp\Vuser  -file
c:\tmp\Vuser\Vuser.ci
```

需要有 **.usr** 文件，因为它将通知驱动程序正在使用的数据库。这样，便可以了解运行时需要加载哪些库。

10 将创建 **output.txt** 文件 （位于由 °×out°± 变量定义的路径下），其中包含运行的所有输出消息。此输出与 VuGen 运行时输出窗口以及 VuGen 主窗口下半部分显示的输出相同。

### options.txt 文件示例

```
-DCCI
-D_IDA_XL
-DWINNT
-Ic:\tmp\Vuser                    （Vuser 包含文件的名称和位置）
-IE:\LRUN45B2\include              （LoadRunner 包含文件的名称和位置）
-ec:\tmp\Vuser\logfile.log        （输出日志文件的名称和位置）
 c:\tmp\Vuser\VUSER.c             （要处理的文件的名称和位置）
```

### Vuser.c 文件示例

```
#include "E:\LRUN45B2\include\lrun.h"
#include "c:\tmp\web\init.c"
#include "c:\tmp\web\run.c"
#include "c:\tmp\web\end.c"
```

## 从 Unix 命令行运行 Vuser

LoadRunner 中包括 Unix shell 脚本实用程序 *run_db_Vuser.sh*，该程序可以像虚拟用户一样自动执行相同的操作，但它只能在命令行中执行。它可以有选择且独立地执行每个回放步骤。对于要在 Unix 上回放的调试测试而言，这是一个非常有用的工具。

将文件 *run_db_Vuser.sh* 置于 $M_LROOT/bin 目录中。要回放 Vuser 类型，请键入：

```
run_db_Vuser.sh   Vuser.usr
```

您还可以使用下列命令行选项：

| | |
|---|---|
| *-cpp_only* | 此选项用于启动预处理阶段。此进程的输出为 “Vuser.c” 文件 |
| *-cci_only* | 此选项用于运行编译阶段。“Vuser.c” 文件将用作输入，而产生的输出为 “Vuser.ci” 文件 |
| *-exec_only* | 此选项通过将 “Vuser.ci” 文件作为输入并通过回放驱动程序运行它，从而运行 Vuser |
| *-ci ci_file* | 通过此选项可以指定要运行的 .ci 文件的名称和位置。第二个参数包含该 .ci 文件的位置 |
| *-out output_directory* | 通过此选项可以确定执行各种进程时所创建的任何输出文件的位置。第二个参数就是目录名和位置 |
| *-driver driver_path* | 通过此选项可以指定用来运行 Vuser 的实际驱动程序可执行文件。默认情况下，该驱动程序可执行文件是根据 VuGen.dat 文件中的设置确定的 |

注意，在运行 run_db_vuser 时只能使用前 3 个选项之一。

## 指定 Vuser 行为

由于 VuGen 将 Vuser 脚本和 Vuser 行为作为两个独立的源进行创建，因此您可以配置用户行为，而无需直接引用 Vuser 脚本。例如，等待时间、步时间、循环迭代、日志记录等。这意味着可以很容易地更改 Vuser 的配置，并可以针对同一个 Vuser 脚本存储多个这样的 “配置文件”。

默认情况下，“Vuser.cfg”文件将负责按照 VuGen 的 “运行时设置”对话框中指定的方式定义此行为。您可以针对不同的用户行为保存此文件的几个版本，然后运行引用相关 *.cfg* 文件的 Vuser 脚本。

您可以从 LoadRunner Controller 中运行具有相关配置文件的 Vuser 脚本。要执行此操作，请在 Vuser 命令行中添加以下内容：

```
-cfg c:\tmp\profile2.cfg
```

有关命令行参数的信息，请参阅第 812 页的 “命令行参数”。

注意，您无法在 VuGen 中控制行为文件。 VuGen 将自动使用与 Vuser 同名的 *.cfg* 文件。（当然，您可以将该文件重命名为 “Vuser.cfg”。）但是，您可以通过在驱动程序命令行末尾添加上面提到的 -cfg 参数，在命令行中手动完成此操作。

---

**注意**：Unix 实用程序 *run_db_vuser* 尚不支持此选项。

---

## 命令行参数

Vuser 在调用时可以接受命令行参数。有几个 LoadRunner 函数可供引用（**lr_get_attrib_double** 等）。使用 LoadRunner Controller 可以向 Vuser 发送命令行参数，方法是将参数添加到脚本窗口的命令行条目中。

在 VuGen 中运行 Vuser 时，将无法控制命令行参数。但是，可以在 Windows 命令行中手动执行此操作，方法是在行结尾处 （其他所有驱动程序参数之后）添加参数，例如：

```
mdrv.exe -usr c:\tmp\Vuser\Vuser.usr    -out c:\tmp\vuser
vuser_command_line_params
```

---

**注意**：Unix 实用程序 *run_db_vuser* 尚不支持此选项。

---

# 录制 OLE 服务器

VuGen 当前不支持 OLE 应用程序的录制。这些应用程序的实际进程不是由标准的进程创建例程启动的，而是由 OLE 自动系统启动的。不过，您可以基于下列规则为 OLE 应用程序创建 Vuser 脚本。

有两种 OLE 服务器：可执行服务器和 DLL。

## DLL 服务器

如果服务器为 DLL，则它最终会加载到应用程序进程空间中，并且 VuGen 将录制对 LoadLibrary 的调用。在这种情况下，您甚至不会意识到它是 OLE 应用程序。

## 可执行服务器

如果服务器是可执行的，则必须采用特殊方式在 VuGen 中调用该可执行服务器：

- 首先，确定实际需要录制的进程。在大多数情况下，客户知道应用程序可执行文件的名称。如果客户不知道该应用程序的名称，请从 NT 任务管理器中调用该程序并确定其名称。
- 标识所需的进程后，在 VuGen 中单击 “开始录制”。当系统提示您输入应用程序名时，输入 OLE 应用程序的名称，后接 “/Automation” 标志。然后，采用常规方式 （不通过 VuGen）启动用户进程。 VuGen 将录制正在运行的 OLE 服务器，且不会调用它的其他副本。在大多数情况下，这些步骤足以启用 VuGen 录制 OLE 服务器的操作。
- 如果在录制时仍然遇到困难，则可以使用 *CmdLine* 程序确定未直接启动的进程的完整命令行。（该程序可从 CSO 网站的 “Patches” 部分获得，其网址为 *http://support.mercuryinteractive.com*）

### 使用 CmdLine

在下例中， *CmdLine.exe* 用于确定进程 MyOleSrv.exe （由某个其他进程启动）的完整命令行。

**要确定其完整的命令行，请执行下列操作：**

1. 将 *MyOleSrv.exe* 重命名为 *MyOleSrv.orig.exe*。
2. 将 *CmdLine.exe* 与应用程序置于同一目录中，并将其重命名为 *MyOleSrv.exe*。

3 启动 *MyOleSrv.exe*。它将发出一个弹出式消息，其中包含原始应用程序的完整命令行 （包括附加信息），并将这些信息写入 *c:\temp\CmdLine.txt* 中。

4 还原为旧名称，并采用正确的命令行参数从 VuGen 中启动 OLE 服务器 *MyOleSrv.exe*。采用常规方式 （不通过 VuGen）启动用户应用程序。在大多数情况下， VuGen 会正确录制。

**如果您在录制时仍然有困难，请继续执行下列步骤：**

1 将 OLE 服务器重命名为 MyOleSrv.1.exe，并将 CmdLine 重命名为 MyOleSrv.exe。

2 将环境变量 “CmdStartNotepad”和 “CmdNoPopup”设置为 1。有关 CmdLine 环境变量的列表，请参阅第 814 页的 “CmdLine 环境变量”。

3 启动应用程序 （不是从 VuGen 中）。将打开记事本，其中包含完整的命令行。检查命令行参数。多次启动应用程序，并比较命令行参数。如果每次调用应用程序时的参数都相同，则可以重置 CmdStartNotepad 环境变量。否则，请将其设置保留为 “1”。

4 在 VuGen 中，采用命令行参数调用 MyOleSrv.1.exe 程序 （在 “记事本”窗口中使用 “复制 / 粘贴”）。

5 启动应用程序 （不是从 VuGen 中）。

### CmdLine 环境变量

您可以通过下列环境变量控制 CmdLine 的执行：

| | |
|---|---|
| CmdNoPopup | 如果设置，则不会显示弹出窗口 |
| CmdOutFileName | 如果设置，且它不为空，则 CmdLine 将尝试创建此文件而不是 c:\temp\CmdLine.txt |
| CmdStartNotepad | 如果设置，记事本中将显示输出文件。（最好是和 CmdNoPopup 一起使用） |

# 检查 .dat 文件

VuGen 使用两个 .dat 文件：vugen.dat 和 mdrv.dat。

## vugen.dat

此 vugen.dat 文件驻留在 M_LROOT\dat 目录下，包含关于 VuGen 的一般信息，VuGen 和 Controller 都将用到这些信息。

```
[Templates]
RelativeDirectory=template
```

**Templates** 部分指明 VuGen 协议的模板的位置。默认条目指明它们位于相对的 *template* 目录下。在 *template* 目录下，每个协议都有一个子目录，其中包含该协议的模板文件。

接下来是 **GlobalFiles** 部分。

```
[GlobalFiles]
main.c=main.c
@@TestName@@.usr=test.usr
default.cfg=test.cfg
default.usp=test.usp
```

**GlobalFiles** 部分包含当您新建测试时， VuGen 复制到 test 目录中的文件的列表。例如，如果您的测试名为 “user1”，则 VuGen 会将 *main.c*、 *user1.usr* 和 *user1.cfg* 复制到 test 目录中。

**ActionFiles** 部分包含特定文件的名称，该文件包含 Vuser 执行的操作以及执行循环的操作。

```
[ActionFiles]
@@actionFile@@=action.c
```

除上述设置之外， *vugen.dat* 还包含指明操作系统的设置以及与编译有关的其他设置。

## mdrv.dat

在 mdrv.dat 文件中，定义库文件和驱动程序可执行文件的位置的每个协议都分别对应一个部分。下一部分描述为定义新的协议需要在此文件中添加的内容。

# 添加新的 Vuser 类型

要向 VuGen 中添加新的 Vuser 类型 / 协议，需要：

- 使用新协议的设置编辑 *mdrv.dat* 文件
- 添加 *.cfg* 文件
- 插入 *.lrp* 文件
- 创建模板目录

## 编辑 mdrv.dat 文件

首先，编辑驻留在 *M_LROOT\dat* 目录中的 *mdrv.dat* 文件。使用以下列表中的所有适用参数添加新的 Vuser 类型部分。

[<*extension_name*>]
ExtPriorityType=< {internal, protocol}>
WINNT_EXT_LIBS=< 适用于 Windows NT 的 dll 名称 >
WIN95_EXT_LIBS=< 适用于 Windows 95 的 dll 名称 >
SOLARIS_EXT_LIBS=< 适用于 Solaris 的 dll 名称 >
LINUX_EXT_LIBS=< 适用于 Linux 的 dll 名称 >
HPUX_EXT_LIBS=< 适用于 HP 的 dll 名称 >
AIX_EXT_LIBS=< 适用于 IBM 的 dll 名称 >
LibCfgFunc=< 配置函数名 >
UtilityExt=< 其他扩展名列表 >
WINNT_DLLS=< 要加载到解释器上下文中的 dll （适用于 Windows NT） >
WIN95_DLLS=< 要加载到解释器上下文中的 dll （适用于 Windows 95） >
SOLARIS_DLLS=< 要加载到解释器上下文中的 dll （适用于 Solaris） >
LINUX_DLLS=< 要加载到解释器上下文中的 dll （适用于 Linux） >
HPUX_DLLS=< 要加载到解释器上下文中的 dll （适用于 HP） >
AIX_DLLS=< 要加载到解释器上下文中的 dll （适用于 IBM） >
ExtIncludeFiles=< 额外的包含文件。多个文件可以用逗号分隔 >
ExtCmdLineConc=< 额外的命令行 （如果 attr 存在连接值） >
ExtCmdLineOverwrite=< 额外的命令行 （如果 attr 存在覆盖值） >
CallActionByNameFunc=< 解释器 exec_action 函数 >
GetFuncAddress=< 解释器 get_location 函数 >
RunLogicInitFunc=<action_logic 初始化函数 >
RunLogicRunFunc=<action_logic 运行函数 >
RunLogicEndFunc=<action_logic 结束函数 >

例如， Oracle NCA Vuser 类型可以表示为：

```
[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 无需修改代码即可处理新的 Vuser 类型。但是可能需要添加特殊的视图。

VuGen 没有附带通用的驱动程序，但是您可以自定义某个现有的驱动程序。要使用自定义的驱动程序，请修改 *mdrv.dat*。使用平台和现有驱动程序添加行，然后使用自定义驱动程序名 （以 < 平台 >_DLLS=<my_replay.dll 名 > 格式）添加新行。例如，如果 SAP 回放 dll 名为 SAPPLAY32.DLL，请在 *mdrv.dat* 的 [sap] 部分中添加下列两行：

```
WINNT=sapdrv32.exe
WINNT_DLLS=sapplay32.dll
```

### 添加 CFG 文件

您可以有选择地指定配置文件，以便为协议设置默认的 “运行时设置”和 “录制”选项。您可以在 *mdrv.dat* 文件的 **LibCfgFunc** 变量中定义此文件，也可以在模板下的新协议子目录中放置一个名为 *default.cfg* 的配置文件。以下是 default.cfg 的示例。

```
[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
```

## 插入 LRP 文件

在 dat/protocols 目录中，可以插入定义协议的 *lrp* 文件。此文件的“Protocol”、“Template”、“Vugen”和 “API”部分中包含与协议对应的配置信息。某些协议可能包含一些附加部分，对应于附加的运行时设置选项。

“Protocol”部分包含协议的名称、类别、描述和位图位置。

```
[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”部分指明脚本各部分的名称和默认的测试名。

```
[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”部分包含关于录制和回放引擎的信息，以及必要的 DLL 和运行时文件。

“API”部分包含内部宏的信息。

您可以使用协议目录中的某个现有 lrp 文件作为新协议的基础。

### 指定模板

添加 *lrp* 文件后，请在 *M_LROOT*/template 下插入一个子目录，其名称对应于 *lrp* 文件中定义的协议名。在此子目录中，插入用来定义默认的常规设置和运行时设置的 *default.cfg* 文件。

如果要对协议的所有脚本使用全局头文件，请添加文件 *globals.h*。此文件应该包含指向新协议的头文件的 include 语句。例如， template/http 子目录可以包含 *globals.h* 文件，该文件将 VuGen 定向至 include 目录中的 *as_web.h* 文件。

#include #as_web.h"

# 第 XVIII 部分

## 附录

# A

# Java 环境：综合指南

本章讨论 Java 环境。其中将解释某些术语，这些术语对于理解和配置某些 Java 应用程序的现有 Java 环境是必不可少的。尽管 Java 语言是跨平台的，但本文档只描述适用于 Windows NT 的问题。

- 术语
- JDK 版本
- 浏览器
- Java 插件
- 其他环境
- 常见问题

## 关于 Java 环境

Java 编程语言是一种面向对象的高级语言。该语言以易于开发、分布式、解释性、安全、可移植和多线程而闻名。

尽管对于开发人员来说， Java 显得易于使用，但随同它不断发展的技术却使得环境的设置十分复杂。 SUN （Java 的开发者）不断发布 Java、 API 和工具的新版本，使得 Java 世界更加难以管理。

# 术语

**JDK：**Java 开发工具包。一种由 SUN Microsystems 开发的软件开发环境，用于生成 Java 程序。JDK 的每个版本都包含 Java 编译器、Java 解释器、Java 类库、Java 小程序查看器、 Java 调试器和其他工具。 JDK 不能重新发布。

**JRE：**Java 运行时环境。 JDK 的一个子集，提供给想要重新发布运行时环境的最终用户和开发人员。 JRE 由 Java 虚拟机、核心类和支持文件组成。

**JVM：**Java 虚拟机。 Java 运行时环境中负责解释字节码的部分。

**CLASSPATH：**一种环境变量，用于通知 Java 虚拟机到何处查找类文件。它应该包含 JDK 核心库和 Java 程序使用的类库。classpath 变量由一组以分号隔开的条目组成。每个条目可以是一个目录或者一个存档文件名 （jar 或 zip 文件）。当前目录 “.” 也应包含在内。在 JDK 1.1.x 中，类路径变量还应包含 *<JDK>\lib\classes.zip*。在 CLASSPATH 中，类是按照条目的顺序进行搜索的。

**PATH：**一种包含目录列表的环境变量，如果操作系统在工作目录中找不到可执行文件，就在该目录列表中查找。请使用分号隔开目录名。使用 JDK 时，应将 *<JDK>\bin* 放在 PATH 变量中，以使 JDK 工具 （*java.exe*、 *javac.exe*、 *appletviewer.exe* 等）可用。

**JAR：**Java 存档文件。一种文件格式，用于将 Java 小程序或应用程序 （类文件、图像、声音等）需要的组件和资源打包。此外， JAR 支持数据压缩，这进一步减少了下载时间。根据约定， JAR 文件以 *.jar* 扩展名结束。可以使用 JDK 安装时自带的 jar.exe 工具或者标准的 WinZip 工具查看和解压缩 Jar 文件。

**JIT：**实时编译器。 JIT 编译器是可以代替使用标准 Java 解释器的一种选择。它 “实时”（Java 程序运行时）将全部字节码转换成本机代码。这使得运行时的速度高于 Java 虚拟机解释代码的速度。现在所有版本的 JVM 和浏览器都包含 JIT。在某些情况下，您可能希望禁用 JIT 编译器，例如，当您想要获取堆栈跟踪中的行数而不是 “（编译过的代码）” 打印时，或者当您怀疑 JIT 引起错误时。

**小程序：**一种可以包含在 HTML 页中的 Java 程序，在支持 Java 的浏览器或小程序查看器的上下文中运行。安全管理器可以限制 Java 小程序的操作。

**应用程序：**一种独立的可执行程序。它没有安全性限制或 I/O 限制 （除非特殊的应用）。 Java 应用程序需要解释器来执行。

**字节码：**由 Java 编译器生成，并通过 Java 解释器执行的与计算机无关的代码。

**类文件：**一种包含与计算机无关的 Java 字节码的文件。 Java 编译器生成 ***.class** 文件，供 Java 解释器读取。

**异常：**一种程序执行过程中出现的事件，阻止程序继续执行正常的流程。一般来说，这是意外错误，紧接着会打印输出堆栈跟踪，以便在代码中查找出现异常的确切位置。程序员可以使用 *try*、*catch* 和 *throw* 关键字来创建响应特定类型异常的异常处理程序。程序员可以在执行了异常处理程序后选择恢复执行。

**垃圾回收：**自动检测并释放不再使用的内存。 Java 运行时系统执行垃圾回收，以便程序员不必显式地释放对象。

**java.exe：**SUN 的 JDK 中自带的 Java 解释器。

**appletviewer.exe：**无需浏览器即可运行小程序的工具。它是 SUN Microsystems 的 JDK 自带的。 Appletviewer 将 HTML 页作为参数接收，并搜索该页查找其中的 <Applet> 标记。然后加载该小程序。

**javap.exe：**用于分解 Java *.class* 文件的工具。它是 SUN 的 JDK 自带的。在缺少源代码的情况下，该工具可以用来验证 API 类。

**javac.exe：**随 SUN 的 JDK 一起提供的 Java 编译器。

**jview.exe：**Microsoft 的 Java 虚拟机。它集成在 Internet Explorer 中。

**HTML 属性：**加载 Java 小程序的 HTML 页，必须包含一个指定有关该小程序的名称、位置及其他属性的详细信息的 <Applet> 标记。该标记中的三个属性提供了查找主小程序类文件的位置的完整规范，这三个属性是：

*code* 属性指定主小程序类文件的名称。

*codebase* 属性指定包含该文件和其他文件的目录的 URL。

*archives* 属性指定其他任何存档文件：包含类文件和资源文件的 *jar/zip/cab* 文件。这些存档文件放在基本代码下。

**classes.zip：**包含 JDK1.1.x 的所有 Java 核心类的 zip 文件。可以在 *<JDK>\lib\classes.zip* 下找到它。这些类对于运行任何 Java 应用程序都是必不可少的，而且必须将其放在 CLASSPATH 中。

**rt.jar：**包含 JDK1.2.x 和 JDK1.3 的所有 Java 核心类的 jar 文件。可以在 *<JDK>\jre\lib\rt.jar* 下找到它。该 jar 文件不需要位于 CLASSPATH 中，因为 JVM 会自动检索它。然而，如果 Xboot classpath（见下文）被替换了，请确保 rt.jar 存在。

**JavaScript：**一种 HTML 的扩展，通过它可以在网页中加入一些动态功能。不要把该术语与 *Java* 语言混淆。 JavaScript 通常由浏览器中的一种特殊机制来解释，这种机制与 Java 语言虚拟机没有任何关系。但可以在 JavaScript 内调用 Java 的标准 JDK 中的静态方法。这将最终导致加载 JVM 及其 DLL。

# JDK 版本

以下部分描述 Java 开发工具包 (JDK) 的各种版本。

## JDK 1.1

下面是 JDK1.1 中 *java.exe* 可执行文件的命令行参数：

| 命令行参数 | 描述 |
|---|---|
| *-version* | 打印准确的内部版本 |
| *-verbose* | 打开详细模式，并从系统类加载器生成有关每个正在被加载的类的名称和路径的打印输出 |
| *-noasyncgc* | 禁用异步垃圾回收 |
| *-verbosegc* | 发生垃圾回收时打印一条消息 |
| *-noclassgc* | 禁用类垃圾回收 |
| *-ms< 数字 >* | 设置初始的 Java 堆大小。<br>例如：-ms256M |
| *-mx< 数字 >* | 设置 Java 堆的最大大小 |
| *-classpath < 用 ";" 隔开的目录 >* | 列出在其中查找替换 CLASSPATH 变量的类的目录 |
| *-noverify* | 不验证任何类 |
| *-nojit* | 禁用 JIT 编译器 |
| *-D< 名称 >=< 值 >* | 设置系统属性 |

**JDK1.1 版本：**

第一代 JDK（版本 1.0.1 和 1.0.2）使用旧的事件模型。JDK1.1 引入了新的事件模型。目前， SUN 不再正式支持 JDK1.15 之前的老版本。

支持的版本是：1.1.6、1.1.7a、1.1.7b 和 1.1.8。JDK1.1.6 据说不稳定。其他版本之间有很小的差别，主要在于安全性修复方面。

**JDK 1.1 示例：**

下例将演示如何不使用 JIT 编译器运行一个名为 TestApp 的应用程序，以及从 jdk 和当前目录中取得类。

```
java –nojit –classpath d:/jdk1.1.7b/lib/classes.zip;.  TestApp
```

下例将演示将 Java 初始堆大小和最大堆大小都设为 256M 时如何运行 TestApp 应用程序。该测试应用程序有两个参数。类从全局 CLASSPATH 变量取得。

```
java –ms256M –mx256M TestApp apparg1 apparg2
```

示例 3

下例将演示如何检查 PATH 中放置的当前 Java 版本：

```
java –version
```

## JDK 1.2

在 JDK1.1.x 中，对类的搜索是根据 CLASSPATH 环境变量完成的。对于小程序，类的搜索也是在 HTML 中指定的基本代码和存档文件中进行的。在 JDK1.2.x 中，一种新的元素决定类的位置："bootstrap classpath"。 *bootstrap classpath* 位于 classpath 之前。默认情况下，它包含放在 *<JDK>\jre\lib\rt.jar* 下的 JDK 核心类。可以使用 ñ*Xbootclasspath* 命令行变量来更改 bootstrap classpath。此外， *javac* 支持一个类似的选项 （"-bootclasspath"），该选项可以用来更改您编译的平台类。

JDK1.2.x 包括 JFC1.1 库的类以及 Java IDL CORBA 类。所有这些都放在系统的 *rt.jar* 文件中。

在 JDK1.2 或更高版本中，有两种类型的虚拟机：*Classic* 虚拟机 （位于 *<JDK>/jre/bin/classic/jvm.dll* 下）和 *HotSpot* 虚拟机 （位于 *<JDK>/jre/bin/hotspot/jvm.dll* 下）。 HotSpot 技术可以获得比 Classic VM 更卓越的性能。要使用 Classic VM，请使用 *-classic* java 命令行选项。

在 JDK1.2 中，可以通过仅指定 jar 名称来运行 Java 应用程序：*java.exe –jar <jar 文件名>*。在这种情况下， JVM 打开 jar 并搜索清单文件 (Manifest.mf)。该文件应该指定包含 Main 方法的类的名称。然后 JVM 加载该特定类并运行 Main 方法。

同 Java 平台的旧版本相比， JDK 1.2 在功能、性能、安全性和全球化支持方面都有所提高。

下表显示了 JDK 1.2 中 java.exe 的非标准命令行选项。

| 命令行参数 | 描述 |
| --- | --- |
| -Xbootclasspath:< 由 “;” 隔开的目录和 zip/jar 文件 > | 设置 bootstrap 类和资源的搜索路径 |
| -Xbootclasspath/p:< 由 “;” 隔开的目录和 zip/jar 文件 > | 前置 bootstrap 类和资源的默认搜索路径 |
| -Xbootclasspath/a:< 由 “;” 隔开的目录和 zip/jar 文件 > | 后置 bootstrap 类和资源的默认搜索路径 |
| -Xnoclassgc | 禁用类垃圾回收 |
| -Xms< 大小 > | 设置初始 Java 堆大小。<br>例如：-Xms128M |
| -Xmx< 大小 > | 设置 Java 堆的最大大小 |
| -Xrunhprof[:help]|[:< 选项 >=< 值 >, ...] | 执行堆、 CPU 或监视器配置 |

## -Xbootclasspath 的用法

*_Xbootclasspath* 参数的功能非常强大，使用时要小心。确保使用适当的 Java 平台类。这些类放在 *<JDK>\jre\lib\rt.jar* 中。

如果使用 *–Xbootclasspath:<…>*，您必须指定 rt.jar 文件的完整路径。如果不指定，将无法启动 VM。您也不能在该参数中指定属于另一版本的 JDK 的类。如果使用前置 (/p) 或后置 (/a) 格式，则可以忽略 rt.jar 路径。

该参数的语法与 CLASSPATH 环境变量的用法类似。当条目包含空格时需要使用引号。

## 安装 JDK1.2.x

安装 JDK1.2.x 需要将 java.exe 文件放到 *<Winnt>\system32* 目录下，以便用户不必在 PATH 变量中放入 *<JDK>\bin* 目录。使用其他 JDK 版本时尤其需要注意这一点。 *<Winnt>\system32* 目录位于 PATH 变量中，并可能出现在其他版本的 *<JDK>\bin* 目录之前。在 JDK 版本之间切换时，这很可能导致意外的运行时冲突 （运行时类与 JVM 不匹配）。

### JDK1.2 版本：

现有的 JDK1.2 版本包括 1.2、 1.2.1 和 1.2.2。它们之间只有很细微的差别。

### JDK1.2 示例：

下例将阐释如何运行 TestApp 应用程序，以及在 JDK1.2.2 Java 平台类之前使用 LoadRunner 类。这两条语句执行相同的任务：

```
java  -Xbootclasspath:c:\LoadRunner\classes;c:\jdk1.2.2\jre\lib\rt.jar
TestApp
java  -Xbootclasspath/p:c:\LoadRunner\classes TestApp
```

下例将阐释将 Java 初始堆大小和最大堆大小都设为 256M 时如何运行 TestApp 应用程序。在 Java 应用程序中使用 肭 参数设置系统属性：

```
java  -Xms256M ñXmx256M  -Dtestprop=propvalue  TestApp
```

## JDK 1.3

JDK1.3 与旧版本的不同之处主要在于它的虚拟机 (VM)。 JDK1.3 作为默认的 VM 实现随 Java HotSpot 客户端 VM 一起提供。要切换到 Classic VM，请使用 *-classic* 选项。 Java 插件 1.3 VM 只提供 HotSpot VM。要在该插件中使用 Classic VM，请将 <JDK>*/jre/bin/classic* 目录从 JDK 目录复制到 <plug-in>/*bin* 目录中，并在控制面板中设置 -classic 标志。

JNDI 类和 RMI/IIOP 包含在 *rt.jar* 文件中。

同 Java 平台的旧版本相比，该版本在功能、性能、 GUI、安全性和全球化支持方面都有所提高。

### JDK1.3 版本：

现有的 JDK1.3 版本包括 1.3 和 1.3.0_01。它们之间只有很细微的差别。

### JDK1.3 示例

下例将阐释不使用 JIT 编译器时如何运行 TestApp 应用程序，并提供有关类的源目录的详细打印输出。

```
java -verbose –Djava.compiler=NONE TestApp
```

下例将阐释如何使用 Classic VM TestApp 运行 TestApp 应用程序：

```
java -classic TestApp
```

## 浏览器

以下部分描述如何将 Netscape 和 Internet Explorer 用于 Java 小程序或应用程序。

### Appletviewer

Appletviewer 是一个可以不使用浏览器而运行 Java 小程序的命令行工具。它接受含有 <Applet> 标记的 URL 地址 （HTML 或文件）。

Appletviewer 工具支持大多数 java.exe 命令行参数。但是，要使用这些参数，必须使用 “-J” 前缀。例如：*-J-verbose*、 *-J-Xbootclasspath:<…>* 等。

### Appletviewer 示例

下例将显示如何运行一个从 Web 上的 HTML 文件引用的小程序。

```
appletviewer http://www.apptest.com/test/test.html
```

下例将显示如何运行一个从本地 HTML 文件引用的小程序，并将类置于 rt.jar 之前（仅 JDK1.2 及更高版本）。

```
appletviewer -J-Xbootclasspath/p:c:\LoadRunner\classes
file:c:\apptest\test.html
```

下例将显示如何运行一个从当前目录中的 HTML 文件引用的小程序，在详细打印输出中显示类，并指定堆大小的最大值和最小值 （JDK1.1 样式）。

```
appletviewer -J-verbose -J-ms256M -J-mx256M test.html
```

## Internet Explorer

### Java 控制台

Java 小程序向 *stdout* 或 *stderr* 发送的输出消息被打印到 Internet Explorer Java 控制台中。可以通过在浏览器中使用 “查看” 菜单打开 Java 控制台。

如果在浏览器中看不到 Java 控制台选项，则必须在 Internet Explorer 配置中启用 Java 控制台选项。要实现这一点，请右键单击 Internet Explorer 图标并选择 “属性” 选项。然后，在 “高级” 选项卡中找到 “Java 虚拟机” 部分。选择 “已启用 Java 控制台” 复选框，并重新启动浏览器。

### Internet Explorer Java 虚拟机版本

在 Internet Explorer 中安装的组件之一是 Microsoft Java VM。在浏览器中安装的 JVM 的版本是独立的，与浏览器的版本不同。它在 Java 控制台的第一行指定。例如，“Microsoft (R) VM for Java, 5.0 版本 5.0.0.3176” 表示您的 JVM 支持 IE5，而且它特定的内部版本号是 3176。

您可以选择只更新 Internet Explorer 中的 JVM 组件以使用较新的内部版本，但是不能降级。

### Internet Explorer Java 类

Internet Explorer 的 JVM 有自己的系统类。可以在位于 *<Winnt>\Java\Packages* 目录下的 zip 文件中找到这些类。浏览器自动将所有这些 zip 文件插入到 CLASSPATH 中。

某些小程序可能会下载 zip 文件并把它们放到浏览器目录下。这些 zip 文件中可能包含导致冲突的类。

## Netscape

### Java 控制台

Java 小程序向 *stdout* 或 *stderr* 发送的输出消息被打印到 Netscape Java 控制台中。要打开 Java 控制台，请选择 “通讯器” > “工具” > “Java 控制台”。

Java 控制台提供几个通过在键盘上键入某个字母就可以激活的选项。键入 “?” 可以查看所有选项。这些选项包括：

b: 中断至调试器 （仅限 Windows）
c: 清除控制台窗口
d: 将小程序的上下文状态转储到控制台
f: 终止在终止队列中的对象
g: 垃圾回收
h: 打印该帮助消息
l: 捕获小程序加载到目录中的所有类
m: 将当前内存的使用情况打印到控制台
q: 隐藏控制台
s: 将内存概要转储到 “*memory.out*”
t: 将线程信息转储到 “*memory.out*”
x: 将内存转储到 “*memory.out*”
X: 将内存 （详细内容）转储到 “*memory.out*”
0-9： 将小程序调试级别设置为 <n>

注意，控制台刷新得很频繁，因而很难从中复制文本。如果您选择了一块文本并且浏览器发送了打印输出，它将替换选定的文本。请耐心等待，直到浏览器再没有内容要发送。

### Netscape Java 类

Netscape 的安装附带它自己的 JVM 类。可以在位于 *<Netscape 文件夹>\Communicator\Programs\java\classes* 目录下的所有 jar 文件中找到这些类。包含 Java 系统类的主 jar 名为 java40.jar。 Netscape 的一个名为 *iiop10.jar* 的 jar 文件中也包含 CORBA Visigenic3.0 类。有时，该 jar 文件会在使用其他 CORBA 类的 CORBA 应用程序中引起冲突。用户可能因此而选择从此目录中删除它。浏览器自动将所有这些 jar 文件插入到 CLASSPATH 中。

注意，某些环境中可能包括其他的 jar，它们不是 Netscape 自带的，而是进行其他安装时放在那里的。这些 jar 中可能包含导致冲突的类。可以分析 verbose 标志（调试级别：0-9）以了解某个特定类的源。

# Java 插件

Java 插件是一种工具，通过该工具，可以使用 SUN 的 JRE 在 Internet Explorer 或者 Netscape 中运行 Java 小程序，不需要使用浏览器的内部 JVM。当 Web 浏览器第一次遇到指定使用 Java 插件的网页时，该浏览器必须下载并安装需要的文件。当以后遇到指定使用 Java 插件的网页时，它将立即从用户的硬盘上调用出来，并且使用安装的 SUN 的 JRE 呈现该小程序。

## Java 插件控制面板

Java 插件是控制面板自带的，通过它可以配置某些选项。从 “开始菜单” > “程序” > “Java Plug-in 控制面板” 打开控制面板。

可以配置下列选项：启用 Java 控制台、使用命令行参数、更改 SUN 的 JVM 运行时版本、启用 JIT 编译器等。

从插件 1.3 开始，可以通过选择 Windows 控制面板中的 “Java Plug-in 控制面板” 项来打开该控制面板。

## Java 控制台

Java 小程序向 *stdout* 或 *stderr* 发送的输出信息将打印到插件 Java 控制台中。启用 Java 控制台是从 Java Plug-in 控制面板中的 “基本” 选项卡实现的。使用插件运行小程序时，唯一需要调用的控制台是插件控制台，而不是浏览器控制台。

## Java 插件 JVM 版本

使用 Java 插件时，可以在不同的 SUN JVM 版本之间切换。选择要运行哪个 JVM 是在 Java 插件控制面板的 “高级” 选项卡中执行的。可以选择使用计算机上安装的任何 JRE/JDK。要验证 Java 插件中当前运行的是哪个版本，请查看 Java 控制台中的前两行。例如，下例将显示版本为 1.2.2 的 Java 插件的输出，当前运行的是 1.1.7B JVM：

```
Java™ 插件：版本 1.2.2.px
使用 JRE 版本 1.1.7B
```

### HTML Java 插件标记

为了使浏览器能够识别使用 Java 插件的网页，必须用某些特定标记替换 <Applet> 标记。每一种浏览器都提出了自己用来处理 Java 插件的 HTML 标记。Internet Explorer 查找具有特定的 Java 插件和小程序属性的 <Object> 标记，然后使用 Microsoft 的 COM/ActiveX 技术加载该标记。为了使用 Navigator 的插件体系结构加载小程序， Netscape Navigator 查找具有特定的 Java 插件和小程序属性的 <Embed> 标记。可以在 html 页中结合使用这两种标记，以便使用 Netscape 和 Internet Explorer 都能通过 Java 插件加载小程序。

有一种称为 Java 插件 HTML 转换器的简单工具，它可以非常方便地转换 HTML 页以便使用 Java 插件。

## 其他环境

### IBM

IBM 拥有自己的 Java 工具。这些工具包括 Java 编译器、虚拟机和略有不同的 JDK 运行时类。

### Oracle Jinitiator

Oracle 拥有自己的虚拟机，称为 Jinitiator。该虚拟机可以用于应用程序，或者作为浏览器的插件。它有自己的 JDK 类，这些类基于 SUN 的 JDK 1.1.5 或 JDK 1.1.7，并且做了各种更改。可以通过使用 *“java -version”* 来识别该虚拟机。这将会得到以“o”结尾的 JDK 版本（例如，“java 版本 1.1.5o”）。Jinitiator 插件可以通过查看 HTML 页，或者通过查找目录 *“< 驱动器 >:\Program Files\Oracle\Jinitiator < 版本 > ”* 来识别。

### BEA WebLogic

尽管没有虚拟机，但 BEA 有自己的工具集。它拥有自己的 RMI 编译器 (rmic.exe)。该编译器用来生成存根和骨架，它们与 SUN 的 *rmic* 工具产生的存根和骨架不同。

### VBJ、 OWJAVA

Vusigenic 和 Iona 拥有自己的 java.exe。通常它们是由其 CORBA 包和安装附带提供。这些不是最新实现的虚拟机，而是 SUN 的虚拟机的包装，该包装带有某些与 CORBA 相关的附加功能。它们接收的参数与 SUN 的虚拟机相同。

### Gemstone/J

Gemstone 拥有自己的 JVM 和 JDK 版本。如果运行 java -version 命令，将会发出下列消息：

```
java 版本 “3.2p2”
HotSpot VM （3.2p2，混合模式，内部版本 3.2p2-Wed-Feb-23-17:58:31-
PST-2000-build-97）
```

## 常见问题

**问题 1：**如何将应用程序的输出重定向到某个文件？

要将 Java 应用程序的 *stdout* 和 *stderr* 重定向到文件，请在 java.exe 命令行的末尾添加 “**>out.txt 2>&1**”。这就将输出重定向到一个名为 out.txt 的文件了。（仅适用于 NT）

**解答：**在 Internet Explorer 中，可以通过启用 Java 记录选项来重定向输出：右键单击 Internet Explorer 图标，选择 “属性”。单击 “Java 虚拟机” 部分中的 “高级” 选项卡，然后选择 “已启用 Java 日志记录” 复选框。重新启动浏览器。

现在，输出已经重定向到文件 *<Winnt>\Java\javalog.txt*。

**问题 2：**如何判断某个特定类是否在类路径中？

**解答：**您可以使用 *javap.exe* 命令行工具。运行 javap.exe < 完整的类名 >（不带 “.class” 后缀）。如果该类不在 CLASSPATH 中，您将收到以下错误消息：“找不到类 < 名称 >”。如果该类在 CLASSPATH 中，您将收到它的全部字段和方法。

**问题 3：**如何从浏览器中转储类？

**解答：**通过 Netscape，可以在运行小程序时捕获服务器类，并可以将它们转储到本地计算机上。在 Java 控制台上键入字母 “L” 启用和禁用该选项。在 Java 控制台中，您将看到以下消息：“# 已启用类文件捕获”，并且所有的服务器类将被转储到 *<Netscape 文件夹>\Communicator\Programs\< 以 URL 地址建立的目录名 >*。

**问题 4：**在我的浏览器中，我看不到用来打开 Java 控制台的选项。它在哪？

**解答：**在 Internet Explorer 中，如果在 “查看” 菜单下看不到该选项，那么您需要在 Internet Explorer 配置中启用该选项。要启用该选项，请右键单击 Internet Explorer 图标，选择 “属性” 选项。然后，在 “Java 虚拟机” 部分的 “高级” 选项卡中，选择 “已启用 Java 控制台” 复选框。重新启动浏览器。在 Netscape 中，该选项通常位于 “通讯器” > “工具” 菜单下，或者在早期版本中，直接在 “工具” 或者 “通讯器” 菜单下。

**问题 5：**与 Java 插件一起运行时，为什么在浏览器的 Java 控制台中看不到任何打印输出？

**解答：**浏览器的 Java 控制台仅显示从浏览器的 VM 发送来的消息。使用插件时，您应该使用 Java 插件的 Java 控制台。

**问题 6：**使用 Java 插件时如何打开 Java 控制台？

**解答：**在该插件的控制面板中，选择 “开始菜单” > “程序” > “Java Plug-in 控制面板”。在 “基本” 选项卡中，选择 “显示 Java Console” 复选框。

**问题 7：**我的计算机上同时安装了 JDK1.1 和 JDK 1.2。“PATH” 指向 *<JDK1.1>\bin*。运行 *“java -version”* 时，为什么仍能看到 JDK1.2？如何在不卸载的情况下否定 JDK1.2？

**解答：**在安装 JDK1.2.x 期间，它的 java.exe 也放在 *<Winnt>\system32* 目录下。因为该目录也在 PATH 中，所以会使用 JDK1.2 的 java.exe。您可以重命名或删除 *\system32* 目录下的 java.exe ，这样就可以在不卸载的情况下否定 JDK1.2。

**问题 8：**我遇到一个 *stack-trace* 错误。如何判断 JIT 是否在运行？

**解答：***stack-trace* 包含导致错误的调用的堆栈。每一行都包含类名和涉及到的方法。调用的行数应该显示在圆括号中。如果圆括号中包含的却是 “编译的代码”，这就表示字节码被编译成了本机代码，并且 JIT 正在运行。

**问题 9：**如何在 java.exe 中禁用 JIT？

**解答：**使用 java.exe 的 “-nojit” 命令行选项。在 JDK1.2.x 和 JDK1.3 中没有 “-nojit” 选项。在这些版本中，使用带有以下参数的 java.exe 命令行：

*-Djava.compiler=NONE*

**问题 10：**如何在浏览器中禁用 JIT？

**解答：**解决方案取决于您的浏览器：

**Internet Explorer** – 右键单击 Internet Explorer 图标，选择 “属性” 选项。在 “Java 虚拟机” 部分的 “高级” 选项卡中，选择 “已启用虚拟机的 JIT 编译器” 复选框。重新启动浏览器。

**Netscape** – 重命名负责 JIT 的 DLL，以便不会找到它。该 DLL 的完整路径是：*<Netscape 文件夹>\Communicator\Programs\java\bin\jit3240.dll*

**问题 11：**为什么会遇到 “NoClassDefFoundError”？

“找不到类 java.lang.NoClassDefFoundError。（错误的类路径？）”

“线程 “Main” 中出现异常 java.lang.NoClassDefFoundError：文件”

**解答：**这些错误消息的出现很可能是因为错误地使用了 JDK1.2 的 ñ*Xbootclasspath* 参数。请检查是否指定了 rt.jar 文件的完整路径，该文件位于 *<JDK>\jre\lib\rt.jar* 下。检查参数中的空格。如果有空格，则必须使用引号，如下所示：

```
-Xbootclasspath:“<…>”
```

**问题 12：**为什么会遇到 “无法初始化线程” 错误？

**解答：**如果使用的是 JDK1.1.x，请检查 *<JDK>\lib\classes.zip* 是否在 CLASSPATH 环境变量中。如果使用的是 JDK1.2.x 和 *-Xbootclasspath* 选项，请检查是否指定了 rt.jar 文件的完整路径，该文件位于 *<JDK>\jre\lib\rt.jar* 中。

**问题 13：**为什么会遇到 “UnsatisfiedLinkError”？

如果 Java 应用程序使用了某个库 （或者 DLL）的本机方法，并且该库文件无法找到、已损坏或者由于 Java 的安全性约束不能从特定的类访问，则该错误就会发生。

**问题 14：**如何解决 “OutOfMemoryError”？

**解答：**如果在应用程序执行期间，虚拟机用尽了堆栈或者堆大小，则 OutOfMemoryError 就可能发生。大多数虚拟机将它们的堆和堆栈大小初始化为 16M 到 64M。某些应用程序由于大量的内存使用或者较长的递归调用可能需要更多的内存。加载大量类 （或者 jar）的应用程序在启动时可能也会消耗内存。为了解决该问题，请用 “-ms”和 “-mx”参数 （在 JDK 1.2 中为 “-Xms”和 “-Xmx”）运行虚拟机。注意，堆栈和堆大小的最大值是 512M （虚拟内存）。

---

**注意：**本部分所涉及到的几个问题可能很快就过时了。尽管我们会努力保持该信息是最新的，但我们仍然感谢有关本文档的各种反馈，请将反馈信息发送到 documentation@merc-int.com。

---

# B

# EJB 体系结构和测试

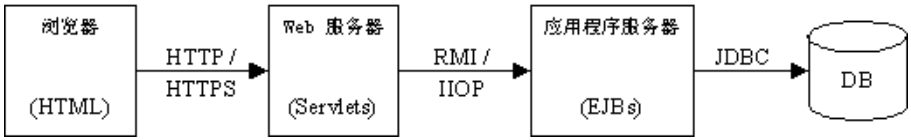
## EJB 简介

Enterprise Java Beans (EJB) 是一种组件体系结构，用于开发和部署基于组件的分布式商业应用程序。

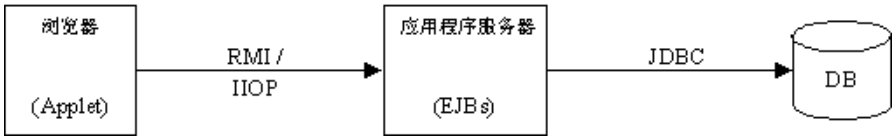
EJB 系统提供了复杂的中间件企业功能，使开发人员可以集中关注模型的实际商业体系结构，并快速创建可伸缩的、事务性的和多用户安全企业应用程序。

## EJB 体系结构

今天的许多 IT 系统都是使用定义完善的 4 层体系结构或 n 层体系结构来建立的。这些体系结构一般都包含基于 Java 技术的某些组件。注意下图中 Web 服务器如何使用 Servlet 以及应用程序服务器如何使用 EJB。EJB 本身使用实体 EJB 包装其他层或者直接使用 JDBC 与数据库对话。




某些系统实现 3 层体系结构，其中 Web 服务器和应用程序服务器在同一主机上运行。虽然这两种功能在同一主机上运行，但是它们仍然使用 Servlet 和 EJB 进行工作。尽管比较少见，但是仍然有 “纯” 3 层体系结构，其中浏览器将加载一个 Java 小程序，Java 小程序使用 RMI、IIOP 或通过 IIOP 的 RMI 协议，直接与应用程序服务器上的 EJB 对话。



# EJB 结构和机制

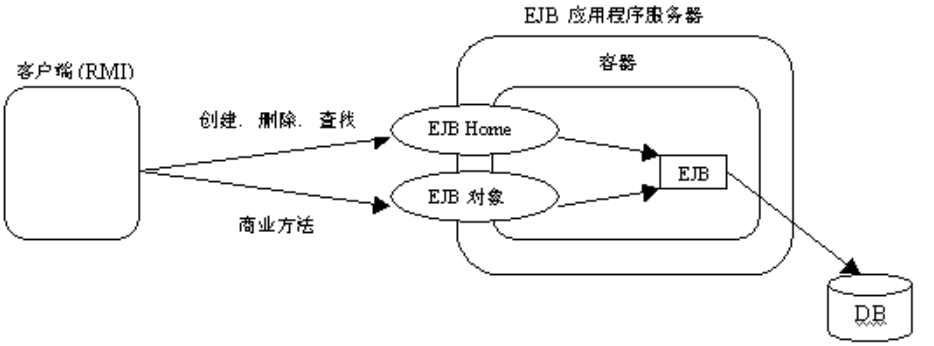
简单的 EJB 是从 3 个基本的 Java 对象 （Java 类）构建的。这些类由 EJB 开发人员创建，它们描述 EJB 及其功能。

这些 Java 对象是：Home 接口、 Remote 接口和 Bean 实现。

- **Home 接口：**使用类似下面的方法为 EJB 提供生存周期服务：create()、remove()、 find()。
- **Remote 接口：**包含 EJB 的商业操作。
- **Bean 实现：**实现 Home 接口和 Remote 接口方法。

开发人员可以使用这些对象，并结合使用部署描述符文件和 EJB 编译器 (ejbc)，来生成那些在远程访问称为商业方法的 EJB 函数时所要求的存根和骨架。 EJB 编译器将整个 Bean 存储在称为 JAR 或 EAR 文件的存档文件中，类似于 WinZip 存储文件和目录的方式。生成的存根和骨架可以在没有编程人员的情况下包装所有较低级别的通信，这样 EJB 对象不必担心客户端 / 服务器通信所涉及的细节。将 EJB 部署到应用程序服务器时，服务器的 EJB 容器将使用存根和骨架类来连接 EJB 和客户端或其他相关的 EJB。该容器还提供 EJB 服务类，如：事务、安全等。

应用程序服务器启动时，它将向部署在系统中的每个 EJB 的 JNDI 树中添加或部署对 Home 存根的命名引用。 JNDI 或 Java 命名目录接口是对 Java 语言的扩展，它提供了通过类似于硬盘文件结构的树形结构中的名称来引用资源的抽象方法。然后应用程序服务器的客户端即可通过 JNDI 来使用这些 Home 存根。每个客户端都将下载一份 Home 存根的副本，以便客户端在本地计算机上使用。客户端使用 Home 存根在应用程序服务器上创建 EJB 实例。在应用程序服务器上，Home 骨架 （由容器控制）将创建 EJB 实例、赋予其状态、与客户端上下文关联、创建 Remote 骨架和创建 Remote 存根。Home 骨架将 Remote 存根返回给客户端。现在客户端即可本地使用该 Remote 存根来调用 EJB 上的商业方法。



可部署的 EJB JAR 应该包含 Home 接口、Remote 接口、Bean 实现、存根和骨架类、部署描述符文件和不是容器的 Classpath 的组成部分的任何被引用的 Java 类文件。（Classpath 是一个环境变量， Java 使用它来查找类文件和包含类文件的存档文件）。部署描述符文件包含诸如 EJB 名称、类、 Home 和 Remote 接口、Bean 类型 （会话或实体）、环境条目、资源工厂引用、 EJB 引用和 JNDI 名称。

有两种部署描述符：

- **序列化的部署描述符：**包含关于某个 EJB 的序列化信息的文件。此文件具有“.ser”扩展名。 [EJB1.0 规范 ]
- **XML 部署描述符：**包含 JAR 文件中所有 EJB 对象相关信息的 XML 文件。该文件的名称是 ejb-jar.xml。这些描述符存储在 jar 或 ear 文件的 META-INF 目录下。某些 EJB 容器具有 EJB 部署所需的其他部署描述符文件。例如， Weblogic 要求 weblogic-ejb-jar.xml。 [EJB1.1 规范 ]

EJB 对象有下列 3 种类型：

- **无状态会话 Bean：**提供单一使用服务，在代表客户端的多次调用中不保持任何状态。
- **有状态会话 Bean：**提供对话交互并在多次调用中保持某个状态。
- **实体 Bean：**永久 EJB，一般用于代表数据库中的数据。

# 在常见应用程序服务器中部署

| 类型 | EJB 规范 | 部署描述符 | 部署方法 |
|---|---|---|---|
| WebLogic4.x | 1.0 | .ser 文件 | - 使用 ejbc 工具创建可部署的 Jar。<br>- 通过在 <WL>/weblogic.properties 属性中指定到 EJB JAR 文件的完整路径来部署：weblogic.ejb.deploy |
| WebLogic5.x | 1.1 | ejb-jar.xml<br>weblogic-ejb-jar.xml | |
| WebLogic6.x | 1.1, 2.0 | ejb-jar.xml<br>weblogic-ejb-jar.xml | - 使用 ejbc 工具创建可部署的 Jar。<br>- 通过将 EJB JAR/EAR 复制到以下目录进行部署：<WL>/config/< 域 >/applications |
| WebSphere3.0 | 1.0 | .ser 文件 | - 使用 jetace 工具创建可部署的 Jar。<br>- 通过管理控制台 UI 部署。<br>（已部署的 EJB 将放在以下目录中：<WS>/deployedEJB） |
| WebSphere3.5 | 1.0, 1.1 （部分） | | |
| WebSphere 4.0 | 1.0, 1.1 | ejb-jar.xml | - 使用应用程序汇编工具来创建 Bean 并进行部署 |
| Oracle OC4J | 1.1, 2.0 （部分） | ejb-jar.xml<br>orion-ejb-jar.xml | - 创建 .ear 文件，或者按照 .ear 文件结构创建目录。<br>- 在 server.xml 配置文件内注册 .ear/ 目录，或者将其放在 OC4J 安装的应用程序目录下。 |
| Sun J2EE | 1.1 | ejb-jar.xml | - 创建 .ear 文件并使用应用程序部署工具部署应用程序。 |

## EJB 单元测试

直到现在，测试 EJB 始终作为 RMI-Vuser 的一部分来进行，方式是通过完成 GUI 来录制 EJB Home 接口和 Remote 接口 API。如果没有实现客户端，就无法进行此类测试。在这种情况下， Java Vuser 用于手动创建 EJB 测试脚本。后一种解决方案很复杂，要求完全理解并可访问源代码。

可以合理假设某些客户将从应用程序服务器建立系统并由此扩展。这意味着他们将首先实现应用程序服务器端并检查其可伸缩性，然后才实现后层（如 Web 服务器 / 客户端）。 LoadRunner 可以用于在早期开发阶段从功能上测试 EJB 的方法和接口，此时仅仅使用 VuGen 的参数化能力和 java 功能，因为如果没有用户界面，将难以使用传统的功能测试工具（如 WinRunner）。另外，LoadRunner 通过结合使用 EJB Vuser 与 Controller 和 Analysis 工具，可以辅助测试可伸缩性问题。

通过 EJB 单元测试解决方案可以浏览在应用程序服务器上安装的 EJB，并为选定的 EJB 自动生成完整的可以运行的脚本。

# C

# 调用外部函数

使用 VuGen 时，可以调用在外部 DLL 中定义的函数。通过从脚本调用外部函数，可以降低脚本的内存印记以及总体运行时间。

要调用外部函数，需要加载定义了该函数的 DLL。

可以用下列方法加载 DLL：

- 本地 – 用于一个脚本，使用 **lr_load_dll** 函数。
- 全局 – 用于所有脚本，方法是向 vugen.dat 文件中添加语句。

## 加载 DLL – 本地

使用 **lr_load_dll** 函数可以在 Vuser 脚本中加载 DLL。DLL 加载成功之后，可以调用 DLL 中定义的任何函数，而无需在脚本中对其进行声明。

**要调用 DLL 中定义的函数，请执行下列操作：**

1 使用 **lr_load_dll** 函数在脚本的开头加载 DLL。将语句放在 *vuser_init* 部分的开头。 **lr_load_dll** 将替换 **ci_load_dll** 函数。

请使用以下语法：

lr_load_dll(*library_name*);

注意，在 UNIX 平台中， DLL 被称为共享库。该库的扩展名取决于平台。

**2** 在脚本的适当位置调用 DLL 中定义的函数。

在下例中，创建表 Test_1 之后，调用了在 *orac1.dll* 中定义的 *insert_vals* 函数。

```
int LR_FUNC Actions(LR_PARAM p)
{
lr_load_dll("orac1.dll");

lrd_stmt(Csr1, "create table Test_1 (name char(15), id integer)\n", -1,
        1 /*Deferred*/, 1 /*Dflt Ora Ver*/, 0);
lrd_exec(Csr1, 0, 0, 0, 0, 0);

/* 调用 insert_vals 函数向该表中插入值。 */
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);
…
```

---

**注意：**您可以为 DLL 指定完整路径。如果不指定路径，**lr_load_library** 将使用 Windows 平台上的 C++ 函数 LoadLibrary 所使用的标准序列来搜索 DLL。在 UNIX 平台上，可以设置 LD_LIBRARY_PATH 环境变量 （或平台上与其等效的变量）。 lr_load_dll 函数使用与 *dlopen* 相同的搜索规则。详细信息，请参阅 *dlopen* 或与其等效的手册页。

---

## 全局加载 DLL

可以全局加载 DLL，使其函数能够用于所有 Vuser 脚本。 DLL 加载成功之后，可以调用 DLL 中定义的任何函数，而无需在脚本中对其进行声明。

**要调用 DLL 中定义的函数，请执行下列操作：**

**1** 向 *mdrv.dat* 文件 （位于 LoadRunner/dat 目录）的相应部分中添加要加载的 DLL 列表。

请使用以下语法：

*PLATFORM*_DLLS=*my_dll1*.dll, *my_dll2*.dll, …

用您的特定平台替换单词 *PLATFORM*。有关平台的列表，请参阅 *mdrv.dat* 文件的开始部分。

例如，要在 NT 平台上为 Winsocket Vuser 加载 DLL，请向 *mdrv.dat* 文件中添加下列语句：

```
[WinSock]
ExtPriorityType=protocol
WINNT_EXT_LIBS=wsrun32.dll
WIN95_EXT_LIBS=wsrun32.dll
LINUX_EXT_LIBS=liblrs.so
SOLARIS_EXT_LIBS=liblrs.so
HPUX_EXT_LIBS=liblrs.sl
AIX_EXT_LIBS=liblrs.so
LibCfgFunc=winsock_exten_conf
UtilityExt=lrun_api
ExtMessageQueue=0
ExtCmdLineOverwrite=-WinInet No
ExtCmdLineConc=-UsingWinInet No
WINNT_DLLS=user_dll1.dll, user_dll2.dll, …
```

**2** 在脚本的适当位置调用 DLL 中定义的函数。

# D

# 在 UNIX 平台上编写脚本

UNIX 平台上的 LoadRunner 用户可以通过编程创建 Vuser 脚本。要通过编程创建脚本，请使用 LoadRunner 模板。

本附录描述：

- 生成模板
- 将 Vuser 操作编入脚本
- 配置 Vuser 运行时设置
- 定义事务和集合点
- 编译脚本

## 关于编写要在 UNIX 平台上运行的 Vuser 脚本

有两种办法创建在 UNIX 平台上运行的 Vuser 脚本：使用 VuGen 或者通过编程。

| | |
|---|---|
| *VuGen* | 您可以使用 VuGen 创建在 UNIX 平台上运行的 Vuser 脚本。可以在 Windows 环境中录制应用程序而将其在 UINX 上运行 （在 UNIX 上不支持录制） |
| *编程* | 工作在单 – UNIX 环境中的用户可以编写 Vuser 脚本。脚本可以用 C 或 C++ 编写，并且必须编译到动态库中 |

本附录描述如何以编程的方式开发 Vuser 脚本。

要通过编程创建脚本，可以使用 LoadRunner 模板作为大型 Vuser 脚本的基础。该模板提供：

- 正确的程序结构
- LoadRunner API 调用
- 创建动态库的源代码和生成文件

用模板创建基本的脚本后，您可以增强该脚本以提供运行时 Vuser 信息和统计信息。详细信息，请参阅第 6 章 “增强 Vuser 脚本”。

## 生成模板

LoadRunner 中包含用于将模板复制到您的工作目录下的实用程序。该实用程序称为 *mkdbtest*，位于 $M_LROOT/bin。要运行该实用程序，请键入：

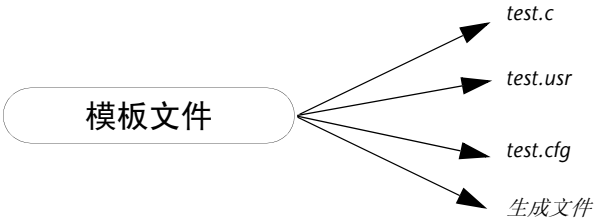
```
mkdbtest name
```

运行 *mkdbtest* 时，将创建一个名为 *name* 的目录，其中包含模板文件 。例如，如果键入

```
mkdbtest test1
```

*mkdbtest* 将创建一个名为 *test1* 的目录，该目录包含模板脚本 *test1.c*。

运行 *mkdbtest* 实用程序时，将创建一个包含下列四个文件的目录：*test.c*、*test.usr*、 *test.cfg* 和 “生成文件”，其中 *test* 是为 mkdbtest 指定的测试名称。



## 将 Vuser 操作编入脚本

可以为您的 Vuser 自定义 Vuser 脚本文件 *test*.c、 *test*.usr 和 *test*.cfg。

实际的 Vuser 操作将编写到 *test.c* 文件中。此文件具有编写的 Vuser 脚本所需的结构。 Vuser 脚本包括三部分：vuser_init、 Actions 和 vuser_end。

注意，该模板为 C++ 用户定义了 *extern C*。此定义对所有 C++ 用户都是必需的，用以确保不会无意中修改任何导出的函数。

```
#include “lrun.h ”
#if defined(__cplusplus) || defined(cplusplus) extern “C ”
{
#endif
int LR_FUNC vuser_init(LR_PARAM p)
{
                lr_message( “vuser_init done\n ”);
                return 0;
}
 int Actions(LR_PARAM p)
{
                lr_message( “Actions done\n ”);
                return 0;
}
 int vuser_end(LR_PARAM p)
{
                lr_message( “vuser_end done\n ”);
                return 0;
}
#if defined(__cplusplus) || defined(cplusplus)}
#endif
```

您可以直接将 Vuser 操作编写到空脚本中，置于每部分的 **lr_message** 函数之前。

在初始化过程中， *vuser_init* 部分将最先执行。此部分中包括连接信息和登陆过程。*vuser_init* 部分仅在每次运行脚本时执行。

*Actions* 部分在初始化之后执行。此部分中包括由 Vuser 执行的实际操作。您可以将 Vuser 设置为重复执行 Actions 部分 （在 *test*.cfg 文件中）。

*vuser_end* 部分在所有 Vuser 操作之后最后执行。此部分中包括清理和注销过程。*vuser_end* 部分仅在每次运行脚本时执行。

---

**注意：**LoadRunner 通过发送 SIGHUP、SIGUSR1 和 SIGUSR2 UNIX 信号控制 Vuser。不要在 Vuser 程序中使用这些信号。

---

# 配置 Vuser 运行时设置

要配置 Vuser 运行时设置，请修改随脚本创建的 *default.cfg* 和 *default.usp* 文件。这些运行时设置对应于 VuGen 的运行时设置。(请参阅第 107 页的 “配置运行时设置”) *default.cfg* 文件包含 “常规”、“思考时间”和 “日志”选项设置。*default.usp* 文件包含 “运行逻辑”和 “步”的设置。

## 常规选项

Unix Vuser 脚本有一个常规选项：

ContinueOnError 指示 Vuser 出现错误时仍继续。要激活该选项，请指定 1。要禁用该选项，请指定 0。

在下例中， Vuser 将在出现错误时仍继续。

```
[General]
ContinueOnError=1
```

## 思考时间选项

您可以设置思考时间选项控制 Vuser 在脚本执行过程中使用思考时间的方式。可以根据下面的图表设置 “Options”、“Factor”、“LimitFlag” 和 “Limit” 参数。

| 选项 | Options | Factor | LimitFlag | Limit |
|---|---|---|---|---|
| 忽略思考时间 | NOTHINK | N/A | N/A | N/A |
| 使用录制思考时间 | RECORDED | 1.000 | N/A | N/A |
| 将录制思考时间乘以 ... | MULTIPLY | 数字 | N/A | N/A |
| 使用录制思考时间的随机百分比 | RANDOM | 范围 | 最低百分比 | 上百分比 |
| 将思考时间限制为 ... | RECORDED / MULTIPLY | 数字 （对于 MULTIPLY） | 1 | 以秒为单位的值 |

要限制在执行过程中使用的思考时间，请将 LimitFlag 变量设置为 1，并以秒为单位指定思考时间 Limit。

下例中的设置指示 Vuser 用范围为 50 到 150 的随机百分比乘以录制思考时间。

```
[ThinkTime]
Options=RANDOM
Factor=1
LimitFlag=0
Limit=0
ThinkTimeRandomLow=50
ThinkTimeRandomHigh=150
```

## 日志选项

您可以设置日志选项为脚本的执行创建简要或详细的日志文件。

```
[Log]
LogOptions=LogBrief
MsgClassData=0
MsgClassParameters=0
MsgClassFull=0
```

可以根据下面的图表设置参数 “LogOptions”、“MsGClassData”、“MsgClassParameters”和 “MsgClassFull”变量。

| 日志记录类型 | LogOptions | MsgClassData | MsgClassParameters | MsgClassFull |
|---|---|---|---|---|
| 禁用日志记录 | LogDisabled | N/A | N/A | N/A |
| 标准日志 | LogBrief | N/A | N/A | N/A |
| 仅参数替换 | LogExtended | 0 | 1 | 0 |
| 仅服务器返回的数据 | LogExtended | 1 | 0 | 0 |
| 仅高级跟踪 | LogExtended | 0 | 0 | 1 |
| 全部 | LogExtended | 1 | 1 | 1 |

下例中的设置指示 Vuser 记录所有从服务器返回的数据和用于替换的参数。

```
[Log]
LogOptions=LogExtended
MsgClassData=1
MsgClassParameters=1
MsgClassFull=0
```

## 循环和运行逻辑

您可以设置循环选项执行多次循环，并控制循环间的步。也可以手动地设置操作的次序和权重。要修改脚本的运行逻辑和循环属性，必须编辑 *default.usp* 文件。

要指示 Vuser 执行 Actions 部分的多次循环，请将 RunLogicNumOfIterations 设置为相应的值。

要控制循环之间的步，请根据下面的图表设置 “RunLogicPaceType”变量及其相关的值。

| 步 | RunLogicPaceType | 相关变量 |
|---|---|---|
| 尽快 | Asap | N/A |
| 循环之间等待约设置的时间 | Const | RunLogicPaceConstTime |
| 循环之间等待约随机的时间 | Random | RunLogicRandomPaceMin, RunLogicRandomPaceMax |
| 每次循环后等待约设置的时间 | ConstAfter | RunLogicPaceConstAfterTime |
| 每次循环后等待约随机的时间 | After | RunLogicAfterPaceMin, RunLogicAfterPaceMax |

下例中的设置指示 Vuser 在两次循环之间等待随机的秒数时，执行四次循环。随机数字的范围为 60 到 90 秒。

```
[RunLogicRunRoot]
MercIniTreeFather=""
MercIniTreeSectionName="RunLogicRunRoot"
RunLogicRunMode="Random"
RunLogicActionOrder="Action,Action2,Action3"
RunLogicPaceType="Random"
RunLogicRandomPaceMax="90.000"
RunLogicPaceConstTime="40.000"
RunLogicObjectKind="Group"
RunLogicAfterPaceMin="50.000"
Name="Run"
RunLogicNumOfIterations="4"
RunLogicActionType="VuserRun"
RunLogicAfterPaceMax="70.000"
RunLogicRandomPaceMin="60.000"
MercIniTreeSons="Action,Action2,Action3"
RunLogicPaceConstAfterTime="30.000"
```

## 定义事务和集合点

在未使用 VuGen 的情况下编写 Vuser 脚本时，必须手动配置 Vuser 文件，以便启用事务和集合。配置设置列在 *test*.usr 文件中。

```
[General]
Type=any
DefaultCfg=Test.cfg
BinVuser=libtest.libsuffix
RunType=Binary

[Actions]
vuser_init=
Actions=
vuser_end=

[Transactions]
transaction1=

[Rendezvous]
Meeting=
```

每个事务和集合都必须在 *usr* 文件中定义。请将事务名添加到 “Transactions” 部分 （后接 “=”）。将每个集合名添加到 “Rendezvous” 部分 （后接 “=”）。如果该部分不存在，则添加到上面所示的 *usr* 文件。

## 编译脚本

修改模板后，请使用脚本目录中的相应 “生成文件” 对其进行编译。注意，用 C++ 编译时，必须使用本机编译器 （非 gnu）。该编译器将创建一个具有下列名称的动态库：

- libtest.so (solaris)
- libtest.a (AIX)
- libtest.sl (HP)

您可以修改生成文件，并通过修改相应的部分来分配其他编译器标志和库。

如果使用的是通用模板，则必须将应用程序的库和头文件包括进来。例如，如果应用程序使用了名为 *testlib* 的库，请将其包括到 LIBS 部分中。

```
LIBS        = \
      -testlib \
      -lLrun50 \
      -lm
```

修改生成文件之后，请在工作目录中的命令行键入 Make 为 Vuser 脚本创建动态库文件。

现在即可通过 LoadRunner Controller 运行脚本。 Vuser 脚本是位于脚本的目录中的 *script*.usr 文件。有关如何将 Vuser 脚本集成到场景中的详细信息，请参阅《LoadRunner Controller 用户指南》。

在将脚本集成到场景之前，应该在命令行中运行它，验证是否能正常工作。

要在 UNIX 命令行中运行 Vuser 脚本，请键入：

```
mdrv -usr ‘pwd’ test.usr
```

其中 *pwd* 是包含 Vuser 脚本的目录的完整路径， *test*.usr 是 Vuser 文件的名称。请检查脚本与服务器的通信，以及是否执行所有必需的任务。

# E

# 使用键盘快捷键

下面的列表描述在虚拟用户生成器中可用的键盘快捷键。

| | |
|---|---|
| ALT+F8 | 比较当前快照 （仅限于 Web Vuser） |
| ALT+INS | 新建步骤 |
| CTRL+A | 全选 |
| CTRL+C | 复制 |
| CTRL+F | 查找 |
| CTRL+G | 转到行 |
| CTRL+H | 替换 |
| CTRL+N | 新建 |
| CTRL+O | 打开 |
| CTRL+P | 打印 |
| CTRL+S | 保存 |
| CTRL+V | 粘贴 |
| CTRL+X | 剪切 |
| CTRL+Y | 重复 |
| CTRL+Z | 撤消 |
| CTRL+F7 | 录制选项 |
| CTRL+F8 | 扫描关联 |
| CTRL+SHIFT+ 空格键 | 显示函数语法 (Intellisense) |

| | |
|---|---|
| CTRL+ 空格键 | 完成向导 （完成函数名） |
| F1 | 帮助 |
| F3 | 向下查找下一个 |
| SHIFT+F3 | 向上查找下一个 |
| F4 | 运行时设置 |
| F5 | 运行 Vuser |
| F6 | 在窗格之间移动 |
| F7 | 显示 EBCDIC 转换对话框 （对于 WinSocket 数据） |
| F9 | 切换断点 |
| F10 | 分步运行 Vuser |

# 索引

## 字母


“ABC”图标 74
Accept-Language 请求标头 434
Acrobat Reader xix
Action
  部分 26
  方法 292
Actions 类 291
ANSI C 支持
  对于增强脚本 69
  在自定义脚本中 283
Astra LoadTest 虚拟用户录制器 7
Baan Vuser 脚本
  概述 629
  函数列表 630
  了解 635
  思考时间 636
  自定义 636
Baan 协议的异常处理 636
C Vuser 282
C 函数
  Vuser 脚本中的限制 69
  调用 libc 函数 69
  附加关键字 790
  用于调试 790
  在 Vuser 脚本中使用 17
C 语言支持
  解释器 18
  约定 283
CARRAY 缓冲区 707
CHECK_HRES （COM 错误检查）323
cHTML 723
Citrix ICA Vuser 脚本
  录制 193
Citrix Vuser 脚本
  查看 204
  录制和回放提示 211
  录制日志 205
  入门 194
  使用 ctrx 函数 202
  同步回放 199
  显示设置 206
  运行时设置 207
Citrix 服务器 , 断开连接 211
COM
  概述和接口 308
  数据类型 309
COM Vuser 脚本
  CHECK_HRES 323
  DONT_CHECK_HRES 323
  IDispatch 接口 326, 334
  lrc_ *类型* 函数 335
  Visual Basic 集合 341
  变量类型转换函数 336
  参数化函数 338
  创建对象实例 324, 334
  错误检查 323
  调试函数 342
  函数 333
  检索接口 325
  脚本结构 322
  接口指针 322
  开发 307
  类型分配函数 335
  类型库 308
  了解 322
  录制选项 312
  入门 309
  扫描关联 328
  实例化对象 324, 334
  选择要录制的 COM 对象 310
  选择要录制的对象 309
  要调试的日志文件 310

Controller
场景 139
发送消息到 765
发送消息到 (GUI) 765
Corba 录制选项。173
Corba-Java Vuser 脚本 343
调试选项 170, 173
关联选项 169
录制 344
录制器选项 164
序列化选项 168
CtLib 217
记录服务器消息 119
结果集错误 234
选项 223
ctrx 函数 202
DB2-CLI 217
DbLib 217
DCOM 308
“DCOM”选项卡 314
declare_rendezvous 函数 764
declare_transaction 函数 764
DLL, 从 Vuser 脚本调用 845
DN (LDAP) 366
DNS Vuser
概述 241
函数 242
DNS 缓存
Web 434
DSL 128
EBCDIC 转换 270
EJB
Vuser 脚本 535
代码生成选项 543
方法 548
实例 546
体系结构 839
EJB 检测器 546
安装 536
命令行 537
日志文件 539
end 方法 291
end_transaction 函数 764
error_message 函数 764, 765
EUC 编码的网页 420
Expect 属性 , Web 检查 454
FIELDTBLS 环境设置 704
FLDTBLDIR 环境设置 704
Forms 侦听器 572
Frame 属性 , 用于对象检查 (Web) 454
FTP 协议
函数列表 360
录制 359, 775
get_host_name 函数 766
get_master_host_name 函数 766
GUI Vuser 脚本
概述 758
简介 758
开发 757–766
了解 759
入门 760
使用 GUI 函数 764
使用 WinRunner 创建 761
GUID 308
gzip 436
HotSync 396
HTML
参数的最大长度 507
HTML 参数的最大长度 507
HTML 视图 （Web 快照）494
HTML 压缩 (gzip) 436
HTTP
缓冲区大小 (Web) 435
HTTP 录制模式 , WAP 746
ica 文件 210
IDispatch 接口 326, 334
If-Modified-Since 标头
Web 430
IIOP 352
IMAP 协议 677
i-mode
概述 723
工具包 724
Informix 217
init 方法 291
intellisense 21
Internet 消息传输 (IMAP) 677
ISDN 128
IUnknown 接口 308

Jacada Vuser 脚本 687
回放 691
了解 691
录制 689
Java Vuser (Corba, RMI)
Java VM 运行时设置 188
类路径运行时设置 189
Java Vuser （Corba 和 RMI）
录制提示 349
录制选项 , Java VM 162
录制选项 , 关联 169
录制选项 , 序列化 168
Java Vuser （全部）
编程 289
编辑 Java 方法 291
插入集合点 296
环境设置 301
运行时设置 187–188
Java Vuser （所有）
关联语句 175
Java Vuser （自定义）
创建模板 290
使用 Java 代码 284
Java 插件 349
Java 虚拟机
录制选项 162
运行时设置 188
JavaScript Vuser 287
JNDI 属性
查找 EJB Home 545
高级 , 上下文工厂 542
指定 541
Jscript 44
Keep-Alive 连接 , Web 434
LDAP 协议
函数列表 364
录制 363
通过 WinSock 243
libc 函数 , 调用 69, 283
lr_whoami 函数 (GUI Vuser) 764
lrbin.bat 实用程序 773
lrc 函数 321
lrd （数据库）函数 224
lreal 函数 715
lrs 函数 248
lrt 函数 697
MAPI 函数 681
MatchCase 属性 454
Media Player 716
mkdbtmpl 脚本 (UNIX) 850
MMS 函数 (MS Media Player) 716
MMSC 738
MS
Exchange 协议 (MAPI) 681
SQL Server, 录制 217
MTS 组件 316
NCA Vuser, 请参阅 Oracle NCA
Nokia 工具包 731
ODBC 录制 217
OnFailure 属性 , Web 检查 454
Oracle 217
Oracle 8.0 版 227
Oracle Configurator 572
Oracle NCA Vuser 脚本
Servlet 测试 572
安全应用程序 571
创建 555
关联 573
检查连接模式 573
录制指南 558
使用 Vuser 函数 563
运行时设置 569
Oracle 应用程序调试 793
OTA, 空中 736
output_message 函数 764
Palm
录制应用程序 396
协议 387
PAP, 推访问协议 736
PeopleSoft8 369, 375
PeopleSoft-Tuxedo Vuser 695
运行 805
POP3 （邮局）协议 682
PPG, 推代理网关 736
Pragma 模式 570
QuickTest Professional 7
Radius 运行时设置 (WAP) 752
RADIUS 支持 735
RealPlayer 713
Repeat 属性 , Web Vuser 454
Report 属性 , Web 检查 454

RMI-Java Vuser 脚本
调试选项 170, 173
关联选项 169
录制 352
录制 RMI over IIOP 352
录制器选项 164
序列化选项 168
RTE Vuser 脚本
创建步骤 643
读取屏幕中的文本 673
概述 641
介绍 642
录制 647
入门 643
使用 te 函数 644
同步 665
映射 PC 键盘 645
运行时设置 661
run_db_vuser shell 脚本 137
S_SSA_ID 表 804
safearray 日志 (COM) 318
SAPGUI Vuser 脚本
常规录制选项 592
代码生成录制选项 593
录制 579
设置录制选项 592, 619
使用 sapgui 函数 603
运行时设置 600
sapgui 函数 603
SAP-Web Vuser 脚本
录制 617
运行时设置 622
SED 实用程序 374
Shift 日本工业标准 (SJIS) 420
Siebel
2 层脚本提示 802
以 36 为基数的键值 804
Siebel 中的后缀值 802
Siebel-Web
关联 481, 624
录制 624
疑难解答 626
Siebel （所有类型）217
SMS - 短消息服务 750
SMTP 协议 683
SOAP Vuser 脚本 387
Solaris
ASCII 转换 247
start_transaction 函数 (GUI) 765
SWECount, 关联 625
te (RTE) 函数 644
TE 系统变量 670
TestDirector
打开 Vuser 脚本 144
断开与 TestDirector 的连接 144
管理 Vuser 脚本 141
将脚本保存到 TestDirector 146
连接 TestDirector 141
treeview
Web Vuser 脚本 383
TSL, 已定义 761, 763
TUXDIR 环境设置 704
Tuxedo Vuser 脚本 695
Tuxedo 6, 7 695
版本 705
查看数据文件 703
环境设置 704
了解 701
日志文件 702
使用 lrt (Tuxedo) 函数 697
数据缓冲区 703
系统变量 704
运行 702
UNIX
命令行 137
URL 步骤
定义的 (Web Vuser) 384
修改 460
“URL 步骤属性”对话框
Web 461
user_data_point 函数 765
UTF-8 转换 403
VB Vuser 285
VBA 引用 126
VBA 运行时设置 126
VBScript Vuser 286
Visual Basic
Vuser 脚本 769
脚本选项 44
Visual C, 使用 Visual Studio 769
Visual Studio 769
VM （虚拟机）162

VoiceXML
概述 724
*另请参阅*无线项
VuGen
概述 11
工具栏 36
脚本录制选项 43
介绍 11
录制 Vuser 脚本 12, 25
用户指南概述 7
运行 Vuser 脚本 15
Vuser
介绍 3
类型 5
另请参阅 “ Vuser 脚本”
Vuser ID, 参数值 84
Vuser 函数
Baan 630
ctrx (Citrix) 202
GUI 764
imap 679
Java 293
lrc (COM) 323
lrd （数据库）224
lreal 715
lrs (WinSock) 248
lrt (Tuxedo) 697
lr （C 函数）16
mapi 681
mms (MS Media Player) 716
Oracle NCA 563
pop3 682
sapgui (SAP) 603
smtp 683
te (RTE) 644
*另请参阅*联机函数参考
外部 , 用户定义的 845
Vuser 脚本
C 支持 283
COM 323
Corba-Java 343
EJB 测试 535
GUI Vuser 761
Jacada 687
Java Vuser （编程）289
Java 语言录制 149
Media Player 713
Real Player 713
TestDirector 集成 141
TSL 761
UNIX, 编译 857
UNIX, 创建脚本 849
UNIX, 运行于 137
编程 281, 849–858
部分 26
参数化 71
场景集成 139
从命令提示符运行 137
调试功能 133
获取 GUI Vuser 信息 766
集合点 59
开发步骤 6
开发工具 7
流式数据 713
事务 57
添加函数 55
向 Controller 发送消息 765
选择生成语言 43
运行 129
运行时设置 107
运行时设置 Java 187–188
在 UNIX 上编程 849
在 UNIX 上创建 849–858
增强 55
重新生成 39
注释 , 插入 60
自定义 281
Vuser 脚本部分 26
Vuser 脚本的 vuser_end 部分 26
Vuser 脚本的 vuser_init 部分 26
Vuser 生成器 , *参阅* VuGen
Vuser 信息 , 获取 61
Vuser 信息 , 获取 (Java) 297
WAP Vuser 脚本
调试信息 435
工具包节点 743
介绍 733
了解 733
*另请参阅*无线脚本
运行时设置 745
指定要录制的内容 740
Wdiff 105

Web Vuser 脚本
Internet 录制选项 399
“结果概要”报告 515
“可视日志”选项 131
部分 373
代理服务器设置 400
调试工具 134
调试功能 , 启用 135
高级提示 523
关联 481
函数 375
检查 444
脚本视图 385
介绍 369
了解 371
录制选项 409
内容筛选 405
删除步骤 459
设置“可视日志”选项 134
树视图 383
添加步骤 458
图像检查 451
文本检查中的常规表达式 455
修改 457
验证文本和图像 443
运行时查看器 134
运行时设置 127, 423
指定用于录制的浏览器 410
自定义标头 404
自定义请求步骤 471
Web 捕获 393
“Web 捕获”选项卡 394
Web 到 Java 转换器 374
Web 关联 479
Web 函数 , 使用 377
Web 性能图
为 Web Vuser 生成 433
Web/WinSock Vuser 脚本 387
Web 捕获选项 393
代理服务器设置 391
录制 394
入门 389
双与多 388
Windows 套接字 Vuser 脚本
查看数据文件 267
脚本和树视图 244
录制 243
排除套接字 247
入门 244
使用 lrs 函数 248
使用数据 255
数据缓冲区 267
数据文件 268
WinInet 引擎（Internet 协议）433
WinRunner
*另请参阅* WinRunner 用户指南
使用 WR 创建 GUI 脚本 761
WinSock 录制选项 246
WSP
录制选项 740
通过电话录制会话 734
运行模式 746
WS*xxx* Tuxedo 变量 704
XML
测试 509
属性 781
自定义请求 511
X-SYSTEM 消息 (RTE) 666
zip 文件选项 37

## A

安全 WAP 740

## B

报告树 , 结果概要 (Web) 517
“报告”工具栏 517
比较 Vuser 105
比较方法 498, 532
边界
定义关联 (Web) 507
编程
Vuser 操作 851
使用模板 770, 850
在 Visual Studio 中 769
编辑器字体 14
编码 , EUC 420
编译 Vuser 脚本 (UNIX) 857
标记 , 参数化 483
“标记替换测试板”对话框 486

标头
危险 404
自定义 404
标准日志选项 117
表单提交步骤 384
表格图标 75
并发组函数 379
捕获 393
不可打印的字符 273
步设置 113
“步骤”按钮 134

## C

参数的类型 , 了解 78
参数化
Java 75
Tuxedo 脚本 702
参数列表 96
撤消 (Web) 77
大括号类型 98
定义属性 77
概述 72
给参数命名 74
更新参数值 86
更新文件中的值 89
还原初始值 77
了解参数类型 78
内部数据类型格式 85
全局目录 99
设置数据文件的属性 87
使用内部数据 78
使用唯一值更新 91
使用种子的随机顺序 90
数据文件 85
新建参数 74
修改现有参数 106
选项 98
选择参数类型 75
参数类型
Vuser ID 84
负载生成器名 81
列表 78
内部数据 78
日期 / 时间 78
数据文件 85
循环编号 80
组名 80
参数属性
定义 77
定义数据文件 87
“参数属性”对话框 77
操作
导入 38
录制多个 35
重新排序 39
操作步骤
函数列表 - Web 378
修改 - Web 460
测试脚本语言 , 请参阅 TSL
测试结果 517
“插入注释”对话框 60
拆分操作 44
场景
从 VuGen 创建 139
将 Vuser 脚本集成到 139
常规表达式
在文本检查中 455
常规选项
Citrix 显示 206
“参数化”选项卡 98
“环境”选项卡 14
“回放”选项卡 131
“显示”选项卡 （仅限于 Web）135
对话框 100
关联选项卡 497
所有 Vuser 99
“常规选项”对话框
“参数化”选项卡 98
超时 , 设置默认值 (Baan) 637
超图形链接步骤 , Web Vuser 384
超文本链接步骤
定义的 384
修改 462
撤消缓冲区 , 清空 (WinSock) 262
“撤消参数”命令 77
持续连接 , Web 434
重复键冲突
Oracle, MSSQL 800
Siebel 802
重新生成 Vuser
Web/WinSock 协议 396

所有协议 39
出现错误时仍继续 64, 122
错误处理 64, 122
COM Vuser 脚本 323
本地修改 （严重级别）233
全局修改 232
“错误消息”对话框 63

## D

“打开”命令 520, 521
“打印”对话框 （Web 报告）521
打印 “结果概要”报告 521
大括号 , 在参数化中使用 98
代理服务器
录制选项 (Web) 400
录制选项 (Web/WinSock) 391
运行时设置 (Internet) 424
“代理服务器身份验证”对话框 401
代码生成选项 (EJB) 543
导出到 zip 文件 37
导入
操作 38
数据库中的数据 92
等待终端稳定下来 672
电话 , 录制 734
定义参数属性 77
数据文件 87
定义属性 , 文本检查 454
动态端口 274
动态运行
定义 130
启用 131
独立模式 , 运行 Vuser 脚本 129
端口映射设置 47
断点 134
断开与 TestDirector 的连接 144
对 RTE Vuser 进行复制和粘贴 652
多个操作 28
多个协议 28
多线程 125

## E

二进制编码数据 458

## F

返回代码 231
方法 , Java 291
非标准 HTTP 应用程序 530
非资源 407
分隔符 , 在数据表格中 88
服务步骤
在树视图中进行修改 (Web) 476
“服务步骤属性”对话框 476
负载平衡 , Oracle NCA 573
负载生成器名 , 参数值 81

## G

高级复制和粘贴 (WinSock) 263
高级录制选项 402
“高级关联属性”对话框 489
高速缓存 , 浏览器 （Web, 无线）430
格式
为参数化 85
显示缓冲区中的数据的 272
更新方法 , 在参数化中 86, 89
“关闭全部”命令 136
关键字 , 添加附加的 790
关联
COM Vuser 328
HTML 语句 (Web) 479
Java 语句 175
Siebel-Web 481, 624
SWECount 625
Tuxedo 709
Web Vuser 的规则 482
对于已知上下文 (Web) 481
概述 101
函数 (C) 103
函数 (Java) 104
脚本语言选项 45
录制后 （Web、无线）491
录制选项 -Java 169
缺少的关联 （数据库）805
扫描数据库 Vuser 脚本 236
使用快照 (Web) 491
修改现有参数 106
关联结果选项卡 500
关联选项
在脚本录制选项中 45

关联选项卡 487
“关联查询”选项卡 236

## H

函数
Baan 630
ctrx (Citrix) 202
GUI 764
imap 679
Java 293
lrc (COM) 321
lrd （数据库）224
lreal (Real Player) 715
lrs (WinSock) 248
lrt (Tuxedo) 697
lr （C 函数）16
mapi 681
pop3 682
sapgui (SAP) 603
smtp 683
te (RTE) 644
WAP 730
在 Web Vuser 脚本中 377
行计数 , 获取 799
行信息 , 数据库 Vusers 230
环境设置
Java 301
Tuxedo Vuser 704
“环境”选项卡 14
还原参数的初始值 77
缓冲区导航器 (WinSock) 258
缓冲区中数据的偏移量 (WinSock) 270
“回放”选项卡 , “常规选项”对话框 131

## J

基于 HTML 的模式 411
集合
“集合”对话框 59
集合点
GUI Vuser 762
Java Vuser 296
插入 59
函数 (GUI) 764
在 Web 脚本中进行修改 475
加密文本 68
加载 DLL
本地 845
概述 845
全局 847
检测器 , EJB 536
检查 (Web)
定义其他属性 454
概述 444
函数 379
类型 445
图像检查 451
文本 446
在脚本中进行修改 477
检查属性对话框 477
简要日志运行时设置 118
键盘快捷键
快捷键列表 859
录制选项 46
运行时设置 108
键盘映射 (RTE) 645
键入样式 (RTE Vuser) 657
脚本生成器 , *参阅* VuGen
脚本视图
Oracle NCA 脚本 568
Web Vuser 脚本 385
Windows 套接字脚本 252
“结果概要”报告 515
打开 520
打印 521
调试 Web 脚本 515
了解 517
筛选信息 519
树分支 517
搜索 520
“结束事务”对话框 58
解密文本 68
禁用日志记录日志选项 117

## K

“开始录制”对话框 33
所有 33
“开始事务”对话框 57
可分辨名称 366

“可视日志”选项
设置 (Web) 131
客户端证书 523
控制步骤
函数 (Web) 382
修改 (Web) 473
库 , 用于脚本 126
快照
XML 510
缓冲区的 , Winsock 256
网页 492
扩展结果集 223
扩展日志选项 117

## L

类路径
运行时设置 189
联机函数参考 xix
联机图书 xix
联机浏览器 134, 528
连接 TestDirector 对话框 141
连接尝试 , 修改 (RTE) 662
“连接”对话框 (RTE) 651
“链接步骤属性”对话框 462
浏览器
录制选项 (Web) 410
启动 (Web/WinSock) 390
使用默认值 (Web) 410
手动启动 (Web) 410
指定位置 (Web) 410
浏览器仿真设置 , Web 428
浏览器高速缓存 （Web, 无线）430
流量转发 50
流式数据协议
mms 函数 716
RealPlayer 函数 715
录制 713
录制 Vuser 脚本
Baan 630
Citrix ICA 193
Corba-Java 344
DNS 241
FTP 359
LDAP 363
Oracle NCA 557
Rmi-Java 351
SAPGUI 579
SAP-Web 617
Tuxedo 695
Web/WinSock 387
Window 套接字 243
代理服务器设置 400
使用 VuGen 25
数据库 220
无线 727
邮件服务 677
录制模式
i-mode, VoiceXML 741
WAP 740
录制选项
Corba 选项节点 173
Internet 协议 399
Java 语言 161–172
RTE 653
WAP 工具包 743
Web 409
Web 设置 409
WinSock 节点 246
调试节点 (Java) 170
端口映射节点 47
高级 （Web 和无线）402
键盘快捷键 46
脚本 （FTP、 COM、 Mail）44
录制 (Web) 421
录制代理服务器 (Web/WinSock) 391
录制代理服务器 （Web 和无线）400
录制器节点 (Java) 164
数据库节点 221
无线 Vuser 739
浏览器 (Web) 410
“录制”按钮 33
“录制日志”选项卡 38

## M

命令提示符 137
命令行参数
UNIX Vuser 脚本 137
处理 (Java Vuser) 300
处理 （C Vuser）67

模板
Java Vuser 290
使用 “C” 编程 850
新建 33
用 “C” 编程 770

## N

内部数据 , 参数化 78
内容检查设置 (Web) 437
内容类型筛选 (Web) 405
“内容类型筛选器” 对话框 406

## Q

其他运行时设置 122
启动屏幕 13
“全部展开” 命令 518
“全部折叠” 命令 518
全局目录 99
全局唯一标识符 (GUID) 308

## R

日期 / 时间 , 参数值 78
日志
设置详细级别 - PC 117
设置详细级别 - UNIX 854
日志高速缓存大小 117
“日志消息” 对话框 61, 62
日志运行时设置 116

## S

扫描关联命令
COM 329
“扫描关联” 命令
数据库 Vuser 236
筛选
报告信息 (Web) 519
高级 （Web 和无线） 405
“筛选器” 对话框 （Web 报告） 519
删除步骤
从 Web 脚本 459
上下文相关帮助 xix
设备名 (RTE) 663
设置 , 请参阅运行时设置
深度关联 (Java) 177
时间戳 （数据库） 223
实例化 COM 对象 324
“使用现有参数” 命令 76
事务 132
GUI Vuser 762
插入 57
用于 Web Vuser 125
在 Web 脚本中进行修改 473
自动 , 对于 LRD 函数 221
自动 , 用于 Web Vuser 脚本 125
首选项运行时设置 432
输出窗口 298
显示 / 隐藏 136
隐藏 132
“输出消息” 对话框 63
属性
Expect (Web) 454
Frame (Web) 454
MatchCase (Web) 454
OnFailure (Web) 454
Repeat (Web) 454
Report (Web) 454
文本检查 454
树视图
Citrix 脚本 204
Oracle NCA 脚本 568
Windows 套接字脚本 252
数据的二进制视图 (WinSock) 256
数据缓冲区
Tuxedo Vuser 脚本 703
WinSock Vuser 脚本 267
数据库 Vuser 脚本
查看网格 229
处理错误 232
返回代码 231
关联 235
开发 217
入门 220
使用 lrd 函数 224
提示 798
行信息 230
“数据库查询向导” 对话框 92
数据库录制选项 221
数据网格 , 显示 / 隐藏 136

数据文件
Windows 套接字 Vuser 脚本 268
用来参数化 85
数据向导 92
数据中的书签 (WinSock) 260
双协议 (Web/WinSock) 387
顺序参数分配 89
思考时间
插入 66
定义的 120
对话框 (Web treeview) 475
函数 (C) 20
函数 (Java) 295
运行时设置 120
在 Web 脚本中进行修改 475
阈值 , WinSock 248
阈值 , 数据库 221
“思考时间”对话框 66
“搜索和替换”对话框 76
搜索屏幕中的文本 (RTE) 673
速度模拟设置 128
随机参数分配 90

## T

提交表单步骤
对话框 466
修改 465
提交数据步骤 384
对话框 (Web) 469
修改 - Web 468
提取数据 230
提示
数据库相关 798
特定于 Siebel 802
“替换其他出现位置”命令 76
“添加新列”对话框 88
调试
Oracle 应用程序 793
Web Vuser 脚本 515
回放期间 134
获取信息 (WAP) 435
设置调试级别 118
数据库应用程序 791
为 Web Vuser 启用 135
调试录制设置 (Java) 170
“调试消息”对话框 63
停止 Vuser 132
同步 Vuser 脚本
等待光标出现 669
等待文本出现 (RTE) 670
等待终端处于静止状态 672
概述 66
概述 (RTE) 665
块模式 (IBM) 终端 666
字符模式 (VT) 终端 669
同步函数 (Baan) 632
图
为 Web 启用 433
“图像步骤属性”对话框 464
图像检查
Web Vuser 脚本 451
修改 (Web) 463
推支持 736

## W

外部函数 845
完成字 21
完全运行时跟踪 118
网格
查看 229
隐藏 136
网关设置 (WAP) 746
网络 (Internet) 上的缓冲区大小 435
网络设置 128
网络速度 , 运行时设置 128
唯一编号 , 参数值 83
唯一的列名 802
唯一值参数分配 91
文本
读取屏幕中的文本 (RTE) 674
搜索屏幕中的文本 (RTE) 673
在快照中进行比较 (Web) 532
文本检查
定义的 445
定义其他属性 454
“文本检查属性”对话框 447
文本视图 (WinSock) 256
文本视图 , *参阅*脚本视图 385
文档集 xx
无线 Vuser 脚本

Internet 录制选项 399
WAP 工具包 743
代理服务器设置 400
介绍 721
了解 721
录制 727
录制选项 739
自定义标头 404

## X

系统变量
RTE 670
Tuxedo 704
显示函数 21
“显示”选项
查看日志 (Web) 516
“显示”选项卡 , 常规选项 135
线程 , 主 （Java 编程）304
线程安全代码 303
消息
发送到 Controller (GUI) 765
发送到输出 61
协议 , *参阅* Vuser
“新建”按钮 648
“新建虚拟用户”对话框
RTE 648
修改 Web 脚本
URL 步骤 460
集合点 475
事务 473
思考时间 475
提交表单步骤 465
提交数据步骤 468
图像步骤 463
虚拟用户生成器 , *参阅* VuGen
序列化 （Java 关联）180
序列化选项 169
选项
CtLib 223
lrd 日志 223
参数化 98
录制 (RTE) 653
循环
更新参数，为每次 86
运行时设置 113
循环编号 , 参数值 80
“选择或创建参数”对话框 74
“选择结果目录”对话框 136

## Y

验证检查
RTE 673
Web 432
页面视图 （Web 快照）494
疑难解答
Oracle 应用程序 793
Siebel 协议 802
VuGen 789
Web Vuser 脚本 523
两层数据库 798
应用程序服务器 , Oracle NCA 558
映射键盘 645
用户代理浏览器仿真 429
用于生成脚本的语言 43
邮件服务协议
IMAP 679
MAPI 681
POP3 682
SMTP 683
录制 678
语法 , 为函数显示 22
运行 Vuser 脚本
动态模式 130
分步 133
使用 VuGen 129
在独立模式下 129
运行逻辑设置 109
“运行”命令 132
运行时查看器
使用提示 (VuGen) 528
在 VuGen 中启用 134
运行时设置
Internet 协议 （Web 等）423
Java 187–188
Oracle NCA 569
Radius 节点 (WAP) 752
RTE 667
RTE 节点 661
VBA 节点 126
WAP 745

步节点 113
代理服务器节点 （Internet 协议）424
调试信息 (WAP) 435
对话框 108
键盘快捷键 108
客户端仿真 (Oracle NCA) 570
快捷键 108
内容检查节点 (Web) 437
其他 122
日志节点 116
手动配置 852
首选项 - 高级 433
首选项节点 （Internet 协议）432
思考时间 120
速度模拟 (Oracle NCA) 569
速度模拟 （Internet 协议）128
所有协议 107
网关节点 (WAP) 746
网络 127
运行逻辑节点 109
载荷节点 (WAP) 750
浏览器仿真节点 428

## Z

载荷设置 (WAP) 750
载体支持 (WAP) 735
在 WinSock 数据中导航 258
暂停 Vuser 132
支持信息 xx
执行报告 （仅限于 Web）531
“执行日志”选项卡 132
终端仿真 647
终端服务器会话 214
“终端设置”对话框 650
注释
插入到 Vuser 脚本 60
屏幕标题注释 (RTE) 655
“注释”按钮 60
转换
Web 函数到 Java 374
自定义请求到 C 471
转换 , UNIX 上的 ASCII 247
转换表设置 246
转义序列 272
资源 , 排除 407
自定义标头
Web, 无线 404
自定义脚本
C Vuser 282
Java Vuser 284
JavaScript Vuser 287
VB Vuser 285
VBScript Vuser 286
自定义请求 419
“自定义请求”对话框 (Web) 472
自定义请求步骤
XML 511
定义的 384
修改 (Web) 471
自动代理服务器脚本 426
自动恢复 14
自动兼容 282
自动检测协议 51
自动事务
Web 和无线协议 433
常规 125
数据库 Vuser 脚本 221
字段分界字符 659
组名 , 参数值 80

# C-Interpreter Copyright Agreement

The Virtual User Generator generates standard C code which can be compiled with any ANSI C compiler. However, for the convenience of our customers we have provided a C Interpreter for running the generated code, without charge. The cci executable which is the front end of the interpreter is based on the freely available "lcc Retargetable C Compiler" by Christopher Fraser and David Hanson, and is covered by the lcc Copyright included below. Any bugs in cci should be reported to Mercury Interactive. The author's copyright notice is below.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHORS NOR MERCURY INTERACTIVE MAKE ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

The authors of this software are Christopher W. Fraser and David R. Hanson.

Copyright (c) 1991,1992,1993,1994,1995 by AT&T, Christopher W. Fraser, and David R. Hanson. All Rights Reserved.

Permission to use, copy, modify, and distribute this software for any purpose, subject to the provisions described below, without fee is hereby granted, provided that this entire notice is included in all copies of any software that is or includes a copy or modification of this software and in all copies of the supporting documentation for such software.

THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHORS NOR AT&T MAKE ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

lcc is not public-domain software, shareware, and it is not protected by a `copyleft` agreement, like the code from the Free Software Foundation. lcc is available free for your personal research and instructional use under the `fair use' provisions of the copyright law. You may, however, redistribute the lcc in whole or in part provided you acknowledge its source and include this COPYRIGHT file.

You may not sell lcc or any product derived from it in which it is a significant part of the value of the product. Using the lcc front end to build a C syntax checker is an example of this kind of product.

You may use parts of lcc in products as long as you charge for only those components that are entirely your own and you acknowledge the use of lcc clearly in all product documentation and distribution media. You must state clearly that your product uses or is based on parts of lcc and that lcc is available free of charge. You must also request that bug reports on your product be reported to you. Using the lcc front end to build a C compiler for the Motorola 88000 chip and charging for and distributing only the 88000 code generator is an example of this kind of product. Using parts of lcc in other products is more problematic. For example, using parts of lcc in a C++ compiler could save substantial time and effort and therefore contribute significantly to the profitability of the product. This kind of use, or any use where others stand to make a profit from what is primarily our work, is subject to negotiation.

Chris Fraser / cwf@research.att.comDavid Hanson / drh@cs.princeton.edu

Mercury Interactive **上海代表处**
**地址**：上海市南京西路 **1266** 号恒隆广场 **35** 层 **3501** 室 **(200040)**
**电话**：**8621 - 62882525**
**传真**：**8621 - 62883030**

主页：**www.mercury.com/cn**
客户支持：**support.mercury.com**