



# Silk Performer 21.0

Web 负载测试教程

**Micro Focus**  
**The Lawn**  
**22-30 Old Bath Road**  
**Newbury, Berkshire RG14 1QN**  
**UK**  
<http://www.microfocus.com>

© Copyright 1992-2020 Micro Focus 或其关联公司。

**MICRO FOCUS**、**Micro Focus** 徽标和 **Silk Performer** 是 **Micro Focus** 或其关联公司的商标或注册商标。

All other marks are the property of their respective owners.

2020-10-21

# 内容

<b>Web 负载测试教程</b> .....	<b>4</b>
Web 负载测试概述 .....	4
Web 2.0 应用程序示例 .....	4
定义 Web 负载测试项目 .....	5
创建测试脚本 .....	6
录制测试脚本 .....	6
脚本试运行 .....	6
尝试使用您的测试脚本 .....	6
分析测试脚本 .....	7
利用 TrueLog Explorer 执行可视化分析 .....	7
查看虚拟用户摘要报告 .....	8
在 TrueLog 中查找错误 .....	8
查看页面统计信息 .....	8
比较录制并回放 TrueLog .....	9
自定义测试脚本 .....	10
用户输入数据自定义 .....	10
验证 .....	12
解析函数 .....	15
会话处理 .....	15
用户配置文件 .....	18
配置监控 .....	19
定义监控选项 .....	19
调整负载 .....	20
负载模型 .....	20
定义负载 .....	21
运行多次负载测试 .....	21
运行一次负载测试 .....	21
监控负载测试 .....	21
使用图形监控服务器性能 .....	22
浏览测试结果 .....	23
TrueLog On Error .....	23
Performance Explorer 概述 .....	24

# Web 负载测试教程

本教程将帮您了解使用 Silk Performer 执行 Web 应用程序负载测试的过程，帮助您尽快开始投入相关工作。本教程将帮助您充分利用 Silk Performer 的全部易用、先进的功能。

## Web 负载测试概述

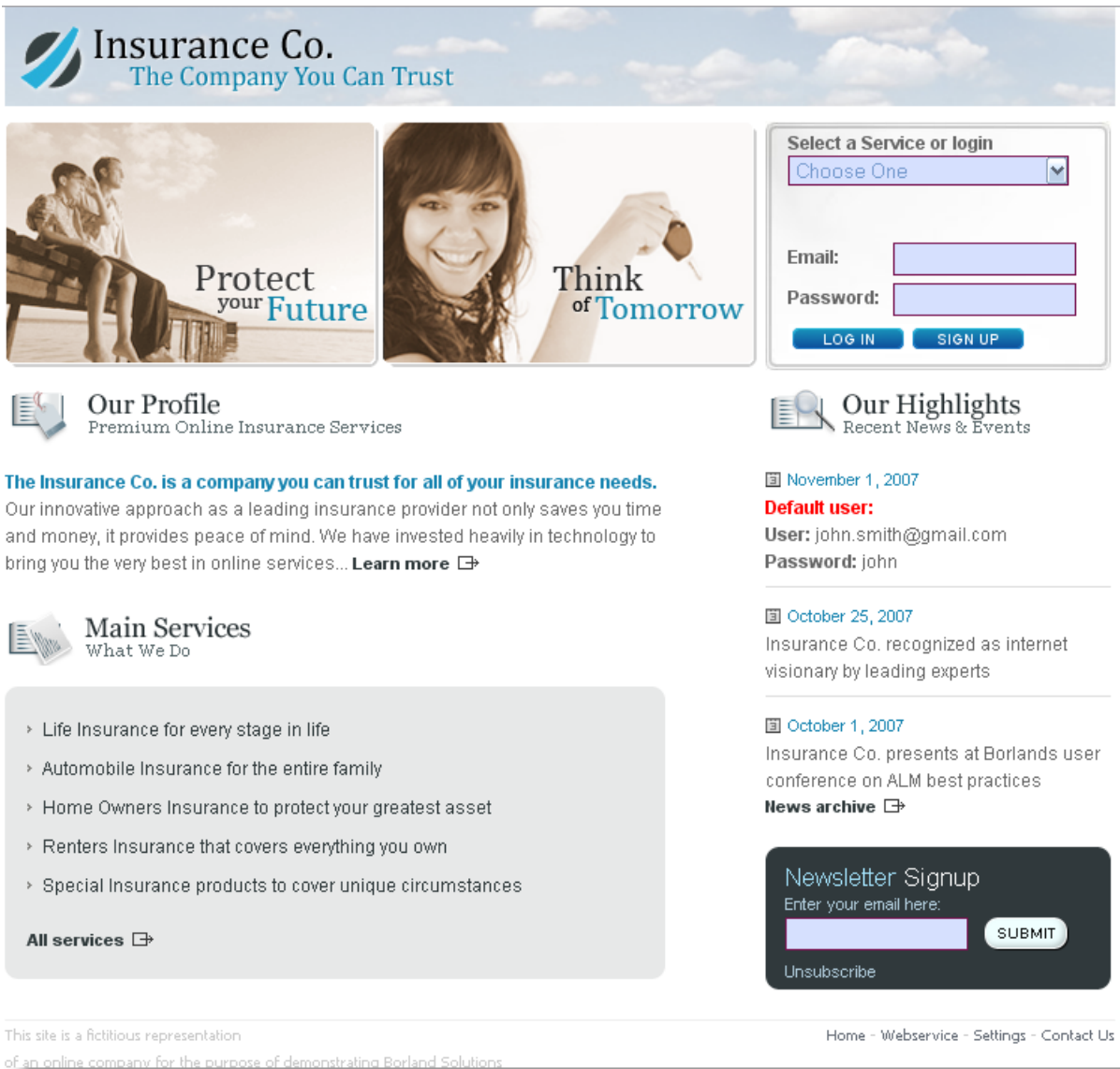
在测试当今的主流 Web 应用程序时，最简单快捷的方法莫过于在协议级别上 (HTML/HTTP) 执行测试。这种方法将生成包含高级功能的简单脚本。当测试包含 AJAX 逻辑的 Web 应用程序，并且在协议级别的测试经证明未成功时，建议使用浏览器驱动的 Web 负载测试。

除了方便执行主流 Web 应用程序的协议级别 (HTTP) 测试之外，Silk Performer 现在还可让您使用实际的 Web 浏览器 (Internet Explorer、Firefox 和 Chrome) 生成负载。通过这种方式，您可以利用 Web 应用程序内置的 AJAX 逻辑，在测试过程中准确模拟复杂的 AJAX 行为。这种强大的测试方法的结果可以反应真实用户的浏览体验，包括展现时间和协议级别的统计信息。

## Web 2.0 应用程序示例

Silk Performer 提供了一个主流 Web 应用程序示例，可用于了解 Web 2.0 应用程序测试。InsuranceWeb 示例 Web 应用程序基于 ExtJS 和 JSF 框架构建，使用 AJAX 技术，通过 JSON 和 XML 进行通信。

示例应用程序托管在 <http://demo.borland.com/InsuranceWebExtJS/> 上。



## 定义 Web 负载测试项目

1. 单击 Silk Performer workflow 栏上的**从此处开始**。



**注:** 如果已经打开了另一个项目, 请从菜单栏中选择**文件 > 新建项目**, 确认关闭当前打开的项目。

随即显示**workflow - 项目大纲**对话框。

2. 在**名称**文本框中, 输入项目的名称。
3. 在**说明**中输入项目说明, 可选。
4. 在类型菜单树中, 选择**Web 业务事务 (HTML/HTTP)**。
5. 单击**下一步**, 根据您的设置创建项目。

随即显示**workflow - 模型脚本**对话框。

# 创建测试脚本

要创建测试脚本，最便捷的方法就是使用 Silk Performer Recorder，Silk Performer 捕获/录制 Web 交互和生成测试脚本的引擎是基于通信报文。

## 录制测试脚本

1. 单击 workflow 栏上的**模型脚本**。随即显示 **workflow - 模型脚本** 对话框。
2. 根据您希望用于录制的浏览器，选择**录制配置文件**列表中列出的浏览器之一。
3. 在 **URL** 字段中，输入要录制的 URL。



**注:** 可通过 <http://demo.borland.com/InsuranceWebExtJS/> 获得 InsuranceWeb 示例 Web 2.0 应用程序。**Select a Service or login** 列表中，自动配额 Auto Quote 和 Agent Lookup 服务可用于执行测试，此列表中的其他服务则未提供任何功能。

4. 单击**开始录制**。Silk Performer Recorder 对话框将打开，并显示为最小化形式，客户端应用程序随之启动。
5. 要查看录制期间发生的操作的报告，请单击**更改 GUI 大小 (Change GUI size)** 按钮最大化 Recorder 对话框。最大化的 Recorder 将打开**操作 (Actions)** 页面。
6. 在客户端应用程序上，执行希望在测试中模拟的与目标服务器交互的动作。Recorder 将捕获和录制交互。关于您的操作以及所下载数据的报告将显示在**操作 (Actions)** 页面中。
7. 若要结束录制，请单击**停止录制 (Stop Recording)** 按钮。
8. 输入 .bdf 文件的名称并保存文件。此时将显示**捕获文件**页面。单击**生成脚本**来根据捕获文件生成脚本。

## 脚本试运行

生成测试脚本后，通过执行脚本试运行来确定脚本是否能无错运行。脚本试运行将确定脚本是否能精确重现您使用 Recorder 录制下的操作。它还会确定脚本中是否包含任何必须参数化的上下文特定会话信息，以保证脚本无错运行。

在脚本试运行时，仅运行一个虚拟用户，为了在脚本执行过程中没有思考时间和延迟，可以启用 stress-test 选项。

## 尝试使用您的测试脚本

1. 单击 Silk Performer workflow 栏上的**尝试使用脚本**按钮。随即显示 **workflow - 尝试使用脚本** 对话框。
2. 从**脚本**列表框中选择一个脚本。
3. 在**配置文件**列表框中，当前活动配置文件将被选中（如果您尚未配置其他配置文件，这将是默认配置文件）。
  - a) 要为选定配置文件配置模拟设置，请单击列表框右侧的**设置**。
  - b) 要配置项目属性，请选择**项目属性**链接。
4. 在包含用户组和虚拟用户的**用户组**列表中，选择您要用以运行虚拟用户的用户组。  
由于这属于脚本试运行，因此仅运行一个虚拟用户。
5. 为实时查看脚本试运行期间从 Web 服务器下载的实际数据，选中**使用 TrueLog Explorer 动态运行 TrueLog Explorer** 复选框。  
如果测试的并非 Web 应用程序，就应该禁用此选项。
6. 单击**运行**。脚本试运行将开始。

在脚本试运行过程中，所有录制的思考时间都将被忽略。此时将打开**监控**窗口，展示有关脚本试运行进度的详细信息。如果选择了**动态**选项，TrueLog Explorer 将打开。在这里可以查看脚本试运行过程中下载的实际

数据。如果在脚本试运行过程中发生任何错误，TrueLog Explorer 可帮助您迅速找到错误并自定义会话信息。使用 TrueLog Explorer 检查和自定义脚本后，脚本应可保证无误运行。

## 分析测试脚本

Silk Performer 提供了多种在执行脚本试运行后评估测试脚本的方法。

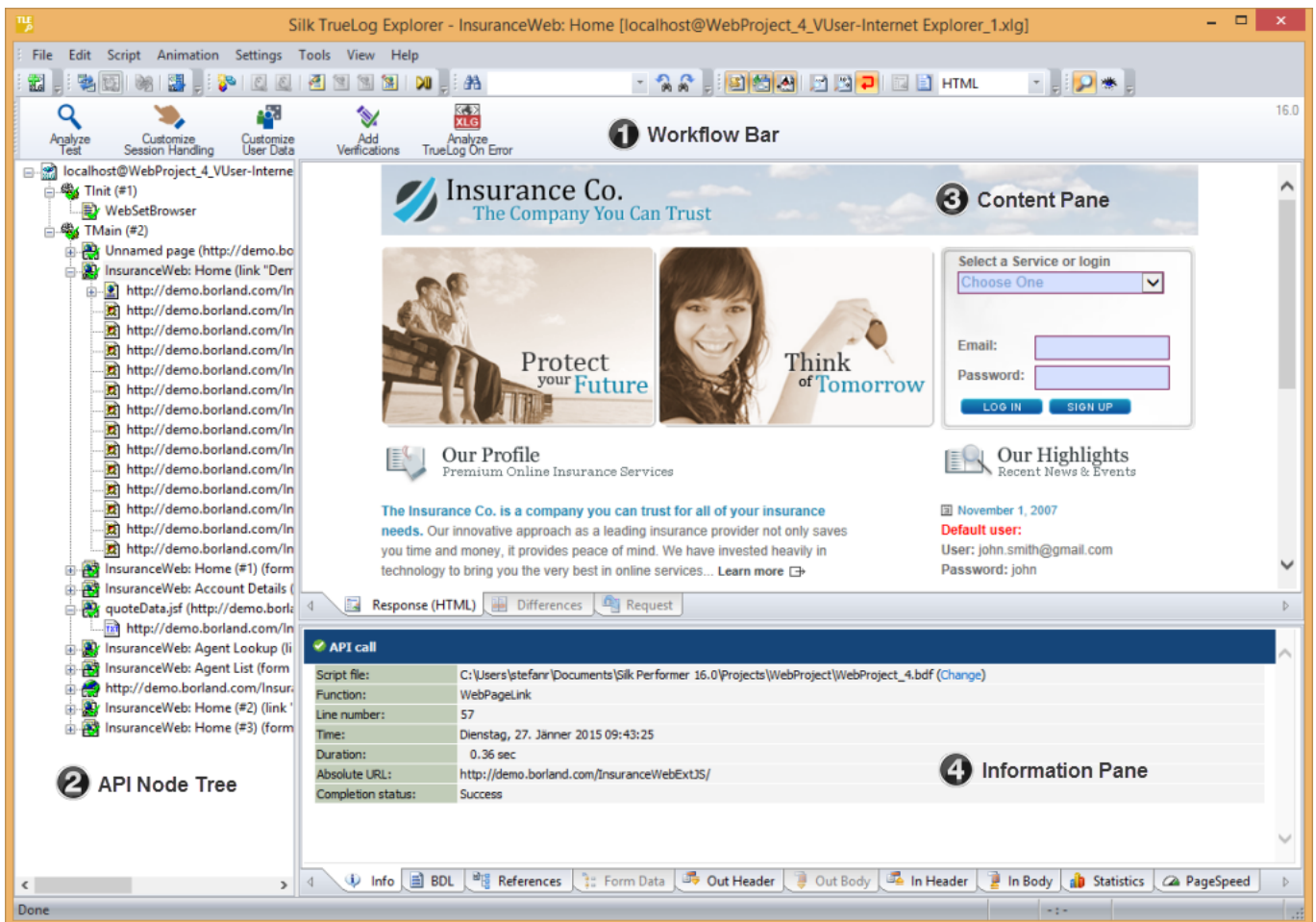
## 利用 TrueLog Explorer 执行可视化分析

TrueLog Explorer 最强大的功能之一就是能够以可视化的方式呈现所测试应用程序显示的 Web 内容。实际上，它将为您展示虚拟用户与应用程序交互时看到的内容。

TrueLog Explorer 接口包含以下部分：

- 在您使用 TrueLog Explorer 的过程中， **workflow 栏**将是您的主要工作接口。Workflow 栏支持 TrueLog Explorer 的五项主要任务，体现了它的内置测试方法论。
- 界面左侧的  **API 节点树 (API Node Tree)** 菜单允许展开和折叠测试过程中下载的 TrueLog 数据。每个已加载的 TrueLog 文件都会显示在此处，同时还会显示所有相关 API 节点的链接。您可以单击一个节点，在  **屏幕 (Screen)** 窗格中显示屏幕快照，在  **信息 (Information)** 视图中显示历史录制详细信息。
- **内容 (Content)** 窗格提供了所有已接收数据的多种视图。
- **信息 (Information)** 窗格显示有关测试脚本和测试运行的数据，包括关于已加载 TrueLog 文件、所选 API 节点、BDL 脚本和统计信息的常规信息。

 **注：**要从 Silk Performer 启动 TrueLog Explorer，请选择**结果 > 浏览 TrueLog**。



The screenshot displays the TrueLog Explorer interface. On the left is the **API Node Tree** (2), showing a hierarchy of test nodes. The main area is divided into three panes: **Workflow Bar** (1) at the top with navigation icons; **Content Pane** (3) in the middle showing a screenshot of the 'Insurance Co.' website; and **Information Pane** (4) at the bottom showing details for an API call. The API call details include:

Script file:	C:\Users\stefanr\Documents\Silk Performer 16.0\Projects\WebProject\WebProject_4.bdf (Change)
Function:	WebPageLink
Line number:	57
Time:	Dienstag, 27. Jänner 2015 09:43:25
Duration:	0.36 sec
Absolute URL:	http://demo.borland.com/InsuranceWebExtJS/
Completion status:	Success


The bottom status bar shows various tool icons like Info, BDL, References, Form Data, Out Header, Out Body, In Header, In Body, Statistics, and PageSpeed.

## 分析测试运行

1. 将脚本试运行中的 TrueLog 加载到 TrueLog Explorer，单击 workflow 栏上的**分析测试**按钮。随即显示**分析测试**对话框。
2. 继续选择以下选项之一：
  - 查看虚拟用户摘要报告
  - 查找 TrueLog 中的错误
  - 比较回放的测试运行与录制的测试运行

## 查看虚拟用户摘要报告

虚拟用户摘要报告是单次脚本试运行的摘要报告，提供基本说明和平均计时。每份报告跟踪单独一个虚拟用户。数据显示为表格的格式。

 **注：**由于虚拟用户摘要报告需要大量数据处理，因此默认不生成。


要在动态脚本试运行结束时显示虚拟用户报告，或者在菜单树中单击一个 TrueLog 文件的根节点时自动显示虚拟用户报告，请在**设置 (Settings) > 选项 (Options) > 工作空间 (Workspace) > 报告 (Reports)** 中选中**显示虚拟用户报告 (Display virtual user report)** 复选框。

虚拟用户摘要报告包含以下内容的详细信息：

- 虚拟用户
- 未覆盖的错误
- 针对测试脚本中定义的每个事务跟踪得到的响应时间信息
- 所下载的各 Web 页面的时间度量
- 度量测试脚本中定义的每个 web 组件，包括使用 POST、GET 和 HEAD 方法提交时的响应时间度量和吞吐率。
- 脚本中使用的各计时器和计数器（度量函数）
- 包括 IIOP、Web 表单、TUXEDO、SAP 和其他方面的信息

## 在 TrueLog 中查找错误

在脚本试运行后，TrueLog Explorer 可帮助您迅速查找错误。检查所有出错信息，判断是否有必要进行自定义修正。

 **注：**在菜单树中查看时，包含回放错误的 API 节点将标有红色“X”标记。

1. 打开要分析或修改的 TrueLog。
2. 在 workflow 栏上单击**分析测试 (Analyze Test)**。随即打开**工作流 - 分析测试 (Workflow - Analyze Test)** 对话框。
3. 单击**查找错误**链接。此时将显示**逐步查看 TrueLog** 对话框，**错误**选项将处于选中状态。
4. 单击**查找下一个逐步查看 TrueLog 结果文件**，每次检查一个错误。

您可以选择在 TrueLog 中浏览时使用不同前进增量，通过可视化的方式验证脚本是否如预期工作（**整页、HTML 文档、表单提交或 API 调用**）。

## 查看页面统计信息

验证测试运行的准确性之后，TrueLog Explorer 可以通过**信息 (Information)** 窗口下的**统计信息 (Statistics)** 选项卡，分析应用程序在“无负载”条件下的性能。**概述 (Overview)** 页面详述了总体页面响应时间、文档下载时间（包括服务器繁忙时间）以及接收嵌入对象所用的时间。



详细的 Web 页面统计信息显示了各 Web 页面组件的准确响应时间。这些详细统计信息能帮助您查明错误和页面下载缓慢的根本原因。

详细的 Web 页面向下钻取结果包含针对各页面组件的以下数据：

时间	说明
DNS	用于将主机名称转换为 IP 地址的时间。
连接	用于与服务器建立连接的时间。
SSL 握手	用于在现有连接上建立安全层的时间。客户端和服务器交换证书和商定加密技术（包括相关密钥），以保护数据传输的安全。
发送	用于将请求数据从 Silk Performer 移交到操作系统的时间。此值不包括将请求数据从操作系统发送到服务器所需的时间。
服务器正忙	用于将请求数据从操作系统发送到服务器的时间、用于计算服务器上响应数据的时间、以及用于将响应数据的第一个字节从服务器发送到客户端的时间（包括网络延迟）。
接收	用于接收整个响应数据的时间。
缓存统计信息	


## 查看概述页面

1. 在 API 节点树菜单中，选择要查看统计信息的 API 节点。
2. 单击**统计信息**选项卡打开**统计信息**视图。
3. 选择 URL 列中的特定组件，以查看详细分析和页面向下钻取。

## 比较录制并回放 TrueLog


通过将脚本开发过程中生成的 TrueLog 与最初录制的对应 TrueLog 进行对比，即可验证测试脚本运行是否准确。

1. 单击 workflow 栏上的**分析测试**按钮。随即显示**workflow - 分析测试**对话框。
2. 单击**比较测试运行**。
3. 所录制的对应 TrueLog 将在比较视图中打开，显示**逐步查看 TrueLog**对话框，**浏览器节点**选项处于选中状态，允许您对 TrueLog 执行逐节点的比较。
4. 单击**查找下一个逐步查看 TrueLog 结果文件**，每次检查一个页面。

 **注：**显示回放内容的窗口左上角带有绿色三角形。显示原始录制内容的窗口左上角带有红色三角形。

## 同步录制和回放 TrueLog

在比较模式下，您可以同步回放与录制 TrueLog 之间对应的 API 节点，以确定录制值与回放值之间的差异。

 **注：**启用 TrueLog 的自动同步时，此功能将禁用。

1. 要启用比较模式，请执行以下操作之一：
  - 选择**视图 > 比较模式**。
  - 单击工具栏上的**比较模式**按钮。
2. 打开录制和回放 TrueLog 的对应集。
3. 右键单击一个 API 节点，选择**同步 TrueLog**。TrueLog Explorer 将在匹配的 TrueLog 中定位与选定 API 节点关系最为密切的 API 节点。

# 自定义测试脚本

使用 Silk Performer 生成测试脚本并执行脚本试运行后, TrueLog Explorer 可帮助您通过以下方式自定义脚本 (以下所列仅为部分方式) :

- **参数化输入数据** - 通过用户数据自定义, 您可以将静态、录制的、用户输入的数据替换为在每次事务中都有所变化的动态、参数化的用户数据, 让测试脚本更贴近现实。运行此类数据驱动型测试时无需手动执行脚本。此功能适用于 Web、数据库、XML、Citrix、SAPGUI、终端模拟和 Oracle Forms 应用程序。
- **为测试脚本添加验证** - 利用**添加验证**工具, 您可以更清晰地看到测试期间所下载的数据, 让您能够验证客户端确实已接收到服务器发送的内容。在系统部署完成后, 验证仍然非常有用, 可用于持续性能管理。此功能适用于 Web、数据库、XML、SAPGUI、Citrix、终端模拟和 Oracle Forms 应用程序。对于 Citrix 应用程序, TrueLog Explorer 支持位图和窗口验证。

## 用户输入数据自定义

如果没有自定义用户输入的数据, 所有模拟事务都将完全相同, 不能体现真实环境中的变量值。

例如, 您可以使用**参数向导**, 自定义测试过程中输入到表单中的用户输入数据。**参数向导**允许您指定要在测试过程中输入窗体字段的值。这能将录制的用户输入数据替换为随机化的、参数化的用户数据, 从而让测试脚本更贴近真实环境。

### 使用新参数自定义 HTML 用户数据

继续之前, 请确保已从测试脚本中移除所有静态会话信息, 并且最近一次脚本试运行生成的 TrueLog 已在 TrueLog Explorer 中打开。

对于基于 HTML 的应用程序, 用户数据自定义的目标是自定义提交到表单字段的值。

此任务解释了使用随机变量创建参数的过程。

1. 单击 workflow 栏上的**自定义用户数据**。随即将打开 **workflow - 自定义用户数据** 对话框。
2. 单击**自定义 HTML 表单中的用户输入数据** 链接。TrueLog Explorer 随即执行以下操作 :

- 选择菜单树中的第一个 **WebPageSubmit API 调用** 节点。
- 打开**逐步查看 TrueLog** 对话框**表单提交** 选项按钮处于选中状态)。
- 显示**请求**视图

**请求**视图将显示一个页面, 其中包含选定 WebPageSubmit 调用提交的 HTML 表单。光标扫过表单控件时, 将显示工具提示, 其中包含控件的名称及其初始值和提交的值; 橙色线条表示下方**表单数据**视图中的对应 BDL 表单字段声明。

3. 在**逐步查看 TrueLog** 对话框中单击**查找下一个或查找上一个**, 浏览 TrueLog 中的所有 WebPageSubmit 调用 (这些也是可能需要参数化的候选调用)。

 **注:** 请求视图中突出显示的 HTML 控件标识可自定义的表单字段。

4. 在**请求**页面上, 右键单击要自定义的表单控件, 选择**自定义值**。

您可以将录制的值替换为各种类型的输入数据 (包括来自文件的预定义值和常规随机值), 并生成测试脚本代码, 使用自定义数据代替录制的输入数据。

此时将打开**参数向导 (Parameter Wizard)**。

利用**参数向导 (Parameter Wizard)**, 您可以通过两种方式修改脚本值 :

- 使用脚本的 dclparam 或 dclrand 部分中定义的现有参数。
- 根据新常量值、随机变量或多列数据文件中的值新建参数。

新建参数后, 该参数将添加到现有参数中, 可在后续自定义中使用。

5. 单击**新建参数 (Create new parameter)** 选项按钮，单击**下一步 (Next)** 新建参数。此时将打开**新建参数 (Create new parameter)** 页面。
6. 单击**基于随机变量的参数 (Parameter from Random Variable)** 选项按钮，然后单击**下一步 (Next)**。此时将打开**随机变量 (Random Variable)** 页面。
7. 在列表框中选择要插入到测试脚本中的随机变量类型，然后单击**下一步 (Next)**。  
下方窗口中将显示所选变量类型的简短说明。  
将开**命名变量并指定其属性 (Name the variable and specify its attributes)** 页面。
8. 在**名称 (Name)** 文本框中输入变量名称。
9. 指定值的调用规则：**随机**还是**顺序**。  
**文件中的字符串**随机变量类型将生成可从指定文件中随机或顺序选择的数据字符串。
10. 在**文件组合框**中，从**名称**列表框中选择预配置的数据源，然后单击**下一步**。此时将显示**选择使用情况类型** 页面。
11. 选择以下选项之一，指定要使用的新随机值：
  - **按使用情况 (Per usage)**
  - **按事务 (Per transaction)**
  - **按测试 (Per test)**
12. 单击**完成**。现在，您的测试脚本将为给定表单字段使用随机变量，取代原有的录制值。新随机变量函数将出现在 **BDL** 页面上。

使用测试脚本中的随机变量函数启动脚本试运行，确认脚本无错运行。

## AJAX 和脚本自定义

Silk Performer Recorder 可以录制和回放使用 AJAX (异步 JavaScript 和 XML) 请求的 Web 应用程序。这是因为 Silk Performer 能识别以 HTML 响应内的 XML 或 JSON 形式传入的异步 AJAX 请求和响应。Silk Performer 会将其遇到的 AJAX 请求作为 `WebPageUrl` 调用编入脚本。

Silk Performer 和 TrueLog Explorer 支持访问 AJAX 请求中的值，支持各种脚本中自定义的数据，例如输入数据参数化、验证、解析和 AJAX 响应中的自定义会话信息。



**提示:** 为了测试，InsuranceWeb 演示应用程序（可通过 <http://demo.borland.com/InsuranceWebExtJS/> 获得）提供了在 JSON 与 XML 序列化方法之间切换的功能。在**选择服务或登录**列表中，为 JSON 选择代理查找，或为 XML 选择代理查找（XML）。

### 格式良好的 JSON 和 XML 数据

JSON 和 XML 是 AJAX 应用程序、REST 技术和其他环境中常用的数据结构格式。Silk Performer 支持以良好的格式查看 BDF 脚本中 XML 和 JSON 格式的字节流。增强的 JSON 格式数据支持通过 TrueLog Explorer 的字符串自定义函数更轻松地自定义字符串值。

当 JSON 格式的数据被录制或插入到 BDF 脚本时，Silk Performer 将显示原始 JSON 字节流。可以使用 Silk Performer 的参数向导轻松自定义以 JSON 格式显示的 XML 数据。

Silk Performer 提供以格式良好的 JSON 呈现视图或原始 JSON 字节流的形式查看 JSON 数据的选项。

### 理解 JSON

根据 JSON.org ([www.json.org](http://www.json.org)) 的定义：“JSON (JavaScript 对象注释) 是一种轻量级数据交换格式。”人类可以轻松读写此格式的数据。机器可以轻松解析和生成此格式的数据。它是基于 JavaScript 编程语言的子集...” “JSON 是一种完全独立于语言的文本格式，但它采用一种 C 系列语言（包括 C、C++、C#、Java、JavaScript、Perl、Python 等其他许多语言）程序员熟悉的约定。这些特点让 JSON 成为理想的数据交换语言。”

## 支持在 TrueLog Explorer 中查看格式良好的 JSON 和 XML 数据

如果 TrueLog Explorer 无法检测到 JSON 或 XML 数据, 启用支持在响应 (Response) 页面中查看格式良好的 JSON 和 XML 数据。如果 TrueLog Explorer 检测到 JSON 和 XML 数据, 这些数据将自动在响应 (Response) 页面上设置为良好格式。

1. 在 TrueLog 菜单树中选择包含 JSON 或 XML 格式数据 (例如 HTTP Post 命令节点) 的节点。
2. 单击响应选项卡。
3. 右键单击 JSON 或 XML 数据, 选择渲染为 > JSON 或渲染为 > XML, 为数据设置良好格式。

## 支持在中查看格式良好的 JSON 和 XML 数据 Silk Performer

1. 在包含 JSON 或 XML 格式数据的 BDL 脚本中, 在要查看的屏幕数据内右键单击。
2. 从上下文菜单中选择首选查看格式。
  - 选择格式为 > JSON, 将数据设置为增强 JSON 格式。
  - 选择格式为 > XML, 将数据设置为原始 XML 格式。

您还可以从上下文菜单中选择格式为自动套用格式, 由 Silk Performer 自动为屏幕数据类型确定最佳格式设置选项。默认情况下, JSON 与 XML 数据在 BDF 脚本中均采用良好格式。

3. 右键单击格式化的数据字符串, 可访问的标准字符串自定义命令。



**注:** 由于格式化的 JSON 数据已集成到 Silk Performer 代码编辑器, 因此可以使用工具栏上的撤销/恢复按钮撤销/恢复 JSON 格式设置。



**注:** 在 TrueLog Explorer 中也可以查看格式良好的 JSON 和 XML 数据。

## 验证

通常, 应用程序错误不会导致 HTTP 响应出错。而是会导致应用程序发出包含错误数据值的响应, 或者在 HTML 内容中整合错误消息, 例如发生 Servlet 异常...或者服务器过于繁忙...。简单地检查 HTTP 状态代码无法发现此类错误。因此, 如果您未在测试脚本中建立其它检查, 往往会忽视应用程序错误。而验证函数可帮助您检查这些并不简单、不标准的 HTTP 错误类型的应用程序错误。

在测试脚本中内置验证后, 您的测试将从简单负载测试演进为综合的负载和功能测试。即便在大负载测试场景中, 您也可以使用这些内置验证的脚本, 而不会对性能造成重大的不利影响。因此, 用户可以检测到仅在负载下才会发生的新类型的应用程序错误, 而这些错误往往会被无额外验证检查的简单负载测试脚本所忽视。

Silk Performer 提供了三种方法来帮助您使用验证功能从而强化测试脚本:

- 由 Recorder 在录制期间自动生成验证函数。
- 通过手动编写验证函数直接强化脚本。
- 通过 TrueLog Explorer 以可视化的方式来实现验证, 完全无需编写 BDL 代码。TrueLog Explorer 将自动在脚本中生成验证函数。



**提示:** 有关详细信息, 请参阅 TrueLog Explorer 帮助。


## 在录制过程中自动生成验证

若要启用在 Silk Performer 脚本录制期间自动生成验证, 请执行以下操作:

1. 在 Silk Performer 中选择设置 > 激活配置文件。此时将显示配置文件 - [<配置文件名称>] - 模拟对话框。
2. 在快捷方式列表中选择录制 > Web, 然后单击验证选项卡。
3. 在录制部分中, 选中录制标题验证和录制摘要验证复选框。
4. 单击确定。

## 标题验证

标题验证提供简单、自动的方法来检查应用程序是否返回了正确的 Web 页面。检索 HTML 文档时, Recorder 可为每个页面级别 Web API 调用或低级别 Web API 调用自动生成标题验证函数。

 **注:** 如果未设置 HTML 页面标题 (<TITLE HTML>标记的内容), 或者 HTML 页面标题共用公用名称, 标题验证将无法使用。

### 标题验证

```
transaction TMain
begin
    ...
    WebVerifyHtmlTitle("ShopIt - Greetings",
    WEB_FLAG_IGNORE_WHITE_SPACE |
    WEB_FLAG_EQUAL | WEB_FLAG_CASE_SENSITIVE, 1,
    SEVERITY_ERROR, bVerifyTitleSuccess1);
    WebPageUrl("http://myHost/shopit");
```

## 摘要验证

在验证相对静态的 HTML 页面的内容时, 摘要验证非常有用。摘要验证不适用于动态的信息, 或是在超链接、嵌入式对象或隐藏表单字段里包含会话信息的 HTML 页面 (此类页面会在每次调用时发生更改)。

函数调用 WebVerifyDataDigest 和 WebVerifyHtmlDigest 来检查 Silk Performer 在回放过程中获取的数据是否不同于在回放过程中获取的数据是否不同于 Recorder 在对应的录制会话中捕获到的数据。利用这些函数调用, Silk Performer 可以生成包含所有字符的出现与否的摘要信息, 并将这些结果与 Recorder 生成的摘要信息对比。此函数使得您可以确定 Silk Performer 在回放期间捕获的数据与录制时是否相同。

Recorder 将自动为每个页面级别 Web API 调用和低级别 Web API 调用生成摘要验证函数, 检索包含摘要验证指定内容类型的文档 (默认内容类型为文本/HTML)。

### 摘要验证

```
const
    gsVerDigest_ShopIt_Greetings := "\h014200000B..."
                                   "\h016000500A..."
                                   "\h030001000A..."
                                   "\h0900410020...";

transaction TMain
begin
    ...
    WebVerifyDataDigest(gsVerDigest_ShopIt_Greetings, 162);
    WebPageUrl("http://myHost/shopit", "ShopIt - Greetings");
    ...
end TMain;
```

## 在回放过程中启用验证检查

参阅验证设置选项, 了解可在脚本回放过程中启用或禁用的验证检查列表。




**提示:** 可以使用 WebSetOption BDL 函数重写脚本中的配置文件设置。

1. 在 Silk Performer 中选择 **设置 > 激活配置文件**。此时将显示 **配置文件 - [<配置文件名称>] - 模拟** 对话框。
2. 在快捷方式列表中选择 **回放 > Web**, 然后单击 **验证** 选项卡。
3. 在 **HTML/XML** 和 **数据** 区域中, 选中您想在回放过程中启用的验证检查。

## 插入内容验证函数


1. 打开要分析或修改的 TrueLog。
2. 选择包含待验证内容（例如文本或图像）的 TrueLog API 节点。
3. 在源页面上选择待验证的内容。

 **注:** 对于页面标题函数和页面摘要验证函数，此步骤并非必需。

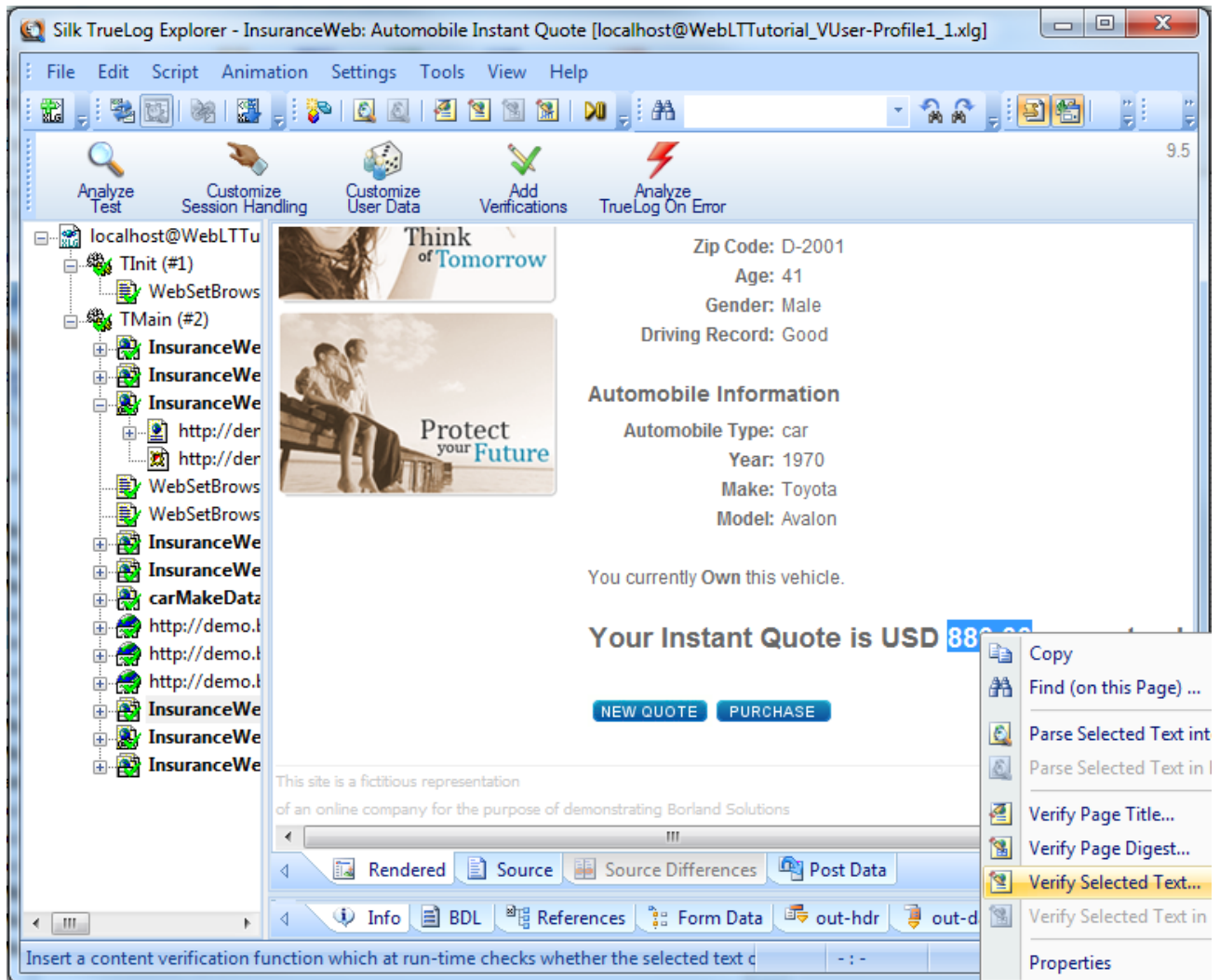
4. 单击 workflow 栏上的**添加验证**。此时将打开 workflow - 添加验证对话框。
5. 选择预先启用的验证：

- 验证页面标题
- 验证选定文本
- 验证 HTML 表格中的选定文本
- 验证摘要

6. 完成以下对话框。  
详细说明验证函数应如何插入到 BDL 脚本。

 **注:** 系统将自动识别左右边界。

7. 对您希望添加到 BDL 脚本的各验证重复此流程。
8. 在 workflow - 添加验证对话框中单击**是**。此时将启动脚本试运行。
9. 确认证验成功通过。  
包含验证的 API 节点以蓝色“V”型符号表明验证通过。



# 解析函数

与验证函数相似的是，解析函数常用于解析从 Web 返回的内容并检查值是否符合您的测试条件。与验证函数不同的是，解析函数解析完从服务器返回的内容之后将解析的值存入 BDL 变量中，而验证函数主要验证制定输入值是否出现。通常，您将使用解析函数处理以下任务：

- 解析会话 ID，使用解析后的动态会话 ID 取代脚本中的静态会话 ID，从而在 Web 应用程序中维护状态信息。这是解析函数的主要应用领域之一。
- 利用解析函数可以强化脚本的验证，而这是无法通过验证函数实现的。例如，如果您要验证 HTML 表中第 3 行第 2 列的值是否等于第 1 行第 2 列与第 2 行第 2 列值之和。通过使用解析函数可以解析出三个值，在脚本中进行比较，从而构建这种强化的验证检查。
- 根据服务器的返回值来有条件的执行部分脚本。例如，一个 HTTP 请求将返回一个 HTML 页面，其中包含：结果：Results: <nnn> items found。基于<nnn> 的值，您想要执行不同的操作。假设您要在 <nnn> = 0 时退出事务，在 <nnn> = 1 时转到[详细信息 \(Details\)](#)的链接，并在 <nnn> is greater than 0 时转到[下一步 \(Next\)](#) 的链接。为实现此目标，您需要解析 HTML 页面的 <nnn> 值，并根据解析出的值将操作编入脚本。

Silk Performer 提供两种方法，通过解析功能来强化测试脚本：

- 通过手动编写验证函数代码直接强化脚本。
- 通过 TrueLog Explorer 以可视化的方式应用解析函数，完全无需编写 BDL 代码。TrueLog Explorer 将自动在脚本中生成验证函数。



**提示：**有关详细信息，请参阅 TrueLog Explorer 帮助。

## HTML 内容解析

HTML 内容解析函数将解析部分可见的 HTML 显示内容。HTML 内容验证和解析函数允许验证和/或解析浏览器中显示的文本内容。您可以在 TrueLog Explorer 的 **HTML 视图 (HTML view)** 中应用如下 HTML 内容解析函数：

- WebParseHtmlBound
- WebParseHtmlTitle
- WebParseTable
- WebParseResponseTag
- WebParseResponseTagContent

## 响应数据解析

数据解析函数将解析部分从服务器返回的响应数据。如果服务器返回了 HTML 文档，这些函数将解析出 HTML 文档的完整源代码。您可以在 TrueLog Explorer 的 **源视图 (source view)** 中应用如下数据验证函数：

- WebParseDataBound (此函数取代已不推荐使用的 WebParseResponseData 函数)
- WebParseResponseHeader

## 会话处理

Web 应用程序通常使用唯一 Web 会话 ID，让 Web 服务器能更进一步处理客户端请求。

有关详细信息，请参阅 TrueLog Explorer 帮助。

## 自定义会话处理

为了标识客户端的进一步的请求，Web 服务器应用程序通常会在运行时生成必要的信息。在服务器发送给浏览器的响应中，可能包含通常称为会话 ID 的唯一字符串。浏览器在每次后续请求中将此字符串返回给服务

器，服务器会通过此字符串来标识不同的 Web 请求会话。会话 ID 通常在 Web 服务器应用程序标识独立用户，验证之前访问的用户状态信息时使用。

会话 ID 可通过多种方法发送到客户端。经常可以在 Cookie、超链接或嵌入式对象所用 URL 的 HTML 内，或者隐藏 HTML 表单字段中找到会话 ID。会话 ID 将包含在 Cookie、URL 和 HTTP 发布数据中发回到服务器。

### Cookie 中的会话信息

从服务器发出：

```
Set-Cookie: SessionID=LGIJALLCGEBMIBIMFKOEJIMM; path=/
```

发往服务器：

```
Cookie: SessionID=LGIJALLCGEBMIBIMFKOEJIMM
```

### HTML 链接的 URL 中的会话信息

从服务器发出：

```
<html>
...
<a href="/ShopIt/acknowledge.asp?
SessionID=LGIJALLCGEBMIBIMFKOEJIMM" >
  Enter Shop
</a>
...
</html>
```

发往服务器：

```
GET /ShopIt/acknowledge.asp? SessionID =
LGIJALLCGEBMIBIMFKOEJIMM HTTP/1.1
```

### 隐藏表单字段中的会话信息

从服务器发出：

```
<html>
...
<form action="kindofpayment.asp" method="post" >
  Currently we only accept Credit Cards
  <input type="hidden" name="SessionID"
value="LGIJALLCGEBMIBIMFKOEJIMM">
  <input type="text" name="name" value="Jack " >
  <input type="submit" name="paymentButton" value="Submit">
</form>
...
</html>
```

发往服务器：

```
POST /ShopIt/kindofpayment.asp HTTP/1.1
...
SessionId=LGIJALLCGEBMIBIMFKOEJIMM&name=Jack&paymentButton=Submit
```

## 何时使用自定义会话处理

假设您脚本中的 `WebPageUrl` 调用使用一个 URL，而该 URL 的查询字符串中包含会话 ID。在回放脚本时，这个硬编码的静态会话 ID 将发送到服务器。由于会话 ID 未能正确标识回放的会话（而是仍在标识录制的会话），因此回放无法正确完成。如果未使用运行时生成的动态值取代脚本中的静态的会话 ID，Web 应用程序



通常会生成错误, 例如: 非常抱歉, 您的会话已过期。请返回登录屏幕并重试 (We are sorry, your session has expired. Please return to the login screen and try again)。

好消息是, 使用 Silk Performer 时通常不必执行会话自定义, 即便需要手动自定义会话, Silk Performer 的 TrueLog Explorer 也能指导您完成此过程。

录制脚本时, 往往不需要修改会话处理自定义即可顺利完成任务。Silk Performer 因此使用多个高度复杂的方法, 阻止用户手动处理硬编码的会话 ID:

- **Cookie 管理**: 如果您的服务器使用 cookie 交换会话信息, Silk Performer 将自动处理动态会话 ID 值。由于 Silk Performer 能准确模拟浏览器的 Cookie 管理, 因此将按照与浏览器向服务器发送 Cookie 的相同方式发送 Cookie。状态管理过程中无需手动干预。
- **页面级 Web API**: 在录制时使用页面级 API (这是默认设置) 将生成脚本, 该脚本主要生成具有完整上下文的 Web API 函数调用, 例如 WebPageLink 和 WebPageSubmit。具有完整上下文的 Web API 调用在 HTML 级别上工作, 而非 HTTP 级别, 因此不会使用 URL 作为参数。因为所有具有完整上下文的 API 调用均无需手动自定义会话。如果您在 **项目大纲 (Outline Project)** 对话框中选择了 **Web 业务事务 (HTML/HTTP) (Web business transaction (HTML/HTTP))** 应用程序类型, 页面级 API 将会被使用。因此, 强烈建议使用 Silk Performer 的页面级 API, 而非低级别的 Web API。

由于在动态生成 HTML 的过程中大量使用到客户端的 **JavaScript**, 因此 Silk Performer Recorder 有时会丢失 HTML 上下文, 编写无上下文的 Web API 调用脚本。例如 WebPageUrl 和 WebPageForm 等无上下文的 Web API 调用包含 URL 并将其作为参数。在此类罕见的情况下, 您的脚本可能包含硬编码的会话 ID。您可以在 Web API 调用的 URL 参数以及脚本的 dclform 部分中声明的表单字段中找到这些会话 ID。

#### 具有完整上下文的脚本 (无需要自定义的内容)

```
transaction TMain
begin
    WebPageUrl("http://lab38/ShopIt/"); // first call is always
context-less
    WebPageLink("Join the experience!");
    WebPageSubmit("Enter", SHOPIT_MAIN_ASP001);
    WebPageLink("Products");
end TMain;

dclform
SHOPIT_MAIN_ASP001:
    "SessionID" := "" <USE_HTML_VAL>, // hidden value:
    "LGIJALLCGEBMIBIMFKOEJIMM"
    // recognized as a hidden form
    // field, the value is taken from
    // the actual HTML form field.
    "name" := "Jack", // changed
    "New-Name-Button" := "" <USE_HTML_VAL> ; //unchanged
    value: "Enter"
```

#### 采用无上下文函数的脚本 (部分中包含需要被自定义的静态会话数据)

```
transaction TMain
begin
    WebPageUrl("http://lab38/ShopIt/"); // first call is always
context-less
    WebPageUrl("http://lab38/ShopIt/main.asp", NULL,
SHOPIT_MAIN_ASP001);
    WebPageForm("http://lab38/ShopIt/main.asp",
SHOPIT_MAIN_ASP002);
    WebPageUrl("http://lab38/ShopIt/products.asp");
end TMain;

dclform
SHOPIT_MAIN_ASP001:
    "from" := "welcome";
```

```
SHOPIT_MAIN_ASP002:
"SessionID"      := "LGIJALLCGEBMIBIMFKOEJIMM",
"name"           := "Jack",
"New-Name-Button" := "Enter";
```

## 自定义会话处理

为了自定义会话处理，必须执行以下步骤。

有关详细信息，请参阅 [TrueLog Explorer 帮助](#)。

1. 标识需要自定义的会话 ID。
2. 搜索从服务器向客户端发送的包含会话 ID 的第一条响应（脚本中的 Web API 调用）。
3. 从找到的 Web API 调用响应中解析出会话 ID，将其提取到一个变量中。
4. 使用此变量替换脚本中出现的所有硬编码会话 ID。

## 用户配置文件

### 浏览器类型

在测试中，虚拟用户可以自定义选择众多 Web 浏览器中的任何一个及其包含的功能。当今大多数主流的浏览器均可被模拟。此外也支持一些为移动设备用户提供服务的浏览器。可以通过使用不同版本的线程技术和 HTTP 协议来定义定制化的浏览器。

### 带宽

家庭用户通常通过调制解调器访问 Internet。由于用户的调制解调器有时是网络通信链中速度最慢的节点，因此可以在负载测试中模拟调制解调器。

可以对虚拟用户使用带宽进行自定义，使用当今消费者广泛应用的任意主流连接类型的相应带宽。可以自定义设置带宽，为下行（从服务器到客户端）和上行（从客户端到服务器）通信设置不同的连接带宽。

## 添加配置文件

1. 选择项目 > **新建配置文件**。此时将打开**新建配置文件**对话框。
2. 为新配置文件输入**名称**，然后单击**确定**。在**项目**菜单树中，**配置文件**文件夹将展开，新配置文件将可用。

## 配置浏览器设置

1. 在 Silk Performer 中，展开**项目**树中的**配置文件**节点。
2. 右键单击要配置的配置文件，选择**编辑配置文件 (Edit Profile)**。



**提示:** 此外，也可以从菜单栏中选择**设置 > 激活配置文件**。

此时将打开**配置文件 - [<配置文件名称>] (Profile - [<profile name>])**对话框，左侧快捷方式列表中选择**回放 (Replay)**类别。

3. 在快捷方式列表中，单击 **Web** 图标。
4. 单击**浏览器 (Browser)**选项卡。  
使用**浏览器类型 (Browser type)**区域中说明特定的浏览器设置。
5. 在**浏览器 (Browser)**列表框中选择想要模拟的 Web 浏览器类型。

您所作出的选择将决定模拟的 HTTP 请求中标头信息的格式，以及模拟所用的线程模型。




**注:** 对于移动 Web 应用程序测试 (iPhone、iPad、Android、Windows Phone 或 Blackberry)，您可以更改录制所用的用户代理字符串。

6. 单击**确定**以保存您的设置。

## 配置浏览器模拟设置

1. 在 Silk Performer 中，展开项目树中的**配置文件**节点。
2. 右键单击要配置的文件，选择**编辑配置文件 (Edit Profile)**。

 **提示:** 此外，也可以从菜单栏中选择**设置 > 激活配置文件**。

此时将打开**配置文件 - [<配置文件名称>] (Profile - [<profile name>])** 对话框，左侧快捷方式列表中选择**回放 (Replay)** 类别。

3. 在快捷方式列表中，单击 **Web** 图标。
4. 单击**模拟**选项卡。

使用**模拟**区域，通过设置选项来实际地模拟用户访问网站。

5. 选中**为每个事务模拟用户行为**复选框，让每个虚拟用户在每次事务后重置浏览器模拟。

根据您选择的其他选项，Silk Performer 可以模拟用户初次访问网站，也可以模拟用户再次访问站点。初次访问站点的用户尚未存储持久性 Cookie、无缓存文档，再次访问页面的用户通常会在两次访问网站之间关闭浏览器、缓存文档、设置持久性 Cookie。禁用此选项将允许虚拟用户模拟在测试结束之前不会关闭的 Web 浏览器，从而在多个事务中重复使用缓存的信息。

6. 单击**初次访问用户**选项按钮，从而模拟用户初次访问某网站。

在每次事务完成后，持久性连接将关闭，Web 浏览器模拟将被重置，文档缓存、文档历史记录、Cookie 浏览器模拟将重置，文档缓存、文档历史录制、Cookie 数据、身份验证数据和 SSL 上下文缓存将被清空。在此类情况下，Silk Performer 将从服务器下载整个站点，包括所有文件在内。

7. 单击**再次访问用户**选项按钮，从而模拟用户再次访问某网站。

在每次事务完成后，持久性连接将关闭，文档历史录制、非持久性 Cookie 数据、身份验证数据和 SSL 上下文缓存将被清空。在这种情况下，用户不必清空文档缓存。有关更多详细信息，请参阅 BDL 函数参考中的 WebSetUserBehavior 函数。

使用**用户公差**区域调整用户容忍模拟的高级选项。

8. 单击**确定**，保存**配置文件 - [<配置文件名称>]** 对话框中的设置。

## 配置监控

运行测试之前，您需要定义 Silk Performer 的服务器监控工具 Performance Explorer 来监控测试中用到的本地和远程服务器。服务监控将暴露和定位、协助解决服务器瓶颈问题，允许您检测操作系统和应用程序服务器的性能表现。

有三种监控选项可用：

- **默认监控** - 此选项指导 Performance Explorer 根据所测试的应用程序类型监控推荐的数据源集合。这与在 Performance Explorer 工作空间启用**自动启动监控**和**使用默认监控模板**设置的效果完全相同（**设置 > 激活配置文件 > 回放 > 监控 > 使用默认监控模板**）。
- **自定义监控** - 此选项将以监控模式打开 Performance Explorer，显示**数据源向导 - 选择数据源**对话框，允许您手动配置数据源。Performance Explorer 监控项目设置将与 Silk Performer 项目设置一同保存。
- **无监控** - 此选项使得您在不监控任何本地或远程服务器的情况下运行测试。使用此选项时，**自动启动监控**设置将禁用（**设置 > 激活配置文件 > 回放 > 监控 > 使用默认监控模板**）。

## 定义监控选项

1. 单击 workflow 栏上的**配置监控**。此时将显示**工作流 - 配置监控**对话框。
2. 选择以下任一选项，单击**下一步**：

- **默认监控** - 此选项指导 Performance Explorer 根据所测试的应用程序类型监控推荐的数据源集合。这与在 Performance Explorer 工作空间启用**自动启动监控**和**使用默认监控模板**设置的效果完全相同（**设置 > 激活配置文件 > 回放 > 监控 > 使用默认监控模板**）。
- **自定义监控** - 此选项将以监控模式打开 Performance Explorer，显示**数据源向导 - 选择数据源**对话框，允许您手动配置数据源。Performance Explorer 监控项目设置将与 Silk Performer 项目设置一同保存。
- **无监控** - 此选项使得您在不监控任何本地或远程服务器的情况下运行测试。使用此选项时，**自动启动监控**设置将禁用（**设置 > 激活配置文件 > 回放 > 监控 > 使用默认监控模板**）。

（仅适用于默认监控和自定义监控）如果您启用了日志记录，将有一个确认对话框显示通知。日志记录可能会造成您的测试结果出现偏差。

3. 单击**确定**接受日志记录设置，或者单击**取消**调整日志记录选项（**设置 > 激活配置文件 > 结果 > 日志记录**）。
4. （仅适用于自定义监控）Performance Explorer 将启动，**数据源向导**将打开。完成向导中说明的步骤。
5. 随即将打开**工作流 - 负载配置**对话框。单击**确定**接受监控设置。

## 调整负载

配置负载是实施负载测试的流程之一。Silk Performer 提供可用作负载测试基础的各种负载模型。配置负载之前，必须选择最符合您的需求的模型。

您可以在负载测试项目中定义一个以上的负载模型，并保存这些模型以便后续使用，但每次只能激活一个负载模型。认可的基线结果与一种负载模型相关联。复制或重命名负载模型时，认可的基线结果将相应地进行复制或重命名。

## 负载模型

Silk Performer 提供以下工作负载模型：

- **递增** - 在负载测试开始时，Silk Performer 不会模拟所定义的全部用户。而是仅模拟指定的部分用户。工作负载将逐渐递增，直至用户列表中指定的所有用户均投入运行。

如果您希望确定系统在何种负载级别下崩溃，或者响应时间不可接受、出现错误阈值，这种工作负载模型将特别有用。

- **稳态** - 在这种模型中，整个测试过程中均使用相同的虚拟用户数量。所有虚拟用户均执行负载测试脚本中定义的事务。当业务完成时，虚拟用户重新开始执行事务。事务间不会出现延迟，达到指定模拟时间后，测试即完成。

在您要查明所测试系统在特定负载级别下的行为时，此工作负载模型尤为有用。


- **动态** - 在测试运行的过程中，您可以手动更改测试中的虚拟用户数量。设置最大虚拟用户数后，可以在不超过此限制的情况下在测试中随时增加或减少数量。这种情况下，未指定测试时间，您必须手动完成测试。

在您要尝试不同负载级别以及需要在负载测试过程中控制负载级别时，此工作负载模型尤为有用。

- **全天** - 此工作负载模型允许您灵活地分配负载。您可以为负载测试安排不同的虚拟用户执行任意间隔下的测试，并未这些不同类型的用户分配虚拟用户数。因此，您可以设计复杂的负载情景，例如工作日负载和周负载。在负载测试运行期间，您还可以在测试间隔中为尚未开始的测试调整负载量。

在您希望以最切合实际的方法建模复杂、长期的工作负载方案时，此工作负载模型尤为有用。

- **队列** - 在这种模型中，事务将按照规定的完成率执行。该比率是基于平均间隔的随机值，此间隔通过模拟时间和脚本 `dcluser` 部分中定义的每用户的事务数计算得出。所有虚拟用户都完成规定的任务后，负载测试即完成。

 **注：**使用此模型时，由于随机完成率方面的原因，测试所需时间可能要长于指定的模拟时间。例如，如果您指定 3,000 秒的模拟时间，希望执行 100 个事务，则可能会观察到 30 秒的平均事务完成率。

在您要模拟使用排队机制处理多个并发请求的工作负载时，此工作负载模型尤为有用。通常情况下，从 Web 服务器而非最终用户处接收请求的 Servlet 引擎或事务服务器等应用程序服务器可利用排队模型准确进行测试。

- **验证** - 在与验证扩展的功能结合使用时，验证测试特别有用。这样的组合适用于基于 Web 的应用程序的回归测试。验证测试针对特定用户类型执行指定的运行次数。

在您要自动执行 Web 应用程序的验证时，以及要从命令行界面启动验证测试时，这种工作负载尤为有用。

## 定义负载

在执行负载测试之前，必须指定要使用的负载模型并进行配置。要选择工作负载模型，请在工作流栏中单击**调整负载**。

1. 单击工作流栏上的**调整负载**。此时将显示**工作流 - 选择和调整负载**。
2. 单击恰当的选项按钮，选择以下负载模型之一：
  - 递增
  - 稳态
  - 动态
  - 全天
  - 队列
  - 验证
3. 单击**下一步**。
4. 根据帮助中列举的步骤配置特定工作负载类型。

## 运行多次负载测试

在负载测试中，通过多个虚拟用户在目标服务器上运行测试脚本，来确定负载对服务器性能的影响。大型负载测试要求您的 LAN 中具备恰当的测试环境，包括一整套管理虚拟用户的代理计算机。

您必须完成以下任务：

- 为要运行的测试设置恰当的测试类型选项
- 准确定义所需负载量
- 支持生成测试结果以评估服务器性能

不要在负载测试过程中完全启用结果记录，因为可能干扰负载测试结果。然而，在出错时，**TrueLog 出错**日志记录选项会将必要的日志文件写入到磁盘，允许您直观地检查回放错误。

## 运行一次负载测试

1. 在工作流栏上单击**运行测试**。随即将打开**工作流 - 负载配置**对话框。
2. 单击**运行**启动负载测试。
3. 在**新建结果文件子目录**对话框中单击**确定**。  
(可选) 要为结果子目录指定名称，请取消选中**自动生成唯一子目录**复选框，在**为结果文件指定子目录**文本框中输入新子目录的名称。

通过查看 Silk Performer 表格监控视图和 Performance Explorer 的图形监控视图可监控测试进度和服务器活动。

## 监控负载测试

运行 Silk Performer 负载测试时，测试人员可以得到详细的实时信息。客户端和服务器端活动的图形显示和全文本报告提供直观地监控结果，监控实时的测试运行的进度。

在执行测试的工作台中，测试人员可以查看有关测试的所涉及到的代理计算机和虚拟用户的综合概述信息。测试人员可以控制所显示信息的详细信息级别，包括测试中所用全部代理计算机的全局进度视图模式，以及

各虚拟用户执行的事务的具体信息的细节模式。各代理和各用户的进度信息显示为多个类别。各用户的运行时详细信息包括可自定义的彩色图标、计时器、函数和错误读数。

此外，还能通过图形的形式查看目标服务器的实时性能监控结果。图表将显示当前应用最广泛的所有 OS 上运行的全系列 Web 服务器、应用程序服务器和数据库服务器的最相关性能信息。可以同时打开多个图表，这些图表可以并排显示，为测试人员提供相关的比较和对照结果。树菜单编辑器支持在图表中组合来自任意数据源。来自客户端应用程序的响应时间和其他性能信息可以与来自服务器的性能数据放在相同的图表中。这项功能提供直接、直观地执行比较，让测试能够确定服务器问题对客户端行为的影响。

## 在负载测试过程中监控代理计算机

在负载测试运行过程中，使用**监控**窗口查看进度。该窗口的上半部分显示代理计算机和用户组的运行的相关信息。

该窗口提供了丰富的信息选项，您可以查看的信息包括：

- 特定代理的状态
- 在代理服务器上完成的测试的进度百分比
- 所执行事务的数量

## 在负载测试过程中监控特定代理

在**监控**窗口的上半部分中，选择要监控的特定代理。

在选定代理服务器后，以下有关在代理上运行的虚拟用户的信息将显示在**监控**窗口的下半部分：

- 状态
- 当前事务的名称
- 已完成业务的百分比
- 所执行事务的数量

## 在负载测试过程中监控特定虚拟用户

在**监控**窗口的下半部分，右键单击要监控的虚拟用户，选择**显示虚拟用户的输出**。

在**虚拟用户**窗口中，Silk Performer 将显示有关选定用户的详细运行时信息，例如用户执行的事务和函数，以及用户发送到服务器、从服务器接收的数据。



**提示:** 右键单击虚拟用户区域，选择**选择列**以选择要查看的列。

## 使用图形监控服务器性能

1. 单击 workflow 栏上的**确认基线**。此时将打开 workflow - **确认基线**对话框。
2. 单击**定义监控选项**指定联机接收性能数据的设置。此时将打开**配置文件结果**对话框。
3. 在**配置文件结果**对话框中，选中**自动启动监控**复选框，以便在运行负载测试时自动开始监控，随后选择以下选项之一：
  - 单击**使用默认监控模板**选项按钮。
  - 单击**使用自定义监控模板**选项按钮，创建自定义监控模板。
4. 单击**创建/编辑自定义监控模板**。随即将出现 Performance Explorer。
5. 关闭目前未使用的所有监控窗口。
6. 单击 Performance Explorer workflow 栏上的**监控服务器**。  
此外，也可以从菜单栏中选择**结果 > 监控服务器**。  
此时将打开**数据源向导/选择数据源**对话框。此时将打开对话框
7. 执行以下步骤之一：

- 如果您确定服务器提供的数据库源，请单击**从预定义的数据源中选择**选项按钮，随后从预定义的数据源列表中进行选择。
  - 如果您不确定服务器提供的数据库源，单击**检测数据库源**选项按钮，由 Performance Explorer 扫描服务器中的可用数据库源。
8. 单击**下一步**。  
在菜单树中，展开对应于服务器和应用程序运行的 OS 的文件夹。
  9. 从所显示的列表中选择要监控的服务器应用程序。  
例如，要监控 OS，请选择**系统**。
  10. 单击**下一步**。此时将打开**连接参数**对话框。
  11. 在**连接参数**区域中，指定连接参数，例如服务器系统的主机名或 IP 地址、端口号，以及连接到数据库源所需的其他数据。  
指定的数据依赖于在您监控的计算机上运行的 OS。
  12. 单击**下一步**。此时将打开**选择显示的度量**对话框。
  13. 展开菜单树，选择要监控的指标。
  14. 单击**完成**。监控器图形将以彩色图标的方式实时地显示服务器性能的指定指标。图形下方显示的是所含元素列表、彩色图标键和各元素相关的性能信息。

## 浏览测试结果

Silk Performer 提供多种显示、报告和分析测试结果的方法。所定义的指标将在测试过程中产生，可通过多种图形和表格的形式显示。选项包括：

- **Performance Explorer**：这是用于查看测试结果的主要工具。综合全面的图形化功能将在用户定义的图形中显示结果以及所需的多种元素。不同测试的结果是可以比较的。其中包含丰富的服务器监控功能。此外，还可以使用基于 HTML 的综合概述报告，其中整合了用户类型统计信息与测试运行时间信息。
- **TrueLog 出错**：Silk Performer 为各种类型的应用程序提供完全可视化的负载验证功能。在负载测试过程中，它允许您将多内容验证检查与完整的深层错误分析相结合使用。
- **虚拟用户报告文件**：启用时，这些文件将包含对应于各用户的模拟结果。各用户的详细指标信息将以表格的形式显示。
- **虚拟用户输出文件**：启用时，这些文件将包含测试脚本中所用写入语句的输出结果。
- **基线报告**：基于 XML/XSL 的详细报告，显示摘要表、事务响应时间详细信息、所访问的全部 HTML 页面的计时器、Web 表格和错误。此信息对参与基线测试的所有用户类型均可用。
- **Silk Central 报告**：Silk Performer 项目可集成到 Silk Central (Silk Central) 的测试计划中，直接通过 Silk Central 执行。这可实现强大的测试结果分析和报告。有关 Silk Central 报告的详细信息，请参阅 *Silk Central 帮助*。

## TrueLog On Error

利用 Silk Performer TrueLog 技术，您可以发现当应用程序处于大负载下错误通常发生在某个用户子集中。大多数应用程序在部署到真实环境后，都会经历类似的负载情况。典型错误包括 Web 页面上错误的文本内容、错误的计算值和显示值，或者应用程序的相关错误消息，例如 Servlet 错误或者服务器过于繁忙的错误。这些不属于系统级错误，并且以 HTTP 200 状态代码显示在 Web 页面上。


TrueLog Explorer 提供 Silk Performer 在负载下验证的视图功能，其中包含以下功能：

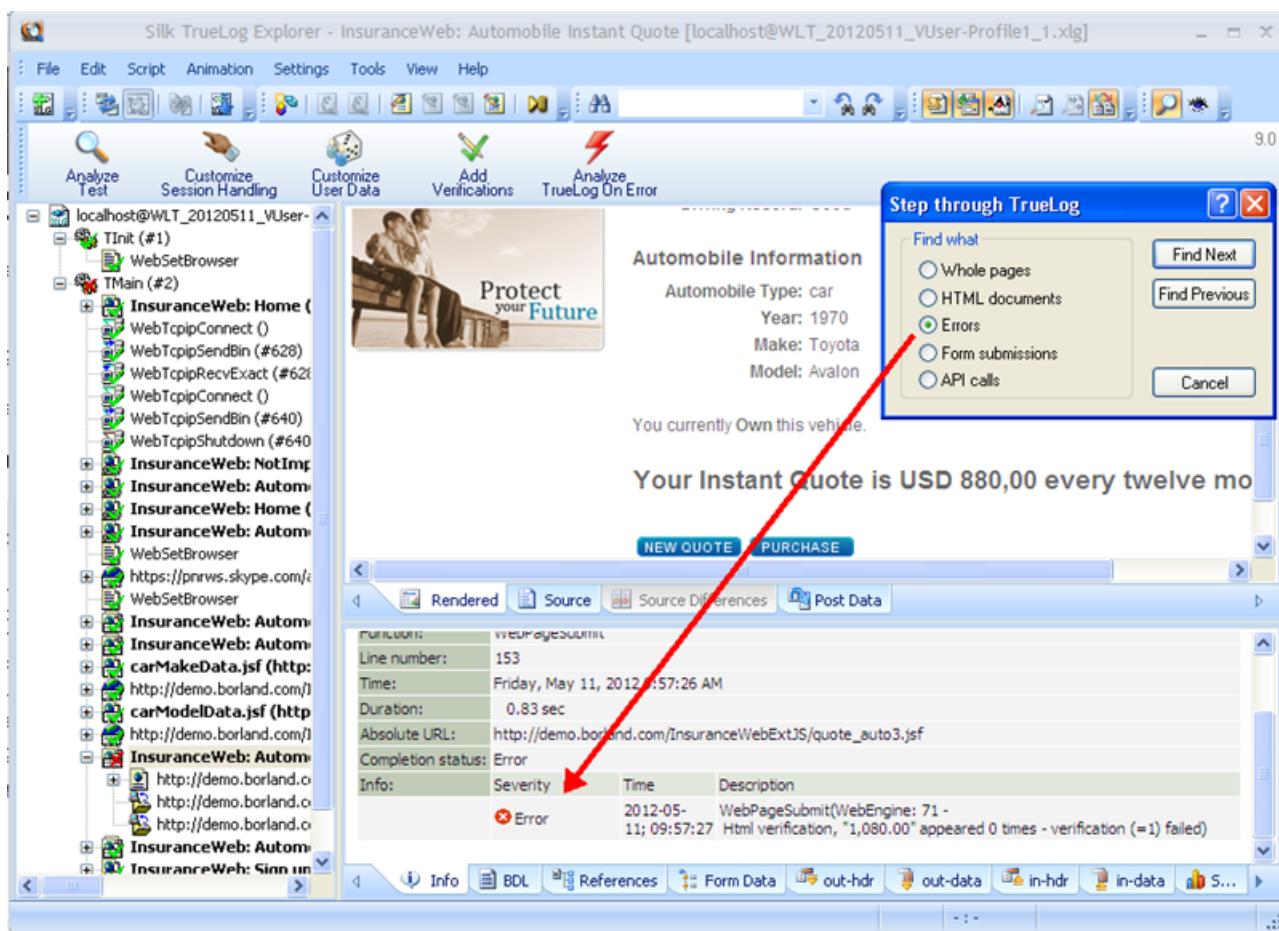
- 可视化内容验证允许您直观地定义待验证的内容。
- TrueLog 错误生成和 TrueLog 错误分析允许您直观地分析错误，以确定其根本原因。

## 在 TrueLog Explorer 中查看错误

1. 在负载测试过程中发生错误时，您可以单击 workflow 栏上的**浏览结果**按钮，查看 TrueLog 的可视化内容。此时将打开 **workflow - 浏览结果**对话框。

- 单击 Silk TrueLog Explorer 链接。TrueLog Explorer 将打开,逐步查看 TrueLog 对话框处于激活状态。选择错误。
- 从一处错误导航到下一处错误。

 **提示:** 要显示错误历史记录, 只需在菜单树中单击浏览前面的 API 节点。



## Performance Explorer 概述

Performance Explorer 提供了两种主要功能：

- 结果分析
- 实时监控

### 结果分析

借助 Performance Explorer 的结果分析功能, 您可通过创建图表、表格和报告来分析已完成的负载测试的结果。您可使用图表显示负载测试过程中收集的数据, 而报告和表格可帮助您汇总重要数据和发现。所有图表、表格和报告的基础是 .tsd 文件中的数据。在每次负载测试过程中, Silk Performer 会捕获大量数据并将其存储在多个时间序列数据 (.tsd) 文件中。完成负载测试后, 您可在 Performance Explorer 中加载数据 (.tsd 文件), 并根据需要显示和编辑这些数据。所有图表均可完全自定义, 并可根据需要包含任意数量的测量工具。您可使用一个或多个测试中的测量工具打开多个图表, 以显示异同点。Performance Explorer 提供了各种图表、表格和报告模板。此外, 您可以将客户端和服务器性能的信息置于同一个图表内, 以便直观地查看服务器性能对客户端行为所产生的影响。将更改保存为模板后, 您就可以重用您的各个设置。

### 实时监控

借助 Performance Explorer 的实时监控功能, 您可创建和配置实时显示系统性能的图表, 以监视广泛的系统。您可以同时打开多个图表以同时查看两个或更多系统的性能, 例如, Web 服务器性能和操作系统性能。可以



在 Performance Explorer 中轻松直观地向图表中添加测量工具：您可将单个测量工具或一组测量工具从树中拖至图表上。

## 概述报告指标

概述报告包含以下部分：

- 总体说明
- 摘要表
- 用户类型
- 自定义图表
- 自定义表
- 详细信息图表

### 总体说明

总体说明包含表格格式的管理信息，以及图形格式的重要负载测试结果。

管理信息包括项目名称、项目说明、负载测试编号、负载测试说明、负载测试日期、负载测试的持续时间、所用代理计算机的数量以及所运行的虚拟用户数量。

这些图表显示了虚拟用户数、事务的响应时间以及错误数。事物的响应时间由执行成功的事务、失败的事务和取消的事务来提供。

根据负载测试项目的不同，一些附加的图表会显示相关的指标。例如，在 Web 应用程序测试中，网页的响应时间指标图表会显示出来。

### 摘要表

本部分包含以表格格式展现的指标的摘要信息，即所有虚拟用户的指标集合。第一个表提供了总体说明，例如执行的事务数和错误数。后续的表提供了与所测试应用程序的类型相关的摘要信息。

### 用户类型

对于每个用户组，本部分以表格格式提供了详细测量。这些指标包括与所测试的应用程序类型（Web、数据库、CORBA 或 TUXEDO）相关的事务响应时间、独立的计时器、计数器以及响应时间和吞吐量等指标。此外还会列出针对所有用户组的错误和警告。

### 自定义图表

本部分包含手动添加的图表。您可以随时在此部分中添加和移除图表。您可以将更改保存为模板，以便为其他摘要报告所用。

### 自定义表

本部分包含手动添加的表。您可以随时在此部分中添加和移除表。您可以将更改保存为模板，以便为其他摘要报告所用。

### 详细信息图表

本部分提供报告中所含图表的放大功能。单击一个图表的缩小版本即可跳转到放大版本，反之亦然。

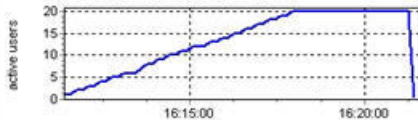
Overview Report  
Click [here](#) to edit text.

general information    summary tables    ranking    user types    custom charts    custom tables    detailed charts

Project: 05142012    Default Project  
Load test: 4    None  
Start date/time: 5/14/2012 4:11:15 PM  
Duration of simulation: 00:10:20  
Agents: 1  
Users: 20  
Report Description:

#### General Project Settings

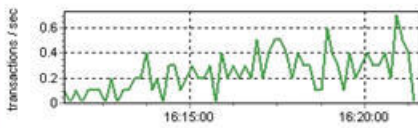
Application type: Web business transaction (HTML/HTTP)  
Workload: Workload1  
Workload model: Increasing



#### Active Users

This chart shows the overall number of active virtual users. A virtual user is considered as active if the user has started and is currently in one of the following states: executing, wait DB, document downloading, and thinktime.  
Click [here](#) to edit text.

number of concurrent users: 20



#### Transactions

The number of SilkPerformer transactions per second.  
Click [here](#) to edit text.

number of transactions: 149  
average number of transactions/sec: 0.24

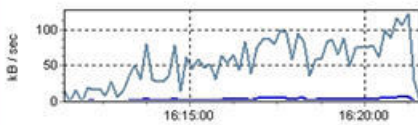


#### Errors

This chart shows the number of API errors per second, including Internet, database, and middleware APIs. A problem is considered an error if its severity is defined as "error" or worse, that is, of higher severity ("transaction exit" or "process exit"). A problem is ignored if its severity is defined as "informational" or "warning".  
Click [here](#) to edit text.

number of errors: 0

#### Throughput / Concurrency



#### Throughput[kB]

The amount of data sent to and received from the server; this includes header and body content information, all TCP/IP-related traffic (HTTP, native TCP/IP, IIOP, POP3, SMTP, FTP, LDAP and WAP), and secure traffic over SSL/TLS. This measurement does not include data overhead caused by SSL/TLS encryption and WTLS encryption in case of WAP.  
Request data sent  
Response data received  
Click [here](#) to edit text.

throughput[kB]: 36,139  
average throughput[kB]/sec: 58.29

## 查看概述报告

- 单击 workflow 栏上的 **浏览结果** 按钮。此时将打开 **workflow - 浏览结果** 对话框。
- 单击 **Silk Performance Explorer** 按钮或链接。如果在 **设置/选项/报告** 对话框中选择了 **自动生成概述报告**，Performance Explorer 将打开，并显示最新负载测试的概述摘要报告。此外，您还可以在此对话框中选择之前保存的模板生成概述报告。此设置为全局设置，无论使用哪种工作负载项目，均可配合 Performance Explorer 一起使用。
- 如果概述报告未自动显示，请单击 workflow 栏中的 **概述报告** 按钮。
  - 导航到正在处理的负载测试的目录，选择相应的 .tsd 文件，然后单击 **打开**。
  - 单击 **下一步**。
  - 可以选择在 **模板** 字段中单击 **[...]**，导航到要使用的模板文件。
  - 单击 **完成**。
  - 根据您选择的负载测试结果，系统可能会提示您确认是否要将所有相关文件加载到项目中。随即概述报告将打开。

# 索引

## A

### AJAX

Web 2.0 应用程序示例 4

## B

### 变量

输入属性 10

标题验证 13

## C

Cookie 17

CORBA 25

### 参数

随机变量 10

### 测试脚本

Web 测试 7

利用 TrueLog Explorer 执行可视化分析 7

### 测试结果

Web 测试 23

## D

### 代理

在 Web 测试过程中执行监控 22

### 弹出窗口支持

Web 2.0 应用程序示例 4

递增工作负载模型 20

### 定义工作负载

Web 测试 21

动态工作负载模型 20

## G

### 概述报告

Web 测试 25, 26

### 概述页面

Web 测试 9

### 工作负载

Web 测试 20

### 工作负载模型

Web 测试 20

## H

### 回放

浏览器模拟设置 19

### 汇总报告

Web 测试 8

### 会话处理

会话信息 15

使用时机 16

自定义 18

## J

### JSON

良好格式 12

认识 11

### 监控

服务器 19

配置 19

### 监控测试

Web 测试 21

### 解析函数

HTML 上下文解析 15

概述 15

响应数据解析 15

## L

### 连接

并发 18

### 浏览器

Web 测试 18

模拟 19

设置 18

### 浏览器驱动的 Web 测试

Web 2.0 应用程序示例 4

### 录制测试脚本

定义项目 6

## M

### 面向 Web 测试的脚本

定义项目 6

### 模拟

浏览器 19

## P

### Performance Explorer

Web 测试 24

排队工作负载模型 20

### 配置文件

添加 18

## Q

全天工作负载模型 20

## S

### 设置

浏览器 18

数据源向导 22

## T

### TrueLog

比较 Web 录制并回放 9

查找错误 8

分析测试运行 8

### TrueLog Explorer

XML 12

TrueLog On Error

Web 测试 23

Tuxedo 25

添加

配置文件 18

同步录制和回放 TrueLog 9

图形

Web 测试 22

## W

Web 2.0 测试

基于 AJAX 的应用程序示例 4

Web 测试

Performance Explorer 24

TrueLog On Error 23

比较录制并回放 TrueLog 9

测试结果 23

调整负载 20

定义工作负载 21

分析测试脚本 7

分析测试运行 8

概述报告 25, 26

概述页面 9

工作负载模型 20

汇总报告 8

监控测试 21

监控所有代理计算机 22

监控特定代理 22

监控虚拟用户 22

利用 TrueLog Explorer 执行可视化分析 7

浏览器类型 18

录制测试脚本 6

面向 Web 测试的脚本 6

内容验证 14

通过图形监控性能 22

同步录制和回放 TrueLog 9

为 Web 测试 5, 6

虚拟用户摘要报告 8

页面统计信息 8

用户配置文件 18

运行测试 21

在 TrueLog 中查找错误 8

自定义测试脚本 10

自定义用户数据 10

为 Web 测试

定义项目 5, 6

稳态工作负载模型 20

## X

XML

良好格式 12

响应数据解析 15

## Y

验证工作负载模型 21

验证检查

标题验证 13

概述 12

在回放过程中启用 13

在录制过程中自动生成 12

摘要验证 13

验证内容

Web 测试 14

页面统计信息

Web 测试 8

用户配置文件

Web 测试 18

用户数据

自定义 10

运行测试

Web 测试 21

## Z

摘要验证 13

自定义

用户输入数据 10

用户数据 10

自定义测试脚本

Web 测试 10