How-To Design and Execute LoadRunner Scenarios for SAP BusinessObjects Planning and Consolidation (BPC)

Applicable Releases:

- SAP BusinessObjects Planning and Consolidation 7.0, version for the SAP NetWeaver Platform
- SAP BusinessObjects Planning and Consolidation 7.0, version for the Microsoft Platform
- SAP BusinessObjects Planning and Consolidation 5.1, version for the Microsoft Platform

IT Practice / Topic Area:
Performance Testing
Scalability Testing

Version 0.99d
July 2009
## Document History

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.99c</td>
<td>Unofficial review release of this guide</td>
</tr>
</tbody>
</table>
### Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Example Text</em></td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td><em>Example text</em></td>
<td>User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td><code>&lt;Example text&gt;</code></td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td><strong>EXAMPLE TEXT</strong></td>
<td>Keys on the keyboard, for example, <strong>F2</strong> or <strong>ENTER</strong>.</td>
</tr>
</tbody>
</table>

### Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="caution.png" alt="Caution" /></td>
<td>Caution</td>
</tr>
<tr>
<td><img src="note.png" alt="Note or Important" /></td>
<td>Note or Important</td>
</tr>
<tr>
<td><img src="example.png" alt="Example" /></td>
<td>Example</td>
</tr>
<tr>
<td><img src="tip.png" alt="Recommendation or Tip" /></td>
<td>Recommendation or Tip</td>
</tr>
</tbody>
</table>
# Table of Contents

1. **Business Scenario** ........................................................................................................1

2. **Background Information** .............................................................................................1

3. **Prerequisites** ................................................................................................................1

4. **Information Gathering** ....................................................................................................2
   4.1 Required Skill Sets....................................................................................................2
   4.2 Scenario Planning.....................................................................................................2
      4.2.1 What is a Scenario........................................................................................3
      4.2.2 Overview of Scenario Planning ......................................................................3
   4.3 Script Planning..........................................................................................................4
      4.3.1 SAP BusinessObjects Planning and Consolidation Architecture Review ....4
      4.3.2 Overview of Script Planning ..........................................................................5
      4.3.3 Comparison of LoadRunner Protocol Types ..................................................5

5. **Script Design** .................................................................................................................7
   5.1 Microsoft Excel version Considerations .....................................................................7
   5.2 Options for Recording LoadRunner Scripts for BPC ..................................................8
   5.3 Recording a HP LoadRunner Script for BPC with VuGen ........................................8
      5.3.1 HP LoadRunner Recording Settings..............................................................9
   5.4 Recording a HP LoadRunner Script for BPC with Fiddler .........................................16
      5.4.1 How to transfer a Fiddler request into a VuGen Web (HTTP/HTML) or SAP – Web script. ..............................................................19
      5.4.2 How-to transfer a Fiddler request into a VuGen Web Services script ..............25
   5.5 Additional requirements for LoadRunner VuGen scripts...........................................30
      5.5.1 BPC User Authentication in VuGen .............................................................30
      5.5.2 HTTP Timeout Configuration........................................................................30
   5.6 Parameterization........................................................................................................31
      5.6.1 EvDRE Parameterization Example ..............................................................31
   5.7 Transaction Verification..............................................................................................36
      5.7.1 Example EvDRE Report Verification code for the Web (HTML/HTTP) and SAP – WEB protocol ..............................................................36
      5.7.2 Example EvDRE Report Verification code for the Web Services protocol ....37
   5.8 Notes for LoadRunner Support in Planning and Consolidations 7.0 for SAP NetWeaver.................................................................37
   5.9 Notes for LoadRunner Support in Planning and Consolidations 7, for Microsoft ......38
   5.10 Notes for LoadRunner Support in Planning and Consolidations 5.1, for Microsoft ....38
   5.11 Unit Testing Scripts with VuGen .............................................................................38

6. **Scenario Design, Execution and Analysis** ....................................................................40
6.1 Scenario Design Considerations ............................................................................. 40
6.2 Designing the Scenario Schedule ........................................................................... 40
6.3 Configuring Systems Monitoring ........................................................................... 45
6.4 Executing and Monitoring Test Scenario .............................................................. 48
   6.4.1 Starting a scenario...................................................................................... 48
   6.4.2 Monitoring a scenario................................................................................ 48
   6.4.3 Monitoring Data Volumes ......................................................................... 49
6.5 Analyzing Results ................................................................................................. 49
   6.5.1 Load Runner Analysis Overview ............................................................... 50
   6.5.2 Graphs ...................................................................................................... 51
   6.5.3 Interpreting Analysis Results ..................................................................... 56
6.6 Working through Performance Issues ................................................................... 58
   6.6.1 Hardware Bottleneck ................................................................................ 59
   6.6.2 Systems Tuning ......................................................................................... 59
   6.6.3 Application Design .................................................................................... 59

7. Appendix .................................................................................................................. 60
   7.1 Appendix A – Scenario Planning Template ....................................................... 61
   7.2 Appendix B – Script Planning Template ............................................................. 63
   7.3 Appendix C – Scenario/Script Schedule Planning Template ............................... 65
1. Business Scenario

In the course of promoting a new solution to production, many clients verify their landscapes using HP Load Runner for stress testing, volume testing, and/or performance testing.

SAP recommends performing stress/volume testing as a best practice within a normal solution implementation project. This can be essential for establishing a comfort level around expected performance, and to validate that sizing estimates are accurate. Unfortunately, it can be difficult to design real life scenarios.

This guide will provide the information necessary to design HP LoadRunner scripts and scenarios that represent actual user scenarios.

2. Background Information

This How-To provides guidance and suggestions for using HP LoadRunner with SAP BusinessObjects Planning and Consolidation. It also provides advice on what questions to ask your user community and functional experts to ensure you have the necessary details to create representative scripts and scenarios.

3. Prerequisites

- HP Load Runner licensed for one of the following protocols:
  - Web (HTTP/HTML)
  - SAP + Web
  - Web Services
    - Important
      - Not all requests are compatible with the Web Services protocol. Refer to section 5.3.2.1 for more information.

- SAP BusinessObjects Planning and Consolidation, version for SAP NetWeaver
  - or

- SAP BusinessObjects Planning and Consolidation, version for SAP Microsoft

- Experience with HP LoadRunner
4. Information Gathering

The initial step in preparing for stress/volume testing of BPC is gathering information on the scenario(s) you will test, which scripts will be included and finally, what the goal or success criteria will be.

This section provides an overview of what types of information to gather before you start developing scripts and scenarios.

4.1 Required Skill Sets

Load and performance testing projects require collaboration between individuals with a wide range of skill sets. You will find a list outlining these skill sets and their expected contributions to the project below:

- Application Consultant(s)
  - Assists in identifying possible pain points
  - Assists in resolving business process and logic related performance problems
- Load Runner Consultant(s)
  - Documents, Creates and Executes Load Runner Scripts and Scenarios to be executed
  - Analyzes results and engage the appropriate teams for issue remediation
- Business Process Expert(s)
  - Creates and reviews business processes to be tested
  - Provides input on parameterization in scripts
  - Provides details on Scenario design
  - Works with Application Consultants to document, test and approve business process or logic changes
  - Provides details on SLA’s
- BPC Technical Resources
  - Performs any necessary hardware upgrades or system tuning based on test results.
- BPC Implementation Project Management
  - Ensures the appropriate resources get involved at the various stages of the Load Testing engagement
- Customer Implementation Lead
  - Approves testing methodology/design, SLA’s and any changes resulting from the testing
  - Ensures the appropriate resources get involved at the various stages of the Load Testing engagement

4.2 Scenario Planning
4.2.1 What is a Scenario

The term “Scenario” refers to a hypothetical situation you create to mimic a business process or a set of business processes. This scenario will be replayed against a Business Planning and Consolidation landscape to gain additional understanding of its impacts.

A scenario includes a series of LoadRunner scripts. Each script contains a series of operations relevant to the business process that you are testing.

Note
The next section will review LoadRunner scripts in more detail.

The scenario also includes the details of how the scripts should be executed, for example:

- How many virtual users (Vusers) are assigned to each script?
- How should user behavior be emulated for each script in the scenario, i.e.
  - When do they login?
  - How quickly do they login?
  - How long do they execute transactions?
  - When do they logout?
  - How quickly do they logout?

Lastly, you are given the option to add systems to monitor during the execution of the scenario. This is very important as it allows you to determine resource bottlenecks and correlate other metrics, such as response times, to physical system resources.

HP LoadRunner scenarios are maintained with the HP LoadRunner Controller software.

4.2.2 Overview of Scenario Planning

Creating the scenario in the HP LoadRunner Controller software is actually one of the last steps of the process, but it is the first thing that needs to be planned.

When designing a scenario, the first question is “What goals do we hope to achieve through testing?” This is a key question, because without defining it, it will be very difficult to gauge the success of your testing. Some potential answers to that question include:

- Performance testing of a particular business cycle, such as a quarter close
- Stress testing to validate hardware sizing with respect to the business processes that will be supported.
- Performance testing of logic or landscape design where you design a baseline test, make a modification to logic files or your system landscape and re-test the original scenario to verify the impact of your changes.

Next, you should outline which user interactions or scripts need to developed to adequately represent the business process(es) that you are testing. This can be fairly high level at this point, but should include details such as the script owner as well as a descriptive name.

Note
The script owner referenced above refers to a SAP BusinessObjects Planning and Consolidation user that is an expert on the business processes being represented within the script. The script owner need not have HP LoadRunner skills, but will provide more specific information on what the script should include. The next step would be for
someone with HP LoadRunner know-how to take these specifications and create a script to represent them.

Lastly, you should determine what servers in your landscape will be active in your scenario so that they can be monitored during scenario execution.

**Important**

Appendix A includes a template that will assist you in creating a high level plan for your Scenario(s).

### 4.3 Script Planning

**4.3.1 SAP BusinessObjects Planning and Consolidation Architecture Review**

HP LoadRunner records scripts for SAP BusinessObjects Planning and Consolidation at the application layer. More specifically, the HTTP requests from the client tier to the .NET tier are recorded. This section was added to help explain what you are actually capturing with the HP LoadRunner tool.

BPC communicates between the Client and .NET server tiers by sending SOAP requests over HTTP/HTTPS. Most operations, like submitting an input schedule or retrieving a report consist of a single web service request and response.

The SOAP request outlines the information you are requesting from the server. The response details the results or status of your request. Since different users will request or submit distinct information for most reports and input schedules, we must vary some requests to reflect this variation. This is done through parameterization.

The responses will vary depending on the type of call being made. For example, SOAP responses for report requests will generally contain the values returned for your query while SOAP responses to an input schedule will detail how many records were accepted, rejected, etc. Using the information in the SOAP responses we can write functions to validate that the requests were successful. This type of validation should be included in order to create reliable scripts.

**Important**

LoadRunner does not launch Excel, or any other SAP BusinessObjects Planning and Consolidation client during the execution of a script or scenario. Instead, it replays the communications, i.e. SOAP requests, which the client submits to the .NET tier. This accurately reproduces system load as well as server request response times.
4.3.2 Overview of Script Planning

HP LoadRunner scripts are created with a program called “HP LoadRunner Virtual User Generator”.

**Note**
This guide will refer to this program as VuGen for the rest of this document.

These scripts use an underlying protocol to capture what a script does, and later uses the same protocol to replay the script. The protocols that can be used with BPC are the “SAP + Web”, “Web (HTTP/HTML)” and “Web Services” protocols.

**Important**
Ensure that your LoadRunner Controller license includes one of these three protocols.

Before creating the scripts, it is important to plan what their purpose will be and what operations they include. An operation, in this case, refers to a unit of work. An example of an operation could be submitting an input schedule, or running a report. Each operation should be encapsulated in a LoadRunner transaction or sub-transaction after a script is recorded. This will allow LoadRunner to effectively report the time each operations takes.

Another consideration when planning scripts is parameterization. It may be helpful to further expand upon this topic.

During a business process multiple users perform similar operations, like submitting data during a planning process. In order for multiple users to submit data successfully they must submit data for different dimension members. Each user may enter data for a different entity, or account or combination of the two.

The detailed data that is specific for each user (entities and accounts from the example above) must be represented in the script through parameterization. Parameterization allows you to configure values that will change for each iteration or user in a script which allows you to more accurately reflect actual user interaction with BPC. Parameterization is configured for each SOAP call where appropriate.

Another key aspect that must be considered is verification. Each request that is sent to the .NET application server is a SOAP message.

Once your scenario and scripts have been planned you can start recording and refining your scripts.

**Note**
Appendix B includes a template that will assist you in creating a high level plan for your Scripts.

4.3.3 Comparison of LoadRunner Protocol Types

Three LoadRunner protocols have been tested with SAP BusinessObjects Planning and Consolidations. Two of them, the Web (HTTP/HTML) and SAP – Web, are effectively the same – the only difference being the license type. The third, Web Services, works in a slightly different way than the others.

With all three protocols the actual SOAP messages produce the same effects on the backend systems; however, the Web (HTTP/HTML) and SAP – Web protocols are recommended for the following reasons:

a. The Web (HTTP/HTML) and SAP – Web protocols include a richer set of API's for parsing responses (the web_* functions) which is useful when adding custom verification code.
b. The Web Service protocol only works with actual Web Service calls, but the BPC client also makes straight Web calls for some operations. Some examples of this are during login, or when viewing Audit reports. Because of this, it is impossible to write a full coverage LoadRunner script only using the Web Services protocol.

c. Again, because of these reasons, the Web (HTTP/HTML) and SAP – Web protocols are recommended if either of those LoadRunner protocol licenses are available.
5. Script Design

When you have reached the point where you have completed the design phase, you are ready to begin recording LR scripts. There are two primary methods for recording scripts.

The first is LoadRunner VuGen. This tool is supplied as part of the LoadRunner suite of products. It will work for recording reports and input schedules, but will not work with some Admin tasks or Data Manager Packages. The reason some actions can’t be recorded with VuGen is related to the way that BPC executes those operations. For example, Data Manager Packages are executed by spawning a child process (OSoftDMClientExecute.exe), and VuGen does not recognize that it should record the actions of the child process. Refer to the section on recording BPC scripts with Fiddler for details on capturing these operations.

⚠️ CAUTION

Some customers have reported issues getting LoadRunner to properly record BPC scripts. The issues have generally presented themselves during login or after login whereas the action pane has failed to load.

An example root cause for this is an incompatibility with the McAfee Host Intrusion Prevention.

If you encounter these issues and are unable to determine the root cause, try recording scripts on a machine that has as few restrictions as possible. If that does not help or is not possible you can still record scripts using a tool such as Fiddler.

⚠️ CAUTION

Limited LoadRunner Support for BPC 5.1 was recently added in a support package and requires a parameter change in the Application Sets database. For more information review SAP Note 1300873 for current Support Pack requirements and configuration details (https://service.sap.com/sap/support/notes/1300873).

The second recording method involves using a third party tool called Fiddler. This tool is not part of the LoadRunner suite of tools. Instead, it is a generic HTTP proxy application that allows you to capture the requests and responses of any HTTP traffic. Because of this, it is a much less tightly coupled solution and does not suffer from the same issues as VuGen. This added flexibility comes with the cost of more manual steps, as you have to manually transpose the requests from Fiddler into VuGen scripts.

⚠️ Important

The How-To Guide entitled “How-To Use the SAP BusinessObjects Planning and Consolidation Toolkit for LoadRunner” contains a utility that performs all of the manual steps required to transform a Fiddler log into a LoadRunner script. The utility is the recommended recording method for creating LoadRunner scripts for SAP BusinessObjects Planning and Consolidation.

5.1 Microsoft Excel version Considerations

BusinessObjects Planning and Consolidations support multiple versions of Excel, though Excel 2003 and Excel 2007 are most common.

Excel 2003 works well with LoadRunner VuGen versions 8.x and higher.

Excel 2007 does not work with all versions of LoadRunner VuGen. The most common issues are:

- Excel 2007 crashes when the recording begins
- VuGen does not capture any actions during the recording
How To... Plan and Design LoadRunner Scenarios for SAP BusinessObjects Planning and Consolidation

a. If you are using Excel 2007, you have a few options to work around this limitation:
   - Upgrade to the latest LoadRunner release
   - Use Fiddler for the recording process
   - Use Excel 2003 to record the scripts, if available

**Note**
Using Excel 2003 is a valid option because LoadRunner records and replays actions at the network layer. At this layer, both Excel 2007 and Excel 2003 perform the same operations.

### 5.2 Options for Recording LoadRunner Scripts for BPC

There are three options for recording LoadRunner scripts:

- **HP LoadRunner VuGen**
  - This option will work under certain conditions, but has many incompatibilities. Because of this, the recommended option is using the BPC Toolkit for LoadRunner.
  - This is described in detail in section 5.3.

- **Fiddler recordings converted to LoadRunner syntax with the BPC Toolkit for LoadRunner**
  - This option is the recommended option. With it you use a 3rd party tool called Fiddler to record the actual BPC communications between the client and .NET tier. The BPC Toolkit for LoadRunner converts the text files created by Fiddler into LoadRunner syntax.
  - This is described in detail in the guide “How-To Use the SAP BusinessObjects Planning and Consolidation Toolkit for LoadRunner”.

- **Fiddler recordings converted to LoadRunner syntax manually**
  - This option is no longer a recommended option due to the creation of the BPC Toolkit for LoadRunner, but has remained in this guide to provide additional details on the process and can be helpful for debugging.
  - This is described in detail in section 5.4.

### 5.3 Recording a HP LoadRunner Script for BPC with VuGen

This section describes the process of recording a script for BPC using VuGen.

**Important**
Limited LoadRunner support for BPC 5.1 was recently added via support package. For more information review SAP Note 1300873 for current Support Pack requirements and configuration details ([https://service.sap.com/sap/support/notes/1300873](https://service.sap.com/sap/support/notes/1300873)).
Important

BPC 7.0 for SAP NetWeaver does not officially support LoadRunner, although it is possible to record and playback scripts with limitations. Official support for LoadRunner for the BPC for NetWeaver product line is targeted for the 7.5 release.

5.3.1 HP LoadRunner Recording Settings

This section will outline the recording settings required for VuGen.

Note

This section displays settings that are known to work. Other settings/options can be changed, but may affect the ability to record BPC scripts.

1. Launch HP LoadRunner Virtual User Generator.
2. Open a new script (File -> New)
3. Select either Web (HTTP/HTML) or SAP – Web as the Protocol and click “OK”.
4. 
5. The “Application Type” can be either “Internet Applications” or “Win32 Applications”, although you must specify either Excel.exe or the BPC Admin Client (OSoftAdminMain.exe) as the “Program to Record”. Click “Options”.

[Image of New single protocol script window]
6. Verify the following settings in the “Recording Options” dialog.

   o General -> Script Options

   o General -> Protocol Options
General -> Recording Options
How To... Plan and Design LoadRunner Scenarios for SAP BusinessObjects Planning and Consolidation

- General -> Recording -> HTML-Advanced

![Advanced HTML dialog box]

Script type:
- A script describing user actions (e.g. web_link, web_submit_form)
- A script containing explicit URLs only (e.g. web_url, web_submit_data)

Non HTML-generated elements (e.g. JavaScript, VBScript, ActiveX, Applets)
- Record within the current script step
- Record in separate steps and use concurrent groups
- Do not record

Hint:
Move the mouse over any item to see its description.

- Network -> Port Mapping Options

![Network Port Mapping dialog box]

Recording Options:
- General
- Script
- Protocols
- Recording
- Network
  - Port Mapping
    - HTTP Properties
      - Advanced
      - Correlation

Network Port Mapping:
- Servers
- Port
- Service
- SSL

Description:
A brief description of the selected option

- HTTP Properties -> Advanced

![HTTP Properties dialog box]
HTTP Properties -> Advanced -> Headers

- Click the “OK” button in the Headers and Recording Options dialog boxes.

8. Click “OK” to start the recording.
5.3.2 Recording a LoadRunner Script using VuGEN

1. Excel will be launched after clicking “OK” in the “Start Recording” dialog.
2. Create a new transaction named “Login”

3. Note

   It is a best practice to create a transaction for every action you record (Report, Input Schedule, etc).

4. Click the “Log On” button in Excel.

5. [Image of Excel interface]
6. Enter the appropriate information in the Login dialog and click “OK”.

7. Once the Login process is complete, end the “Login” transaction.

8. Create a new LoadRunner transaction for the event you are about to record, and then execute the process. For example, a process could be comprised of executing a report or submitting an input schedule.

9. When the process is complete, end the transaction.
11. Repeat steps 6 and 7 for each report/input schedule that you want to include in this script.

13. End the recording.

14. **Recording... (167 events).**

15. VuGen will compile and display the script to you.

### 5.4 Recording a HP LoadRunner Script for BPC with Fiddler

Fiddler is a 3rd party tool that allows you to capture BPC’s SOAP requests. In order to use it, Fiddler must be downloaded and installed. Fiddler can be downloaded from [http://www.fiddlertool.com/fiddler/](http://www.fiddlertool.com/fiddler/) and should be installed on the system you will use to create scripts.

**Important**

The How-To Guide entitled “How-To Use the SAP BusinessObjects Planning and Consolidation Toolkit for LoadRunner” contains a utility that performs all of the manual steps required to transform a Fiddler log into a LoadRunner script. The utility is the recommended recording method for creating LoadRunner scripts for SAP BusinessObjects Planning and Consolidation.

**Tip**

See the “I get certificate errors or .NET security exceptions when debugging with Fiddler2.” section in the following document for information on recording BPC over SSL with Fiddler. ([http://www.fiddlertool.com/Fiddler/help/knowntissues.asp](http://www.fiddlertool.com/Fiddler/help/knowntissues.asp))

1. Close all applications that are generating HTTP traffic. This limits the amount of extraneous traffic that is captured.

2. Launch fiddler.
3. 

4. Once fiddler has completed initializing, launch the BPC client you want to record.

5. Login to BPC and navigate to the report, input schedule or other process you wish to execute.

6. At this point Fiddler will have captured a number of SOAP requests. If you wish to include the navigation steps in your LoadRunner script, take note of the last Request #. Otherwise, select delete all existing requests.

7. 

8. After you complete executing the process, stop capturing traffic in Fiddler by selecting “File -> Capture Traffic”.

July 2009
10. Delete any HTTP requests that do not have the status code 200 from Fiddler.

11. The remaining requests make up all of the HTTP calls that were made in the process you recorded. These requests will have to be added to a LoadRunner VuGen script manually in order to be replayed later.

12. To view the details for individual request simply double click on it. The request details will be displayed in the top right pane. The response details will be displayed in the bottom right pane.

13. The two most important tabs for creating the LoadRunner requests are the Inspectors -> TextView and Inspectors -> Headers tabs.

14. The Inspectors -> TextView tab displays the body of our SOAP requests.
16. The Inspectors -> Headers tab displays all HTTP headers. If the header “SOAPAction” exists then we add it in our LoadRunner script.

17. The Inspectors -> Headers tab displays all HTTP headers. If the header “SOAPAction” exists then we add it in our LoadRunner script.

5.4.1 How to transfer a Fiddler request into a VuGen Web (HTTP/HTML) or SAP – Web script.

Requests are made in Web (HTTP/HTML) and SAP – Web scripts by using the web_custom_request method. This method takes a number of parameters that detail:

- A request name
- URL – The URL of the request
- Method – The HTTP Method (GET or POST)
- Resource – Indicates whether a URL is a resource – This will always be set to “0” for this task.
- RecContentType – The content type – This will always be “text/xml” for this task.
- Referer – The referring web page – This can be blank for this task.
• Mode – The recording level – This will always be “HTML” for this task.

• Body – This will contain the request body from fiddler.

```javascript
web_custom_request("QueryEngine.asmx 5",
   "URL=http://gbxp0025 sapbco.com 80/OSoft/App/OSoftDataService/QueryEngine.asmx",
   "Method=POST",
   "Resource=0",
   "RecContentType=text/xml",
   "Referer=",
   "Mode=HTML",
   "EncType=text/xml, charset=utf-8",
   "Body="
      "RequestBody",
   LAST);
```

You can use the template below to create new web_custom_requests:

```javascript
web_custom_request("Request_Name_1",
   "URL=http://server.name.com/OSoft/App/OSoftDataService/QueryEngine.asmx",
   "Method=POST",
   "Resource=0",
   "RecContentType=text/xml",
   "Referer=",
   "Mode=HTML",
   "Body="
      "Request_Body",
   LAST);
```

### 5.4.1.1 Creating a web_custom_request

The instructions below will guide you through the process of updating the web_custom_request with the Fiddler information.

1. Double click on the first request in Fiddler that you want to include in your LoadRunner script.
2. Copy and paste the “web_custom_request” template above into the Action section of a Web (HTTP/HTML) or SAP – Web script.
3. <Optional> Update the request name.
4. The URL is composed of three components: the protocol (HTTP), the server name and port (server.name.com) and the remaining URL (/OSoft/App/OSoftDataService/QueryEngine.asmx). All of this information can be retrieved from the Inspectors -> Headers tab in Fiddler.
5. The method can also be obtained from the Inspectors -> Headers tab.

6. The “Resource”, “ResContentType”, “Referer”, “Mode” and “EncType” parameters do not need to be changed from their default values.

7. Obtaining the “Body” parameter is a multi-step process. First, select the Inspectors -> TextView tab for the request (top right), and then select “View in Notepad”.

8. The “Resource”, “ResContentType”, “Referer”, “Mode” and “EncType” parameters do not need to be changed from their default values.

9. Obtaining the “Body” parameter is a multi-step process. First, select the Inspectors -> TextView tab for the request (top right), and then select “View in Notepad”.
10. Replace all double quotes ("") with the characters backslash + a double quote (\"").

11. Add an additional backslash (\) after each occurrence of the domain name.

**Note**

This is required because VuGen scripts are written in C and a double quote is used to denote a string. By adding a backslash before the double quote we are telling the C processor that it is a literal double quote.

13. Add an additional backslash (\) after each occurrence of the domain name.
Note
This is required because a single backslash denotes special characters in C. Adding an additional backslash tells C to treat both backslashes as a single literal backslash.

15. Next, add a double quote to the beginning and end of each line. This is done so that the request will appear as a single string to VuGen.

16. Copy and paste the contents of notepad into the body parameter.

5.4.1.2 Adding Headers for web_custom_request’s

17. You now have a complete web_custom_request, but there are two headers that may need to be configured to make the web_custom_request function properly.

18. For certain requests the SOAPAction header determines which function a Web Service performs. If this header appears on the Headers tab in Fiddler, then you must add a corresponding web_add_header() function call before the web_custom_request.

19. The second header, Content-Type, specifies the mime type of the message. This should be added for all web_custom_request’s.

The SOAPAction Header

20. To determine if you need to add the SOAPAction header:

1. Review the Inspectors -> Headers tab in Fiddler. If there is a “SOAPAction” header in the Miscellaneous section, then add a web_add_header statement before the web_custom_request.

2. To create add the header replace the token “header_value” in the following string with the SOAPAction value from Fiddler.

4. web_add_header("soapaction", "header_value");
5. In this example, the method would be:


7. Add the `web_add_header` method call above the `web_custom_request` in VuGen.

### The Content-Type Header

1. Review the Inspectors -> Headers tab in Fiddler. Copy the value of the “Content-Type” header in the Entity section, and then add a `web_add_header` statement before the `web_custom_request`.

2. To create add the header replace the token “header_value” in the following string with the Content-Type value from Fiddler.

3. `web_add_header("Content-Type", "header_value");`
5. In this example, the method would be:

7. `web_add_header("Content-Type", "application/x-www-form-urlencoded");`

8. Add the `web_add_header` method call above the `web_custom_request` in VuGen.

### 5.4.2 How-to transfer a Fiddler request into a VuGen Web Services script.

Requests are made in Web Services scripts using the `soap_request()` method. This method takes a number of parameters including:

- **StepName** – The name of the step – any text can be used.
- **URL** – The URL of the web service
- **Snapshot** – The name of the inf file containing a snapshot for the step – this can remain t1.inf for our purposes.
- **SOAPEnvelope** – The body of the SOAP message.
- **ResponseParam** – The name of the output parameter to store the response

```python
soap_request("StepName=NewQueryMethod",
"URL=http://server.name.com/OSoft/App/OsoftDataService/QueryEngine.asmx",
"SOAPEnvelope=Request_Body",
"Snapshot=t1.inf",
"ResponseParam=response",
LAST);
```

You can use the template below to create new `soap_requests`:

```python
soap_request("StepName=NewQueryMethod",
"URL=http://server.name.com/OSoft/App/OsoftDataService/QueryEngine.asmx",
"SOAPEnvelope=Request_Body",
"Snapshot=t1.inf",
"ResponseParam=response",
LAST);
```
5.4.2.1 Verifying Compatibility of request with Web Services Protocol

The Web Services protocol does not accommodate all SAP BusinessObjects Planning and Consolidation client requests. The reason for this is that some client requests use standard HTTP calls. The requests are not compatible with the Web Services protocol if the method is “GET” or the Content-Type is something other than “txt/xml”. The Web (HTTP/HTML) and SAP – Web protocols support those request types.

Checking the HTTP Method

9. The HTTP method can be obtained from the Inspectors -> Headers tab for the request in Fiddler. If the HTTP method is not POST, then you will have to skip this request as it is not compatible with the Web Services LoadRunner protocol.

10. Checking Content-Type

The Content-Type header can be obtained from the Inspectors -> Headers tab for the request in Fiddler. If the Content-Type value is not “text/xml”, then you will have to skip this request as it is not
How To... Plan and Design LoadRunner Scenarios for SAP BusinessObjects Planning and Consolidation

compatible with the Web Services LoadRunner protocol.

5.4.2.2 Creating a soap_request

The instructions below will guide you through the process of updating the soap_request with the Fiddler information.

1. Double click on the first request in Fiddler that you want to include in your LoadRunner script.
2. Copy and paste the “soap_request” template above into the Action section of a Web Services script.
3. <Optional> Update the step name.
4. The URL is composed of three components: the protocol (HTTP), the server name and port (server.name.com) and the remaining URL (/OSoft/App/OSoftDataService/QueryEngine.asmx). All of this information can be retrieved from the Inspectors -> Headers tab in Fiddler.
5. Obtaining the “SOAPEnvelope” parameter is a multi-step process. First, select the Inspectors -> TextView tab for the request (top right), and then select “View in Notepad”.

URL

Protocol

Servername and Port
7. Replace all double quotes ("") with the characters backslash + a double quote (\"").

8. Replace all double quotes ('"') with the characters backslash + a double quote (\"").

9. **Note**

   This is required because VuGen scripts are written in C and a double quote is used to denote a string. By adding a backslash before the double quote we are telling the C processor that it is a literal double quote.

10. Add an additional backslash (\) after each occurrence of the domain name.
Note
This is required because a single backslash denotes special characters in C. Adding an additional backslash tells C to treat both backslashes as a single literal backslash.

12. Next, add a double quote to the beginning and end of each line. This is done so that the request will appear as a single string to VuGen.

13. Copy and paste the contents of notepad into the “SOAPEnvelope” parameter.

14. The “Snapshot” parameter does not need to be updated.

15. The “ResponseParam” value indicates the name of the parameter that will contain the text of the response. This will be used during verification.

5.4.2.3 Adding Headers for soap_request's

You now have a complete soap_request, but you may also need to add a SOAPAction header for the request to work.

Note
Unlike the Web (HTTP/HTML) and SAP – Web protocols, the Web Services protocol only supports SOAP requests. The only header pertinent to SOAP requests is the SOAPAction header. You still need to be wary of the “Content-Type” header, because if it is not “text/xml”, then the request is not compatible with the Web Services protocol. See section 5.3.2.1 for more information.

The SOAPAction Header

16. To determine if you need to add this header:

1. Review the Inspectors -> Headers tab in Fiddler. If there is a “SOAPAction” header in the Miscellaneous section, then we need to add a web_add_header statement before the soap_request.

```
2. To create add the header replace the token “header_value” in the following string with the SOAPAction value from Fiddler.

4. web_add_header("soapaction", "header_value");```
5. In this example, the method would be:

7. `web_add_header("soapaction", "http://outlooksoft.com/NewQueryMethod");`

8. Add the `web_add_header` method call above the `soap_request` in VuGen.

### 5.5 Additional requirements for LoadRunner VuGen scripts

There are some additional LoadRunner methods that need to be added to the VuGen scripts in order for them to function properly. This section will review those methods.

#### 5.5.1 BPC User Authentication in VuGen

The `web_set_user` method is used by all three LoadRunner protocols to setup authentication. The syntax of the method is:

```c
web_set_user("domain\username", "password", "server:port");
```

This method should be added before any `soap_request` or `web_custom_request` methods.

```c
web_set_user("domain\username", "PassWrd", "bpc.domain.com");
```

#### 5.5.2 HTTP Timeout Configuration

By default, all requests will time out after 120 seconds. If any requests take longer than the defined timeout period the script will generate an error, specifying a timeout as the root cause.

Large reports can take longer than 120 seconds to process, especially under heavy loads. Because of this it is recommended that you increase the timeout to at least 5 minutes, more if required. The three methods used to do this are:

```c
web_set_timeout("STEP", "300");
web_set_timeout("RECEIVE", "300");
web_set_timeout("CONNECT", "300");
```

The example above increases the timeout to 300 seconds.
These methods should be added before any soap_request or web_custom_request methods.

5.6 Parameterization

Parameterization is a very important part of making scripts realistic by adding variability. It is appropriate for pretty much any script, but is especially important in reporting.

The act of parameterizing a script is relatively straightforward, but can be very difficult as well. The first step is to determine how users actually interact with a report (or other business process). What is it that they change (account member selections, entity member selections, etc)? Once you have determined that, you can add that information into your script by creating LoadRunner parameters.

The example below demonstrates adding parameters to an EvDRE report.

5.6.1 EvDRE Parameterization Example

9. When EvDRE reports are created, the author selects columns and rows to include in the report. These dimension members are generally the target for parameterization.

10. In this admittedly basic example we will parameterize the members for the dimension “Account”.

11. The screenshot below shows the portion of the EvDRE SOAP request that contains the column and row dimensions and members.

```
web_custom_request("Example_Request_i","
  URL:http://appserver150:1080/651t/app/650DataService/QueryEngine.assm",
  Method=POST,
  Resource=0,
  RecContentType=image/text/xml",
  Referer="",
  Encoding=UTF8,
  EncType=text/xml, charset=utf-8",
  BODY="<xml version='1.0' encoding='utf-8'>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/soap-enc/"
  soap:rootElement="soap:Envelope">
  <soap:Header/>
  <soap:Body>
    <QueryViewResult>
      <QueryViewName/>
      <SqlOnly>false</SqlOnly>
      <ResultType>2</ResultType>
      <strParameter>
        <ParameterName>Account</ParameterName>
        <ParameterValue>""</ParameterValue>
      </strParameter>
    </QueryViewResult>
    <!-- Add more parameters here -->
  </soap:Body>
</soap:Envelope>
</xml>"
```

12. When EvDRE reports are created, the author selects columns and rows to include in the report. These dimension members are generally the target for parameterization.

13. In this admittedly basic example we will parameterize the members for the dimension “Account”.

14. The screenshot below shows the portion of the EvDRE SOAP request that contains the column and row dimensions and members.

```
web_custom_request("Example_Request_i","
  URL:http://appserver150:1080/651t/app/650DataService/QueryEngine.assm",
  Method=POST,
  Resource=0,
  RecContentType=image/text/xml",
  Referer="",
  Encoding=UTF8,
  EncType=text/xml, charset=utf-8",
  BODY="<xml version='1.0' encoding='utf-8'>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/soap-enc/"
  soap:rootElement="soap:Envelope">
  <soap:Header/>
  <soap:Body>
    <QueryViewResult>
      <QueryViewName/>
      <SqlOnly>false</SqlOnly>
      <ResultType>2</ResultType>
      <strParameter>
        <ParameterName>Account</ParameterName>
        <ParameterValue>""</ParameterValue>
      </strParameter>
    </QueryViewResult>
    <!-- Add more parameters here -->
  </soap:Body>
</soap:Envelope>
</xml>"
```

5.6.1.1 Creating a Parameter

15. To create a parameter:

1. select / highlight the text you wish to replace with a parameter, right click it and select “Replace with a Parameter”
How To... Plan and Design LoadRunner Scenarios for SAP BusinessObjects Planning and Consolidation

2. Enter a parameter name and click “OK”.

5.6.1.2 Configuring the Parameter

5. The next step is to add other valid Accounts to the “Account” parameter. To do this:

1. Select “Vuser -> Parameter List” from the file menu in VuGen.

3. Select “Vuser -> Parameter List” from the file menu in VuGen.

2. Enter a parameter name and click “OK”.

4. Select “Vuser -> Parameter List” from the file menu in VuGen.

...
3. Select “Edit with Notepad…”

4. Add additional values, one per line and ensure there is a single blank line at the end of the parameter file. When finished, save and close the parameter file.

5. Ensure that you have the appropriate data when setting parameters; otherwise you introduce the possibility of reports not returning data.

6. Next, update the parameter selection criteria. This has two components. The first is “Select Next Row”, which determines how the next row is selected. The valid values are:
Sequential – Parameters will be selected sequentially
Random – Parameters will be selected randomly
Unique – Parameters will be selected uniquely

8.

9. The second criteria is “Update value on”, which determines when you select the next value. The valid values are:

- Each Iteration – updates the parameter at the beginning of each iteration
- Each Occurrence – updates the parameter each time it is referenced
- Once – updates the parameter the first time it is used. The parameter will retain the same value throughout the execution of the script / scenario.
11. Once you have completed setting the selection criteria you can simulate the actual parameter selection by clicking the “Simulate Parameter...” button. Enter the appropriate number of Vusers and iterations and select “Simulate”.

12.
5.7 Transaction Verification

Transaction verification is the process used to ensure that the SOAP requests function properly. The method used to do this can vary drastically depending on the VuGen protocol (Web, SAP – Web or Web Services) and process (Report, Input Schedule, Admin Task, etc) that is being performed.

The primary method for doing this is to analyze the response from the server for each SOAP request. The best way to learn about the expected responses is to step through your scripts with the “Data Returned By Server” Runtime logging option enabled in VuGen.

5.7.1 Example EvDRE Report Verification code for the Web (HTML/HTTP) and SAP – WEB protocol

The Web (HTML/HTTP) and SAP – WEB protocols can analyze request responses using the web_reg_save_param method. This method allows you to set text boundaries, and store the text within the boundaries as a parameter. That parameter will be analyzed later to verify whether the process worked.

For example, the following code snippet validates the session and return statuses from Example_Request_1.

```c
web_reg_save_param("sessionStatus", "LE-\&lt;SessionStatus\&gt;", "RB-\&lt;SessionStatus\&gt;", "ORD=1", "LAST");
web_reg_save_param("returnStatus", "LE-return status="", "RB-"\&gt;", "ORD=1", "LAST");
web_custom_request("Example_Request_1");
if (strncmp(lr_eval_string("(sessionStatus)"),"succeed") != 0)
    lr_fail_trans_with_error("The session layer reported the non-success status: %s", lr_eval_string("(sessionStatus)");
if (strncmp(lr_eval_string("(returnStatus)"),"succeed") != 0)
    lr_fail_trans_with_error("The return status reported the non-success status: %s", lr_eval_string("(returnStatus)");
```
The code in yellow displays two separate `web_reg_save_param` method calls. These methods will scan the response of the `web_custom_request` (not highlighted) and store the results as parameters. Finally, the code in light blue verifies the response by comparing the expected response “succeed” with the “sessionStatus” and “returnStatus” parameters.

This is the most basic level of verification. Depending on the requirements you could also verify additional details, such as the number of cells returned.

**Note**

More information on the `web_reg_save_param` method can be found in VuGen’s help files.

**Note**

Verification code requires some C programming skills.

### 5.7.2 Example EvDRE Report Verification code for the Web Services protocol

The Web Services protocol stores the entire request response as a parameter. In order to verify the response, we must analyze the entire response parameter. This has to be done using C code. Depending on your requirements, the effort can be quite intensive. The following is an example providing the same verification process as the example outlined in section 5.6.1.

```c
if (strstr(lr_eval_string("(response)"), "&lt;SessionStatus&gt;\succeeded\&lt;/SessionStatus&gt;\:")) == NULL
{
    lr_output_message("ERROR");
    lr_error_message("Error with Example_Cell_1: %s", lr_eval_string("(response)"));
    lr_end_transaction("022", LR_FAIL);
    return 0;
}
else if (strstr(lr_eval_string("(response)"), "return status="\succeeded"&gt;\:")) == NULL
{
    lr_output_message("ERROR");
    lr_error_message("Error with Example_Cell_1: %s", lr_eval_string("(response)"));
    lr_end_transaction("022", LR_FAIL);
    return 0;
}
```

In this example, the SOAP response is stored in the parameter “response”. The strstr C method is used to determine if the expected result is found in the “response” parameter. If not, and error is raised.

### 5.8 Notes for LoadRunner Support in Planning and Consolidations 7.0 for SAP NetWeaver

SAP BusinessObjects Planning and Consolidation 7.0, for SAP NetWeaver does not officially support Load Runner as a testing tool. This is due to the fact that some data in the SOAP responses are encrypted, and therefore LoadRunner cannot interpret them directly. The root cause of the issue is that user information is stored as encrypted data, and when the authenticated user does not match the user information in the encrypted data, errors are raised.

**Important**

The BPC Toolkit for LoadRunner provides this functionality and the process is documented in the guide “How-To Use the SAP BusinessObjects Planning and Consolidation Toolkit for LoadRunner”.

July 2009 37
Important
LoadRunner support is currently targeted for SAP BusinessObjects Planning and Consolidation 7.5, version for SAP NetWeaver.

5.9 Notes for LoadRunner Support in Planning and Consolidations 7, for Microsoft

SAP BusinessObjects Planning and Consolidation 7.0, version for the Microsoft Platform supports LoadRunner, although there are some additional steps one has to take if parameterization of users is required.

Important
The BPC Toolkit for LoadRunner provides this functionality and the process is documented in the guide “How-To Use the SAP BusinessObjects Planning and Consolidation Toolkit for LoadRunner”.

5.10 Notes for LoadRunner Support in Planning and Consolidations 5.1, for Microsoft

SAP BusinessObjects Planning and Consolidation 5.1, version for the Microsoft platform provides some LoadRunner support, though some actions may not be 100% compatible. For example, in some Web Service calls, username information is stored in an encrypted fashion. Because of this, you may have to re-run all scripts with the same credentials used to record the scripts.

CAUTION
Limited LoadRunner Support for BPC 5.1 was recently added in a support package and requires a parameter change in the Application Sets database. For more information review SAP Note 1300873 for current Support Pack requirements and configuration details (https://service.sap.com/sap/support/notes/1300873).

5.11 Unit Testing Scripts with VuGen

It is important that you verify the proper operation of scripts directly in VuGen before attempting a full scale scenario test. It is recommended that scripts are executed with both the “Parameter Substitution” and “Data Returned by Server” extended log options enabled.
Next, execute the script and verify that your parameterization and verification code are working as expected.
6. Scenario Design, Execution and Analysis

Proper scenario design requires a great deal of knowledge regarding the business processes to be executed. It is best tackled by a team consisting of members with business process expertise and members that know how to translate those requirements into a meaningful scenario.

This section will outline how to create and execute a scenario. Lastly, some details regarding analyzing the results are provided.

Note
Appendix C contains a Scenario/Script Schedule Planning Template that is useful in creating the scenario.

6.1 Scenario Design Considerations

Once you have determined the business processes to test and the target concurrency rate you can begin defining different scenarios to test. Here are some suggestions that can help in isolating performance bottlenecks:

- Execute each business process separately, within its own scenario to determine how it functions by itself.
- Execute the scenario including all business processes with a varying Virtual User count (25%, 50%, 100%, etc) to isolate concurrency levels where bottlenecks occur.
- If you encounter poor performance in scenarios, then you may want to focus on business logic (default logic or data manager packages) as well as the system metrics (CPU, memory, database locking, etc). This can be done by commenting out sections in default logic, or disabling scripts that execute data manager packages and re-executing the scenario where you encountered poor performance. You will be in a much better position to improve performance issues after you have isolated the underlying cause. At that point you can determine if the problem is best addressed by additional hardware, a logic or business process change, etc.

Important
It is important to note that not all performance problems can be solved with additional hardware or performance tuning. In some cases business process changes or application logic changes must be applied. For this reason, it is crucial that your load testing initiative includes your lead application consultants and business process experts as well as your technical team(s).

6.2 Designing the Scenario Schedule

The scenario schedule defines many details including:

- Which scripts are included in a scenario
- When each script begins
- How users are “Ramped Up”, or logging in, for each script
- How long users execute scripts before “Ramping Down”, or logging out, for each script
- How users are “Ramped Down”.

This section includes an overview of how to design the scenario schedule. This should be a review for anyone who has worked with LoadRunner in the past.
1. Open the LoadRunner Controller application.

2. Create a new Scenario. Select the scenario type of "Manual Scenario".

   ![New Scenario dialog box]

   **Note**
   You can also select the “Use the Percentage Mode...” if you want to allocate users to scripts based on percentages instead of user count.

3. Browse to and select the scripts to include in the scenario from the bottom pane. Click OK when complete.
4. After selecting “OK” you will enter the design tab for the scenario. Scripts can be added and removed/disabled in the top pane. You design the schedule in the Global Schedule pane in the bottom right.

5. After selecting “OK” you will enter the design tab for the scenario. Scripts can be added and removed/disabled in the top pane. You design the schedule in the Global Schedule pane in the bottom right.
7. There are two options when setting up the scenario schedule, Scenario and Group.

8. Schedule by Scenario: Allows you to configure the schedule for all scripts at once. This is easier to configure, but less flexible.

   Schedule by Group: Allows you to configure schedules for each script individually.

9. There are two options when configuring the Run Mode, Real-life schedule or Run until complete.

10. Real-life schedule: Allows you to start and stop users and set durations for different activities throughout the scenario.

11. Run-until complete: Allows you to configure how many users to start. They will run until they have completed the number of iterations specified in the runtime settings of each script.

12. Once you determine the schedule and run mode types, you can define the actual schedules. This is done by adding and configuring “Actions”. These actions include:

   - Start Group – Determines when to start the group of users (only available for “Schedule by Group”).

13. Initialize – Determines how users are initialized.
• Start Vusers - Determines how many users to start, and what the ramp up schedule is (i.e. all users log in at once, 1 user logs in each 5 seconds, etc).

```
Start Vusers
10 Vusers:
  Simultaneously
  Vusers every 00:00:15
```

14.

• Duration – how long to run with the current Vuser count until the next Action

```
Duration
Run for 0 days and 00:05:00
```

15.

• Stop Vusers – Determines how many users to stop, and how they ramp down (i.e. stop all users at once, or one user every 5 seconds, etc).

```
Stop Vusers
5 Vusers:
  Simultaneously
  Vusers every 00:00:30
```

16.

17.

18.

**Example**

The screenshot below displays an example schedule.
6.3 Configuring Systems Monitoring

Systems monitoring is configured in the Load Runner Controller software before executing a scenario. This allows you to add any WMI counters you wish to monitor. At a minimum, you should ensure that you are capturing the following metrics for each server type:

- **.NET Application Server (Microsoft or SAP NetWeaver release)**
  - %Processor Time (Processor_Total)
  - Processor Queue Length (System)
  - Memory (Available Mbytes)
  - Physical Disk \ Current Disk Queue Length
  - Physical Disk \ Average Disk Queue Length
  - Process (w3wp.exe) \ % Processor Time
  - Process (w3wp.exe) \ Working Set

- **Microsoft Analysis Server (BPC for Microsoft)**
  - %Processor Time (Processor_Total)
  - Processor Queue Length (System)
  - Memory (Available Mbytes)
  - Physical Disk \ Current Disk Queue Length
  - Physical Disk \ Average Disk Queue Length
  - Process (msmdsrv) \ %Processor Time
  - Process (msmdsrv) \ Working Set
  - Analysis Services \ Current Locks
  - Analysis Services: Current Lock Waits

- **Microsoft SQL Server (BPC for Microsoft)**
  - %Processor Time (Processor_Total)
  - Processor Queue Length (System)
  - Memory (Available Mbytes)
How To... Plan and Design LoadRunner Scenarios for SAP BusinessObjects Planning and Consolidation

- Physical Disk \ Current Disk Queue Length
- Physical Disk \ Average Disk Queue Length
- Process (sqlservr) \ %Processor Time
- Process (sqlservr) Working Set
- SQL Server \ Buffer Manager
- SQL Server \ Buffer Cache Hit Ratio

- **ABAP Application Server (BPC for SAP NetWeaver)**
  - %Processor Time (Processor_Total)
  - Processor Queue Length (System)
  - Memory (Available Mbytes)
  - Physical Disk \ Current Disk Queue Length
  - Physical Disk \ Average Disk Queue Length

**Note**

Also add the % Processor Time and Working Set counters for the database engine that you are using.

**Important**

You are not limited to the counters mentioned above. These should be considered the baseline set of counters to include in your test, however, be aware that as more counters that are added, more system resources will be on both the systems being monitored and the LoadRunner controller and generators.

To add WMI counters in the Load Runner Controller:

1. Launch Load Runner controller, load your scenario and select the Run tab.
2. Select the Windows Resources graph in the “Available Graphs” section:
3. Right click on the Windows Resources graph and select “Add Measurements”
5. Click the “Add” button in the Monitored Server Machines section. Enter each Application Server,

6. After adding a monitored server, a default set of WMI counters will be added. To add additional counters click the “Add” button in the Resource Measurements section with the appropriate server select in the Monitored Server Machines section. This will launch an additional dialog box providing access to all available WMI counters for the specified machine.

7. 

8. Once you have completed adding counters, click the “OK” buttons.
6.4 Executing and Monitoring Test Scenario

Once your scenario is configured, and WMI counters are setup you can start the scenario. This process is quite simple.

6.4.1 Starting a scenario

1. To start a scenario, simply select the “Start Scenario” button on the Run tab.

2. 

6.4.2 Monitoring a scenario

You can also monitor the scenario through the Load Runner controller software. There are various tools available for this task. The first are the graphs, which are updated during runtime. Some particularly useful graphs are the “Windows Resources” and “Transaction Response Time” graphs. Using these you can view current system utilization and transaction response times, respectively. There are a number of additional graphs available as well.

In addition to the graphs, there is also a Scenario Status section that details the current testing status. It includes information on:

- Running Vusers
• Elapsed Time
• Hits/Second
• Passed Transactions
• Failed Transactions
• Errors

You can select the magnifying glass next to “Passed Transactions”, “Failed Transactions” and “Errors” to get additional details.

6.4.3 Monitoring Data Volumes

Data volumes can grow rapidly when executing LoadRunner scenarios which can adversely impact performance. Because of this, it is definitely worth monitoring during testing.

6.4.3.1 Monitoring Data Volumes for Microsoft Releases

High record counts in the WriteBack (>100,000) and FAC2 tables (>2-4 million records) will adversely affect performance. To address the WriteBack table size, ensure that you have a maintenance plan that will routinely perform lite optimizations when record counts climb above 100,000 records. Lite optimizations can occur with the system online, but Full optimizations require system downtime. Monitoring the record counts in the FAC2 tables will enable you to determine how often you will need to perform tasks like Full optimizations that will require system downtime.

⚠️ Note

There is a delivered data manager package that allows you to perform and schedule optimization operations.

6.4.3.2 Monitoring Data Volumes for SAP NetWeaver Releases

The SAP NetWeaver release(s) handle data in a different way than Microsoft. In SAP NetWeaver releases, data is added to requests. Each request contains a somewhat fixed number of records, so multiple requests will likely be created during scenario execution. Ensure that the number of requests does not exceed the amount recommended for your underlying database. To close these requests, compress the cube.

6.5 Analyzing Results

Result analysis is the key concept that allows you to judge the success level of your testing. The process you take for analyzing your results will likely differ based on your success criteria. This section will provide some general advice on analyzing your results with Load Runner Analysis.
Once your test is complete, make sure you store the results by selecting “Results -> Analyze Results” in the Load Runner Controller software.

6.5.1 Load Runner Analysis Overview

The Load Runner Analysis tool offers a wealth of information regarding the executed scenario. This section will explore some basic functionality of the Load Runner Analysis application.

1. Launch the Load Runner Analysis Application and open the appropriate Analysis Session file.

2. Upon loading the Analysis Session, you should see the Session Explorer Window, normally in the upper left hand side of the Analysis application. This window allows you to navigate between the Summary Report and various graphs displaying different aspects of the scenarios results.

3. If you don’t see the Session Explorer window it can be opened by selecting “Windows -> Session Explorer”
6.5.1.1 The Summary Report

The Summary Report, which is the first item displayed in the Session Explorer, provides a good overview of the scenario including:

- The Maximum Number of Running VUsers
- The Total Throughput in bytes
- The Average Throughput (bytes/second)
- Total Hits
- Average Hits per Second
- Total Errors (if any occurred)
- A Transaction Summary section outlining response times and exit status

<table>
<thead>
<tr>
<th>Summary Report</th>
<th>Running VUsers</th>
<th>Hits per Second</th>
<th>Throughput</th>
<th>Transaction Summary</th>
<th>Average Transaction Response Time</th>
</tr>
</thead>
</table>

### Analysis Summary


- **Scenario Name**: C:\LoadRunner\Scenarios\Shanghai_TestScenario_5_2.4
- **Results in Session**: C:\LoadRunner\Emp\Export\ShortRunSummaryReport1
- **Duration**: 1 hour, 19 minutes and 23 seconds

### Statistics Summary

- **Maximum Running VUsers**: 276
- **Total Throughput (bytes)**: 122,760,997
- **Average Throughput (bytes/second)**: 22,779
- **Total Hits**: 20,620
- **Average Hits per Second**: 4.028
- **Total Errors**: 1

You can define SLA data using the SLA configuration wizard.
You can analyze transaction behavior using the Analyze Transaction mechanism.

### Scenario Behavior Over Time

The SLA status of the following measurements displayed over time. You can select a specific time range for each transaction in order to analyze the time range.

<table>
<thead>
<tr>
<th>Measurement Name</th>
<th>Time Ranges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Under Test</td>
<td>0</td>
</tr>
<tr>
<td>Errors</td>
<td>0</td>
</tr>
<tr>
<td>Throughput</td>
<td>0</td>
</tr>
</tbody>
</table>

### Transaction Summary

6.5.2 Graphs

Load Runner Analysis creates five standard graphs:

- **Running Vusers**
  - Displays the number of Vusers that executed Vuser scripts, and their status, during each second of a load test. This graph is useful for determining the Vuser load on your server at any given moment.

- **Hits per Second**
  - Displays the number of hits made on the Web server by Vusers during each second of the load test. This graph helps you evaluate the amount of load Vusers generate, in terms of the number of hits.

- **Throughput**
Displays the amount of throughput (in bytes) on the Web server during the load test. Throughput represents the amount of data that the Vusers received from the server at any given second. This graph helps you to evaluate the amount of load Vusers generate, in terms of server throughput.

- **Transaction Summary**
  - Displays the number of transactions that passed, failed, stopped, or ended with errors.

- **Average Transaction Response Time**
  - Displays the average time taken to perform transactions during each second of the load test. This graph helps you determine whether the performance of the server is within acceptable minimum and maximum transaction performance time ranges defined for your system.

### 6.5.2.1 Adding Additional Graphs

4. There are many additional graphs that can be created to display different types of information. To add a graph:

   1. Right click on the Graphs folder in the Session Explorer and select “Add New Item -> Add New Graph…” or use the shortcut “Ctrl + A”. This will launch the “Open a New Graph” dialog.

   ![Session Explorer Graphs Folder](image)

   2. Select the graph that you wish to add, set any properties if appropriate and click the “Open Graph” button.
4. **Tip**

You can alter the reporting period for many graphs by editing the Scenario Elapsed Time property. This can be very useful for isolating the impact of different events like a period of very high concurrency or a very intensive process.

### 6.5.2.2 Merging Graphs and Auto Correlate

While most of the graphs add value to scenario analysis, often times the data for two graphs can be much more compelling than one. I often merge the Windows Resources and Average Transaction Response Times graphs to aid in correlating high system utilization with transactions or vise versa.

Load Runner Analysis can also help you determine the root cause of different events, like high response times, by Auto Correlating results.

To merge graphs:

1. Select one of the graphs you wish to merge in the Session Explorer
2. Right click on the graphs canvas and select “Merge Graphs”
3. Select the graph to merge with, the type of merge and enter a graph title then select “OK”.

4. To Auto Correlate a graph:
   1. Right click on the graphs canvas and select “Auto Correlate”.
2. The Auto Correlate dialog will be displayed. Here you can configure Correlation Options and time ranges for each measurement. If unsure, keep the defaults and select “OK”.

4. The resulting graph allows you to select the measure you wish to correlate to. Upon selecting a new measurement the graph data will be updated to display the remaining measures and their % correlation match.
6.5.3 Interpreting Analysis Results

Interpreting the analysis results is a pretty complex topic and requires a broad understanding of both the transactions you are testing and the underlying systems architecture. This section will outline general areas to review and how to analyze some counters / measures.

**Important**

The Auto Correlation feature of the Load Runner Analysis tool is very helpful in determining what measures are relevant in determining the root cause of poor performance.

**Note**

Load Testing is an iterative process. It may be necessary to add additional counters and re-run scenarios to isolate the root cause. This should be expected and adequate time should be included to allow for this.

6.5.3.1 Average Transaction Response Time

The Average Transaction Response Time graph quite possibly contains the most important measures in the analysis results. By looking in this graph, one can easily see whether response times are within expectations.

If average transaction response times are within expectations then your testing is likely complete. You should re-test a few times to verify consistency and make sure that the basic Windows resources are healthy, but for the most part you are complete.

If the average transaction response times are higher than expectations, then there are two areas to review:

1. Review the Windows Resources to identify the bottleneck / root cause
2. Review the scenario to ensure that it accurately represents the business process you were targeting.

6.5.3.2 System Root Cause Analysis for Poor Performance

If transaction performance does not meet signoff times then it is time to start looking for bottlenecks. One approach for this task is to auto correlate the measures in the Average Transaction Response Time and Windows Resources graphs. Once the graphs have been correlated, review the poor performing transactions and identify the windows resources that correlate to those events.

**Note**
It is also possible for LoadRunner to monitor UNIX systems, which may be the case for SAP BusinessObjects Planning and Consolidation 7.0, for SAP NetWeaver.

The next step is to research the windows resources that appear to be the root cause. If you are unable to get enough information from the current measures, add more detailed metrics to your scenario and re-execute the test.

After taking tuning steps, by either changing parameters or increasing hardware resources, re-execute the scenario. Finally, review the results to see if transaction response times have been brought within signoff times.

**Important**
If you encounter a few transactions exceeding expectations, it may also be helpful to review the process itself to see if it can be redesigned to execute in a more optimal manner.

6.5.3.3 Analysis for Common Measures

This section describes some common measures and some details on how to interpret them.

- **CPU: %Processor Time (_Total)**
  - CPU consistently over 85 – 90% indicates a CPU bottleneck for the server

- **Memory: Available Mbytes, Pages/Sec**
  - A low amount of Available Mbytes indicates a memory shortage for the server

- **Physical Disk: Current Disk Queue Length, Average Disk Queue Length**
  - If this number spikes during poor processing performance then the current disk storage solution may not be adequate to support the needs of the application

- **Process msmdsrv: %Processor Time**
  - If the %Processor Time is consistently about 85 – 90% it indicates a CPU bottleneck for Analysis Services.

- **Analysis Services: Current Locks, Current Lock Waits**
  - The Current Locks counter measures the current number of locked objects, and can indicate when locks are preventing users from accessing data in a timely fashion.
  - Current Lock Waits measures the number of clients waiting for a lock.

- **Process sqlservr: %Processor Time**
  - If the %Processor Time is consistently about 85 – 90% it indicates a CPU bottleneck for SQL.

- **SQL Server: Buffer Manager \ Buffer Cache Hit Ratio**
The buffer cache hit ratio measures the percentage of pages that were in memory and did not require a disk access to get at the data. A buffer cache hit ratio should be nearly 99 percent. If the ratio is lower, there may be memory constraints that affect performance.

**Important**

If you are unsure of what a measure indicates, select the Windows Resources graph in Load Runner Analysis and select “Show measurement description…”

This will display the Measurement Description dialog.

---

### 6.6 Working through Performance Issues

If you encounter poor performance during the execution of a scenario, then your first task is to isolate the root cause of the performance issue. There are a number of tools that can be used to acquire this information including:

- The Systems metrics recorded during the scenario execution
- The Transaction response times recorded during the scenario execution
- BPC Log files (EvDRE, Logic Debug files, etc)

By reviewing the various metrics and log files in conjunction with the results from our baseline tests we should able to determine the root cause. The root causes will likely fall in to one of three broad categories as outlined in the following sections.
6.6.1 Hardware Bottleneck

Hardware bottleneck issues are encountered when you reach a capacity problem with one of the physical components in the system, for example CPU or RAM. They can be identified by reviewing the systems metrics for scenarios at different concurrency levels. If you can identify a point in time where one of the system resources is completely or near completely utilized and that correlates with a drop in transaction performance than it is very likely that you have encountered a hardware bottleneck. This can be addressed by:

a. Adding additional capacity for the over-subscribed resource (adding memory or processors or upgrading the server)

b. Identifying resource intensive business processes and changing logic or methodology to decrease their resource cost.

6.6.2 Systems Tuning

Systems tuning issues occur when a configurable resource, such as memory available to a process, reaches a threshold. In these cases, the physical server has additional capacity for that resource, but tuning is required to allow the process to access it. These can be identified by reviewing BPC logs, system metrics and transaction response times.

6.6.3 Application Design

In some cases, performance problems cannot be addressed by either adding additional hardware or systems tuning. In other cases, the underlying problem can be traced to the way business processes interact with each other. These cases are often visible in metrics such as database or analysis services locks. Most of these issues can be traced back to the way logic or an application feature (such as dimension member formulas) is being used to obtain a business goal. BPC logs and the transaction response time metrics can be helpful in determining which business process(es) are being affected, and a closer analysis of those business processes will lead you to the root cause. It can also be helpful when working with business processes that load data to perform single iteration tests while paying close attention to the number of records that get created. In some cases it can be easy to overlook the effect script / default logic can have on your data volumes. If you cannot find the root cause in this way, try commenting out function calls in default or script logic packages that operate on large volumes of data and re-run the scenario.

**Note**

You can generally find long running transactions by reviewing standard BPC logs.

Once you have identified the root cause, you can determine alternative means to achieving the business goal, decide on the possible go forward strategies, implement them and finally retest them to verify which strategy yields the best outcome.
7. Appendix
7.1 Appendix A – Scenario Planning Template

Scenario Name:
Goal of Scenario (Scalability, Edge Case, etc):

Exit criteria for successful test (Response time, User Count, etc):

BPC Testing Project Manager:
(Responsible for coordinating with the SME and LoadRunner Technicians in testing exercise)

BPC Subject Matter Expert:
(Responsible for ensuring the testing represents real use cases)

HP LoadRunner Technician(s):
(Responsible for Scripts / Scenario Creation)

Expected Project Start Date:
Expected Project End Date:

Scripts Included In Scenario:

<table>
<thead>
<tr>
<th>Script Name</th>
<th>Users Allocated to Script</th>
<th>Script Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Servers to be monitored:

<table>
<thead>
<tr>
<th>Server Name</th>
<th>Operating System Type</th>
<th>BPC Server Role</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.2 Appendix B – Script Planning Template

This section outlines questions consider when planning a Script

Script Name:
Script Owner (LoadRunner):
Script Owner (Functional):
Script Goal / Description:

Script Transaction Overview:

<table>
<thead>
<tr>
<th>Transaction Name</th>
<th>Transaction Type</th>
<th>Parameterized Dimensions (Y/N)</th>
<th>Verification Required (Y/N)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Transaction Details Section

Transaction Name:

List of Dimensions to be Parameterized:

List Valid Dimension Member Criteria for Parameterized Dimensions:

Verification Requirements:

Transaction Name:

List of Dimensions to be Parameterized:

List Valid Dimension Member Criteria for Parameterized Dimensions:

Verification Requirements:
### 7.3 Appendix C – Scenario/Script Schedule Planning Template

**Scenario Name:**

**Total Scenario Duration:**

For each script, complete the following schedule (Applies to user ramp up / Scheduling):

<table>
<thead>
<tr>
<th>Script Name:</th>
<th>Action</th>
<th>User Count</th>
<th>Action Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>