How-To Use the SAP BusinessObjects Planning and Consolidation (BPC) Toolkit for LoadRunner

Applicable Releases:

SAP BusinessObjects Planning and Consolidation 7.0, version for 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 0.97.1
July 2009
<table>
<thead>
<tr>
<th>Document Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.97.1</td>
<td>Internal release – Updated 07/08</td>
</tr>
</tbody>
</table>
## Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example Text</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>Example text</td>
<td>Emphasized words or phrases in body text, graphic titles, and table titles.</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>
</tbody>
</table>

## Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Caution" /></td>
<td>Caution</td>
</tr>
<tr>
<td><img src="image" alt="Note or Important" /></td>
<td>Note or Important</td>
</tr>
<tr>
<td><img src="image" alt="Example" /></td>
<td>Example</td>
</tr>
<tr>
<td><img src="image" 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. How-To use the BPC Fiddler Converter Component ...............................................2
   4.1 Using Fiddler to Capture SAP BusinessObjects Planning and Consolidation
       Processes ................................................................................................................2
   4.2 Converting the Fiddler Trace file to a LoadRunner Script ....................................4
   4.3 Creating the LoadRunner Script .........................................................................9
   4.4 What does the Fiddler Converter do? ................................................................ 12

5. BPC User Checksum Generator Component ....................................................... 13
   5.1 Creating User Checksums .............................................................................. 13
       5.1.1 Creating a Checksum for a single user ................................................ 14
       5.1.2 Creating Checksums for a User Range .............................................. 15
   5.2 Using the Checksums in a LoadRunner Script ................................................ 16

6. BPC User Context Generator Component ......................................................... 19
   6.1 Creating User Contexts .................................................................................... 20
       6.1.1 Creating Contexts for a Single User .................................................... 20
       6.1.2 Creating Contexts for a User Range ..................................................... 21
   6.2 Using the Contexts in a LoadRunner Script ..................................................... 22
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.

A companion How-To guide is available that describes the process of using LoadRunner with SAP BusinessObjects Planning and Consolidation in detail. This How-To guide documents a set of tools created within the EPM RIG that enable rapid script creation and assist with platform specific challenges related to LoadRunner.

2. Background Information

This How-To guide consists of the guide you are reading now, as well as a utility named the BPC LoadRunner Toolkit. ([Download](#)) This application converts currently contains three components:


2. The User Checksum Generator allows you to use multiple user accounts within LoadRunner scripts by generating user specific checksums.

   **Important**
   The User Checksum Generator is specific to the Microsoft releases.

3. The User Context Generator allows you to use multiple user accounts within LoadRunner scripts by generating user specific encrypted contexts. This also allows you to update the application set name with the encrypted context.

   **Important**
   The User Context Generator is specific to the SAP NetWeaver release.

3. Prerequisites

- HP LoadRunner licensed for one of the following protocols:
  - Web (HTTP/HTML)
  - SAP + Web
  - Web Services
- Fiddler (for Fiddler Converter Component)
- SAP BusinessObjects Planning and Consolidation, version for SAP NetWeaver or
- SAP BusinessObjects Planning and Consolidation, version for SAP Microsoft
4. How-To use the BPC Fiddler Converter Component

This section outlines the steps required to create LoadRunner scripts with the BPC Fiddler Converter utility.

4.1 Using Fiddler to Capture SAP BusinessObjects Planning and Consolidation Processes

The first step in this process is to capture the SAP BusinessObjects Planning and Consolidation process that will be converted into a LoadRunner script. This is done with the appropriate SAP BusinessObjects Planning and Consolidation client application and Fiddler.

For more information, refer to the How-To guide entitled “How-To Use the SAP BusinessObjects Planning and Consolidation, version for SAP NetWeaver Load Runner Utility”.

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

2. Launch fiddler.

3. Once fiddler has completed initializing, launch the SAP BusinessObjects Planning and Consolidation client you want to record.

4. Login to SAP BusinessObjects Planning and Consolidation and navigate to the report, input schedule or other process you wish to execute.

At this point Fiddler will have captured a number of SOAP requests.
5. After you complete executing the process, stop capturing traffic in Fiddler by selecting “File -> Capture Traffic”.

6. The remaining sessions make up all of the HTTP calls that were made in the process you recorded.

7. Save the Fiddler sessions in Fiddlers’ native format (ArchiveZip). This can be consulted in the future if you want additional details about the requests and responses included in your recording.

   **Note**
   This is an optional step. It is a recommended practice as the resulting file can be loaded in Fiddler at a later time for additional review.

8. Save the remaining sessions as Text. This text file will be used by the utility to create the LoadRunner script. To do this:
   - Select All Sessions
   - **Tip**
     If you keep track of the request numbers (leftmost field in Fiddlers Web Sessions dialog) you can save subsets of the sessions individually. This gives you the benefit of more specific LoadRunner transactions. For example, requests 1 through 5 may be for the “Login” transaction, while requests 6 through 11 may represent a data manager package submission.
   - Select File -> Save -> Session(s) -> As Text…
4.2 Converting the Fiddler Trace file to a LoadRunner Script

The Fiddler Converter component of the BPC Toolkit for LoadRunner utility reads the details stored in a Fiddler Session file (text) and converts each request into the appropriate LoadRunner request. The resulting output files’ text acts as the content of the LoadRunner Virtual User Generator Action section.
In addition, the utility adds the following:

- A LoadRunner transaction encapsulating all LoadRunner Requests in the Fiddler text file.
- A LoadRunner sub transaction encapsulating each individual LoadRunner Request. The sub transaction name reflects the query type and request index.
- C utility methods are added for convenience and for use in the verification code.
- Each valid request captured by Fiddler will be converted to the LoadRunner request specific to the protocol you select. The conversion process performs the following tasks:
  - Formats the request body appropriately for LoadRunner.
  - Add Header values where appropriate.
  - Determines if requests are compatible with the selected LoadRunner protocol. If not, a comment is added to the output file.
- Verification code is added to EvDRE Report and Input Schedule requests.

1. Launch the BPC LoadRunner Toolkit utility.

2. Enter a “Primary Transaction Name”. The “Primary Transaction Name” is used for the following:
   - As the name of the primary LoadRunner transaction encapsulating all of the actual requests.
   - As the name of the output file (a “.txt” extension is added) which contains the HP LoadRunner Virtual User Generator Action section.

3. Select the appropriate “BPC Version”.

![BPC Toolkit for LoadRunner](image)
Important

The version selected here must match the version you used during the Fiddler recording. For example, this utility cannot convert requests from SAP BusinessObjects Planning and Consolidation 7.0 for Microsoft into SAP BusinessObjects Planning and Consolidation 7.0 for SAP NetWeaver.

4. Select the LoadRunner Protocol that you will use.

5. Select the Transport Protocol (either HTTP/HTTPS) in use.
6. Enter the Fiddler text file created in section 4.1.

7. Enter the “Output Directory”. This is where the converted file will be stored.

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/knownissues.asp)
8. The “Create Action Section” check box determines whether an entire Load Runner Action section gets created, or just the Web or SOAP requests.

9. Click the “Convert to LoadRunner” button.

10. A dialog box will be displayed when the conversion is complete. The message will review the number of requests converted and the location and name of the output file.
4.3 Creating the LoadRunner Script

1. Open HP LoadRunner Virtual User Generator.
2. Create a new script with the LoadRunner protocol you specified in section 4.2.

**Note**
If you are using either the “Web (HTTP/HTML)” or “SAP – Web” protocol, you should select “Cancel” in the “Start Recording” dialog. The information we need is in the text file created in section 4.2.
3. Open the text file created in section 4.2 with your favorite text editor (notepad, etc). Select and copy all of the text.

4. Overwrite the contents of the Action section of the LoadRunner script you created (step 2 of this section) with the text in the clipboard.
5. Search for “web_set_user” in LoadRunner. Once found, update the default arguments.

- “Domain\Username” – Replace with the SAP BusinessObjects Planning and Consolidation users “Domain\Username” you used in section 4.1.

- “Password” – Replace with the users’ password.

**Note**

You can encrypt the password using LoadRunner’s Password Encoder tool. If you use this tool, you will have to replace the string “Password” with the function:

\[
lr_decrypt("Encoded_String")
\]

- “ServernameAndPort” – Replace with the Server name and Port used during the recording process. If unsure, you can obtain it from the URL property in any of the LoadRunner requests.
6. The script is ready to be tested and customized.

### 4.4 What does the Fiddler Converter do?

This section lists the various functions performed by the utility included with this How-To Guide.

- Parses the following from the Fiddler text file for each SOAP request and response:
  - SOAP Request URL
  - SOAP Request body
  - HTTP Request Type (POST/GET/etc.)
  - SOAPAction Header
  - Content-Type Header
  - BPC Query Type (AxisQuery/CellQuery/etc.)
  - HTTP Response Code

- The SOAP responses are checked for the standard return and/or session status messages. If they are found, verification code will be added to the resulting LoadRunner script for this request.

- If the BPC Query Type is an EvDRE report or an Input Schedule then basic verification code will be added to the resulting LoadRunner script for this request.

- Formats the SOAP request so that it is compatible with a “C” style string. For example, backslash characters “\" are automatically escaped and newline characters are added where necessary.

- Checks for standard “Return Status” and “Session Status” messages in responses. Adds verification if for appropriate return codes.

- Creates a LoadRunner script for the requested protocol, either Web Services or Web (HTTP/HTML).

- Wraps all SOAP/HTTP requests in a single LoadRunner transaction. The transaction name is based on the “Script Name” you specify in the utilities interface.

- Creates a LoadRunner sub-transaction for each SOAP/HTTP request, based on the specified Script Name, the BPC query type (if identified) and the numeric ID of the request.

- Ignores any unsuccessful (non HTTP-200 status) SOAP/HTTP requests when creating the LoadRunner script.

- Ignores any requests that are not valid for the Web Service protocol. It places a comment in the LoadRunner script identifying that a request was skipped.
- Ignores any requests that are not valid for the Web (HTTP/HTML) / SAP - Web protocols. It places a comment in the LoadRunner script identifying that a request was skipped.
- Splits the SOAP request body string into shorter string if it exceeds 10,000 characters. The lines are split at known delimiters (; and |) to ensure dimension member names are not truncated.
- Content-Type and SOAPAction HTTP headers are added to the LoadRunner scripts when appropriate.
- Adds code to correlate the ‘__VIEWSTATE’ parameter used in BPC Live Reporting.
- Adds code to correlate for Reporting Services

5. BPC User Checksum Generator Component

The BPC User Checksum Generator component is used to generate username checksums for BPC 5.1 and BPC 7.0 for Microsoft. This is only required if you wish to execute the LoadRunner scripts with a different BPC user (or users) than you used when recording the script. This allows you to use other user accounts by creating the BPC specific checksum values for a range of users. With this, it is possible to parameterize both the username and checksum and avoid any authentication issues.

![Important](image)

This component is only applicable to the BPC 5.1 and 7.0 for Microsoft releases.

5.1 Creating User Checksums

User Checksums are created with the BPC Toolkit for LoadRunner. The component is located on the User Checksum Generator tab.
5.1.1 Creating a Checksum for a single user

1. After the BPC LoadRunner Toolkit has been launched, click on the “User Checksum Generator” tab.

2. Ensure that the “Single User” radio button is checked.

3. Enter the username in the format
   a. domain\username

   **Important**
   The Checksums that are generated are case specific. For example, the checksum for Test1 (6S82IPn/4eVRxAhWRRyMJWup/wg=) is different from the checksum of test1 (1tTKttMgl+NS/74V602gUw5IKCA=). The case of the username in the SOAP message must match the case you used to generate the checksum.

4. Click the “Generate Checksums” button.

5. The resulting checksum will be displayed in the “BPC User Checksum” textbox on the right.
5.1.2 Creating Checksums for a User Range

In many cases you will have a range of test user accounts including many users. The “User Range” option was created to facilitate this. To create checksums for a range of users:

1. After the BPC LoadRunner Toolkit has been launched, click on the “User Checksum Generator” tab.

2. Ensure that the “User Range” radio button is checked.

3. Enter the username prefix in the form:
   a. domain\username_before_index

   **Example**

   ```
   td1\bpcUser_
   ```

4. Enter the Username Suffix if one exists.

   **Note**
   This is optional. For example, if your usernames were:
   
   td1\bpcUser_1_test
   td1\bpcUser_2_test
   
   Then the Username Prefix would be “td1\bpcUser_” and the Username Suffix would be “_test”

5. Enter the Start Index
6. Enter the End Index
7. Click the “Generate Checksums” button.
8. The resulting checksum will be displayed in the “BPC User Checksum” textbox on the right.

5.2 Using the Checksums in a LoadRunner Script

As mentioned in the section outline, some SOAP messages that BPC sends contain Username information and a related checksum value. To get LoadRunner scripts for BPC to work with multiple user accounts you must parameterize all username and checksum values, and ensure that each checksum always refers to the username in use in the SOAP message.

This section will describe how to configure this.

1. Copy the content from the BPC User Checksums section to a new text file and save it in your LoadRunner script directory with the extension “.dat” (the default extension for a parameter file).
2. Replace all instances of the originally recorded username with a new parameter named "{Username}" or something similar. This can be found inside the “UserID” element of most SOAP calls.
   a. Text Before
   
   "<UserID>&lt;![CDATA[&lt;Username&gt;]]&gt;&lt;/UserID&gt;"
   
   b. Replace All

   ![Screenshot of Search and Replace dialog]

   c. Text After
   
   
   &lt;UserID&gt;&lt;![CDATA[{{Username}}]]&gt;&lt;/UserID&gt;

3. Replace all instances of the originally recorded checksum with a new parameter named "{Checksum}" or something similar. This can be found inside the “Checksum” element of most SOAP calls.
   a. Text Before
   
   &lt;Checksum&gt;&lt;![CDATA[&lt;Checksum&gt;]]&gt;&lt;/Checksum&gt;
   
   b. Replace All

   ![Screenshot of Search and Replace dialog]

   c. Text After
   
   &lt;Checksum&gt;&lt;![CDATA[{{Checksum}}]]&gt;&lt;/Checksum&gt;

4. Open LoadRunners Parameter List dialog (ctrl + L or Vuser -> Parameter List)
5. Create a new parameter named “Username” and configure it with the appropriate row selection criteria. Specify the “dat” file created in step 1 as the “File Path”.

Note
If unsure use “Unique” or “Random” for “Select next row” and “Once” for “Update Value On”.

CAUTION
Ensure the parameter names match the case exactly of the parameters added to the script in steps 2 and 3 as “C” is case sensitive.

6. Create a new parameter named “Checksum” and configure it with the appropriate row selection criteria. Specify the “dat” file created in step 1 as the “File Path”.

Note
The critical configuration for “Checksum” is to set the “Select next row” option to “Same as Username”.

CAUTION
6. **BPC User Context Generator Component**

BPC 7.0 for SAP NetWeaver sends an encrypted copy of the context in the body of some SOAP messages (EvDRE queries, for example). The context includes current view information as well as user information. If an attempt is made to execute a LoadRunner script with a user other than the recording user, and the encrypted context has not been updated, then the .NET server will return an authentication failure message.

This component allows you to create encrypted contexts for a range of users, based on the context recorded during the initial scripting phase.

**Important**

This component is only applicable for the BPC 7.0 for SAP NetWeaver release.
6.1 Creating User Contexts

BPC user contexts are created with the User Context Generator component of the BPC Toolkit for LoadRunner.

6.1.1 Creating Contexts for a Single User

1. After the BPC LoadRunner Toolkit has been launched, click on the “User Context Generator” tab.

2. Select the “Single User” radio button and enter the username in the form “domain\username”.

3. (Optional) Enter an application set name if you wish to change that in the encrypted context as well.

   ![Note](Image)

   This option was added to facilitate migrating scripts to a different appset.

4. Copy the strContext element from one of the SOAP calls and paste it in the “Source strContext” text box.

   ![Tip](Image)

   The “strContext” content can be located by searching for “<strContext> <![CDATA[" in your LoadRunner script. Once located, copy the text between “<strContext> <![CDATA[“ and “"]></strContext>“.

   ![Note](Image)

   The context of strContext can be quite long.
5. Click the “Generate Contexts” button.

6. The resulting context will be displayed in the BPC User Contexts section.

6.1.2 Creating Contexts for a User Range

1. After the BPC LoadRunner Toolkit has been launched, click on the “User Context Generator” tab.

2. Select the “User Range” radio button.

3. Enter the username prefix in the form:
   a. domain\username_before_index

   ![Example](image)

4. Enter the Username Suffix if one exists.

   ![Note](image)

   This is optional. For example, if your usernames were:
   td1\bpcUser_1_test
   td1\bpcUser_2_test

   Then the Username Prefix would be “td1\bpcUser_” and the Username Suffix would be “_test”
5. Enter the Start Index
6. Enter the End Index
7. (Optional) Enter an application set name if you wish to change that in the encrypted context as well.

**Note**
This option was added to facilitate migrating scripts to a different appset.

8. Copy the strContext element from one of the SOAP calls and paste it in the “Source strContext” text box.

**Tip**
The “strContext” content can be located by searching for “<![CDATA[“ in your LoadRunner script. Once located, copy the text between “<![CDATA[“ and ”]]>”.

**Note**
The context of strContext can be quite long.

9. Click the “Generate Contexts” button.

10. The resulting contexts will be displayed in the BPC User Contexts section.

---

### 6.2 Using the Contexts in a LoadRunner Script

As mentioned in the section outline, some SOAP messages that BPC sends contain Username information in an encrypted context. To get LoadRunner scripts for BPC to work with multiple user
accounts you must parameterize all username and encrypted context values, and ensure that each encrypted context always refers to the username in use in the SOAP message.

This section will describe how to configure this.

1. Copy the content from the BPC User Contexts section to a new text file and save it in your LoadRunner script directory with the extension ".dat" (the default extension for a parameter file).

   ![Image](image.png)

   Note
   
   In this example the parameter file is named “user_contexts.dat”

2. Replace all instances of the originally recorded username with a new parameter named “{Username}”, or something similar. This can be found inside the “UserID” element of most SOAP calls.
   
   a. Text Before
   
   `<UserID> <![CDATA[ \%d\%bpcUser ]]></UserID>`
   
   b. Replace All

   ![Search and Replace](image.png)

   c. Text After

   `<UserID> <![CDATA[ {Username} ]]></UserID>`

3. Replace all instances of the originally recorded encrypted context with a parameter named “{Context}”, or something similar. This can be found within the CDATA section of the “strContext” element in many, but not all, SOAP calls.
   
   a. Text Before (note – context data is much longer than what is displayed below)

   `<strContext> <![CDATA[ H9xgHuy2iisCqHKM0KX ]]></strContext>`
   
   b. Replace All
4. Open LoadRunners Parameter List dialog (ctrl + L or Vuser -> Parameter List)
5. Create a new parameter named “Username” and configure it with the appropriate row selection criteria. Specify the “dat” file created in step 1 as the “File Path”.

**Note**
If unsure use “Unique” or “Random” for “Select next row” and “Once” for “Update Value On”.
CAUTION

Ensure the parameter names match the case exactly of the parameters added to the script in steps 2 and 3 as “C” is case sensitive.

6. Create a new parameter named “Context” and configure it with the appropriate row selection criteria. Specify the “dat” file created in step 1 as the “File Path”.

Note

The critical configuration for “Context” is to set the “Select next row” option to “Same as Username”.