

# Testing PowerServer Web applications

Appeon® PowerServer® 2020  
FOR WINDOWS & UNIX & LINUX

DOCUMENT ID: ADC40000-01-2020-01

LAST REVISED: March 25, 2020

Copyright © 2020 Appeon. All rights reserved.

This publication pertains to Appeon software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Appeon Inc.

Appeon, the Appeon logo, Appeon PowerBuilder, Appeon PowerServer, PowerServer, PowerServer Toolkit, AEM, and PowerServer Web Component are trademarks of Appeon Inc.

SAP, Sybase, Adaptive Server Anywhere, SQL Anywhere, Adaptive Server Enterprise, iAnywhere, Sybase Central, and Sybase jConnect for JDBC are trademarks or registered trademarks of SAP and SAP affiliate company.

Java and JDBC are trademarks or registered trademarks of Sun Microsystems, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Appeon Inc., 1/F, Shell Industrial Building, 12 Lee Chung Street, Chai Wan District, Hong Kong.

# Contents

1	Testing with UFT .....	1
1.1	Overview .....	1
1.2	Configuring AEM .....	1
1.3	Configuring UFT .....	1
1.3.1	Selecting Web add-in .....	1
1.3.2	Configuring options .....	2
1.3.3	Configuring object identification .....	5
1.4	Considerations for Recording Web Applications .....	12
1.4.1	Mouse scroll ball .....	12
1.4.2	Keeping data consistent .....	12
1.4.3	Recording TreeView DataWindow .....	12
1.4.4	Recording DatePicker .....	12
1.4.5	Recording Menu .....	12
1.5	Known Issues .....	13
1.5.1	Failing to record operations on SDI windows in Internet Explorer 8.0 .....	13
1.5.2	Generating scripts in an incorrect order .....	13
1.5.3	Failing to run tests .....	13
1.6	Appendix .....	13
2	Testing with AscentialTest .....	14
3	Testing with LoadRunner .....	15
3.1	Overview .....	15
3.2	General Limitations on Performance Testing .....	15
3.3	Testing Steps .....	15
3.3.1	Configuring AEM .....	15
3.3.2	Data Preparation (for update only) .....	16
3.3.3	Preparing Test Cases .....	16
3.3.4	Recording Scripts .....	16
3.3.5	Modifying Scripts .....	23
3.3.5.1	Parameterize scripts of SessionID .....	23
3.3.5.2	Parameterize request header .....	24
3.3.5.3	Parameterization of DataWindow SQL statements .....	26
3.3.5.4	Parameterization of Embedded SQL statements .....	28
3.3.5.5	Additional steps for the Update operation .....	29
3.3.6	Playing back scripts to test the correctness of scripts .....	29
3.3.7	Setting Scenarios .....	29
3.3.7.1	Additional steps for the Update operation .....	30
3.3.8	Running Scenarios .....	30
3.4	Troubleshooting .....	30
3.4.1	"The session does not exist" error .....	30
3.4.2	Errors appear when playing back scripts with LoadRunner 8.0 .....	31
3.4.3	The value of sessionID is null .....	31
3.4.4	Error message appears in script playback .....	32
3.4.5	Error message in Apeon Log .....	33
3.4.6	Failed to parameterize scripts .....	33
3.4.7	Out of memory error and the application server shut down .....	35

3.4.8 Field values do not change after parameterization and playback .....	35
3.4.9 Runtime errors causing scenario failure .....	35
3.4.10 Transactions failed .....	36
3.4.11 Unable to connect to remote servers .....	36

# 1 Testing with UFT

## 1.1 Overview

You can automate the testing of PowerServer Web application by using UFT (Unified Functional Testing), formally called QTP (QuickTest Professional), one of the most popular automatic testing tools. This guide will walk you through how to configure UFT. It also introduces several considerations and known issues for recording Web applications with UFT. To make sure that you can follow this guide easily, you would need to know how to use UFT beforehand. Please read the UFT user manual first if you do not know the basis of UFT.

You will find step-by-step instructions for the following tasks in this user guide:

- [Configuring AEM](#)
- [Configuring UFT](#)
- [Considerations for Recording Web Applications](#)
- [Known Issues](#)
- [Appendix](#)

## 1.2 Configuring AEM

To test a Web application converted by Appeon PowerServer, go to AEM and switch the **Run Mode** to **Test Mode** (by default, **Normal Mode** is selected).

For more information, please refer to Section 4.4.6.4, “Run Mode” in *PowerServer Configuration Guide for .NET* or in *PowerServer Configuration Guide for J2EE*.

**Figure 1.1: Configuring AEM**

Run Mode Settings
Select the run mode for Web applications. If test mode has been turned on, Appeon Server will start in the performance test mode, allowing stress testing tools (such as LoadRunner) to read and write parameters in the plain text.
Run Mode: <input type="radio"/> Normal Mode <input checked="" type="radio"/> Test Mode

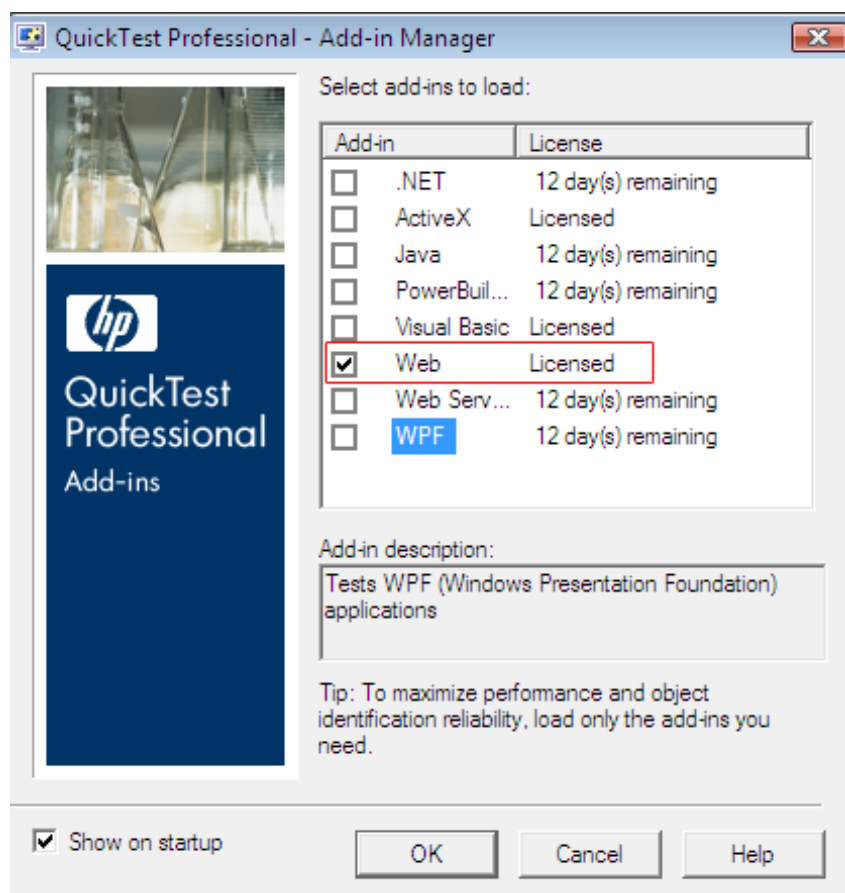
## 1.3 Configuring UFT

Before you can use UFT to record the operations on the controls or functionalities of a Web application, you will need to configure UFT properly.

### 1.3.1 Selecting Web add-in

When you start QuickTest, the **Add-in Manager** dialog box opens. In the Add-in Manager, select the **Web** add-in, and then click **OK** to save the settings and close Add-in Manager.

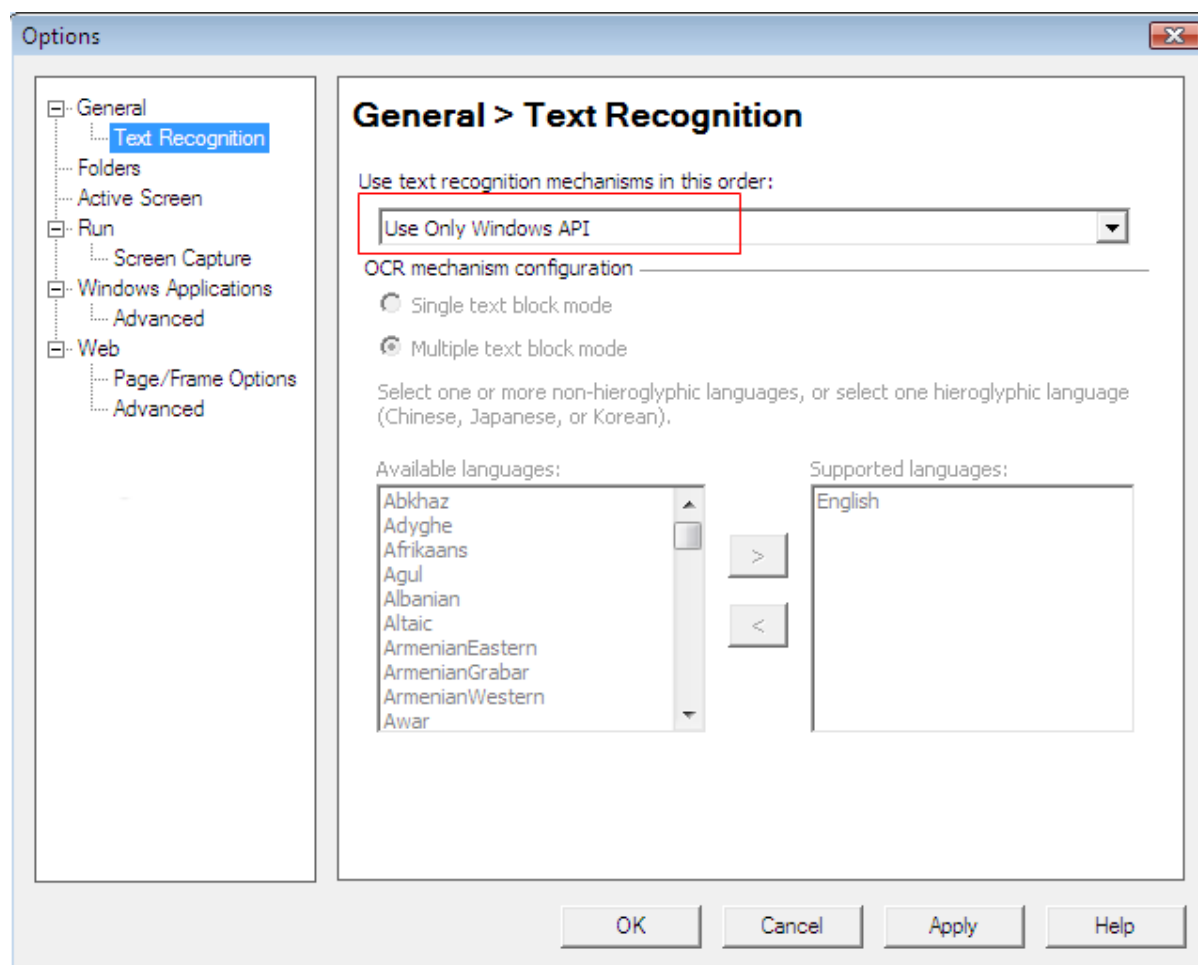
Note: If the Add-in Manager does not open when starting QuickTest, choose **Tools > Options**, click **General** and select **Display Add-in Manager on startup**. When you exit and restart QuickTest, the **Add-in Manager** opens.

**Figure 1.2: Selecting add-ins**

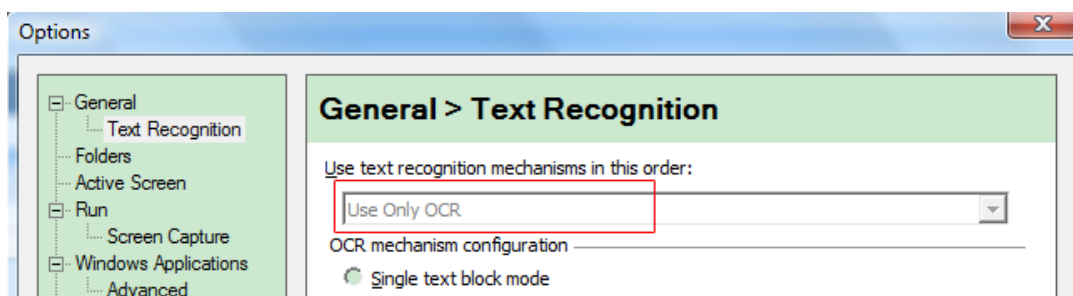
### 1.3.2 Configuring options

Step 1: Choose **Tools > Options**. The **Options** dialog box opens.

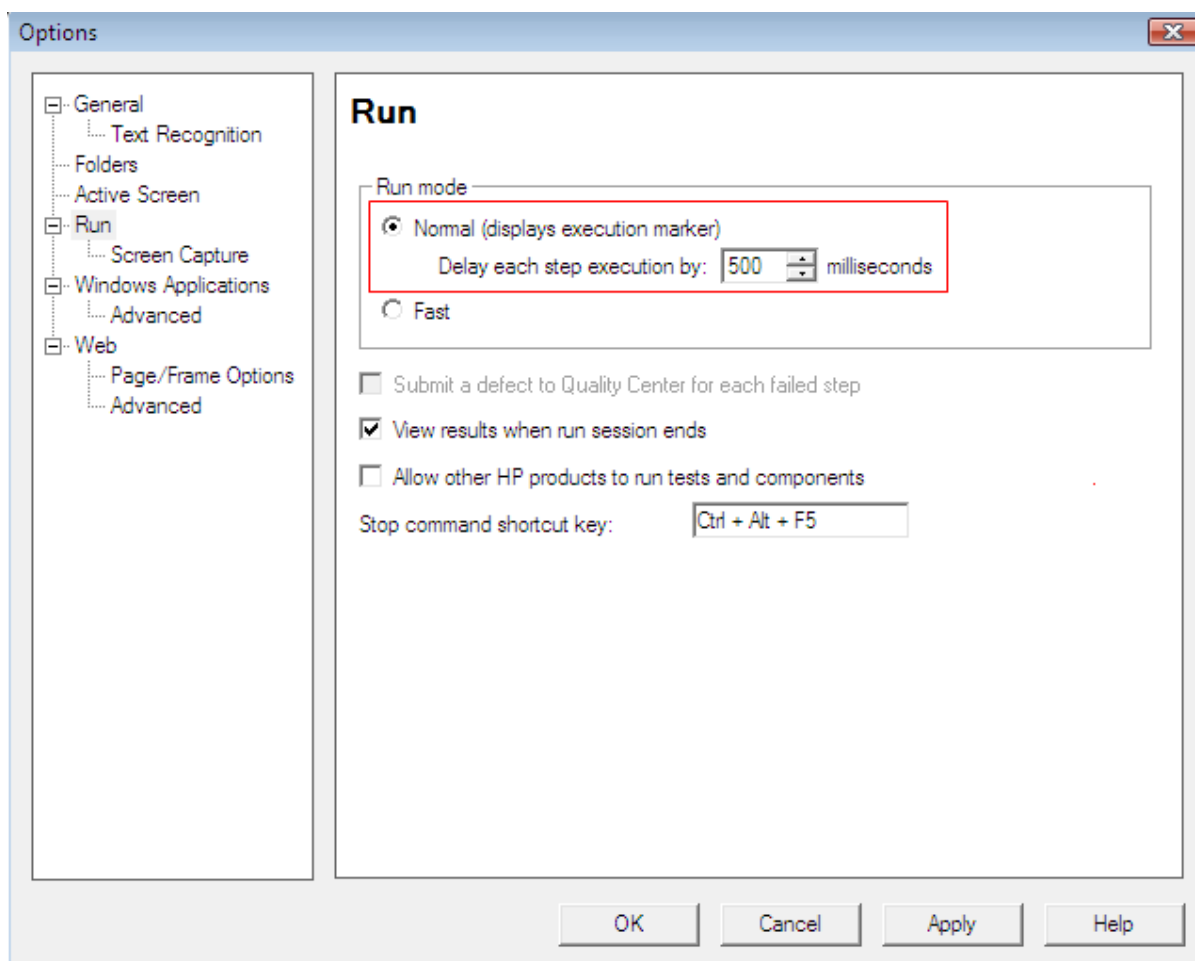
Step 2: In the left pane, click **General**, and in the right pane, select **Use Only Windows API** from the **Use text recognition mechanisms in this order** list.

**Figure 1.3: Text Recognition****Notes:**

1. If there is unrecognized script obtained and the checkpoint is not an image when you insert a Text checkpoint, check the configuration here. Make sure the **Use Only Window API** is selected in the **Use text recognition mechanism in this order** dropdown list box.
2. Windows 7 does not support the Windows API test recognition mechanism. Therefore, when you install relevant QTP 10.00 patches, the **Use text recognition mechanisms in this order** option in the **Options** dialog box (**Options > General > Text Recognition**) is set to **Use only OCR**, and cannot be changed (the option is disabled), as shown in the figure below. However, you can still test a Web application on Windows 7, the only problem you might have is the text that QTP obtain may contain some unrecognizable characters. When this happens, you can manually modify these unrecognizable characters. For more information, refer to the QTP 10.0 user guide.

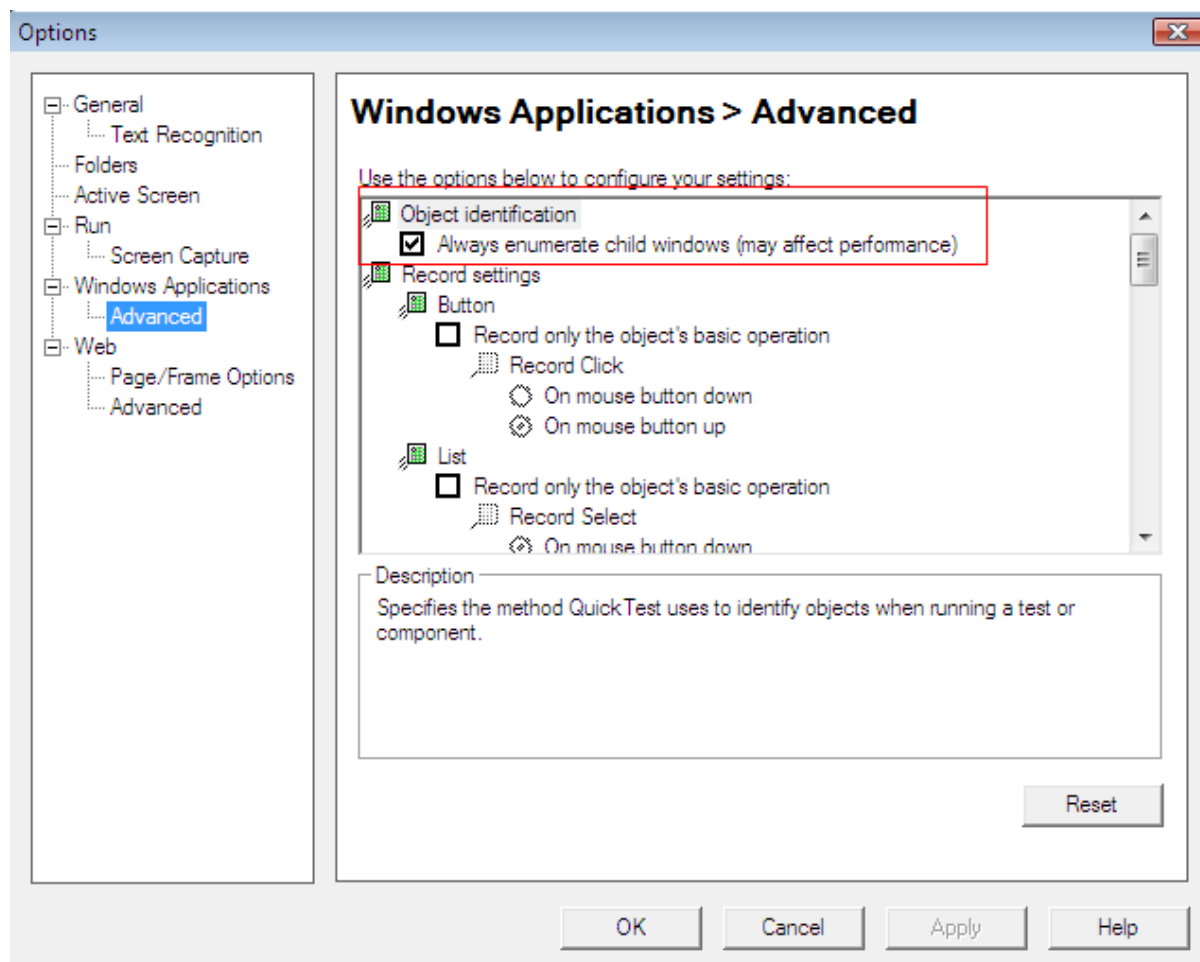
**Figure 1.4: View Text Recognition window**

Step 3: Click **Run** in the left pane. In the right pane, select **Normal (displays execution marker)**, and then type 500 or select 500 using the spin control in the **Delay each step execution by [ ] milliseconds** spin control, as shown in the figure below.

**Figure 1.5: Choose the Run mode**

Step 4: In the left pane, expand **Windows Applications** and click **Advanced**. In the right pane, select **Always enumerate child windows (may affect performance)** option under **Object identification**.



**Figure 1.6: Set the object identification**

Step 5: Click **OK** or **Apply** to save the settings.

### 1.3.3 Configuring object identification

Copy the following scripts to the **Expert View** in UFT and click **Run**.

This is to map the PowerServer Web application objects with standard UFT objects. And it is an "import once, use forever" operation. You just need to copy the following scripts to your UFT **Expert View** for the very first time when you begin to record PowerServer Web applications.

```
Dim App 'As Application
Dim intPosition
Set App = CreateObject("QuickTest.Application")
App.Launch
App.Visible = True

'Configuration of Standard Windows objects
'Object identification configuration for user-defined object "WinEdit"\
intPosition =
App.Options.ObjectIdentification("WinEdit").MandatoryProperties.Find("attached
text")
If intPosition <> -1 Then
App.Options.ObjectIdentification("WinEdit").MandatoryProperties.Remove intPosition
End If

'Object identification configuration for user-defined object "WinEditor"
```

```

intPosition =
  App.Options.ObjectIdentification("WinEditor").MandatoryProperties.Find("attached
  text")
If intPosition <> -1 Then
  App.Options.ObjectIdentification("WinEditor").MandatoryProperties.Remove
  intPosition
End If

'Object identification configuration for test object object "browser"
App.Options.ObjectIdentification("browser").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("browser").AssistiveProperties.RemoveAll()

'Configuration of user-defined objects

'Object identification configuration for user-defined object "eoncheckbox"
App.Options.ObjectIdentification("WinCheckBox").CreateUserDefinedObject("eoncheckbox")
App.Options.ObjectIdentification("eoncheckbox").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eoncheckbox").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eoncheckbox").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eoncheckbox").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eoncheckbox").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eoncheckbox").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eoncheckbox").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eoncheckbox").EnableSmartIdentification = False
App.Options.ObjectIdentification("eoncheckbox").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eoncheckbox").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eoncmbutton"
App.Options.ObjectIdentification("WinButton").CreateUserDefinedObject("eoncmbutton")
App.Options.ObjectIdentification("eoncmbutton").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eoncmbutton").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eoncmbutton").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eoncmbutton").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eoncmbutton").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eoncmbutton").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eoncmbutton").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eoncmbutton").EnableSmartIdentification = False
App.Options.ObjectIdentification("eoncmbutton").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eoncmbutton").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eoncombobox"
App.Options.ObjectIdentification("WinComboBox").CreateUserDefinedObject("eoncombobox")
App.Options.ObjectIdentification("eoncombobox").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eoncombobox").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eoncombobox").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eoncombobox").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eoncombobox").AssistiveProperties.Add("window
id")

'App.Options.ObjectIdentification("eoncombobox").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eoncombobox").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eoncombobox").EnableSmartIdentification = False
App.Options.ObjectIdentification("eoncombobox").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eoncombobox").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eondatawindow"
App.Options.ObjectIdentification("WinObject").CreateUserDefinedObject("eondatawindow")
App.Options.ObjectIdentification("eondatawindow").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eondatawindow").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eondatawindow").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eondatawindow").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eondatawindow").AssistiveProperties.Add("window
id")

```

```
'App.Options.ObjectIdentification("eondatawindow").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eondatawindow").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eondatawindow").EnableSmartIdentification = False
App.Options.ObjectIdentification("eondatawindow").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eondatawindow").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eondatepicker"
App.Options.ObjectIdentification("WinCalendar").CreateUserDefinedObject("eondatepicker")
App.Options.ObjectIdentification("eondatepicker").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eondatepicker").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eondatepicker").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eondatepicker").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eondatepicker").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eondatepicker").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eondatepicker").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eondatepicker").EnableSmartIdentification = False
App.Options.ObjectIdentification("eondatepicker").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eondatepicker").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eondwedit"
App.Options.ObjectIdentification("WinEditor").CreateUserDefinedObject("eondwedit")
App.Options.ObjectIdentification("eondwedit").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eondwedit").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eondwedit").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eondwedit").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eondwedit").AssistiveProperties.Add("window id")
'App.Options.ObjectIdentification("eondwedit").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eondwedit").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eondwedit").EnableSmartIdentification = False
App.Options.ObjectIdentification("eondwedit").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eondwedit").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eondweditmask"
App.Options.ObjectIdentification("WinEditor").CreateUserDefinedObject("eondweditmask")
App.Options.ObjectIdentification("eondweditmask").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eondweditmask").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eondweditmask").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eondweditmask").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eondweditmask").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eondweditmask").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eondweditmask").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eondweditmask").EnableSmartIdentification = False
App.Options.ObjectIdentification("eondweditmask").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eondweditmask").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eondwlistbox"
App.Options.ObjectIdentification("WinList").CreateUserDefinedObject("eondwlistbox")
App.Options.ObjectIdentification("eondwlistbox").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eondwlistbox").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eondwlistbox").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eondwlistbox").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eondwlistbox").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eondwlistbox").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eondwlistbox").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eondwlistbox").EnableSmartIdentification = False
App.Options.ObjectIdentification("eondwlistbox").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eondwlistbox").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eondwstatic"
App.Options.ObjectIdentification("Static").CreateUserDefinedObject("eondwstatic")
App.Options.ObjectIdentification("eondwstatic").OrdinalIdentifier = "location"
```

```
App.Options.ObjectIdentification("eondwstatic").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eondwstatic").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eondwstatic").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eondwstatic").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eondwstatic").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eondwstatic").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eondwstatic").EnableSmartIdentification = False
App.Options.ObjectIdentification("eondwstatic").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eondwstatic").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eoneditmask"
App.Options.ObjectIdentification("WinEdit").CreateUserDefinedObject("eoneditmask")
App.Options.ObjectIdentification("eoneditmask").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eoneditmask").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eoneditmask").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eoneditmask").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eoneditmask").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eoneditmask").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eoneditmask").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eoneditmask").EnableSmartIdentification = False
App.Options.ObjectIdentification("eoneditmask").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eoneditmask").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonhscrollbar"
App.Options.ObjectIdentification("WinScrollBar").CreateUserDefinedObject("eonhscrollbar")
App.Options.ObjectIdentification("eonhscrollbar").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eonhscrollbar").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eonhscrollbar").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonhscrollbar").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonhscrollbar").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eonhscrollbar").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eonhscrollbar").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonhscrollbar").EnableSmartIdentification = False
App.Options.ObjectIdentification("eonhscrollbar").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eonhscrollbar").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonlabeledit"
App.Options.ObjectIdentification("WinEdit").CreateUserDefinedObject("eonlabeledit")
App.Options.ObjectIdentification("eonlabeledit").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eonlabeledit").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eonlabeledit").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonlabeledit").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonlabeledit").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eonlabeledit").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eonlabeledit").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonlabeledit").EnableSmartIdentification = False
App.Options.ObjectIdentification("eonlabeledit").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eonlabeledit").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonlistbox"
App.Options.ObjectIdentification("WinList").CreateUserDefinedObject("eonlistbox")
App.Options.ObjectIdentification("eonlistbox").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eonlistbox").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eonlistbox").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonlistbox").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonlistbox").AssistiveProperties.Add("window id")
'App.Options.ObjectIdentification("eonlistbox").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eonlistbox").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonlistbox").EnableSmartIdentification = False
App.Options.ObjectIdentification("eonlistbox").BaseFilterProperties.RemoveAll()
```

```
App.Options.ObjectIdentification("eonlistbox").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonlistview"
App.Options.ObjectIdentification("WinListView").CreateUserDefinedObject("eonlistview")
App.Options.ObjectIdentification("eonlistview").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eonlistview").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eonlistview").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonlistview").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonlistview").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eonlistview").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eonlistview").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonlistview").EnableSmartIdentification = False
App.Options.ObjectIdentification("eonlistview").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eonlistview").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonmonthcalendar"
App.Options.ObjectIdentification("WinCalendar").CreateUserDefinedObject("eonmonthcalendar")
App.Options.ObjectIdentification("eonmonthcalendar").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eonmonthcalendar").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eonmonthcalendar").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonmonthcalendar").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonmonthcalendar").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eonmonthcalendar").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eonmonthcalendar").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonmonthcalendar").EnableSmartIdentification =
False
App.Options.ObjectIdentification("eonmonthcalendar").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eonmonthcalendar").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonmultilinedit"
App.Options.ObjectIdentification("WinEditor").CreateUserDefinedObject("eonmultilinedit")
App.Options.ObjectIdentification("eonmultilinedit").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eonmultilinedit").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eonmultilinedit").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonmultilinedit").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonmultilinedit").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eonmultilinedit").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eonmultilinedit").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonmultilinedit").EnableSmartIdentification =
False
App.Options.ObjectIdentification("eonmultilinedit").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eonmultilinedit").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonpicbutton"
App.Options.ObjectIdentification("WinButton").CreateUserDefinedObject("eonpicbutton")
App.Options.ObjectIdentification("eonpicbutton").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eonpicbutton").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eonpicbutton").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonpicbutton").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonpicbutton").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eonpicbutton").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eonpicbutton").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonpicbutton").EnableSmartIdentification = False
App.Options.ObjectIdentification("eonpicbutton").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eonpicbutton").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonpiccombobox"
App.Options.ObjectIdentification("WinComboBox").CreateUserDefinedObject("eonpiccombobox")
App.Options.ObjectIdentification("eonpiccombobox").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eonpiccombobox").MandatoryProperties.RemoveAll()
```

```
App.Options.ObjectIdentification("eonpiccombobox").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonpiccombobox").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonpiccombobox").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eonpiccombobox").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eonpiccombobox").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonpiccombobox").EnableSmartIdentification =
False
App.Options.ObjectIdentification("eonpiccombobox").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eonpiccombobox").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonpiclistbox"
App.Options.ObjectIdentification("WinList").CreateUserDefinedObject("eonpiclistbox")
App.Options.ObjectIdentification("eonpiclistbox").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eonpiclistbox").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eonpiclistbox").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonpiclistbox").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonpiclistbox").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eonpiclistbox").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eonpiclistbox").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonpiclistbox").EnableSmartIdentification = False
App.Options.ObjectIdentification("eonpiclistbox").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eonpiclistbox").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonradiobutton"
App.Options.ObjectIdentification("WinRadioButton").CreateUserDefinedObject("eonradiobutton")
App.Options.ObjectIdentification("eonradiobutton").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eonradiobutton").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eonradiobutton").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonradiobutton").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonradiobutton").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eonradiobutton").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eonradiobutton").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonradiobutton").EnableSmartIdentification =
False
App.Options.ObjectIdentification("eonradiobutton").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eonradiobutton").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonrichedit"
App.Options.ObjectIdentification("WinEditor").CreateUserDefinedObject("eonrichedit")
App.Options.ObjectIdentification("eonrichedit").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eonrichedit").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eonrichedit").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonrichedit").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonrichedit").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eonrichedit").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eonrichedit").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonrichedit").EnableSmartIdentification = False
App.Options.ObjectIdentification("eonrichedit").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eonrichedit").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonsinglelinedit"
App.Options.ObjectIdentification("WinEdit").CreateUserDefinedObject("eonsinglelinedit")
App.Options.ObjectIdentification("eonsinglelinedit").OrdinalIdentifier =
"location"
App.Options.ObjectIdentification("eonsinglelinedit").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eonsinglelinedit").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonsinglelinedit").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonsinglelinedit").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eonsinglelinedit").AssistiveProperties.Add("x")
```

```
'App.Options.ObjectIdentification("eonsinglelineedit").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonsinglelineedit").EnableSmartIdentification =
  False
App.Options.ObjectIdentification("eonsinglelineedit").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eonsinglelineedit").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonstatic"
App.Options.ObjectIdentification("Static").CreateUserDefinedObject("eonstatic")
App.Options.ObjectIdentification("eonstatic").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eonstatic").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eonstatic").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonstatic").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonstatic").AssistiveProperties.Add("window id")
'App.Options.ObjectIdentification("eonstatic").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eonstatic").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonstatic").EnableSmartIdentification = False
App.Options.ObjectIdentification("eonstatic").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eonstatic").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eontab"
App.Options.ObjectIdentification("WinTab").CreateUserDefinedObject("eontab")
App.Options.ObjectIdentification("eontab").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eontab").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eontab").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eontab").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eontab").AssistiveProperties.Add("window id")
'App.Options.ObjectIdentification("eontab").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eontab").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eontab").EnableSmartIdentification = False
App.Options.ObjectIdentification("eontab").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eontab").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eontreeview"
App.Options.ObjectIdentification("WinTreeView").CreateUserDefinedObject("eontreeview")
App.Options.ObjectIdentification("eontreeview").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eontreeview").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eontreeview").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eontreeview").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eontreeview").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eontreeview").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eontreeview").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eontreeview").EnableSmartIdentification = False
App.Options.ObjectIdentification("eontreeview").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eontreeview").OptionalFilterProperties.RemoveAll()

'Object identification configuration for user-defined object "eonvscrollbar"
App.Options.ObjectIdentification("WinScrollBar").CreateUserDefinedObject("eonvscrollbar")
App.Options.ObjectIdentification("eonvscrollbar").OrdinalIdentifier = "location"
App.Options.ObjectIdentification("eonvscrollbar").MandatoryProperties.RemoveAll()
App.Options.ObjectIdentification("eonvscrollbar").AssistiveProperties.RemoveAll()
App.Options.ObjectIdentification("eonvscrollbar").MandatoryProperties.Add("nativeclass")
App.Options.ObjectIdentification("eonvscrollbar").AssistiveProperties.Add("window
id")
'App.Options.ObjectIdentification("eonvscrollbar").AssistiveProperties.Add("x")
'App.Options.ObjectIdentification("eonvscrollbar").AssistiveProperties.Add("y")
App.Options.ObjectIdentification("eonvscrollbar").EnableSmartIdentification = False
App.Options.ObjectIdentification("eonvscrollbar").BaseFilterProperties.RemoveAll()
App.Options.ObjectIdentification("eonvscrollbar").OptionalFilterProperties.RemoveAll()
```

## 1.4 Considerations for Recording Web Applications

This section only highlights a few areas that you should pay special attention to when you test PowerServer Web applications with UFT. For step-by-step instructions on how to use UFT, please refer to the UFT user guide.

### 1.4.1 Mouse scroll ball

UFT cannot record operations performed by using the mouse scroll ball, therefore avoid using mouse scroll ball to scroll a window when recording the application with UFT.

### 1.4.2 Keeping data consistent

Make sure that the data of running a test need to be the same data when the test is recorded, otherwise it may cause run failures.

### 1.4.3 Recording TreeView DataWindow

The TreeView DataWindow is not a standard Windows TreeView in UFT, therefore when clicking a TreeView node, its coordinates will be recorded by UFT.

### 1.4.4 Recording DatePicker

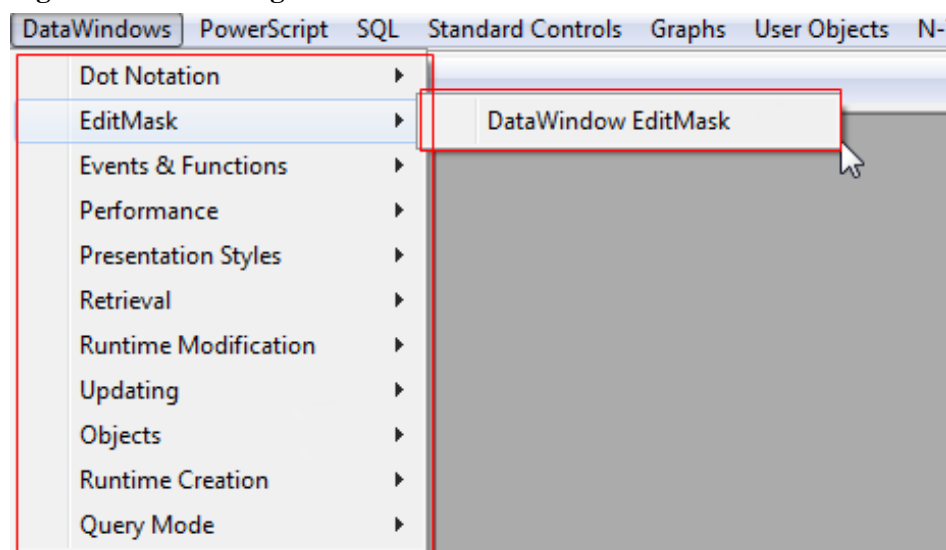
For the DatePicker control or controls of this type (time controls that use x and y coordinates to recognize data), the date (year and month) to run a test need to be the same as the year and month when the test is recorded.

### 1.4.5 Recording Menu

UFT may occasionally record a Menu as a ControlMenu which will cause run failures. When this happens, you can record the operation again, and make sure that you do not click any menu other than the one you intend to record.

Typically, to record a sub-menu, instead of clicking the menu, move the mouse over the menu, for example *DataWindows* and then *EditMask* as shown in the figure below, until the sub-menu you want to record appears, and then click the sub-menu.

**Figure 1.7: Recording Menu**





## 1.5 Known Issues

This section lists the issues one may come across when testing PowerServer Web applications with UFT (or QTP, the former name).

### 1.5.1 Failing to record operations on SDI windows in Internet Explorer 8.0

**Description:** No test script is generated while QTP 10.00 is recording operations on an SDI window (main-type window opened by the PB Open function) in IE 8.0.

**Cause:** It is probably caused by the limited support of QTP 10.00 for IE 8.0.

**Solution:** You can upgrade to QTP 11.00 or UFT 12.5.3 to avoid this problem.

### 1.5.2 Generating scripts in an incorrect order

**Description:** The test scripts are generated in an incorrect order.

**Cause:** This is probably a UFT issue and happens occasionally.

**Solution:** When this happens, you can record the operation again and make sure that the scripts are generated in the correct order; or you can manually change the scripts into the correct order in the Expert View.

### 1.5.3 Failing to run tests

**Description:** Test failed to run in UFT, and errors like "Cannot identify the object ..." occur.

**Cause A & Solution A:** This is caused by the incorrect order in which the record is generated. See [Generating scripts in an incorrect order](#) for details.

**Cause B & Solution B:** Data of running a test is not the same data when the test is recorded. See [Keeping data consistent](#).

## 1.6 Appendix

This section lists the default ID of several objects. They will be used to represent the object in the **Object Repository** and the script. You can refer to the following table and modify the default ID in the **Object Repository** if you find it difficult to understand or remember them.

Control code	Description
[ATL: Eon000202]	MDI Window
[ATL: Eon000203]	MDI Window with help status bar
[ATL: Eon000204]	Popup window
[ATL: Eon000205]	Response window
[ATL: Eon000206]	Child Window
[ATL: Eon000038]	User Object

## 2 Testing with AscentialTest

You can automate the testing of PowerServer Web application by using AscentialTest 9.5.2 or above.

There is no special requirements from PowerServer for how to configure and run AscentialTest. Please visit [this page](#) provided by AscentialTest that explains how automated tests are written for the PowerServer Web application as well as the PowerBuilder application.

There is a requirement for PowerServer before you can test the Web application converted by PowerServer. That is, go to AEM and switch the **Run Mode** to **Test Mode** (by default, **Normal Mode** is selected). For more information, please refer to Section 4.4.6.4, “Run Mode” in *PowerServer Configuration Guide for .NET* or in PowerServer Configuration Guide for J2EE.

**Figure 2.1: Configuring AEM**

Run Mode Settings
Select the run mode for Web applications. If test mode has been turned on, Apeon Server will start in the performance test mode, allowing stress testing tools (such as LoadRunner) to read and write parameters in the plain text.
Run Mode: <input type="radio"/> Normal Mode <input checked="" type="radio"/> Test Mode

## 3 Testing with LoadRunner

### 3.1 Overview

The HP LoadRunner is an automated performance and testing product for examining system behaviors and performance. The HP LoadRunner can simulate hundreds or thousands of concurrent users to put the application through the rigors of real-life user loads, while collecting information from key infrastructure components (Web servers, database servers, etc.). The results can then be analyzed in details, to explore the reasons for a particular behavior.

### 3.2 General Limitations on Performance Testing

1. The Web/HTTP protocol is used;
2. Only Web applications in English can be tested with LoadRunner.

### 3.3 Testing Steps

To put it concisely, performing test of Apeon PowerServer with the HTTP protocol in the test mode is to complete the following steps:

1. Set the record parameters and record the scripts;
2. Change the constants of Session ID in the scripts to variables;
3. If testing the Update operation, you need to parameterize SQL statements (to avoid deadlock of database tables when there are many concurrent access);
4. Configure the Run-Time Settings;
5. Play back the scripts unless the tests succeed, otherwise return to 2), 3), 4);
6. Design and run the scenario according to the testing scheme.

#### 3.3.1 Configuring AEM

1. Select the **Test Mode** as the run mode:

Navigate to **AEM > Application > Client Features > YourApplicationName > Run Mode > Test Mode**, and then click **Save**.

2. Select the **Debug Mode** as the log generation mode:

Navigate to **AEM > Server > Logging > Log Mode > Debug Mode**, and then click **Save**.

After you play back scripts successfully in vugen.exe, change the **Debug Mode** back to **Standard Mode**.

### 3.3.2 Data Preparation (for update only)

This is required only when testing the Update operation.

To simulate large number of concurrent access of Update, sufficient records should be prepared first.

Requirements in the modified table:

1. The Recording Numbers must meet the following condition:

In the table, the number of Records should be greater than the number of (Vusers + 1) times the number of Modified Record/Users ( $\text{Number of Records} > (\text{Vusers} + 1) * \text{Number of Modified Record/Users}$ ).

For example, if there are 10 users concurrently accessing and conducting the DataWindow Update 10r x 7c with the "use update" mode operation, the modified table must at least contain the following number of records:  $(10 + 1) * 10 = 110$ .

2. The initial value of the keyword "id" should be 0 (This is for simplifying the code process for the SQL statement parameterization in scripts).

### 3.3.3 Preparing Test Cases

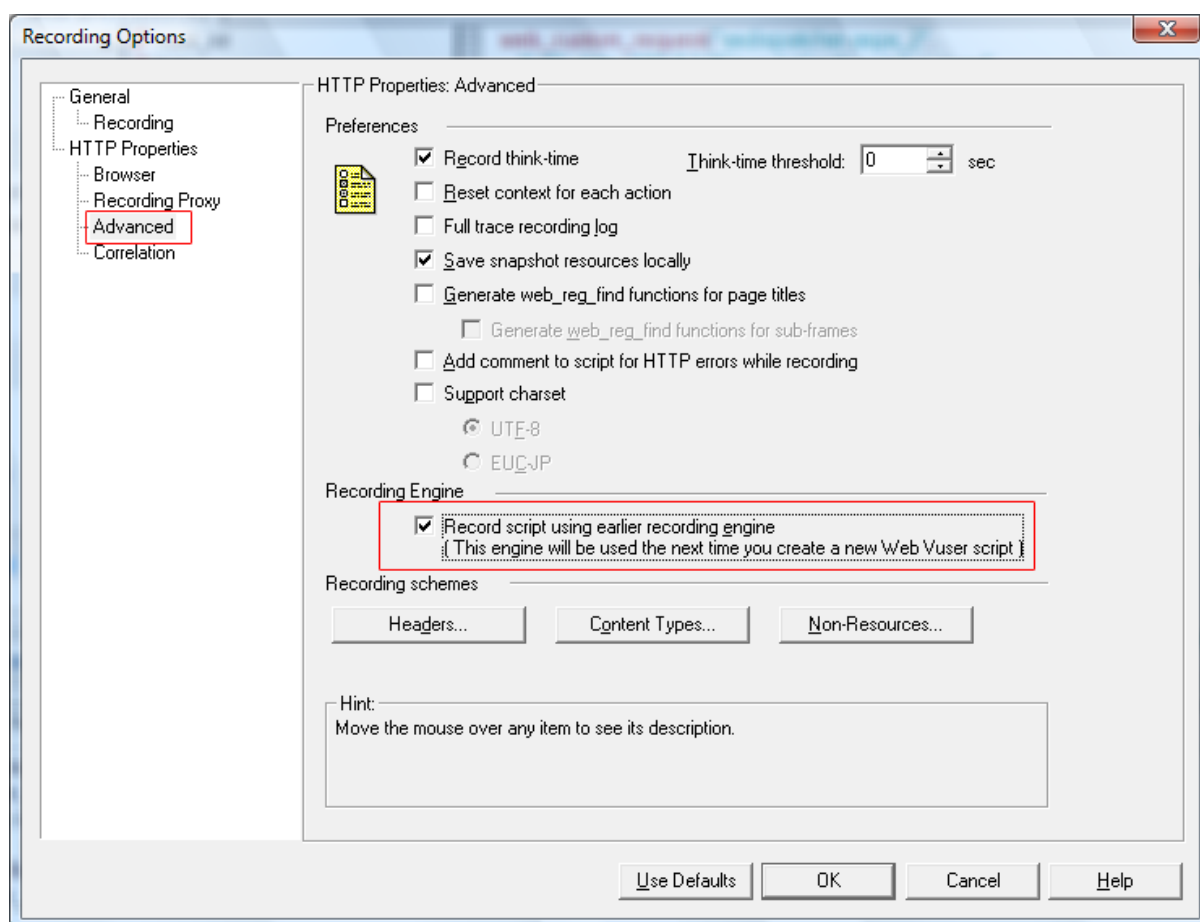
Determine the scripts to be recorded and the actions to be executed in scripts.

### 3.3.4 Recording Scripts

1. Set the record engine: **Record script using earlier recording engine.**

Select from **File > Tools > Recording Options > Advanced.**

Check the **Record script using earlier recording engine** checkbox below **Recording Engine**, and then click **OK** as shown in the following figure.

**Figure 3.1: Recording Options**

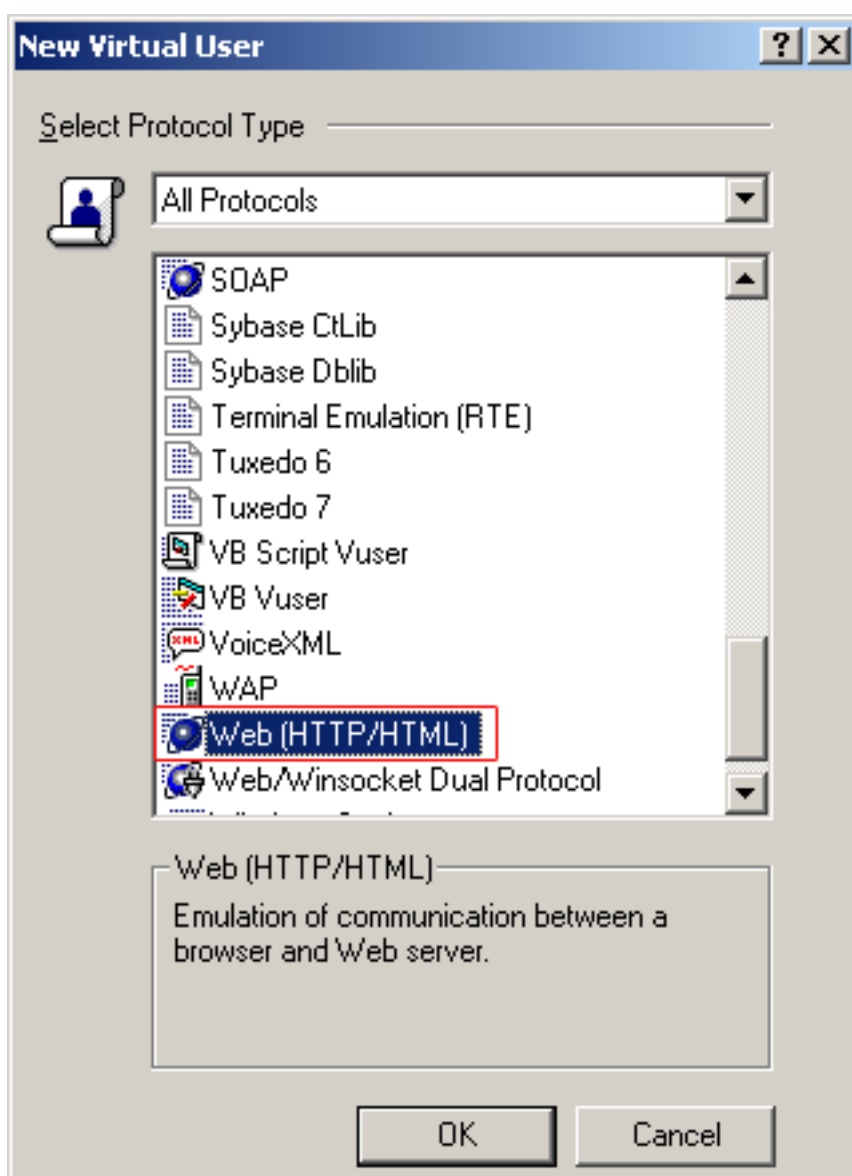
Restart the **LoadRunner**.

**Record script using earlier recording engine** option does not exist in Load Runner 12.53 and later versions. Ignore this setting in such case.

2. Set the protocol for recording: **Web (HTTP/HTML)**.

Select from **File> New**.

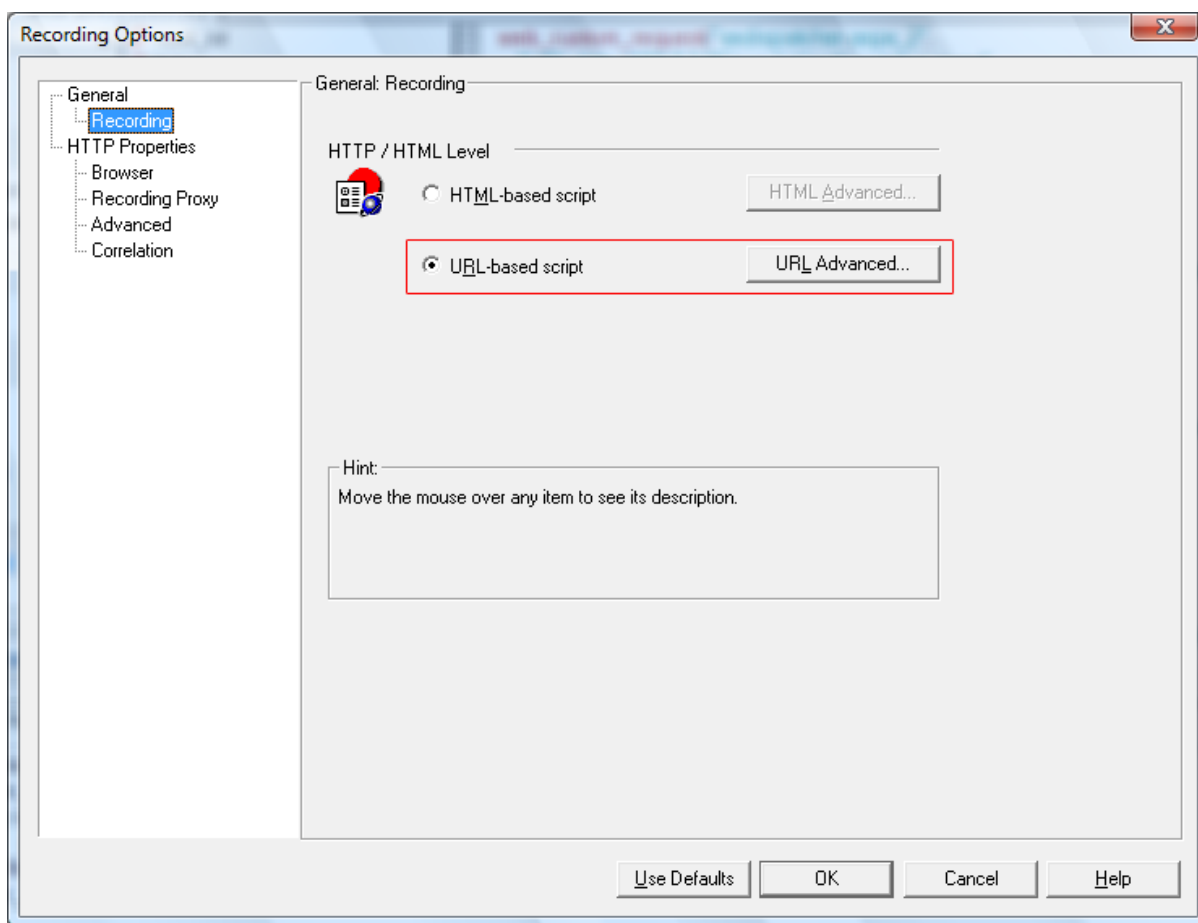
In the **New Virtual User** window, select **Web (HTTP/HTML)**, and then click **OK** as shown in the following figure.

**Figure 3.2: New Virtual User**

3. Set the **Recording** options.

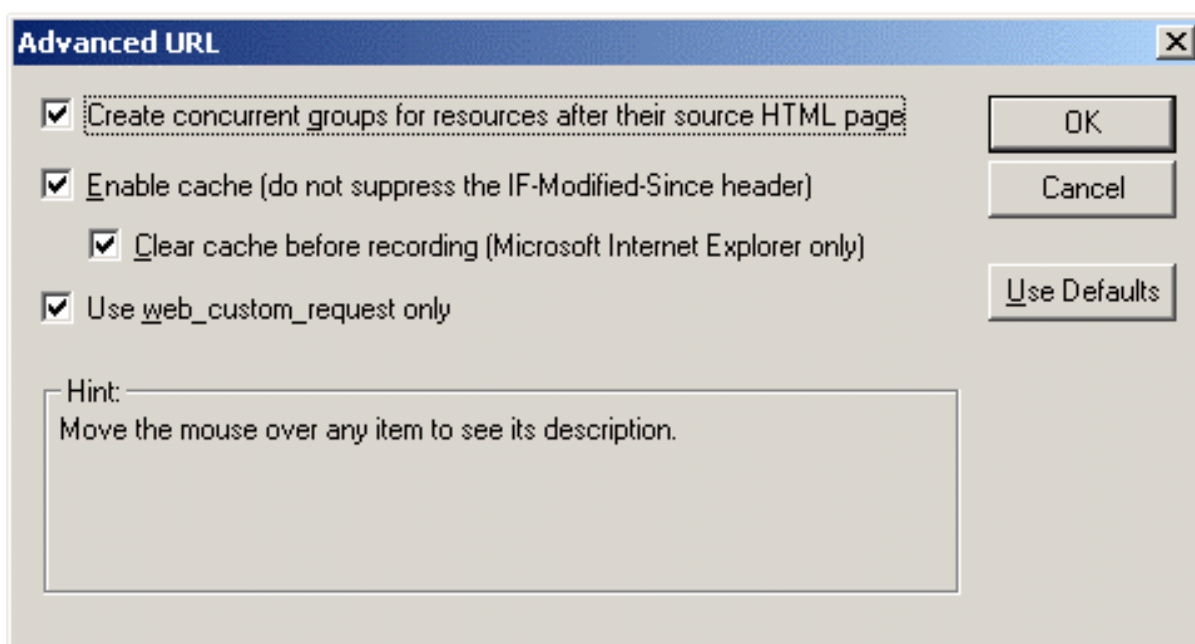
Select from **Tools > Recording Options**.

In the window that appears, select **General > Recording** on the left, and then select the **URL-based scripts** radio button as shown in the following figure.

**Figure 3.3: Recording Options**

Click the **URL Advanced** button in the same window, and the **Advanced URL** window appears.

Check all the checkboxes and click **OK** as shown in the following figure. (The check boxes in this window may vary in different Load Runner versions. For example, in Load Runner 12.53 and later versions, there are only 2 check boxes: **Create concurrent groups for resources after their source HTML page**, and **Use web\_custom\_request only**.)

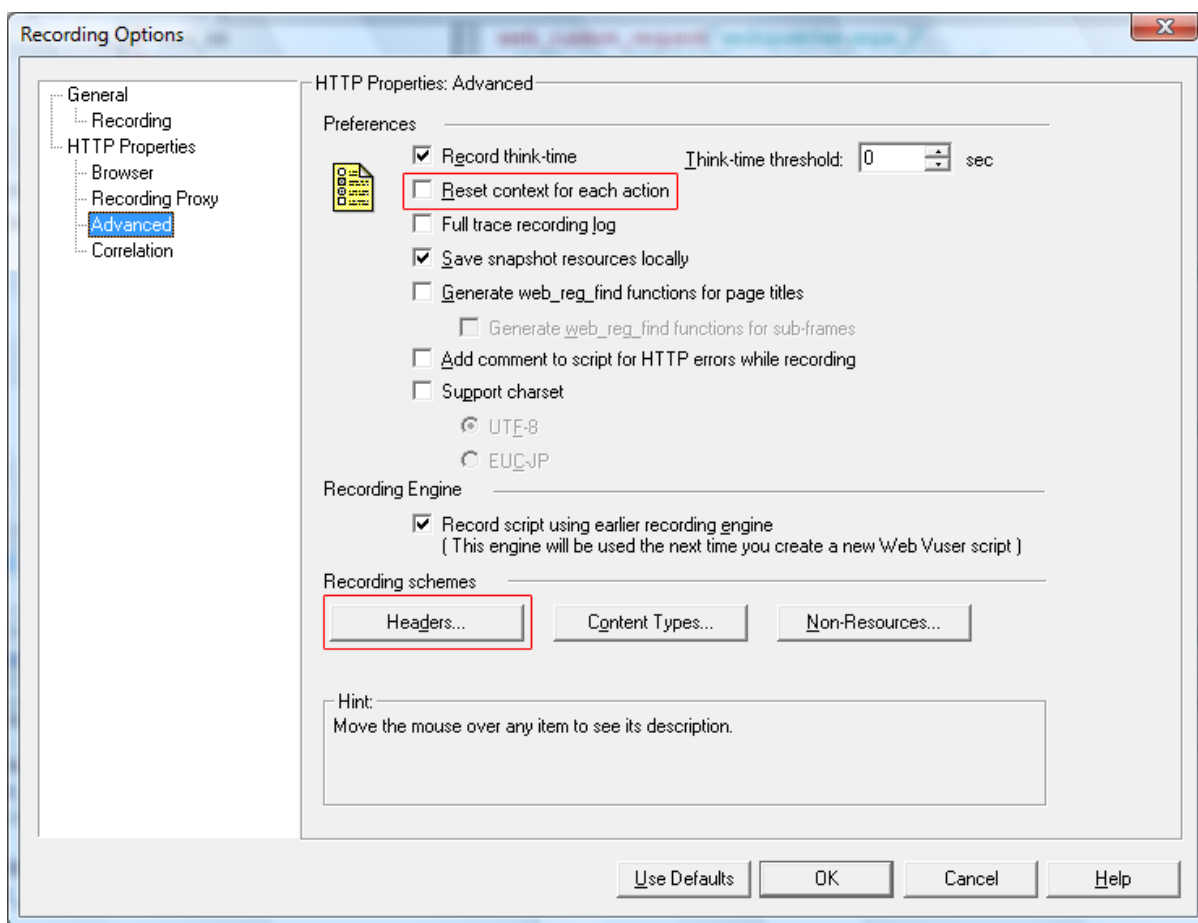
**Figure 3.4: Advanced URL**

4. Set the **Advanced** options.

Select from **Tools > Recording Options**.

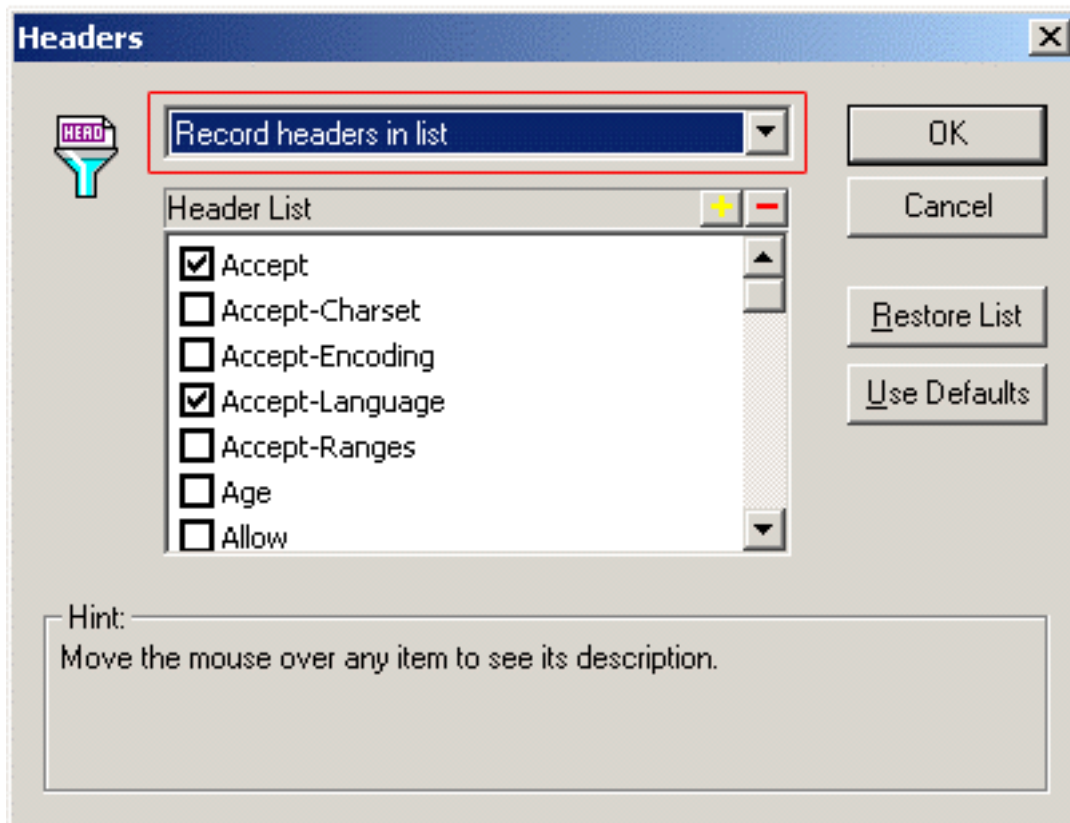
In the window that appears, select **HTTP Properties > Advanced** on the left, and then uncheck the **Reset context for each action** option, as shown in the following figure.



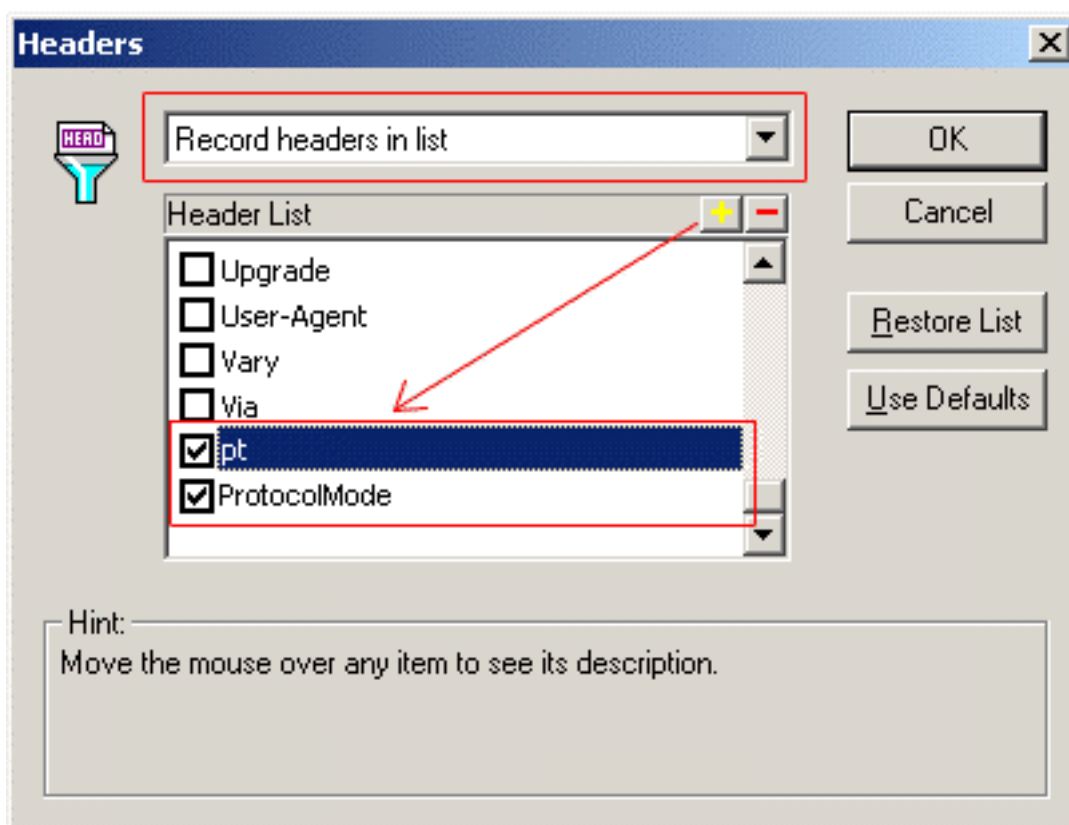
**Figure 3.5: Recording Options**

Click the **Headers** button in the same window.

In the **Headers** window, select the **Record headers in list** from the dropdown list box, and then check the **Accept** and the **Accept-Language** checkboxes, as shown in the following figure.

**Figure 3.6: Headers**

Click the "+" sign to add two headers: **pt** and **ProtocolMode**, then select them in the **Header List**, and then click **OK** as shown in the following figure.

**Figure 3.7: Headers**

### 3.3.5 Modifying Scripts

#### 3.3.5.1 Parameterize scripts of SessionID

Parameterize scripts of SessionID:

1. Use the keyword "servlet" to search in the scripts. Find the function that has the first occurrence of both the "servlet" keyword and the "BodyBinary" parameter, and then add the following script in front of it, so as to capture the current SessionID.

```
web_reg_save_param("SessionID", "LB=ID =", "RB=\r\n", LAST);
```

2. Search all the remaining "servlet" statements, and modify the parameters starting with "Body=" in the following way:

Find the location of "session\_id =", and replace the recorded value of SessionID with the character string "{SessionID}".

Example in PowerServer 2016 and later versions:

Replace

```
"Body=<?xml version=\"1.0\" encoding=\"utf-8\"?><data><header type=\"193\" version=\"35\" session_id=\"2011743809\" />
```

with

```
"Body=<?xml version=\"1.0\" encoding=\"utf-8\"?><data><header type=\"193\" version=\"35\" session_id=\"{SessionID}\" />
```

Example in PowerServer 2015 and earlier versions:

Replace

```
"session_id = -318005152\r\n"
```

with

```
"session_id = {SessionID}\r\n"
```

The following block (*{SessionID}* in "session\_id = {SessionID}\r\n") is an example after changed.

```
web_custom_request("apeon.weblibrary.AXDispatcher_6",
"URL=http://172.17.1.108:8080/servlet/apeon.weblibrary.AXDispatcher",
"Method=POST",
"Resource=1",
"Referer=",
"EncType=",
"Body="
"\x1F\x8B\x08\x00\x00\x00\x00\x00\x00\x0B"
"ST"
"\x16"
"b0a0"
"\x06"
"b"
"\x0B\x06"
"K"
"\x06"
"C"
"\x06"
"3"
"\x06"
"#"
"\x06"
"S"
"\x06"
"1"
"\x86\x00"
"†T"
"\x86"
"\"
"\x86"
"4†|™Ë#È#Ç"
"\x90\x0C\x14\x01\x01"
"1 fb"
"\x00\x00\x12\x02"
"½#8"
"\x00\x00\x00"
"-----apeon data\r\n"
"Name = Kill Session\r\n"
"session_id = {SessionID}\r\n"
"-----apeon data\r\n"
"D"
"\x00\x00\x00"
,
LAST);
```

### 3.3.5.2 Parameterize request header

Check the scripts and make sure each request to the axdispatcher.aspx page contains the "pt" and "protocolMode" parameters in its header. For example,

```

web_add_header("pt","1090");
web_add_header("protocolMode","xml");
web_custom_request("axdispatcher.aspx_1",
  "URL=http://192.0.1.1/servlet/axdispatcher.aspx",
  "Method=POST",
  "Resource=0",
  "RecContentType=text/html",
  "Referer=",
  "Snapshot=t34.inf",
  "Mode=HTTP",
  "EncType=",
  "Body=<?xml version=\"1.0\" encoding=\"utf-8\"?><data><header type=\"192\"
version=\"35\" session_id=\"{SessionID}\" />
<transobject name=\"sqlca\" dbms_id=\"0\" autocommit_mode=\"0\" sql_code=\"0\"
sql_dbcode=\"0\" sql_nrow=\"0\" sql_errortext=\"\"
server_name=\"192.0.2.1\" database=\"apeon\" log_id=\"apeon\" db_password=\"\"
user_password=\"apeon\" user_id=\"\"
identity=\"0\" cache_name=\"apeon\" commiton_disconnect=\"-1\" catalog_owner=
\"58633936\" lock_level=\"65176744\"
capital_identifier=\"0\" associate_tablecolumn=\"0\" driver_id=\"10\" trimspace=
\"-1\" />
<body><sql type=\"1\"><value><![CDATA[@@SQL_6_u_mdi_fxs]]></value></
sql><parameters count=\"1\">
<parameter name=\"ESQL_1\" isnull=\"false\" datatype=\"1\" inouttype=\"1\">DEM</
parameter></parameters></body></data>",
LAST);

```

### To add the "pt" parameter:

Step 1: Search for the scripts that request the axdispatcher.aspx page.

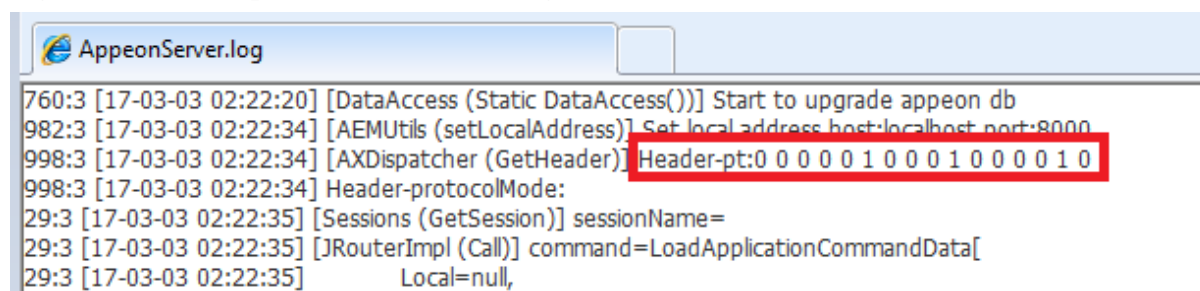
For example, web\_custom\_request("axdispatcher.aspx",.....); indicates the first request to the axdispatcher.aspx page;

web\_custom\_request("axdispatcher.aspx\_1",.....); indicates the second request to the axdispatcher.aspx page.

Step 2: In the PowerServer log file generated via debug mode in AEM, search for the header-pt information according to the sequence of the requests, and then convert its value from binary to decimal.

For example, binary "0000010001000010" can convert to decimal "1090".

### Figure 3.8: header-pt information in the log file



Step 3: Add the "pt" parameter and the decimal value to the scripts before the request starts.

For example, web\_add\_header("pt","1090");

### To add the "protocolMode" parameter:

If the body of the request to the axdispatcher.aspx page is an XML string, add the "protocolMode" parameter to the scripts before the request starts.

For example, `web_add_header("protocolMode","xml");`

### 3.3.5.3 Parameterization of DataWindow SQL statements

1. The recorded DataWindow SQL statements are as follows:

```
"UpDatesSQL_1 = update \"PERF_TEST1\" set \"depid\" =
1,\"code\" = 'modify1',\"name\" = 'modify1',\"born\" = '2006-09-04
14:35:12.843',\"age\" = 1,\"salary\" = 1 where \"id\" = 0;update
\"PERF_TEST1\" set \"depid\" = 1,\"code\" = 'modify1',\"name\" =
'modify1',\"born\" = '2006-09-04 14:35:12.843',\"age\" =
1,\"salary\" = 1 where \"id\" = 1;update \"PERF_TEST1\" set
\"depid\" = 1,\"code\" = 'modify1',\"name\" = 'modify1',\"born\" =
'2006-09-04 14:35:12.843',\"age\" = 1,\"salary\" = 1 where \"id\"
= 2;update \"PERF_TEST1\" set \"depid\" = 1,\"code\" =
'modify1',\"name\" = 'modify1',\"born\" = '2006-09-04
14:35:12.843',\"age\" = 1,\"salary\" = 1 where \"id\" = 3;update
\"PERF_TEST1\" set \"depid\" = 1,\"code\" = 'modify1',\"name\" =
'modify1',\"born\" = '2006-09-04 14:35:12.843',\"age\" =
1,\"salary\" = 1 where \"id\" = 4;update \"PERF_TEST1\" set
\"depid\" = 1,\"code\" = 'modify1',\"name\" = 'modify1',\"born\" =
'2006-09-04 14:35:12.859',\"age\" = 1,\"salary\" = 1 where \"id\"
= 5;update \"PERF_TEST1\" set \"depi
d\" = 1,\"code\" = 'modify1',\"name\" =
'modify1',\"born\" = '2006-09-04 14:35:12.859',\"age\" =
1,\"salary\" = 1 where \"id\" = 6;update \"PERF_TEST1\" set
\"depid\" = 1,\"code\" = 'modify1',\"name\" = 'modify1',\"born\" =
'2006-09-04 14:35:12.859',\"age\" = 1,\"salary\" = 1 where \"id\"
= 7;update \"PERF_TEST1\" set \"depid\" = 1,\"code\" =
'modify1',\"name\" = 'modify1',\"born\" = '2006-09-04
14:35:12.859',\"age\" = 1,\"salary\" = 1 where \"id\" = 8;update
\"PERF_TEST1\" set \"depid\" = 1,\"code\" = 'modify1',\"name\" =
'modify1',\"born\" = '2006-09-04 14:35:12.859',\"age\" =
1,\"salary\" = 1 where \"id\" = 9\r\n"
```

- To make the field value change every time the same user executes it, you need to parameterize the constant after the field: "modify1" in the above example.
- To ensure that different virtual users modify different records, you need to parameterize the id value after the "Where" keyword.

For these purposes, at least two parameters should be set: "Modified" and "ID".

To parameterize fields of the number type and the data type, it is also necessary to set the new parameters.

In this example, the type of parameter "Modified" is iteration number, and the type of parameter "id" is UserID.

- In the scripts, for most field values, you must perform the following two replacements in the DataWindow SQL statements.
  - Replace "modify1" with "{modified}"
  - Replace "where \"id\" = " (**Note:** In this example, the last character included in the quotation mark must be a space) with "where \"id\" = {id}"

If the text to be replaced is split and displayed in two lines, for example, "modify1" is displayed as "modif" and "y1" in two separated lines, you need to manually modify or

programmatically replace it, otherwise transaction failures will occur when there are many concurrent user access.

Make sure that every SQL statement is correctly replaced.

```
.....,\"col8\" = '{modified}',\"col9\" = 'modif
"y1\", \"col10\" = 'modify1' where \"id\" = {id}06;.....
```

### 3. The parameterized SQL statements are as follows:

```
"UpDateSQL_1 = update \"PERF_TEST1\" set \"depid\" = 1,\"code\" =
'{modified}',\"name\" = '{modified}',\"born\" = '2006-09-04
11:25:12.859',\"age\" = 1,\"salary\" = 1 where \"id\" =
{id}0;update \"PERF_TEST1\" set \"depid\" = 1,\"code\" =
'{modified}',\"name\" = '{modified}',\"born\" = '2006-09-04
11:25:12.859',\"age\" = 1,\"salary\" = 1 where \"id\" =
{id}1;update \"PERF_TEST1\" set \"depid\" = 1,\"code\" =
'{modified}',\"name\" = '{modified}',\"born\" = '2006-09-04
11:25:12.859',\"age\" = 1,\"salary\" = 1 where \"id\" =
{id}2;update \"PERF_TEST1\" set \"depid\" = 1,\"code\" =
'{modified}',\"name\" = '{modified}',\"born\" = '2006-09-04
11:25:12.875',\"age\" = 1,\"salary\" = 1 where \"id\" =
{id}3;update \"PERF_TEST1\" set \"depid\" = 1,\"code\" =
'{modified}',\"name\" = '{modified}',\"born\" = '2006-09-04
11:25:12.875',\"age\" = 1,\"salary\" = 1 where \"id\" =
{id}4;update \"PERF_TEST1\" set \"depid\" = 1,\"code\" =
'{modified}',\"name\" = '{modified}',\"born\" = '2006-09-04
11:25:12.875',\"age\" = 1,\"salary\" = 1 where \"id\" =
{id}5;update \"PERF_TEST1\" set \"depid\" = 1,\"code\" =
'{modified}',\"name\" = '{modified}',\"born\" = '2006-09-04
11:25:12.875',\"age\" = 1,\"salary\" = 1 where \"id\" =
{id}6;update \"PERF_TEST1\" set \"depid\" = 1,\"code\" =
'{modified}',\"name\" = '{modified}',\"born\" = '2006-09-04
11:25:12.875',\"age\" = 1,\"salary\" = 1 where \"id\" =
{id}7;update \"PERF_TEST1\" set \"depid\" = 1,\"code\" =
'{modified}',\"name\" = '{modified}',\"born\" = '2006-09-04
11:25:12.890',\"age\" = 1,\"salary\" = 1 where \"id\" =
{id}8;update \"PERF_TEST1\" set \"depid\" = 1,\"code\" =
'{modified}',\"name\" = '{modified}',\"born\" = '2006-09-04
11:25:12.890',\"age\" = 1,\"salary\" = 1 where \"id\" = {id}9\r\n"
```

### 4. When playing back the scripts after replacement, the DataWindow SQL statements executed are as follows:

```
UpDateSQL_1 = update "PERF_TEST1" set "depid" = 1,"code" =
'modify1',"name" = 'modify1',"born" = '2006-09-04
11:25:12.859',"age" = 1,"salary" = 1 where "id" =
10;update "PERF_TEST1" set "depid" = 1,"code" =
'modify1',"name" = 'modify1',"born" = '2006-09-04
11:25:12.859',"age" = 1,"salary" = 1 where "id" =
11;update "PERF_TEST1" set "depid" = 1,"code" =
'modify1',"name" = 'modify1',"born" = '2006-09-04
11:25:12.859',"age" = 1,"salary" = 1 where "id" =
12;update "PERF_TEST1" set "depid" = 1,"code" =
'modify1',"name" = 'modify1',"born" = '2006-09-04
11:25:12.875',"age" = 1,"salary" = 1 where "id" =
13;update "PERF_TEST1" set "depid" = 1,"code" =
'modify1',"name" = 'modify1',"born" = '2006-09-04
11:25:12.875',"age" = 1,"salary" = 1 where "id" =
14;update "PERF_TEST1" set "depid" = 1,"code" =
'modify1',"name" = 'modify1',"born" = '2006-09-04
11:25:12.875',"age" = 1,"salary" = 1 where "id" =
15;update "PERF_TEST1" set "depid" = 1,"code" =
```

```
'modify1',"name" = 'modify1',"born" = '2006-09-04
11:25:12.875',"age" = 1,"salary" = 1 where "id" =
16;update "PERF_TEST1" set "depid" = 1,"code" =
'modify1',"name" = 'modify1',"born" = '2006-09-04
11:25:12.875',"age" = 1,"salary" = 1 where "id" =
17;update "PERF_TEST1" set "depid" = 1,"code" =
'modify1',"name" = 'modify1',"born" = '2006-09-04
11:25:12.890',"age" = 1,"salary" = 1 where "id" =
18;update "PERF_TEST1" set "depid" = 1,"code" =
'modify1',"name" = 'modify1',"born" = '2006-09-04
11:25:12.890',"age" = 1,"salary" = 1 where "id" = 19
```

5. In the iterative operations, the DataWindow SQL statements executed at the third time are as follows:

```
UpDateSQL_1 = update "PERF_TEST1" set "depid" = 1,"code" =
'modify3',"name" = 'modify3',"born" = '2006-09-04
11:25:12.859',"age" = 1,"salary" = 1 where "id" =
10;update "PERF_TEST1" set "depid" = 1,"code" =
'modify3',"name" = 'modify3',"born" = '2006-09-04
11:25:12.859',"age" = 1,"salary" = 1 where "id" =
11;update "PERF_TEST1" set "depid" = 1,"code" =
'modify3',"name" = 'modify3',"born" = '2006-09-04
11:25:12.859',"age" = 1,"salary" = 1 where "id" =
12;update "PERF_TEST1" set "depid" = 1,"code" =
'modify3',"name" = 'modify3',"born" = '2006-09-04
11:25:12.875',"age" = 1,"salary" = 1 where "id" =
13;update "PERF_TEST1" set "depid" = 1,"code" =
'modify3',"name" = 'modify3',"born" = '2006-09-04
11:25:12.875',"age" = 1,"salary" = 1 where "id" =
14;update "PERF_TEST1" set "depid" = 1,"code" =
'modify3',"name" = 'modify3',"born" = '2006-09-04
11:25:12.875',"age" = 1,"salary" = 1 where "id" =
15;update "PERF_TEST1" set "depid" = 1,"code" =
'modify3',"name" = 'modify3',"born" = '2006-09-04
11:25:12.875',"age" = 1,"salary" = 1 where "id" =
16;update "PERF_TEST1" set "depid" = 1,"code" =
'modify3',"name" = 'modify3',"born" = '2006-09-04
11:25:12.875',"age" = 1,"salary" = 1 where "id" =
17;update "PERF_TEST1" set "depid" = 1,"code" =
'modify3',"name" = 'modify3',"born" = '2006-09-04
11:25:12.890',"age" = 1,"salary" = 1 where "id" =
18;update "PERF_TEST1" set "depid" = 1,"code" =
'modify3',"name" = 'modify3',"born" = '2006-09-04
11:25:12.890',"age" = 1,"salary" = 1 where "id" = 19
```

In this way, the value changes every time the same user executes an Update operation.

### 3.3.5.4 Parameterization of Embedded SQL statements

To parameterize the value in the embedded SQL statement, you will have to define the value as a variable.

For example, the following script cannot be parameterized.

```
select a from table where c = '1';
```

You will have to modify the script as below, in order to parameterize the value:

```
int ic
ic = 1
select a from table where c= :ic
```



### 3.3.5.5 Additional steps for the Update operation

For the Update operation, the script parameterization should meet the following three requirements:

1. Parameterization of SessionID (which has already been described in [Parameterize scripts of SessionID](#));
2. Parameterization of SQL statements (which will be described in [Parameterization of DataWindow SQL statements](#) and in [Parameterization of Embedded SQL statements](#));
3. Add the Commit request to the end of each Update request, that is, move the Commit request in the recorded scripts from end.c to action.c.

### 3.3.6 Playing back scripts to test the correctness of scripts

Before playing back the scripts, set as follows from the menu: **Vuser > Run-time Settings > Browser Emulation > Options > Check new resources when simulating browser cache = Yes.**

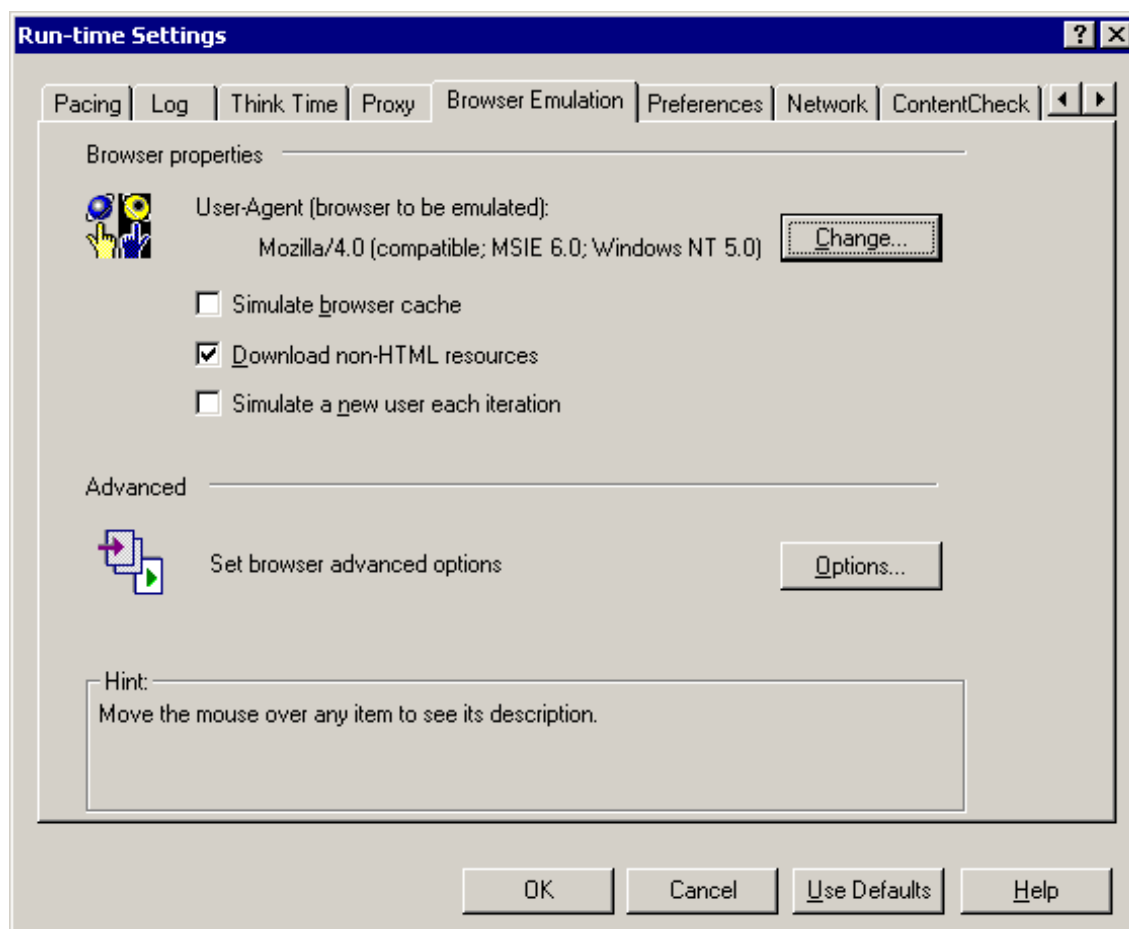
Check the log of AEM to see if the log of recorded scripts is the same as the log of playback (SessionID values can be different), otherwise, go back to the previous step to check and modify the scripts and settings again.

### 3.3.7 Setting Scenarios

1. Select from **Tools > Create controller scenario**;
2. Set the schedule of scenario according to your needs;
3. Configure the parameter settings for scenario:

Navigate to **Browser Emulation**, uncheck the **Simulate browser cache** and the **Simulate a new user each iteration** checkboxes, and then check the **Download non-HTML resources** checkbox, as shown in the following figure.

Click **OK**.

**Figure 3.9: Run-times Settings**

### 3.3.7.1 Additional steps for the Update operation

Set the Update operation being committed when 10 virtual users access concurrently. Make sure that the IDs of virtual users are the successive integers starting from 1, for example, 1~10, and keep running for 5 minutes.

When you run the scenario, ensure that it runs correctly without any error or error messages in the AEM log, and check the corresponding tables in the database and make sure the values of records (id 10-109) have been changed.

### 3.3.8 Running Scenarios

Select to run the scenario and the test report will be generated.

## 3.4 Troubleshooting

### 3.4.1 "The session does not exist" error

**Phenomena:** The scripts run correctly right after they are recorded and modified, but after a while or after the server is restarted, the following error messages appear in the log of Apeon when playing back the scripts:

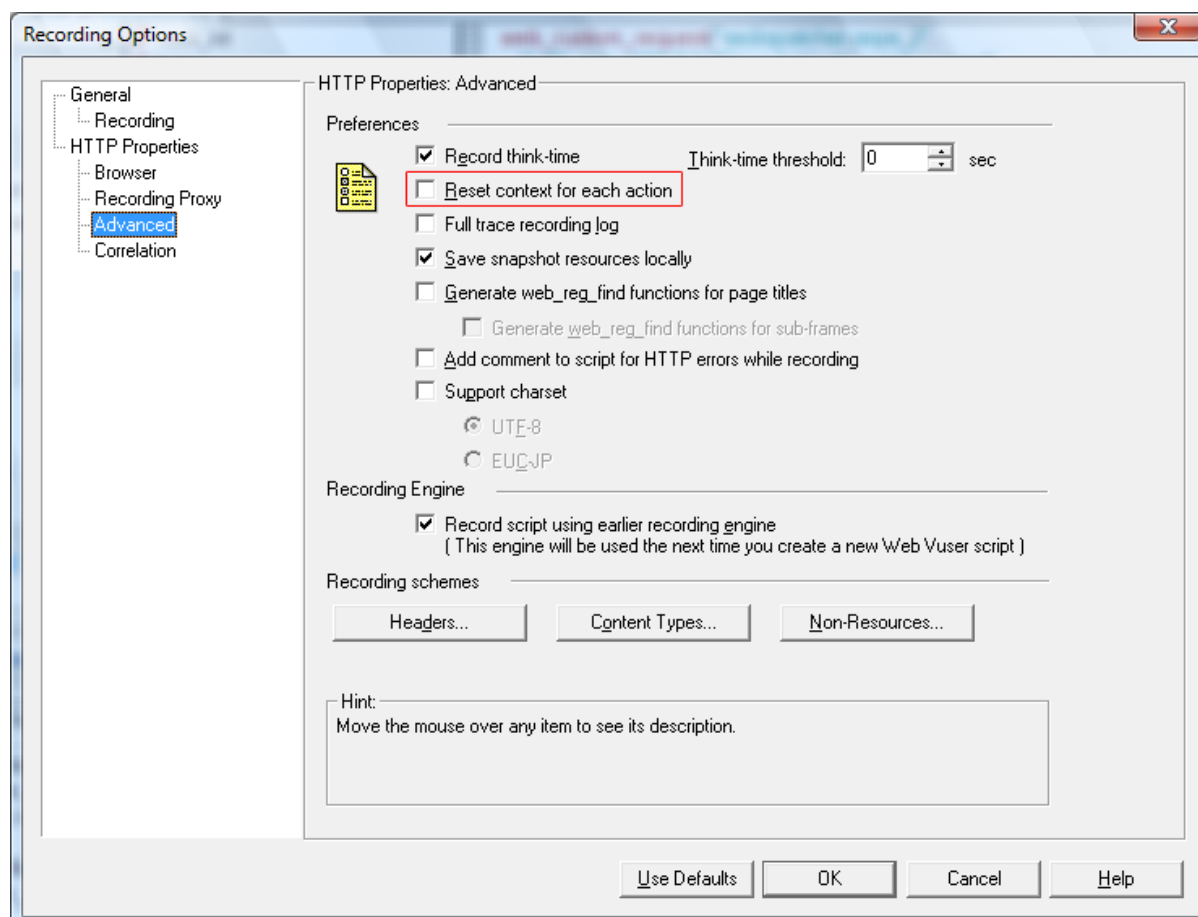
```
1150101302078:0 [06-06-12 16:35:02.078]
```

```
[com.appeon.server.services.controller.JRouterImpl (call())] The
session does not exist."
```

**Solution:** Check and make sure that the **Reset context for each action** option is unchecked as shown in the following figure, and then record the scripts again.

In this way, the scripts will remain usable even after the application server is restarted.

**Figure 3.10: Recording Options**



### 3.4.2 Errors appear when playing back scripts with LoadRunner 8.0

**Phenomena:** The following error messages will appear in the ApeonServer.log or ApeonError.log when the scripts are played back with LoadRunner 8.0:

```
1154707726562:0 [06-08-04 09:08:46.562]
[com.appeon.server.services.controller.JRouterImpl (call())] The
session does not exist.
```

**Solution:** This is a known issue with LoadRunner 8.0.

Please upgrade LoadRunner 8.0 to a later version.

### 3.4.3 The value of sessionId is null

**Phenomena:** If you use the following statements to debug the value of SessionID, the SessionID value will be null in the Execution Log after the script is played back.

```
lr_output_message(lr_eval_string("SessionID----->"));
```

```
lr_output_message(lr_eval_string("{SessionID}"));
```

**Solution:** The function (as follows)

```
web_reg_save_param("SessionID", "LB=ID =", "RB=\r\n", LAST);
```

to obtain Session ID should be placed before the second HTTP request. Do not place it before the first request.

```
web_custom_request("AXDispatcher.aspx_2", "URL=http://172.17.1.29/servlet/ AXDispa
tcher",
"Method=POST",
"Resource=1",
"RecContentType=text/plain",
"Referer=",
"Binary=1",
"EncType=",
"Body="
"\x1F\x8B\x08\x00\x00\x00\x00\x00\x00\x0B"
"SPf`"
"\xD0"
"chd("
"\x00\xC2"
"T"
"\x86"
"| "
"\x86"
"<"
"\x86"
"x"
"\x86"
"d "
"\x9D\x02\xE4\xC5\x03"
"q"
"\x05"
"p."
"\x17"
("("
"\x97\x03"
"d"
"\x17\x03"
"E"
"\x12\x81"
"\ "
"\x8C\x0C"
"@ "
"\xC0\x02\xC4"
"L"
"\x0C\x00\x9D\x84\xA2\xD6"
"? "
"\x00\x00\x00"
", "
LAST);
```

### 3.4.4 Error message appears in script playback

**Phenomena:** After recording, modifying, and playing back the scripts, the following messages appear in the Execution log.

```
vuser_init.c(236): Error: No match found for the requested
parameter "SessionID". If the data you want to save exceeds 256
bytes, use web_set_max_html_param_len to increase the parameter size
vuser_init.c(236): Error: The above "not found" error(s) may be
```

explained by header and body byte counts being 0 and 0, respectively.  
Abort was called from an action

**Solution:** This is probably caused by the Web cache setting. If you will not test Web cache, be sure to cancel the settings for simulating cache in the **Run-time Settings**, by selecting from **Vuser > Run-time Settings > Browser Emulation > Options > Check for newer resources when simulating browser cache = Yes**.

### 3.4.5 Error message in Apeon Log

**Phenomena:** After the script is played back, set the pacing>1 and play back the scripts again, the following error messages appear in the ApeonServer log for AEM:

```
1154707726562:0 [06-08-04 09:08:46.562]
[com.appeon.server.services.controller.JRouterImpl (call())]
The session does not exist.
1154707726593:0 [06-08-04 09:08:46.593]
[com.appeon.server.services.controller.JRouterImpl (call())]
The session does not exist.
1154707726625:0 [06-08-04 09:08:46.625]
[com.appeon.server.services.controller.JRouterImpl (call())]
The session does not exist.
1154707726687:0 [06-08-04 09:08:46.687]
[com.appeon.server.services.controller.JRouterImpl (call())]
The session does not exist.
1154707726718:0 [06-08-04 09:08:46.718]
[com.appeon.server.services.controller.JRouterImpl (call())]
The session does not exist.
```

**Solution:** Make sure that the **Simulate a new user each iteration** option for **Run-time Settings** is unchecked.

### 3.4.6 Failed to parameterize scripts

**Phenomena:** After the scripts are recorded, the plain text cannot be found in all web\_custom\_request as shown below:

```
web_custom_request("apeon.weblibrary.AXDispatcher3",
  "URL=http://172.17.1.29/servlet/apeon.weblibrary.AXDispatcher",
  "Method=POST",
  "Resource=1",
  "RecContentType=text/plain",
  "Referer=",
  "Binary=1",
  "EncType=",
  "Body="
  "\\x1F\\x8B\\x08\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x0B"
  "kQ"
  "\\x16"
  "a"
  "\\xD0"
  "e"
  "\\xB0"
  "d0c0e0"
  "\\x07\\xD3\\xE6\\x0C"
  "&"
  "\\x0C\\xC6\\x0C\\x5C\\xC5\\x0C\\x85\\x0C"
  "g"
  "\\x0C\\xC9\\x0C\\x89\\x0C\\x5C\\x0C\\xFF\\x19\\x19\\x98\\x18"
  "Y"
  "\\x18"
```

```
"8"
"\x81"
"Ln"
"\x06\x1E\x06"
"^"
"\x06"
">"
"\x06\x10\x90\x00"
"b&"
"\x06\x00\xE6\x89"
"_xB"
"\x00\x00\x00"
" ",
LAST);
```

**Solution:** Before running the scenario, make sure that you have selected **Test Mode** in the **Run Mode** settings of AEM and that you have set the log mode to **Standard**. Then the plain text ("-----apeon data\r\n" "Name = Connect\r\n" "session\_id = -1476559241\r\n" "-----apeon data\r\n" "P" "\x00\x00\x00" ) will appear as shown below.

```
web_custom_request("apeon.weblibrary.AXDispatcher3",
  "URL=http://172.17.1.29/servlet/apeon.weblibrary.AXDispatcher",
  "Method=POST",
  "Resource=1",
  "RecContentType=text/plain",
  "Referer=",
  "Binary=1",
  "EncType=",
  "Body="
  "\x1F\x8B\x08\x00\x00\x00\x00\x00\x00\x00\x0B"
  "kQ"
  "\x16"
  "c"
  "\xD0"
  "e0d0a0g0c0"
  "\x05"
  "BK"
  "\x06"
  "# "
  "\xCF\x90\x81\xAB\x98\xA1\x90"
  "!"
  "\x87"
  "!"
  "\x99"
  "!"
  "\x91\x81\x8B\xE1"
  "?#"
  "\x03\x13"
  "#"
  "\x0B\x03"
  "' ' "
  "\x90\xC9\xCD\xC0\xC3\xC0\xCB\xC0\xC7\x00\x02"
  "R@"
  "\xCC\xC4\x00\x00"
  "HO"
  "\xC9"
  "&D"
  "\x00\x00\x00"
  "-----apeon data\r\n"
  "Name = Connect\r\n"
  "session_id = -1476559241\r\n"
  "-----apeon data\r\n"
  "p"
```

```
"\x00\x00\x00"
" ",
LAST);
```

### 3.4.7 Out of memory error and the application server shut down

**Phenomena:** On the Solaris platform, WebLogic reports the out of memory error, and then shuts down. This is because the physical memory of the server is too low.

**Solution:** Increase the virtual memory of the server.

1. For Windows, set the virtual memory to 2G;

2. For Solaris, set the swap space to 2G.

The commands to set swap on Solaris are:

```
Mkfile /beaswap
swap -a /beaswap
```

To add new swap space, this command must be run every time a Solaris server is restarted.

### 3.4.8 Field values do not change after parameterization and playback

**Phenomena A:** The corresponding field values in the records do not change after the SQL parameterization and script playback, when using SAP SQL Anywhere database.

**Solution A:**

Steps to reproduce:

1. Record in the Test Mode: **DataWindow > Updating > Change Employee Salary** function.

```
"UpDateSQL_1 = update \"employee\" set \"salary\" = ? where \"emp_id\" =?....."
```

2. In the record scripts, create a new parameter "Salary" for the incremental change, and parameterize the SQL statements, as shown below:

```
"UpDateSQL_1 = update \"employee\" set \"salary\" = {salary} where
\"emp_id\" = 105....."
```

3. Play back the scripts.

**Phenomena B:** When open another URL, the value of salary in records such as \"emp\_id\" = 105 are found unchanged.

**Solution B:** The connection parameters (DisableBind=1) should be set in PowerBuilder for the SAP SQL Anywhere database, then deploy the application source code and record the scripts again.

```
ConnectionString='DSN=ApeonSample;UID=dba;PWD=sql',DisableBind=1
```

### 3.4.9 Runtime errors causing scenario failure

For the Apeon .NET version, after errors occur at runtime, the scenario cannot run correctly as before.

**Phenomena:** The following errors are reported in the server log:

```
2006-08-17 14:41:36.609 : Object reference not set to an instance of an object.
at Apeon.Server.Commands.CommitCommand.Execute()
at Apeon.Server.Commands.Command.ConstructAndExecute(CommandData commandData)
at Apeon.Server.Services.Controller.JRouterImpl.ExecuteInTransaction
(CommandData commandData)
at Apeon.Server.Services.Controller.JRouterImpl.ExecuteCommand
(HttpResponse response, CommandData superCommand, SessionAccessInfo session)
at Apeon.Server.Services.Controller.JRouterImpl.Call(CommandData
superCommand, HttpRequest request, HttpResponse response)
at Apeon.Server.Services.Controller.JRouterImpl.Call(Int32
clientType, Object requestData, Hashtable clientContext)
```

LoadRunner will report the following error:

```
Action1.c(190): Error: Page download timeout (120 seconds) has expired.
```

**Solution:** Restart IIS, SQL Server, and LoadRunner.

Check the PowerBuilder source code and change `autocommit=false` to `autocommit=true`, or move the **Commit** operation after the **Retrieve** operation, so as to ensure the commit operation will be executed for every retrieval.

### 3.4.10 Transactions failed

Many transactions in the test fail when there are large amount of concurrent access with Update or Retrieve.

**Phenomena:** The script playback is successful in the Virtual User Generator, and the logs in AEM report no errors. But the transaction fails when there are large amount of concurrent access.

**Solution:** Move the **Commit** request after the **Update** request, that is, move the **Commit** request in recorded scripts from `end.c` to `action.c`.

### 3.4.11 Unable to connect to remote servers

**Phenomena:** LoadRunner is not able to generate the scripts after recording and the following error appears in the log:

```
(Service=) NOT PROXIED! (REASON: Unable to connect to remote server: rc = -1 , le = 0)
```

**Solution:** Go to **Tools > Recording Options** and look at the **Port Mapping**.

Select **Socket level and WinINet level data** for **Capture level** as shown in the following figure.



**Figure 3.11: Port Mapping**

