HPE Run Results Viewer

Software Version: 14.00

User Guide

Go to **HELP CENTER ONLINE**
http://uft-help.saas.hpe.com/

Hewlett Packard
Enterprise

Document Release Date: January 2017 | Software Release Date: January 2017

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise Development LP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HPE required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: https://softwaresupport.hpe.com.

This site requires that you register for an HPE Passport and sign in. To register for an HPE Passport ID, go to https://softwaresupport.hpe.com and click **Register**.

Contents

Welcome to the Run Results Viewer User Guide

The HPE Run Results Viewer User Guide explains how to use the Run Results Viewer to interpret and use the test results from your GUI or API tests.

We recommend having some prior knowledge of UFT and its testing capabilities so you can fully evaluate the test results.

Using the Run Results Viewer

After running a test or component, you can view the run results in the HPE Run Results Viewer. The Run Results Viewer contains multiple panes, each of which displays specific types of information. The Run Results Tree Pane and Search Box displays a hierarchical representation of the run results. The remaining panes provide details about a selected node or step, the data used for a particular step, optional screen captures or images (UFTGUI testing only), optional system information (UFTGUI testing only), and so on.

In addition, you can view the run results in HTML format directly in your browser or in UFT. To select the format in which you want to view the run results, open the Run Sessions pane in the Options dialog box (**Tools > Options > General** tab > **Run Sessions** node).

By default, the Run Results Viewer opens automatically at the end of a run session. If you want to change this behavior, in UFT, clear the **View results when run session ends** check box in the Run Sessions pane of the Options dialog box.

The Run Results Viewer contains a description of the steps performed during the run session.

- For a GUI component, or for a GUI test that does not contain Data Pane parameters, the Run Results Viewer shows a single test iteration.
- For tests, if the test contains Data Pane parameters, and the test settings are configured to run multiple iterations, the Run Results Viewer displays details for each iteration of the test run. The results are grouped by the actions in the test.
- For an API test, the individual steps and checkpoints included in the test. If a test is set to run multiple iterations, each iteration is displayed.

You set the test to run for one or all iterations in the Run pane of the Settings dialog box. For details, see the UFT Help Center for your version of UFT.

The Run Results Viewer is installed automatically with UFT.

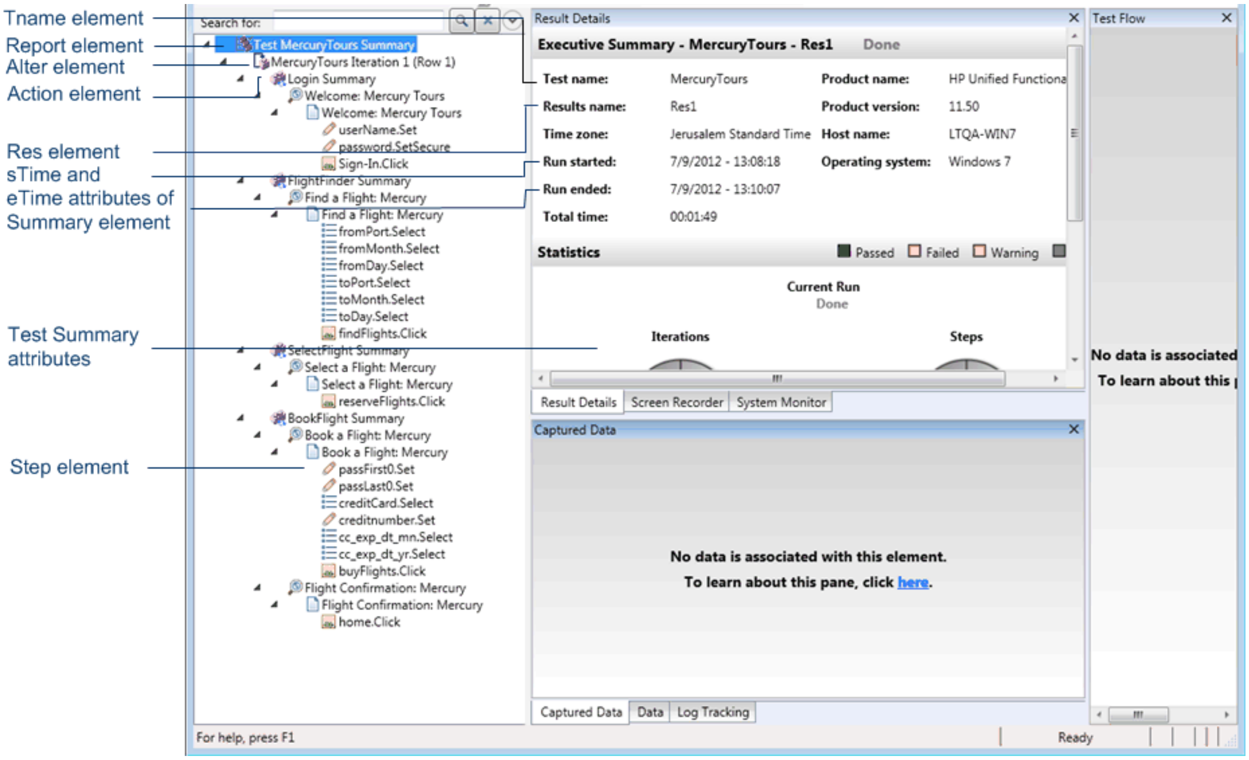
You can also install the Run Results Viewer as a standalone application. This enables you to share the results of your tests with business analysts and developers who do not have UFT installed on their computers.

Run results XML file

The results of each run session are saved in a single **.xml** file (called **results.xml**). This **.xml** file stores information on each of the run result nodes in the display. The information in these nodes is used to dynamically create **.htm** files that are shown in the Result Details pane in the Run Results Viewer.

Each node in the run results tree is an element in the **results.xml** file. In addition, there are different elements that represent different types of information displayed in the run results. You can take run result information from the **.xml** file and use XSL to display the information you require in a customized format (either when printing from within the Run Results Viewer, when displaying run results in your own customized results viewer, or when exporting the run results to an **.html** file).

The diagram below for a GUI test, shows the correlation between some of the elements in the **.xml** file and the items they represent in the run results. These elements are similar for both tests and business components.



Executive Summary - MercuryTours - Res1 Done

Test name:	MercuryTours	Product name:	HP Unified Functiona
Results name:	Res1	Product version:	11.50
Time zone:	Jerusalem Standard Time	Host name:	LTQA-WIN7
Run started:	7/9/2012 - 13:08:18	Operating system:	Windows 7
Run ended:	7/9/2012 - 13:10:07		
Total time:	00:01:49		

No data is associated with this element.
To learn about this pane, click here.

No data is associated
To learn about this

XSL provides you with the tools to describe exactly which run result information to display and exactly where and how to display, print or export it. You can also modify the **.css** file referenced by the **.xsl** file, to change the appearance of the report (for example, fonts, colors, and so forth).

For example, in the **results.xml** file, one element tag contains the name of an action or a component, and another element tag contains information on the time at which the run session is performed. Using XSL, you could tell your customized Run Results Viewer that the action or component name should be displayed in a specific place on the page and in a bold green font, and that the time information should not be displayed at all.

You may find it easier to modify the existing **.xsl** and **.css** files provided with the Run Results Viewer application, instead of creating your own customized files from scratch. The files are located in the **HPE\Run Results Viewer\dat** folder, and are named as follows:

- **PShort.xsl.** Specifies the content of the run results report printed, or exported to an HTML file, when you select the **Short** option in the Print or Export to HTML File dialog boxes.
- **PDetails.xsl.** Specifies the content of the run results report printed, or exported to an HTML file, when you select the **Detailed** option in the Print or Export to HTML File dialog boxes.
- **PStringTable.xsl.** Specifies the string constants to be used in the exported document. For example, **Iteration #** may be used for the iteration number prefix. If you select the **User-defined XSL** option in the Print or Export to HTML File dialog boxes, the **.xsl** file you specify must contain an "include" call to this file. You can localize the strings, if needed.

- **PResults.css.** Specifies the appearance of the run results print preview. This file is referenced by the above **.xsl** files.
- **Results.css.** Specifies the styles, fonts, and colors of the various elements displayed in the run results.

UFT only: For details on the structure of the XML schema, and a description of the elements and attributes you can use to customize the run results reports, see the *HPE Run Results Schema Reference*.

Custom Fields (UFT API Testing Only)

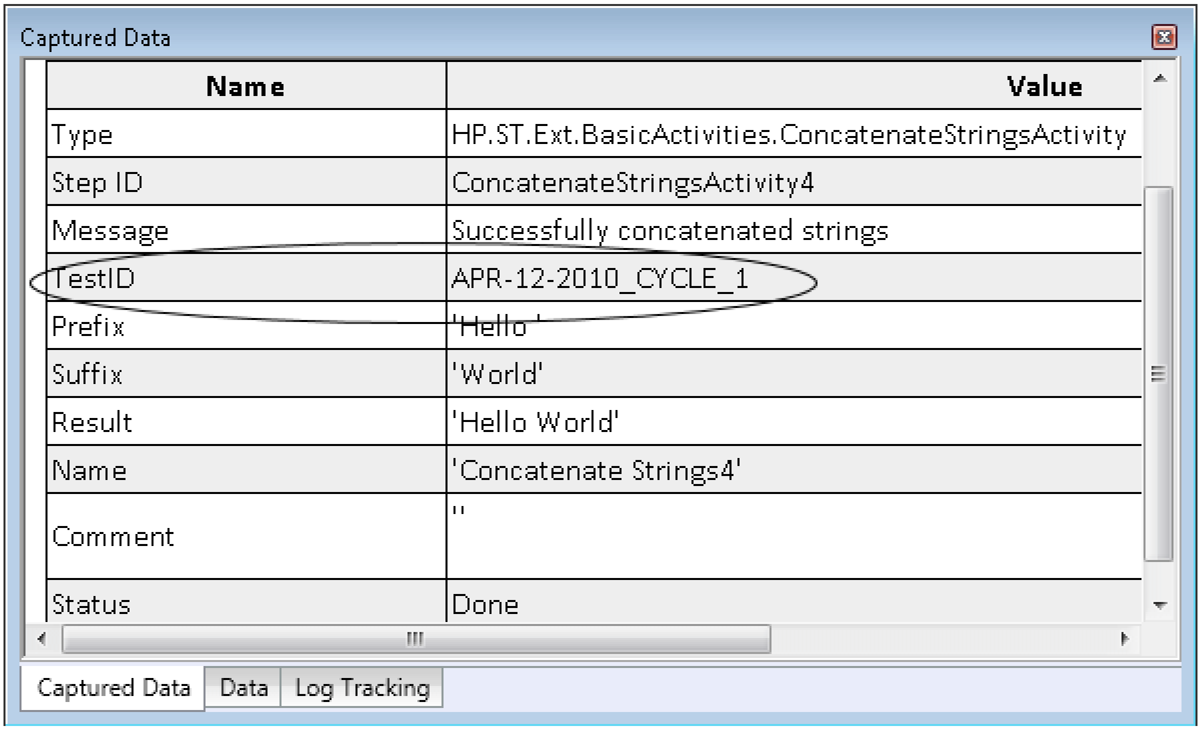
You can use the **Report** function to show custom information in the Run Results Viewer. You can specify strings or existing arguments and display them in the viewer.

You add the **Report** function to the step's events. For details, see the section on Custom Code and Events in the *Unified Functional Testing User Guide*.

The following example prints **APR-12-2010_CYCLE_1** in the **Value** column for **TestID**.

```
args.Activity.Report("TestID","APR-12-2010_CYCLE_1");
```

The report displays the keyword and its value at the Activity level of the results.



Name	Value
Type	HP.ST.Ext.BasicActivities.ConcatenateStringsActivity
Step ID	ConcatenateStringsActivity4
Message	Successfully concatenated strings
TestID	APR-12-2010_CYCLE_1
Prefix	'Hello '
Suffix	'World'
Result	'Hello World'
Name	'Concatenate Strings4'
Comment	''
Status	Done

Navigate the run results tree

This task describes how to collapse or expand a branch in the run results tree to select the level of detail that the tree displays.

When you open run results in the Run Results Viewer for the first time, the tree expands one level at a time. If the child branches under a parent branch were previously expanded, that state is maintained when you expand or collapse the parent branch.

You can do the following to view the results:

- "Expand a specific branch" below
- "Expand a branch and all branches below it" below
- "Expand all of the branches in the run results tree" below
- "Collapse a specific node" below
- "Collapse all of the nodes in the tree" below
- "Move between previously selected nodes within the run results tree" below
- "Find specific steps within the Run Results" on the next page
- "Filter the tree to display only nodes that match certain criteria" on the next page

Expand a specific branch

- Double-click the branch.
- Select the branch and click the arrow to the left of the branch icon.
- Press the plus key (**+**) on your keyboard number pad.

The tree displays the details for the branch, and the expand sign changes to collapse.

Expand a branch and all branches below it

- Select the branch and press the asterisk (*****) key on your keyboard number pad.
- Right-click a branch and select **Expand All**.

Expand all of the branches in the run results tree

- Right-click the top level branch and select **Expand All**.
- Select **View** > **Expand All**.
- Click the **Expand All** button .
- Select the top level of the tree and press the asterisk (*****) key on your keyboard number pad.

Collapse a specific node

- Double-click the node.
- Right-click a node and select **Collapse All**.
- Select it and click the arrow to the left of the node icon.
- Press the minus key (**–**) on your keyboard number pad.

The node's child nodes disappear from the tree.

Collapse all of the nodes in the tree

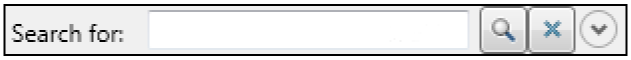
- Right-click the top level branch and select **Collapse All**.
- Select **View** > **Collapse All**.
- Click the **Collapse All** button .

Move between previously selected nodes within the run results tree

Click the **Go to Previous Node** or **Go to Next Node** buttons ⇦ ⇨ .

Find specific steps within the Run Results

Use the **Search** box (located above the run results tree), for example:



You can search for text, status, and/or types of nodes.

Filter the tree to display only nodes that match certain criteria

Use the Filter dialog box **(View > Filters).**

Jump to a step in a GUI test from the run results

You can view the step in UFT that corresponds to a node in the Run Results tree for any node that has a corresponding step in a GUI test.

> **Note:** This feature is disabled for:
>
> - Any testing document other than a GUI test.
> - The Action, Iteration, and Test Summary nodes.
> - Any step that is part of an action that was run using the **LoadAndRunAction** statement.
> - Any step performed by a recovery scenario.
> - Tests that were run in **Fast** mode.
> - Any step run from the Watch or Console debug panes in UFT.

1. Make sure that UFT is open to the test whose results are displayed in the Run Results Viewer.
2. Select a node in the run results tree.
3. Perform one of the following:
    a. Click the **Jump to Step in Test** button ☰ from the Run Results toolbar.
    b. Right-click and select **Jump to Step in Test** from the context menu.
    c. Select **View >Jump to Step in Test**.
4. The UFT window is activated and the step is highlighted.

Manually submit defects to ALM

This task includes the following steps:

- "Prerequisites" on the next page
- "Connect to an ALM project" on the next page
- "Add a new defect" on the next page
- "Results" on the next page

1. **Prerequisites**

   Ensure that the ALM client is installed on your computer. (Enter the ALM Server URL in a browser and ensure that the Login screen is displayed.)

2. **Connect to an ALM project**

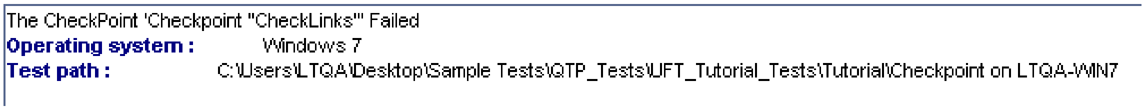
   Select **Tools > ALM Connection** or click the **ALM Connection** button and connect to an ALM project.

   > **Note:** If you are using the Run Results Viewer on a computer running Windows 10 or Windows Server 2016, you must connect to ALM the first time with administrative privileges.

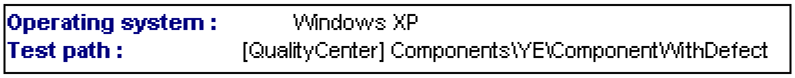
3. **Add a new defect**

   a. Select **Tools > Add Defect** or click the **Add Defect** button to open the New Defect dialog box in the specified ALM project. The New Defect dialog box opens.

   b. Modify the defect details as necessary. Basic information on the test or component and any checkpoints (if applicable) is included in the description:



```
The CheckPoint 'Checkpoint "CheckLinks"' Failed
Operating system :        Windows 7
Test path :              C:\Users\LTQA\Desktop\Sample Tests\QTP_Tests\UFT_Tutorial_Tests\Tutorial\Checkpoint on LTQA-WIN7
```



```
Operating system :        Windows XP
Test path :              [QualityCenter] Components\YE\ComponentWithDefect
```

   > **Tip:** You can attach movies (**.fbr** files) to defects in ALM. If you have the Unified Functional Testing Add-in for ALM installed, you can view the movies from ALM.

4. **Results**

   The defect is added to the ALM project's defect database.

Automatically submit defects to an ALM project

This task describes how to set the Run options in UFT to automatically submit defects to your ALM project for each failed step in your GUI test, eliminating the need to remember to submit these defects after a run session.

This task includes the following steps:

- "Prerequisites" on the next page
- "Modify the Run settings in the Options dialog box" on the next page
- "Results" on the next page

1. **Prerequisites**

   - In UFT, make sure you are connected to the relevant ALM project prior to the run session (**ALM > ALM Connection**).

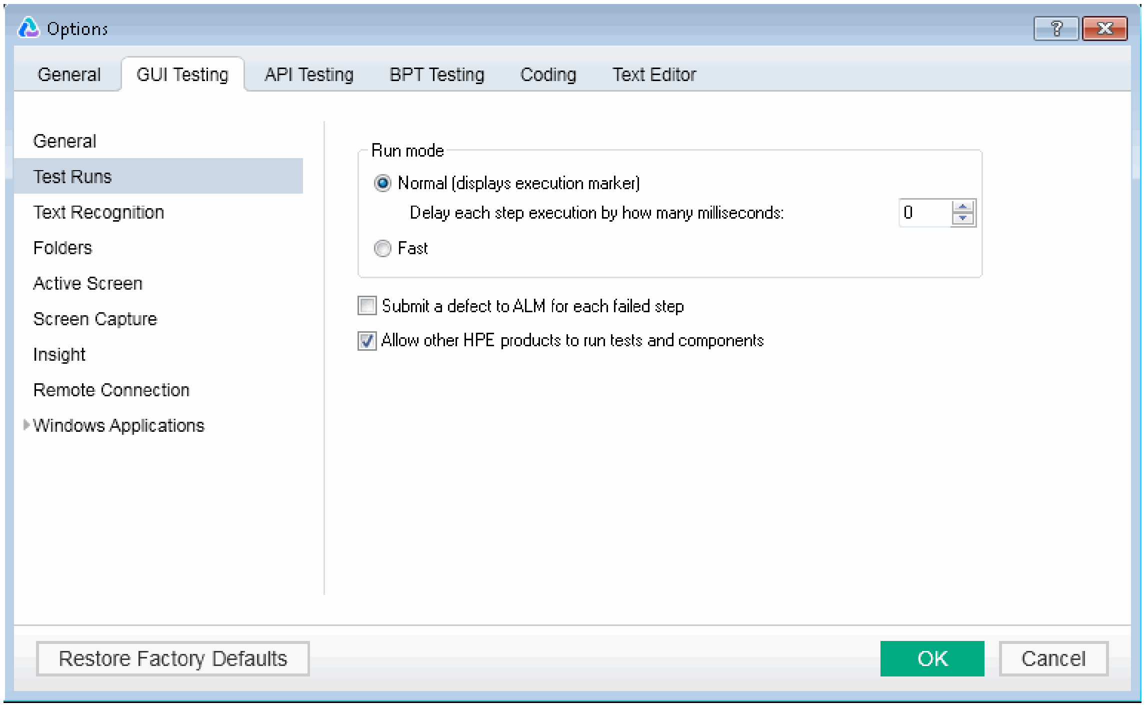
     > **Note:** If you are using the Run Results Viewer on a computer running Windows 10 or Windows Server 2016, you must connect to ALM the first time with administrative privileges.

   - The run results must be stored in this ALM project.

2. **Modify the Run settings in the Options dialog box**

   a. Select **Tools > Options**. The Options dialog box opens.

   b. In the **GUI Testing** tab, click the **Test Runs** node.



Run mode
● Normal (displays execution marker)
Delay each step execution by how many milliseconds: 0
○ Fast
□ Submit a defect to ALM for each failed step
☑ Allow other HPE products to run tests and components

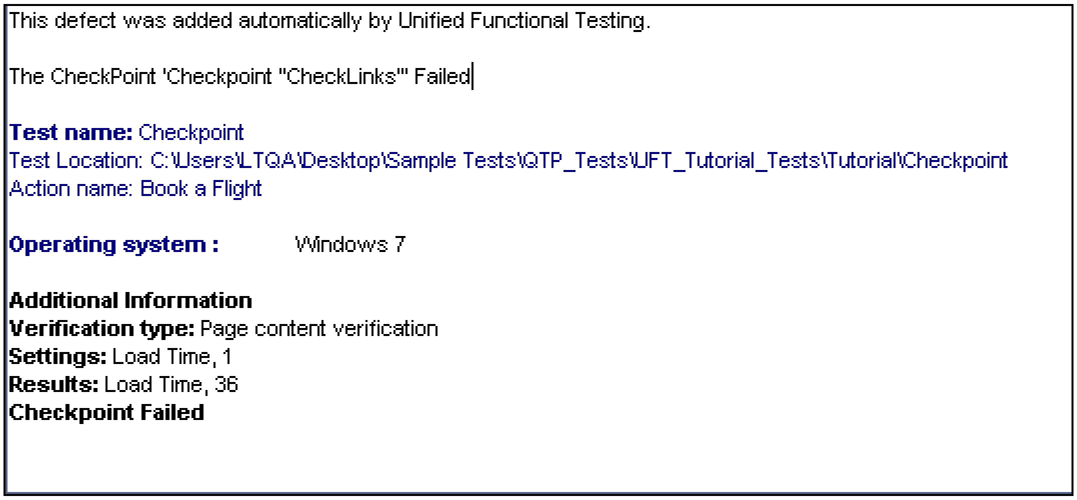
   c. Select the **Submit a defect to ALM for each failed step** check box.

   d. Click **OK** to close the Options dialog box.

3. **Results**

During each run session, the necessary defects are added to your ALM project.

A sample of the information that is submitted to ALM for each defect is shown below:



This defect was added automatically by Unified Functional Testing.

The CheckPoint 'Checkpoint "CheckLinks"' Failed

Test name: Checkpoint
Test Location: C:\Users\LTQA\Desktop\Sample Tests\QTP_Tests\UFT_Tutorial_Tests\Tutorial\Checkpoint
Action name: Book a Flight

Operating system : Windows 7

Additional Information
Verification type: Page content verification
Settings: Load Time, 1
Results: Load Time, 36
Checkpoint Failed

Export run results

This task describes how to export run results to a file.

This task includes the following steps:

- "Open the results in the Run Results Viewer" below
- "Specify the export settings" below
- "Save the file" below
- "Results" on the next page

1. **Open the results in the Run Results Viewer**

   In the Run Results Viewer, select **File > Open**.

2. **Specify the export settings**

   Select **File > Export To File**. The Export Run Results dialog box opens.

3. **Save the file**

   Click **Export**. The Save As dialog box opens. Specify the file name and path, and select the required file type.

Report type	Save as type
Step details	• HTML (***.htm**, ***.html**) (default) • PDF (***.pdf**) • DOC (***.doc**) (Available if Microsoft Word is installed)
Data Table	Excel (***.xls**)

Report type	Save as type
Log Tracking **(UFTGUI Testing only)**	XML (***.xml**)
Screen Recorder **(UFTGUI Testing only)**	FlashBack (***.fbr)**
System Monitor **(UFTGUI Testing only)**	• Text (***.csv**, ***.txt**) (default) • Excel (***.xls**) • XML (***.xml**) • HTML (***.htm**, ***.html**) **Note:** Only the system monitor data is exported, not the graph.

4. **Results**

   When you click **Save**, the file is exported in the specified format to the designated location.

   > **Note:** You can view **.fbr** files in the HPE Micro Recorder. You can also attach **.fbr** files to defects in ALM. If you have the Unified Functional Testing Add-in for ALM installed, you can view the movies from ALM.

Play a Screen Recorder movie in the HPE Micro Player

> **Note:** UFT must be installed on the computer on which you want to use the HPE Micro Player.

1. Perform one of the following:
   - Double-click any **.fbr** file in Windows Explorer.
   - Select **Start > All Programs > HPE Software > HPE Unified Functional Testing > Tools > HPE Micro Player** and then select **File > Open** in the Micro Player to select any **.fbr** file.
   - Open the program at **<UFT installation folder\bin\Free_HPSR_Player.exe**.

   The movie opens in the HPE Micro Player and begins playing.

   > **Note:** For details on accessing UFT and UFT tools and files in Windows 8.X or higher and Windows Server 2012, see "Accessing the UFT in Windows 8.X or Higher Operating Systems" on page 15.

2. Use the controls at the top of the window to access a particular location in the movie or to modify the volume settings.

Delete run results

This task describes how to use the Run Results Deletion Tool to remove unwanted or obsolete run results from the file system, according to specific criteria that you define. For example, you may want to always delete run results older than a certain date or over a minimum file size. This enables you to free up valuable disk space.

Prerequisites

To delete run results from an ALM project, you must first:

- Make sure that you have **Delete Run** permissions for this ALM project.
- Connect to the ALM project.

Delete run results using the Run Results Deletion Tool

In the Run Results Deletion Tool, select the file or folder from which to delete run results and then select the selected run results to delete.

The selected run results are deleted from the file system and/or the ALM project.

Known Issues- Viewing Run Results

Local system monitor	After you run a test or component with the local system monitoring option activated when the test or component is either very short, or the number of seconds entered for the **Enable local system monitoring every: __ seconds** option is high (a high percentage relative to the length of your entire test run), then when you select one of the last steps in the Run Results tree, the **Current Step** indicator in the System Monitor pane may jump to a position outside (to the right) of the graph. **Workaround:** Add a **Wait** statement to the end of the test or reduce the number of seconds entered in the **Enable local system monitoring every: __ seconds** option.
Run session errors	Errors during the run session produce more than one error node in the run results.
Exporting run results	• **Exporting run results.** When UAC is set to ON and you select to export the Run Results to a system folder, the exported file is stored under Virtual Storage rather than under the specified folder. (Relevant for Windows 7, Windows Server 2008 R2, Windows 8.x, and Windows Server 2012)
Viewing run results from ALM	If you installed the Run Results Viewer without installing UFT, then after you run a BPT test from ALM, pressing **Show Last Run Results** might fail to display the results in the Run Results Viewer if the Visual C++ 2005 redistributable is installed on your computer. **Workaround:** Install the Unified Functional Testing Add-in for ALM from the ALM Add-ins page.

Accessing the UFT in Windows 8.X or Higher Operating Systems

By default, the Run Results Viewer is not displayed on the **Start** or **Apps** screen in Windows 8.x or higher.

You can add the Run Results Viewer to the Start screen by navigating to its location on the file system or the **Desktop** screen and pinning it to the **Start** screen.

In addition, you can add other related files, including .exe files and documentation the **Start** screen by pinning these items.

> **Note:** By default, the Start and Apps screens on Windows 8.x or higher are set to open Internet Explorer in Metro Mode. However, if User Account Control is turned off on your computer, Windows 8 will not open Internet Explorer in Metro mode. Therefore, if you try to open an HTML shortcut from the Start or Apps screen, such as the UFT Help or Readme file, an error will be displayed.

> To solve this, you can change the default behavior of Internet Explorer so that it never opens in Metro mode. In the **Internet Properties** dialog box > **Programs** tab, select **Always in Internet Explorer on the desktop** for the **Choose how you open links** option. For more details, see http://support.microsoft.com/kb/2736601 and http://blogs.msdn.com/b/ie/archive/2012/03/26/launch-options-for-internet-explorer-10-on-windows-8.aspx.

Run Results - Understanding Step Results

In the run results, certain steps are displayed differently to help you gain a greater understanding of the results of your test run.

These include:

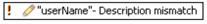
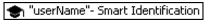
- Steps that used Smart Identification
- Checkpoint and output value steps
- Parameterized steps
- Calls to UFT API or Service Test tests/actions

Smart Identification in the Run Results (UFT GUI Testing Only)

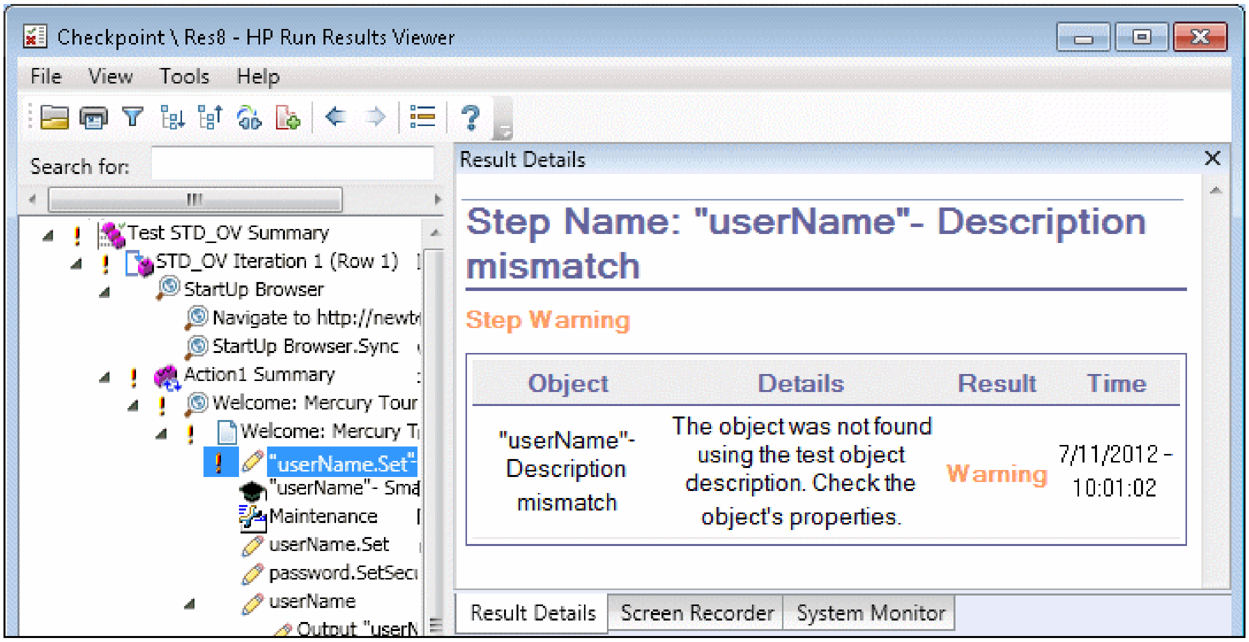
If the learned description does not enable UFT to identify the specified object in a step, and a Smart Identification definition is defined (and enabled) for the object, then UFT tries to identify the object using the Smart Identification mechanism. The following examples illustrate two possible scenarios.

No Object Matches the Learned Description

If UFT successfully uses Smart Identification to find an object after no object matches the learned description, the run results display a warning status and include the following information:

In the results tree	**In the Result Details pane**
A description mismatch icon for the missing object. For example:  ! "userName"- Description mismatch	An indication that the object (for example, the **userName WebEdit** object) was not found.
A Smart Identification icon for the missing object. For example:  "userName"- Smart Identification	An indication that the Smart Identification mechanism successfully found the object, and information on the properties used to find the object. You can use this information to modify the learned test object description, so that UFT can find the object using the description in future run sessions.
The actual step performed. For example: userName.Set	Normal result details for the performed step.

The image below shows the results for a test or component in which Smart Identification was used to identify the **userName WebEdit** object after one of the learned description property values changed.



Step Name: "userName"- Description mismatch

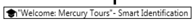
Step Warning

Object	Details	Result	Time
"userName"- Description mismatch	The object was not found using the test object description. Check the object's properties.	Warning	7/11/2012 - 10:01:02

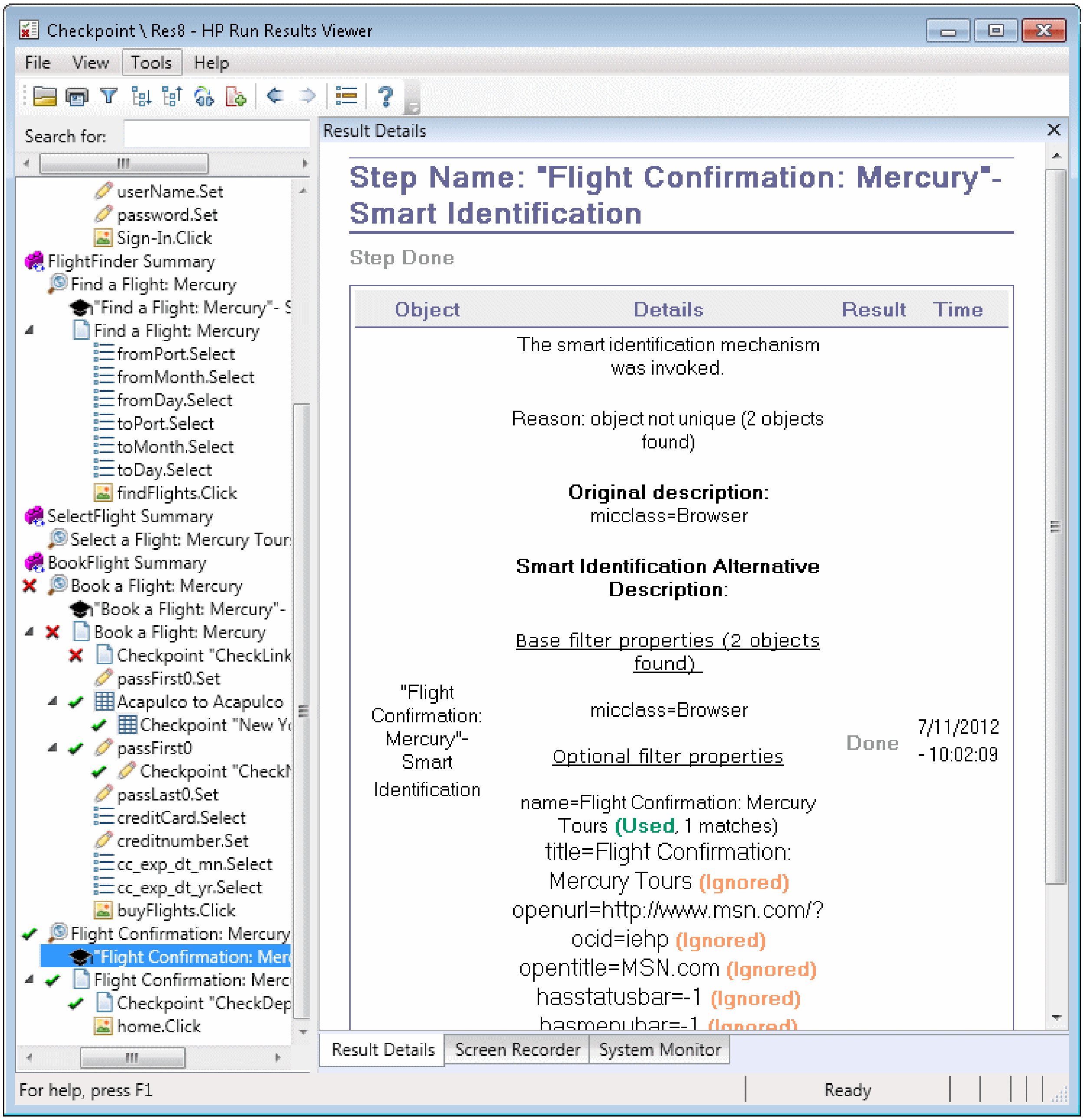
Multiple Objects Match the Learned Description

If UFT successfully uses Smart Identification to find an object after multiple objects are found that match the learned description, UFT shows the Smart Identification information in the Run Results Viewer. The step still receives a passed status, because in most cases, if Smart Identification was not used, the test object description plus the ordinal identifier could have potentially identified the object.

In such a situation, the run results show the following information:

In the results tree	In the Result Details pane
A Smart Identification icon for the missing object. For example:  "Welcome: Mercury Tours"- Smart Identification	An indication that the Smart Identification mechanism successfully found the object, and information on the properties used to find the object. You can use this information to create a unique object description for the object, so that UFT can find the object using the description in future run sessions.
The actual step performed. For example: home.Click	Normal result details for the performed step.

The image below shows the results for a test or component in which Smart Identification was used to uniquely identify the **Flight Confirmation: Mercury** object after the learned description resulted in multiple matches.



Step Name: "Flight Confirmation: Mercury"- Smart Identification

Step Done

Object	Details	Result	Time

"Flight Confirmation: Mercury"- Smart Identification	The smart identification mechanism was invoked.
	Reason: object not unique (2 objects found)
	Original description: micclass=Browser
	Smart Identification Alternative Description:
	Base filter properties (2 objects found)
	micclass=Browser
	Optional filter properties
	name=Flight Confirmation: Mercury Tours (Used, 1 matches)
	title=Flight Confirmation: Mercury Tours (Ignored)
	openurl=http://www.msn.com/?ocid=iehp (Ignored)
	opentitle=MSN.com (Ignored)
	hasstatusbar=-1 (Ignored)
	hasmenubar=-1 (Ignored)
		Done	7/11/2012 - 10:02:09

If the Smart Identification mechanism cannot successfully identify the object, the test or component fails and a normal failed step is displayed in the run results.

Checkpoint and Output Value results

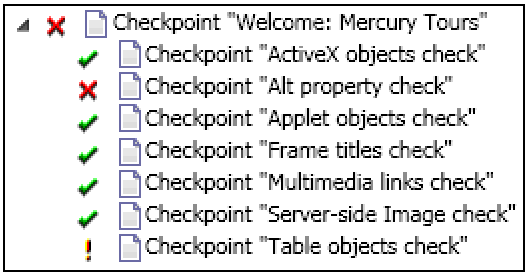
The information displayed in the Run Results Viewer and the available options are determined by the type of checkpoint or output value step you select.

- "Accessibility Checkpoints" below
- "Bitmap Checkpoints Results" on page 24
- "File Content Checkpoints" on page 25
- "Standard Checkpoint Results" on page 26
- "Table and Database Checkpoint Results" on page 27
- "Text and Text Area Checkpoint Results" on page 28
- "XML Checkpoint Results" on page 29
- "Output Value Results" on page 31
- "File Content Output Values" on page 32
- "XML Output Values" on page 32

Accessibility Checkpoints

When you include accessibility checkpoints in your test, the Run Results Viewer displays the results of each accessibility option that you checked.

The run results tree displays a separate step for each accessibility option that was checked in each checkpoint. For example, if you selected all accessibility options, the run results tree for an accessibility checkpoint may look something like this:



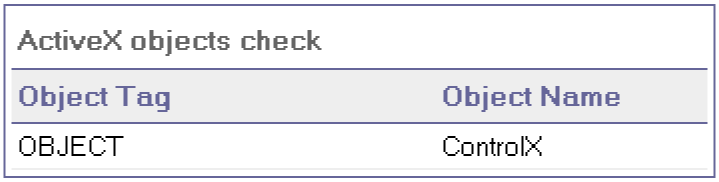
✗ Checkpoint "Welcome: Mercury Tours"
- ✓ Checkpoint "ActiveX objects check"
- ✗ Checkpoint "Alt property check"
- ✓ Checkpoint "Applet objects check"
- ✓ Checkpoint "Frame titles check"
- ✓ Checkpoint "Multimedia links check"
- ✓ Checkpoint "Server-side Image check"
- ! Checkpoint "Table objects check"

The run result details provide information that can help you pinpoint parts of your Web site that may not conform to the W3C Web Content Accessibility Guidelines. The information provided for each check is based on the W3C requirements.

> **Note:** Some of the W3C Web Content Accessibility Guidelines that are relevant to accessibility checkpoints are cited or summarized in the following sections. This information is not comprehensive. When checking whether your Web site satisfies the W3C Web Content Accessibility Guidelines, you should see the complete document at: http://www.w3.org/TR/WAI-WEBCONTENT/.

ActiveX Check

Guideline 6 of the W3C Web Content Accessibility Guidelines requires you to ensure that pages are accessible even when newer technologies are not supported or are turned off. When you select the ActiveX check, UFT checks whether the selected page or frame contains any ActiveX objects. If it does not contain any ActiveX objects, the checkpoint passes. If the page or frame does contain ActiveX objects then the results display a warning and a list of the ActiveX objects so that you can check the accessibility of these pages on browsers without ActiveX support. For example:

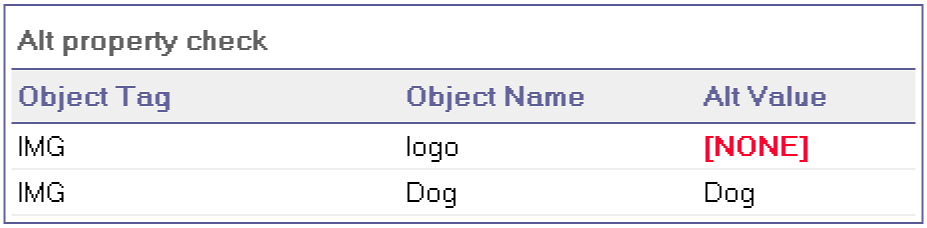


ActiveX objects check

Object Tag	Object Name
OBJECT	ControlX

Alt Property Check

Guideline 1.1 of the W3C Web Content Accessibility Guidelines requires you to provide a text equivalent for every non-text element. The Alt property check checks whether objects that require the Alt property under this guideline, do in fact have this attribute. If the selected frame or page does not contain any such objects, or if all such objects have the required attribute, the checkpoint passes. If one or more objects that require the property do not have it, the test fails and the run result details display a list that shows which objects are lacking the attribute. For example:



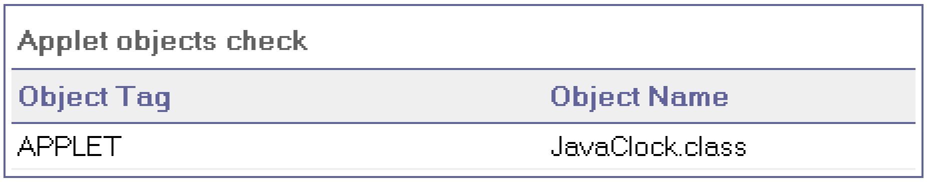
Alt property check

Object Tag	Object Name	Alt Value
IMG	logo	[NONE]
IMG	Dog	Dog

The Captured Data pane displays the captured page or frame, so that you can see the objects listed in the Alt property check list.

Applet Check

The Applet Check also helps you ensure that pages are accessible, even when newer technologies are not supported or are turned off (Guideline 6 of the W3C Web Content Accessibility Guidelines), by finding any Java applets or applications in the checked page or frame. The checkpoint passes if the page or frame does not contain any Java applets or applications. Otherwise, the results display a warning and a list of the Java applets and applications. For example:

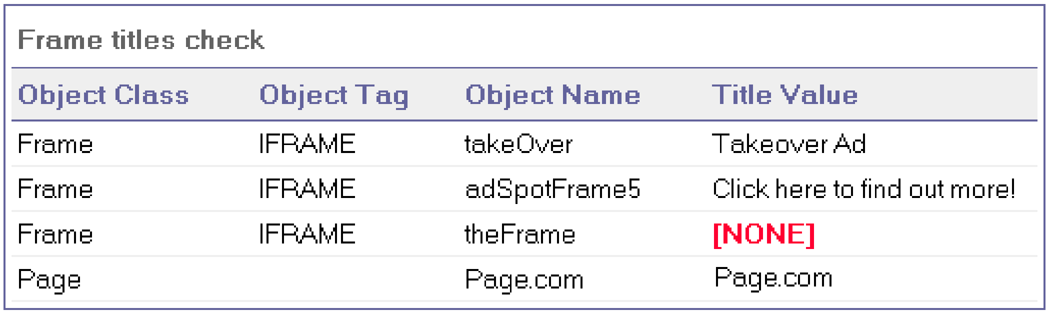


Applet objects check

Object Tag	Object Name
APPLET	JavaClock.class

Frame Titles Check

Guideline 12.1 of the W3C Web Content Accessibility Guidelines requires you to title each frame to facilitate frame identification and navigation. When you select the Frame Titles check, UFT checks whether Frame and Page objects have the **TITLE** tag. If the selected page or frame and all frames within it have titles, the checkpoint passes. If the page, or one or more frames, do not have the tag, the test fails and the run result details display a list that shows which objects are lacking the tag. For example:



Frame titles check

Object Class	Object Tag	Object Name	Title Value
Frame	IFRAME	takeOver	Takeover Ad
Frame	IFRAME	adSpotFrame5	Click here to find out more!
Frame	IFRAME	theFrame	[NONE]
Page		Page.com	Page.com

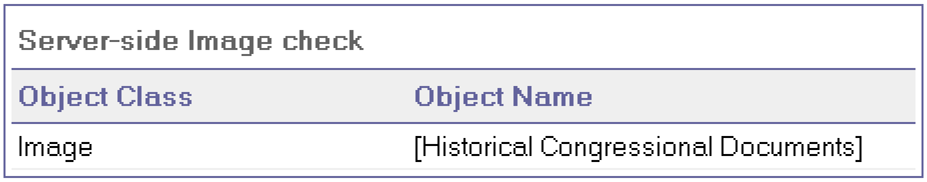
The Captured Data pane displays the captured page or frame, so that you can see the frames listed in the Frame Titles check list.

Multimedia Links Check

Guidelines 1.3 and 1.4 of the W3C Web Content Accessibility Guidelines require you to provide an auditory, synchronized description of the visual track of a multimedia presentation. Guideline 6 requires you to ensure that pages are accessible, even when newer technologies are not supported or are turned off. The Multimedia Links Check identifies links to multimedia objects so that you can confirm that alternate links are available where necessary. The checkpoint passes if the page or frame does not contain any multimedia links. Otherwise, the results display a warning and a list of the multimedia links.`

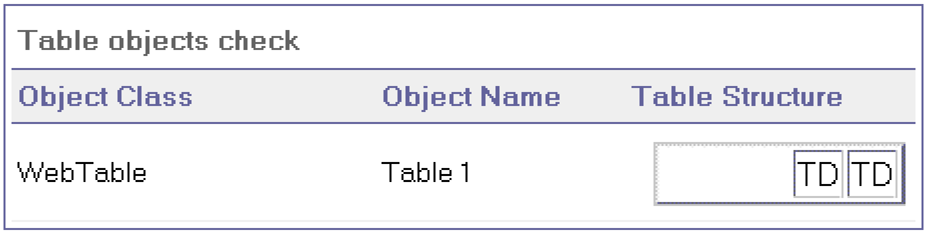
Server-Side Image Check

Guideline 1.2 of the W3C Web Content Accessibility Guidelines requires you to provide redundant text links for each active region of a server-side image map. Guideline 9.1 recommends that you provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape. When you select the Server-side Image check, UFT checks whether the selected page or frame contains any server-side images. If it does not, the checkpoint passes. If the page or frame does contain server-side images, then the results display a warning and a list of the server-side images so that you can confirm that each one answers the guideline requirements. For example:



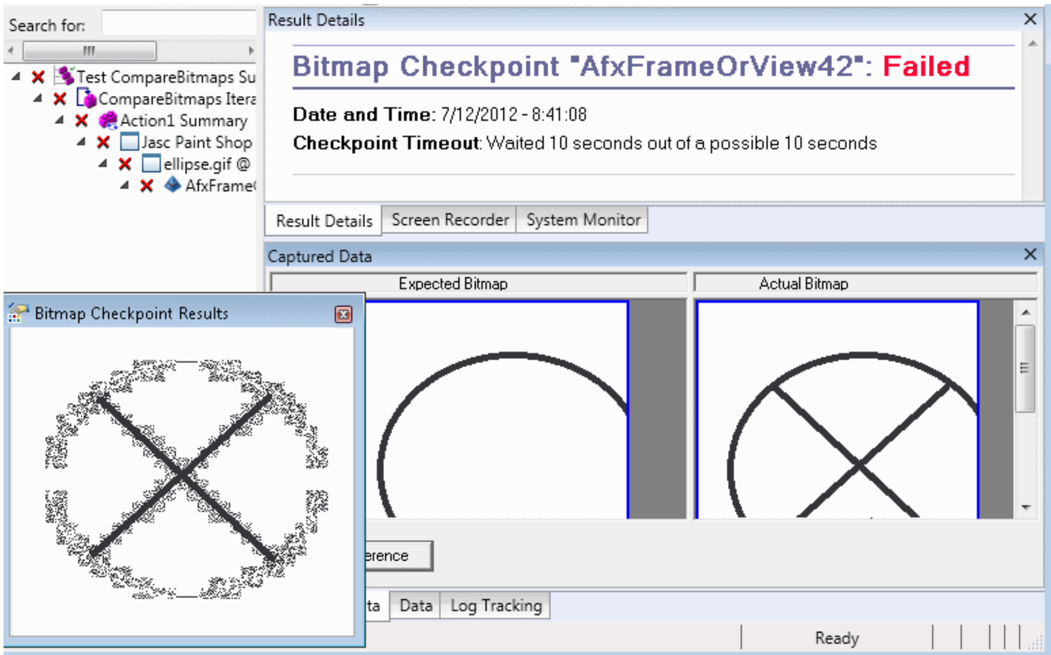
Server-side Image check

Object Class	Object Name
Image	[Historical Congressional Documents]

<table>
<tr>
<td><b>Tables Check</b></td>
<td>
<p>Guideline 5 of the W3C Web Content Accessibility Guidelines requires you to ensure that tables have the necessary markup to be transformed by accessible browsers and other user agents. It emphasizes that you should use tables primarily to display truly tabular data and to avoid using tables for layout purposes unless the table still makes sense when linearized. The <b>TH</b>, <b>TD</b>, <b>THEAD</b>, <b>TFOOT</b>, <b>TBODY</b>, <b>COL</b>, and <b>COLGROUP</b> tags are recommended so that user agents can help users to navigate among table cells and access header and other table cell information through auditory means, speech output, or a Braille display.</p>
<p>The Tables Check checks whether the selected page or frame contains any tables. If it does not, the checkpoint passes. If the page or frame does contain tables, the results display a warning and a visual representation of the tag structure of the table. For example:</p>

<p>Table objects check</p>
<table>
<tr><th>Object Class</th><th>Object Name</th><th>Table Structure</th></tr>
<tr><td>WebTable</td><td>Table 1</td><td>TD TD</td></tr>
</table>
</td>
</tr>
</table>

Bitmap Checkpoints Results

The Result Details pane displays the checkpoint step results, including its status (**Passed** or **Failed**), the date and time the checkpoint was run and the portion of the checkpoint timeout interval that was used (if any).

When Comparing Expected Bitmaps with Actual Bitmaps	The Captured Data pane shows the expected and actual bitmaps that were compared during the run session, and a **View Difference** button. When you click the **View Difference** button, UFT opens the Bitmap Checkpoint Results window, displaying an image that represents the difference between the expected and actual bitmaps. This image is a black-and-white bitmap that contains a black pixel for every pixel that is different in the two images. Similar results would be displayed for a component.  Search for: Test CompareBitmaps Su CompareBitmaps Itera Action1 Summary Jasc Paint Shop ellipse.gif @ AfxFrame Result Details Bitmap Checkpoint Results Result Details Screen Recorder System Monitor Captured Data Expected Bitmap Actual Bitmap erence ta Data Log Tracking Ready Bitmap Checkpoint "AfxFrameOrView42": Failed **Date and Time**: 7/12/2012 - 8:41:08 **Checkpoint Timeout**: Waited 10 seconds out of a possible 10 seconds
When Locating Specified Bitmaps in Actual Bitmaps	The Captured Data pane shows the actual bitmap of the runtime object in the application and the source bitmap that UFT attempted to locate within the object. It may also show the coordinates of a possible candidate that was found, and the image similarity percentage used to find the candidate. **Note:** By default, the information in the Captured Data pane is available only if the bitmap checkpoint fails. You can change the conditions for when bitmaps are saved in the run results, using the **Save still image captures to results** option in the **Screen Capture** pane (**Tools > Options > GUI Testing** tab **> Screen Capture** node) of the Options dialog box.

Considerations

- When comparing bitmaps, if the checkpoint is defined to compare only specific areas of the bitmap, the run results display the actual and expected bitmaps with the selected area highlighted.
- When comparing bitmaps, if the dimensions of the actual and expected bitmaps are different, UFT fails the checkpoint without comparing the bitmaps. In this case the **View Difference** functionality is not available in the results.
- The **View Difference** functionality is not available when viewing results generated in a version of QuickTest earlier than 10.00.
- If the bitmap checkpoint is performed by a custom comparer:
  - UFT passes the bitmaps to the custom comparer for comparison even if their dimensions are different.
  - The Result Details pane also displays the name of the custom comparer (as it appears in the **Comparer** box in the Bitmap Checkpoint Properties dialog box), and any additional information provided by the custom comparer.
  - The difference bitmap is provided by the custom comparer.

File Content Checkpoints

The Result Details pane displays detailed results of the selected checkpoint, including its status (**Passed** or **Failed)**, and the date and time the checkpoint was run. It also displays the number of lines that were checked, the number of changes found in the checked lines, and the total number of changed lines found in the file (including both the lines that were selected in the checkpoint and the lines that were not).

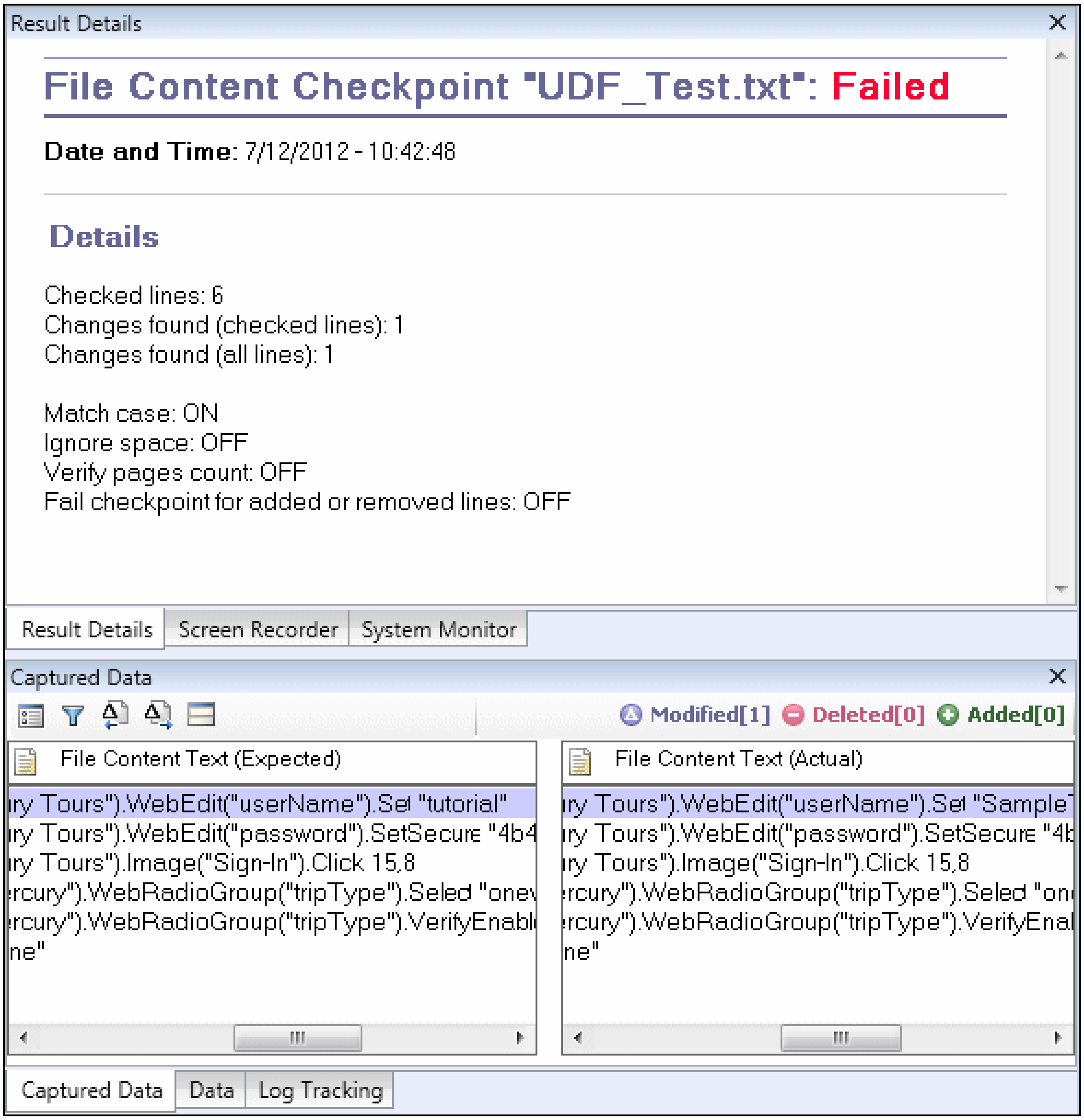
The details area also specifies whether the checkpoint includes the following options: **Match case**, **Ignore spaces**, **Verify page count**, and **Fail checkpoint for added or removed lines**

For failed steps, the Captured Data pane displays any differences found for all lines in the actual file, regardless of whether they were selected for comparison in the checkpoint. An asterisk (*) adjacent to the line number indicates that a regular expression was selected for comparison against the actual file.

In the following example, the details of the failed checkpoint indicate that the expected results and the current results do not match.

- The expected value in line 1 is not identical to the actual value.
- The asterisk (*) in line 1 indicates that a regular expression was selected for comparison against the actual file *and* the lines in the expected and actual files are different.
- Lines that were selected for comparison are displayed in black.
  Lines that were not selected for comparison are displayed in light grey.
- The last line exists in the source (expected) file but is missing from the actual file. This line was not selected for comparison, so its textual content is displayed in grey instead of black.

- No lines were added to the actual file.



File Content Checkpoint "UDF_Test.txt": Failed

Date and Time: 7/12/2012 - 10:42:48

Details

Checked lines: 6
Changes found (checked lines): 1
Changes found (all lines): 1

Match case: ON
Ignore space: OFF
Verify pages count: OFF
Fail checkpoint for added or removed lines: OFF

File Content Text (Expected)

```
ıry Tours").WebEdit("userName").Set "tutorial"
ıry Tours").WebEdit("password").SetSecure "4b4
ıry Tours").Image("Sign-In").Click 15,8
ercury").WebRadioGroup("tripType").Select "onew
ercury").WebRadioGroup("tripType").VerifyEnabl
ne"
```

File Content Text (Actual)

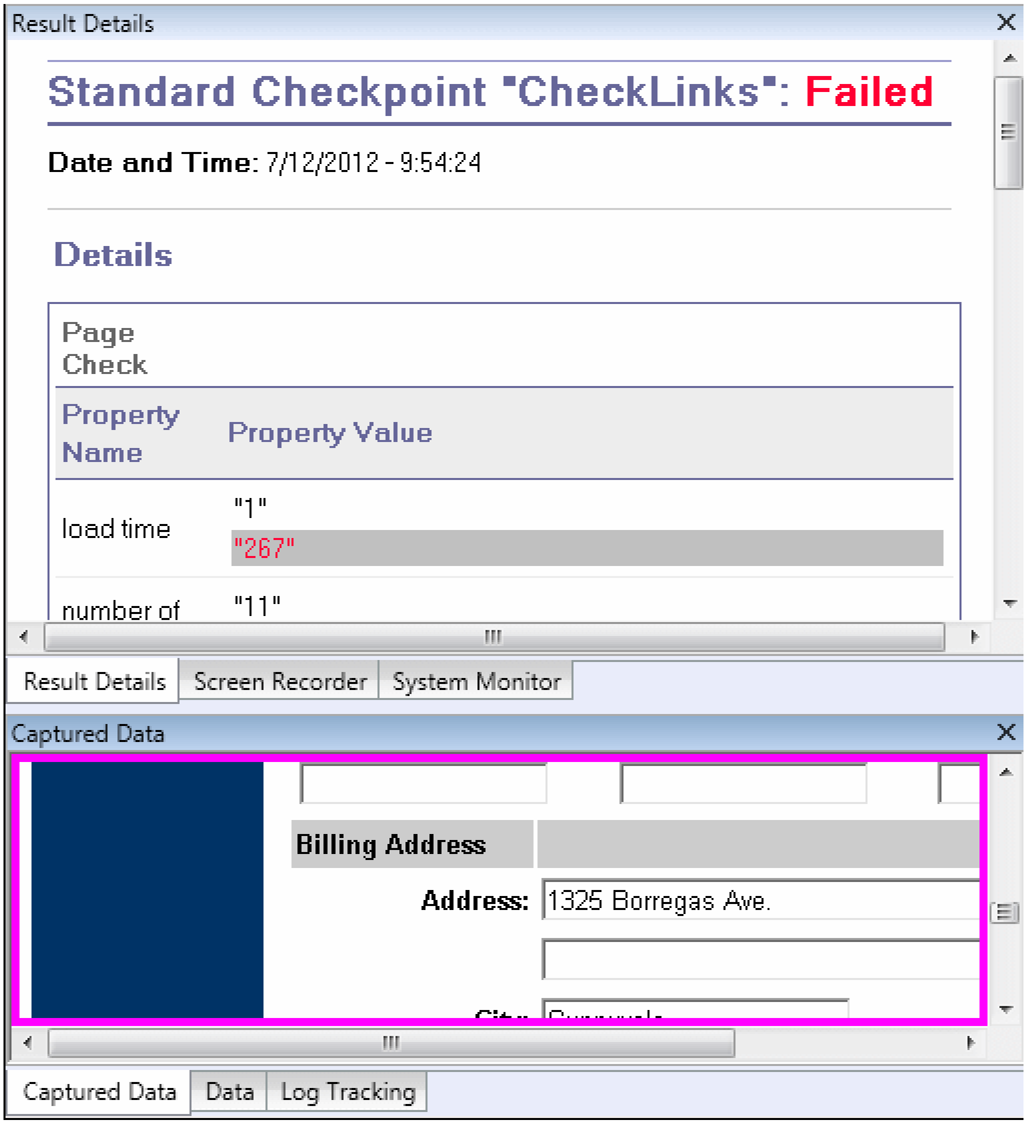
```
ıry Tours").WebEdit("userName").Set "SampleT
ıry Tours").WebEdit("password").SetSecure "4b
ıry Tours").Image("Sign-In").Click 15,8
ercury").WebRadioGroup("tripType").Select "on
ercury").WebRadioGroup("tripType").VerifyEnal
ne"
```

Standard Checkpoint Results

The Result Details pane displays detailed results of the selected checkpoint, including its status (**Passed** or **Failed)**, the date and time the checkpoint was run, and the portion of the checkpoint timeout interval that was used (if any). It also displays the values of the object properties that are checked, and any differences between the expected and actual property values.

The Captured Data pane displays the image capture for the checkpoint step (if available).

In the following example, the details of the failed checkpoint indicate that the expected results and the current results do not match. The expected value of the flight departure is **Paris**, but the actual value is **Frankfurt.**



Standard Checkpoint "CheckLinks": Failed

Date and Time: 7/12/2012 - 9:54:24

Details

Page Check

Property Name	Property Value
load time	"1" "267"
number of	"11"

Billing Address

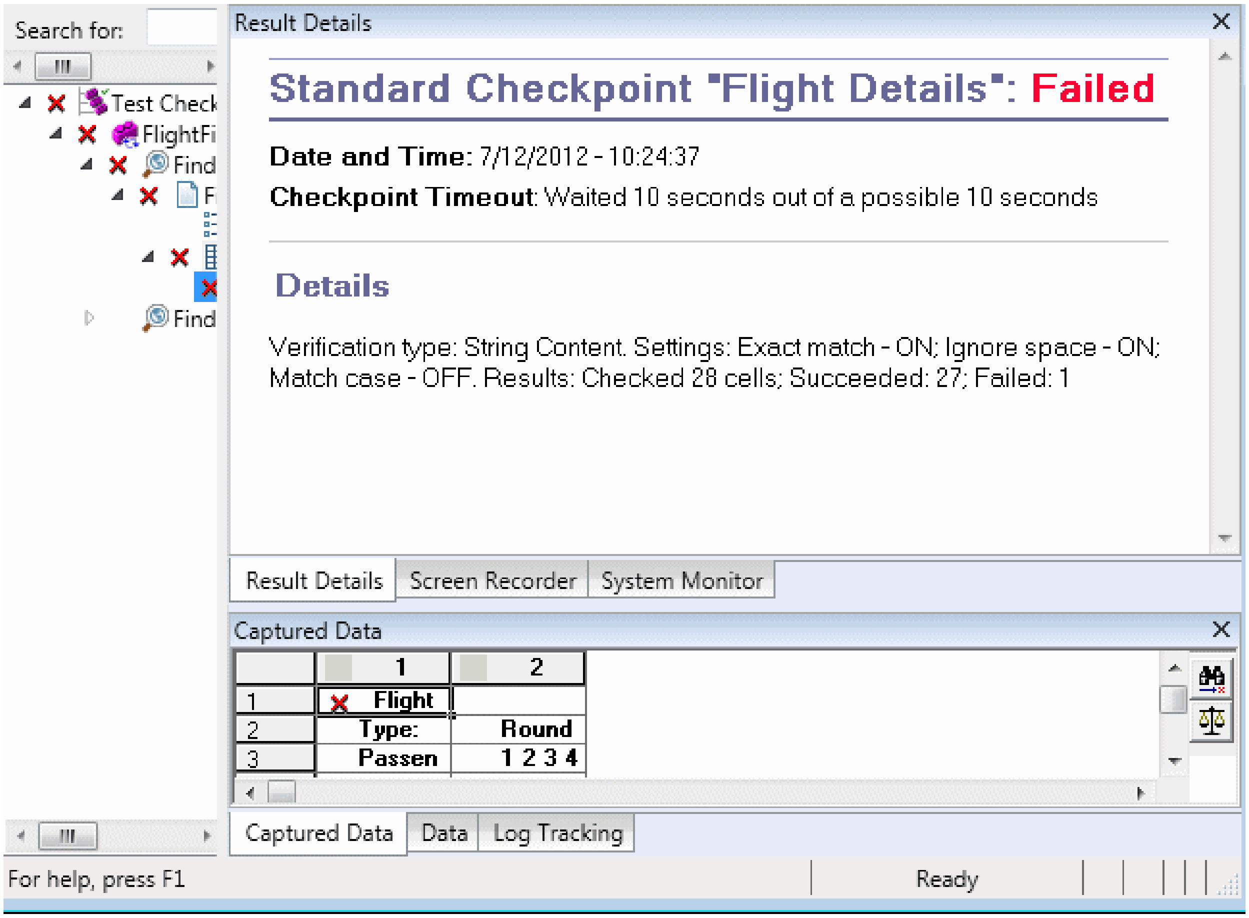
Address: 1325 Borregas Ave.

Table and Database Checkpoint Results

The results displayed for table and database checkpoints are similar. The Result Details pane displays the checkpoint step results, including its status (**Passed** or **Failed**), the date and time the checkpoint was run, the verification settings you specified for the checkpoint, and the number of individual table cells or database records that passed and failed the checkpoint.

If the checkpoint failed, the Captured Data pane shows the table cells or database records that were checked by the checkpoint. Cell values or records that were checked are displayed in black; cell values or records that were not checked are displayed in gray. Cells or records that failed the checkpoint are marked with a failed ✗ icon.

The following is an example of the results for a table checkpoint:



Standard Checkpoint "Flight Details": Failed

Date and Time: 7/12/2012 - 10:24:37

Checkpoint Timeout: Waited 10 seconds out of a possible 10 seconds

Details

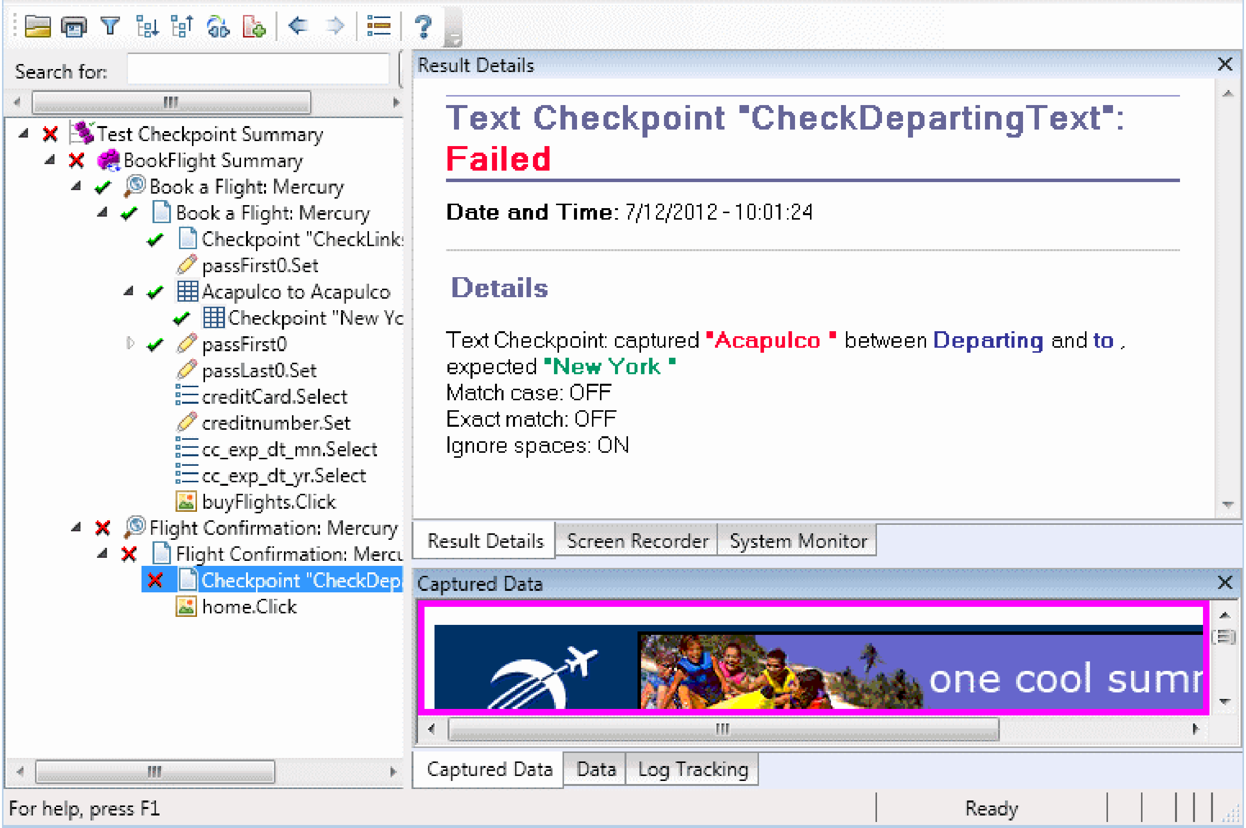
Verification type: String Content. Settings: Exact match - ON; Ignore space - ON; Match case - OFF. Results: Checked 28 cells; Succeeded: 27; Failed: 1

	1	2
1	✗ Flight	
2	Type:	Round
3	Passen	1 2 3 4

Text and Text Area Checkpoint Results

The Result Details pane displays the checkpoint step results, including its status (**Passed** or **Failed**), the date and time the checkpoint was run and the portion of the checkpoint timeout interval that was used (if any). It also shows the expected text and actual text that was checked, and the verification settings you specified for the checkpoint.

The following is an example of the results for a text checkpoint:



Text Checkpoint "CheckDepartingText": Failed

Date and Time: 7/12/2012 - 10:01:24

Details

Text Checkpoint: captured "**Acapulco** " between **Departing** and **to** ,
expected "**New York** "
Match case: OFF
Exact match: OFF
Ignore spaces: ON

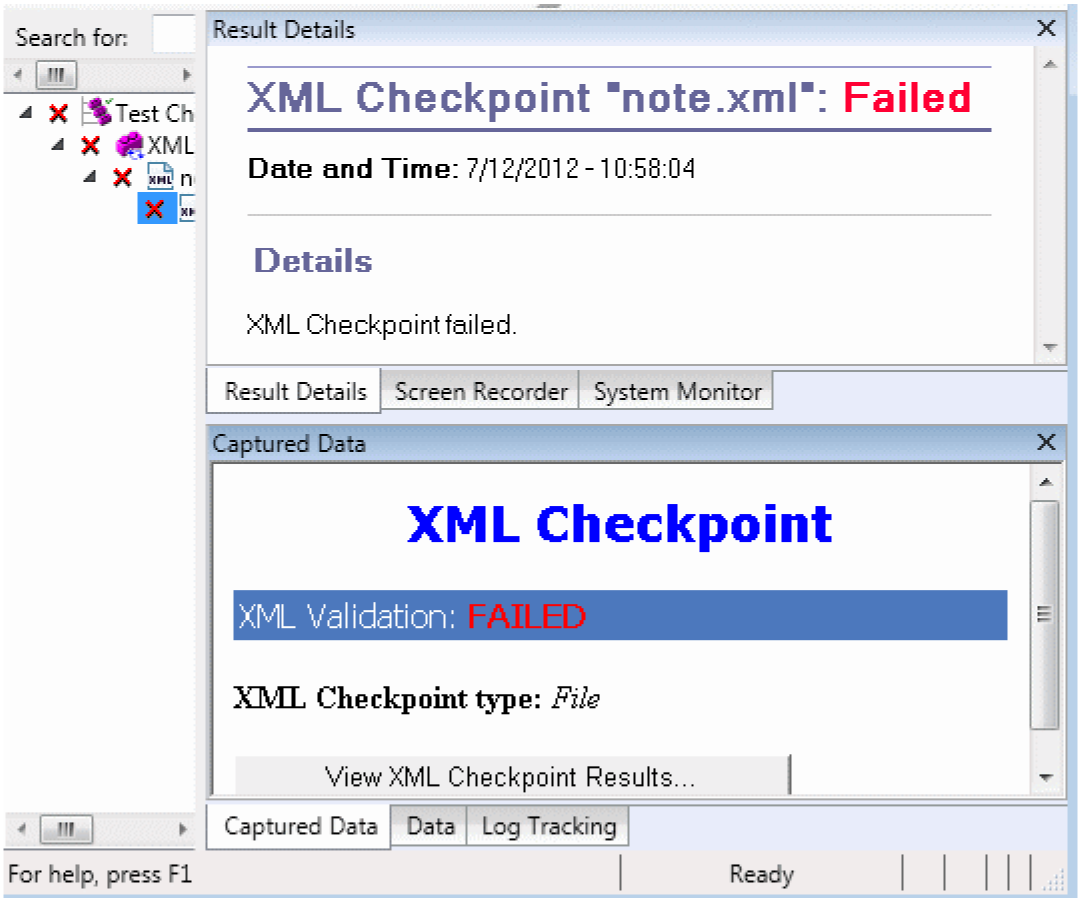
XML Checkpoint Results

The Result Details pane displays the checkpoint step results.

The Captured Data pane shows the details of the schema validation (if applicable) and a summary of the checkpoint results. If the schema validation failed, the reasons for the failure are also shown.

If the checkpoint failed, you can view details of each check performed in the checkpoint by clicking **View XML Checkpoint Results** in the Captured Data pane. The XML Checkpoint Results window opens, displaying details of the checkpoint's failure.

The following is an example of the results for an XML checkpoint:



XML Checkpoint "note.xml": Failed

Date and Time: 7/12/2012 - 10:58:04

Details

XML Checkpoint failed.

XML Checkpoint

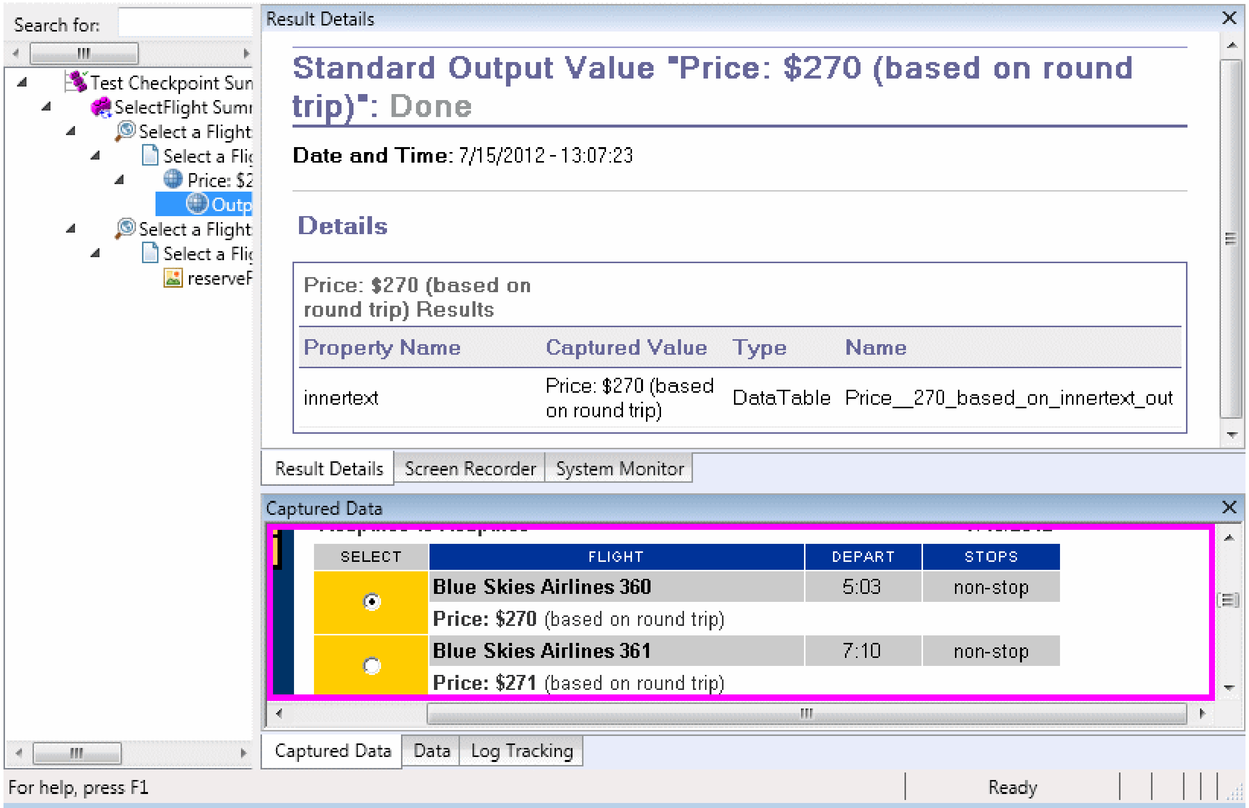
XML Validation: FAILED

XML Checkpoint type: *File*

Note: By default, if the checkpoint passes, the **View XML Checkpoint Results** button is not available. The availability of these detailed results is dependent on the **Save still image captures to results** setting in the **Screen Capture** pane (**Tools > Options > GUI Testing** tab **> Screen Capture** node) of the Options dialog box.

Output Value Results

The Result Details pane displays detailed results of the selected output value step, including its status, and the date and time the output value step was run. It also displays the details of the output value, including the value that was captured during the run session, its type, and its name, as shown in the following example. Similar results would be displayed for a component.



Standard Output Value "Price: $270 (based on round trip)": Done

Date and Time: 7/15/2012 - 13:07:23

Details

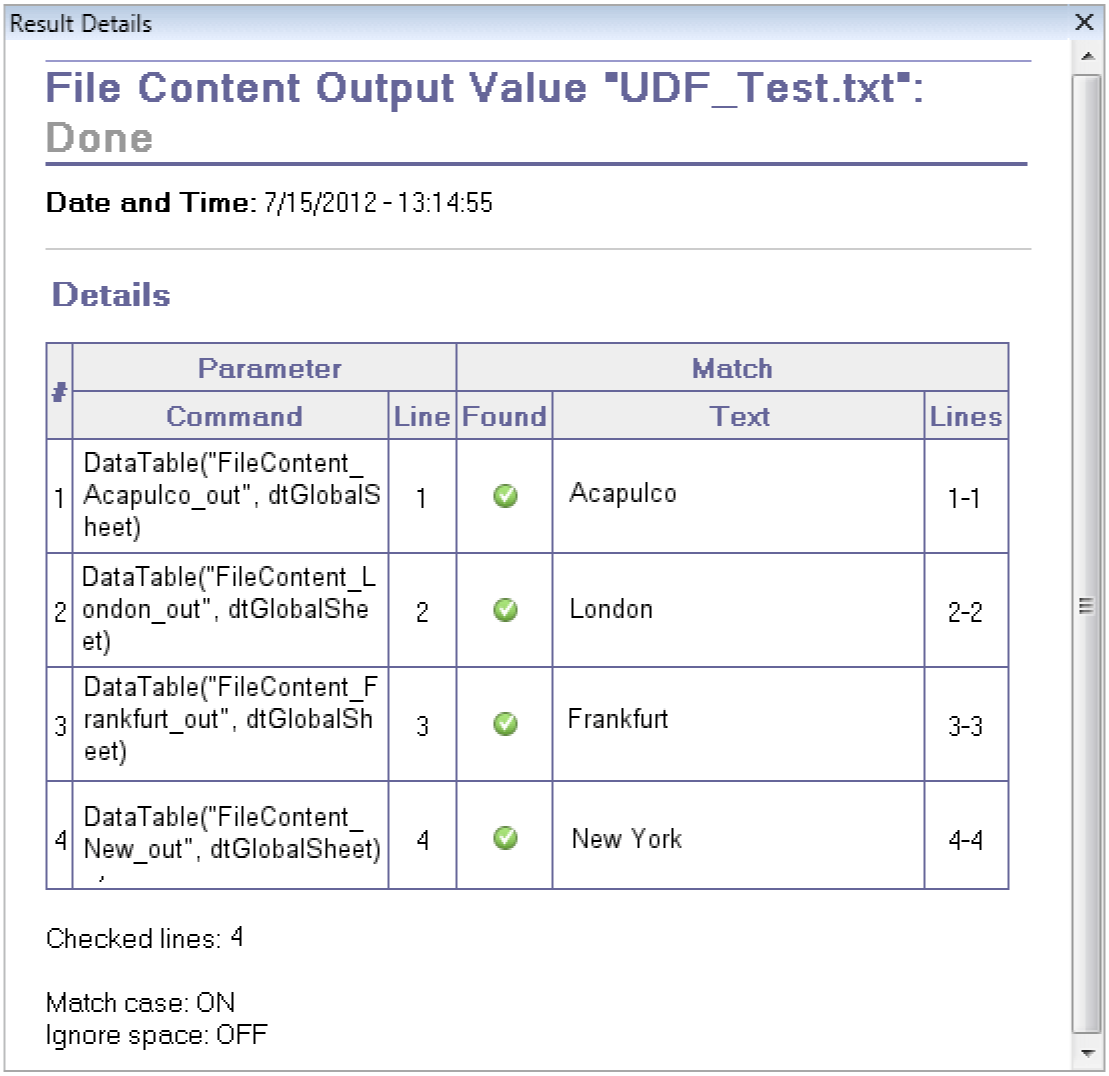
Price: $270 (based on round trip) Results

Property Name	Captured Value	Type	Name
innertext	Price: $270 (based on round trip)	DataTable	Price__270_based_on_innertext_out

SELECT	FLIGHT	DEPART	STOPS
●	**Blue Skies Airlines 360**	5:03	non-stop
	Price: $270 (based on round trip)		
○	**Blue Skies Airlines 361**	7:10	non-stop
	Price: $271 (based on round trip)		

File Content Output Values

The Result Details pane displays the results of the file content output value step, including its status, the date and time the step was run, and details for the parameterized file content output value. This pane also shows the configuration settings (whether **match case** and **ignore spaces** were set), as shown in the following example.



File Content Output Value "UDF_Test.txt": Done

Date and Time: 7/15/2012 - 13:14:55

Details

#	Parameter		Match		
	Command	Line	Found	Text	Lines
1	DataTable("FileContent_Acapulco_out", dtGlobalSheet)	1	✓	Acapulco	1-1
2	DataTable("FileContent_London_out", dtGlobalSheet)	2	✓	London	2-2
3	DataTable("FileContent_Frankfurt_out", dtGlobalSheet)	3	✓	Frankfurt	3-3
4	DataTable("FileContent_New_out", dtGlobalSheet)	4	✓	New York	4-4

Checked lines: 4

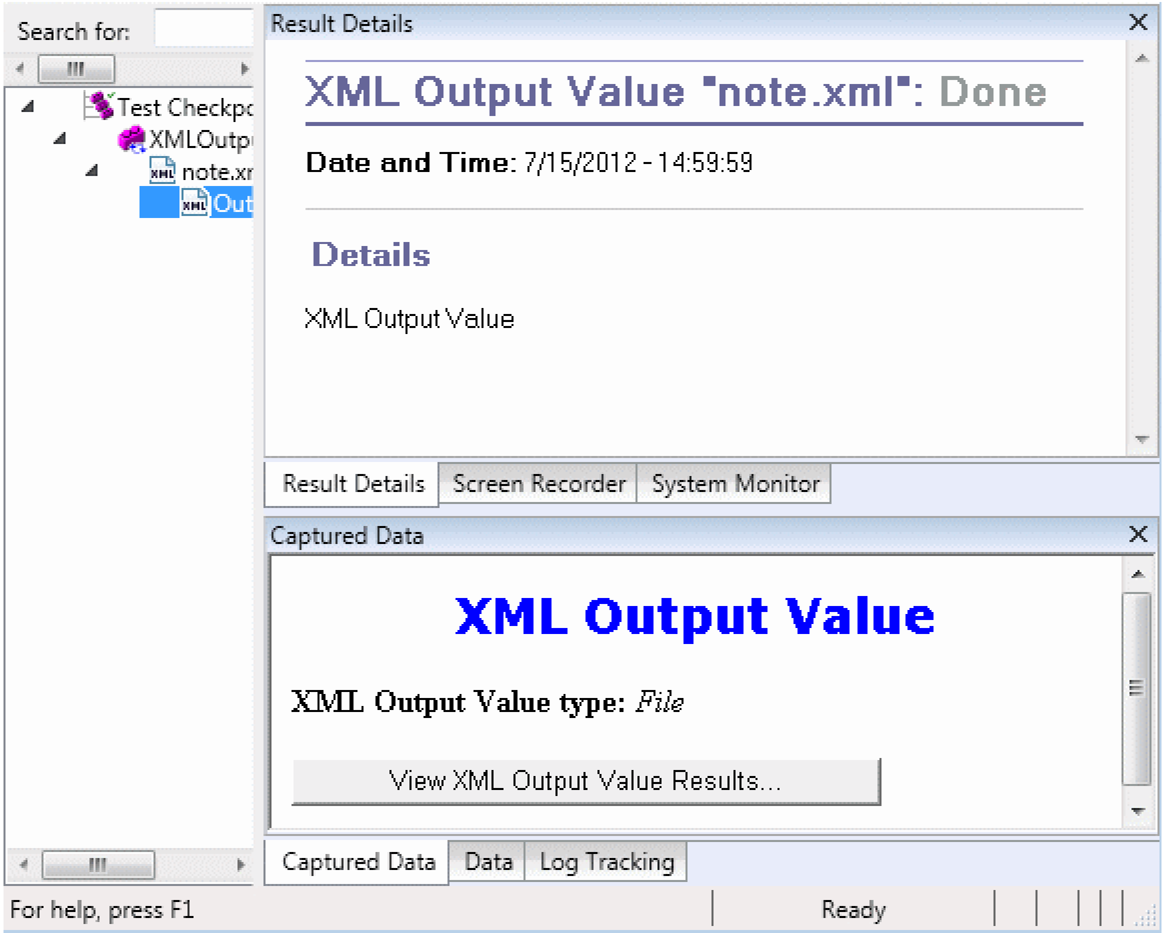
Match case: ON
Ignore space: OFF

XML Output Values

The Result Details pane displays a summary of the output value results.

From the Captured Data pane, you can view detailed results by clicking **View XML Output Value Results** to open the XML Output Value Results window.

The following is an example of the results for an XML output value:



XML Output Value "note.xml": Done

Date and Time: 7/15/2012 - 14:59:59

Details

XML Output Value

XML Output Value

XML Output Value type: *File*

View XML Output Value Results...

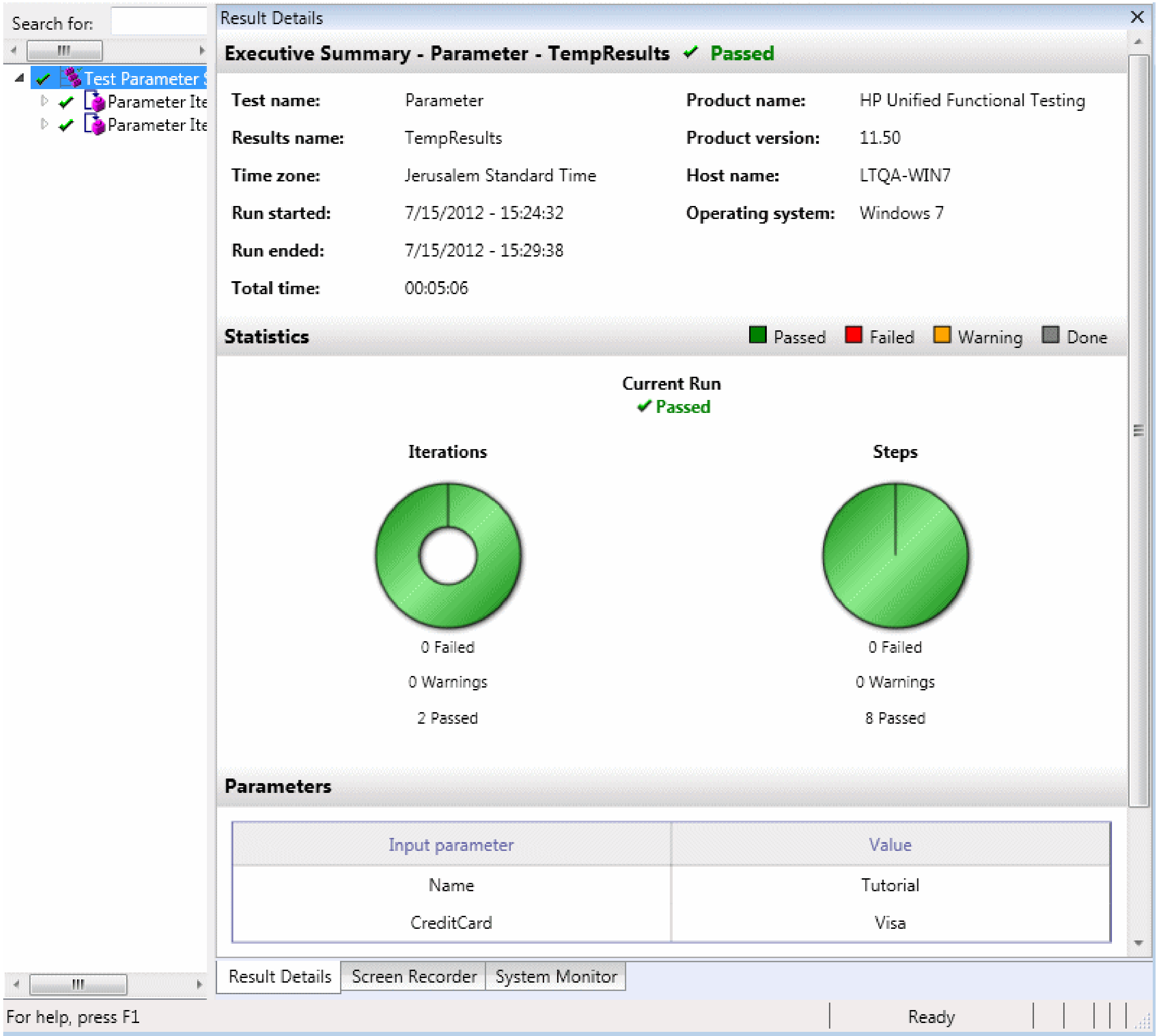
> **Note:** By default, the **View XML Output Value Results** button is available only when an error occurs. The availability of these detailed results is dependent on the **Save still image captures to results** setting in the **Screen Capture** pane (**Tools > Options > GUI Testing** tab **> Screen Capture** node) of the Options dialog box.

Parameterized values in the run results

A **parameter** is a variable that is assigned a value from an external data source or generator for a test, or from within a component. You can view the values for the parameters defined in your test or component in the Run Results Viewer.

To view parameterized values, expand the nodes in the run results tree and click the root node to view test or component input and output parameters, or click an action node that contains parameterized values (tests only).

Test and component parameters are displayed in the **Parameters** section of the Executive Summary area of the Results Details pane, which you display by clicking the root node of the run results tree. The example below shows input test parameters. The **Parameters** section would be identical for input component parameters.



Executive Summary - Parameter - TempResults ✓ Passed

Test name:	Parameter	**Product name:**	HP Unified Functional Testing
Results name:	TempResults	**Product version:**	11.50
Time zone:	Jerusalem Standard Time	**Host name:**	LTQA-WIN7
Run started:	7/15/2012 - 15:24:32	**Operating system:**	Windows 7
Run ended:	7/15/2012 - 15:29:38		
Total time:	00:05:06		

Statistics 🟩 Passed 🟥 Failed 🟧 Warning ⬛ Done

Current Run
✓ Passed

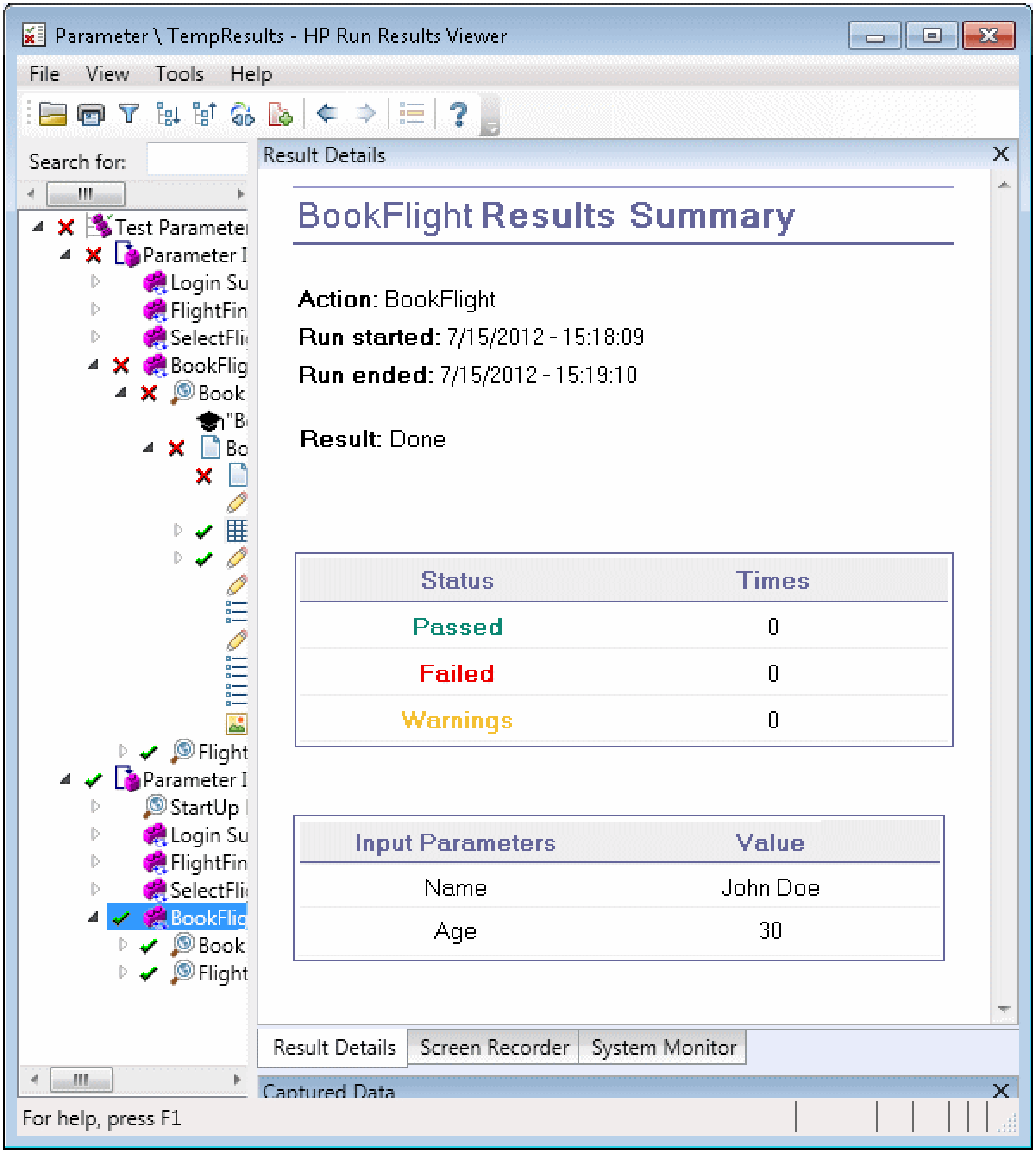
Iterations	Steps
0 Failed	0 Failed
0 Warnings	0 Warnings
2 Passed	8 Passed

Parameters

Input parameter	Value
Name	Tutorial
CreditCard	Visa

If output test or component parameters were defined, they would be displayed in this pane beneath the input parameters.

For action parameters, the name and value of the input and output parameters are displayed in the Result Details pane. Similar results would be displayed for a component.



BookFlight Results Summary

Action: BookFlight
Run started: 7/15/2012 - 15:18:09
Run ended: 7/15/2012 - 15:19:10

Result: Done

Status	Times
Passed	0
Failed	0
Warnings	0

Input Parameters	Value
Name	John Doe
Age	30

The example above shows input parameters that were defined at the action level. If output parameters were defined at this level, they would also be displayed in this pane.W

GUI Tests containing calls to UFT API /Service Test Tests

If your test contains a call to a UFT API or Service Test test, you can view the results of that test in the run results. The run results tree displays all of the GUI test-specific nodes that preceded the call to the UFT API or Service Test test, all of the UFT API or Service Test test-specific nodes from that test call, and all of the GUI test-specific nodes that followed that call.

Send Us Feedback

Let us know how we can improve your experience with the User Guide.

Send your email to: docteam@hpe.com

Hewlett Packard
Enterprise

Java™
COMPATIBLE