Object SPY - Troubleshooting Tool
CBTA 3.0 SP11
1 Table of Contents

2 Object Spy .............................................................................................................................................. 5

3 Spying SAP GUI Transactions .............................................................................................................. 6
  3.1 Selecting a SAP GUI Session ........................................................................................................... 6
  3.2 Spying Embedded HTML Content ................................................................................................. 8

4 Spying Web Sessions ............................................................................................................................ 12
  4.1 Selecting an Internet Explorer Session ........................................................................................... 12
  4.2 Spying the HTML Content ............................................................................................................. 13
  4.3 Spying Table UI Element .............................................................................................................. 15
  4.4 Delayed Activation of the Spy Mode ............................................................................................... 16

5 Troubleshooting URI Issues .................................................................................................................. 18
  5.1 URI Alternatives ............................................................................................................................. 18
  5.2 URI Verification ............................................................................................................................... 19
  5.3 URI Resolution Details .................................................................................................................... 20
  5.4 Troubleshooting URI Issues .......................................................................................................... 21
  5.5 Regular Expressions ......................................................................................................................... 23

6 References ................................................................................................................................................ 25
  6.1 Documentations ............................................................................................................................... 25
  6.2 List of SAP Notes .............................................................................................................................. 25

7 Table of Figures ....................................................................................................................................... 26
2 Object Spy

The Object Spy is a tool that CBTA delivers to troubleshoot test execution issues.

It can be launched with the button “Get Technical UI Information”, which is visible when a component is selected in the Test Script tab of TCE.

The main purpose of the tool is to determine the URI of a UI control (or element) in the hierarchy of UI containers. It can access to:

- SAP GUI sessions
- Browser sessions

It lets you spy the UI controls of the application being tested and generates the URI for you. The test engineers can then copy the generated URI and paste it as input parameter, when using components that target a control.

For complex scenarios, the URI determined automatically might be ambiguous. The tool also provides the ability to troubleshoot the situation by checking URI alternatives.
3 Spying SAP GUI Transactions

3.1 Selecting a SAP GUI Session

Object Spy lets you select the UI technology of the application being tested. There is a tab for SAP GUI and one for the Web Applications running inside Internet Explorer.

![Object Spy - UI Technologies](image)

Figure 2: Object Spy - UI Technologies

When testing SAP GUI applications, the combo box lets you select the SAP logon session, and you can then spy on the SAP GUI window by pressing the Spy button.

You can then move the mouse over the SAP GUI controls, to highlight one. A red box surrounds the control that is pointed to by the cursor. You can click on it to display the control attributes in the properties panel of the Object Spy.
Example of the selection of a GuiToolbarControl control

You can, for instance, select a tool bar, as shown in the screenshot below.

Figure 3: SAP GUI Transaction - GuiToolbarControl example

Spying is only possible when scripting is enabled on the SAP GUI

Control Properties in the Object Spy

The properties of the selected control are displayed in a tree. The URI shown below identifies the toolbar.

Figure 4: Object Spy - URI Example

For composite controls (containers with child controls) the tree may also include additional information about the child controls. In our example, the toolbar includes several buttons.
3.2 Spying Embedded HTML Content

SAP GUI Transactions have the capacity to include the content of external Web applications via the GuiHTMLViewer control.

From a test automation perspective this kind of composition is quite challenging because the native SAP GuiScripting recorder does not record the actions performed against the embedded HTML content.

Embedded HTML Content Example

![Embedded HTML Content Example](image)

Recording the Embedded HTML Content

With CBTA 3.0 SP08, the test recorder detects automatically whether the transaction includes an GuiHTMLViewer container but it does not record it automatically. In such situation, a new button is shown in the test recorder toolbar.

![Test Recorder Toolbar - Attach to HTML Viewer](image)

The test engineer must explicitly click the “Attach to HTML Viewer” button and select the GuiHTMLViewer control that he/she wants to start recording.
Default Component CBTA_GUI Hv_StartWebController

The component CBTA_GUI Hv_StartWebController is the one used at runtime to get access to the HTML content.

This component provides the ability to attach the test player to the embedded Internet Explorer session and perform actions against the embedded HTML content.

Test Script Example

The screenshot below shows a typical example where the new component is used to get access to the HTML content. As shown here the test script can include components for testing SAP GUI applications (CBTA_GUI_*) but also components to perform actions on HTML elements (CBTA_WEB_*).

![Test Script Example](image)

Actions are implicitly performed against the embedded HTML content when using WEB Components. In this example a screenshot of the HTML content is capture using the CBTA_WEB_A_CAPTURESCREEN component. While the previous using the CBTA_GUI_A_CAPTURE screen was capturing the whole SAP GUI window.

Input Parameters

The component expects as input the URI of the GuiHTMLViewer control which can be determined using the Object Spy feature.

![Input Parameters](image)
Determining the URI of an Embedded HTML Element

Starting with SP5 the object spy provides the ability to also spy the embedded HTML content. A new item is available in the contextual menu when a GuiHTMLViewer control is selected.

This new menu item lets you jump to a different spy mode where the session of the embedded content is automatically selected (like shown below).
UI elements of the embedded HTML content can be selected and regular spying capabilities are then available. The collected information is displayed as usually in the tree.

Figure 11: Object Spy - HTML Content Properties
4 Spying Web Sessions

4.1 Selecting an Internet Explorer Session

The approach is the same for Web applications. The tool lets you select the Internet Explorer session displaying the HTML content of the application being tested.

The filtering of Internet Explorer sessions makes it easy to select the ones that are compliant with test automation requirements. These requirements might be different depending on the UI technology used by the application being tested.

- For CRM Web UIs the so-called Test Mode must be enabled.
- For Web Application based on the Unified Rendering Light Speed framework, the Stable ID is a pre-requisite.

![Image of Object Spy - Web Sessions with Test Mode]

Figure 12: Object Spy - Web Sessions with Test Mode

Note:

- Applications started manually are not shown by default.
- Applications must have been started from TCE (Test Composition Environment) – This to make sure the tool can generate consistent URI(s).
- With Internet Explorer 11, the IE5 (Quirks) mode is not supported.
4.2 Spying the HTML Content

Click the **Spy** button to start spying the application. You can move the mouse over the HTML content and the HTML element located under the mouse pointer gets highlighted like shown in the screenshot below.

![CBTA Test: ZCR_FIORI_CHECKPOINT_OPTIONS](image)

You can then click on it to get information about the HTML element and get its properties and attributes displayed in a tree and computes the value of the **URI** property. The **URI** is typically the information that the test script should use to identify the element at runtime. The test engineer may copy its value and paste it as input of one of the steps of a test script.
UI Technology-specific Attributes

The tool also tries to determine the underlying UI technology used by the application and may display attributes that are specific to that UI technology.

In the screenshot below the application being tested is a Web Dynpro Application based on the Unified Rendering Light Speed framework. The tool has been able to retrieve additional information and makes it visible in dedicated nodes such as the Light Speed Data node.

Figure 14: Object Spy - URI for a Web Dynpro UI Element

Figure 15: Object Spy - Light Speed Attributes
4.3 Spying Table UI Element

Some of the default components like CBTA_LS_T_FindRow expect as input parameter the URI of a table container. While spying, the selection of a table UI control is always a bit challenging because, most of the time, it has no border and the corresponding UI element cannot be selected.

In order to make it possible to spy for containers, one mode have been introduced with CBTA 3.0 SP09.

![Spying Table UI Element](image)

When the mode “relevant control” is checked, the controls only relevant for UI technology are highlighted using a different color. (only SAPUI5 and ABAP We Dynpro are supported)

- You may resume to the regular mode by uncheck special highlighting.

Here is an example.

![Object Spy - TABLE Mode](image)
Note that for some SAP UI technologies, the same information is also shown in a dedicated node. Here is an example with a Web Dynpro TABLE:

![Figure 17: Object Spy – URI of the Parent Web Dynpro Table](image)

### 4.4 Delayed Activation of the Spy Mode

Some HTML elements like contextual menu items or combobox items could not be spied because their container gets collapsed as soon as the browser loses the focus. The object spy has now the ability to postpone the activation of the spy mode thanks to a delay.

![Spy Delay](image)

**Example**

The delay gives the test engineer the opportunity to do something just before spying the HTML content. In this example the delay was necessary to let us open the combobox and thus spy its child items.
The result is then displayed and the `innerText` property confirms that the child element could be spied properly.
5 Troubleshooting URI Issues

5.1 URI Alternatives

Most of the generated URI(s) do rely on the ID attribute when searching for a UI element. Unfortunately, there is no guarantee that this information is stable at runtime. In other words, the value of the ID could be bound to the session and change each time the scenario is executed.

To troubleshoot such situation, the Object Spy includes a URI Helper tab where it is possible to check several URIs and search for alternative to using the ID.

![Figure 3B: Object Spy - URI Alternatives](image)

Three alternatives are proposed by the tool:

- By innerText ➔ URI relies on the value of the innerText property (if any)
- By title ➔ URI relies on title HTML attribute (if any) [This corresponds to the tooltip]
- By parentId ➔ URI relies on the ID of the parent element (if any)
5.2 URI Verification

It may happen that the URI matches several HTML elements. The “Locate” buttons let you verify whether the URI is consistent and matches the latest HTML element that has been spied.

Wiki Page Example

With this example the URI relying on the parent ID is ambiguous. It does not provide enough information to identify the expected element.

Pressing the locate button detects such situation and notifies the user with a clear feedback in the “Last Operation Result” status bar.

Figure 19: Object Spy - Wiki Page Example

Figure 20: Object Spy - URI Alternatives
5.3 URI Resolution Details

One may need to get additional information when the resolution does not find the expected element. The "Details" link lets you open a dialog showing technical information about the HTML element that has been found.

Resolution Details:

- The HTML attributes and properties are shown to let you understand which element has been found.
- The HTML content of the element is being displayed.
- The toolbar lets you navigate in the HTML document hierarchy to clarify what can be the root cause of the resolution problem.
5.4 Troubleshooting URI Issues

The purpose of the URI Helper is to assist test engineers when there is a need to use advanced capabilities of the URI syntax.

The URI syntax is quite flexible; it is for instance possible to:

- Search for a particular UI element using several HTML attributes
- Search using regular expressions (instead of providing the exact attribute value)

Note

For more details on the URI syntax, refer to the documentation:
CBTA Runtime Library & Default Components

The Object Spy lets you generate a URI and check it to make sure the syntax is correct.

Example

In this example, we want to search for the link below using part of its \texttt{href} attribute.

Once the element spied, the URI Helper table shows the HTML attributes and properties:

- One or several HTML attributes can be selected
- The appropriate regular expression is generated based on the operator selected
  - Equals
  - Contains
  - Starts
  - Ends
- The value of the attribute can be modified
Each selection or modification triggers the generation of a new URI like shown below:

![URI Generation using a Regular Expression](image)

Once the URI is generated, the “Locate” button let you check whether the generated URI is correct or not.

**Explanations**

With this example the URI resolver will search for an Anchor HTML element having its `href` attribute ending with `/support`.

- Note that the usage of the "html." prefix here
- `~=` informs the URI resolver that the value specified is a regular expression
- `/Support$` Regular Expression to check that the actual `href` value ends with `/Support`
- The `tag` property is selected by default – this is recommended to avoid ambiguities and to improve performance at runtime.
5.5 Regular Expressions

The URI Syntax is quite complex on its own; as you may know some characters have to be escaped using the token syntax (e.g.: %space% can be used to escape a space character). The situation is even more complex when using regular expressions because they have their own escaping logic. It is therefore recommended to first verify the regular expression syntax itself before trying to use it inside a Uri.

**Recommendation**

*Regular expressions can be checked online; a lot of websites provides this feature.*

**Example**

Here is an example showing a concrete use case where we search for a particular text ending with a number in parenthesis.

![Figure 25: Object Spy - URI Challenging Example](https://fot.wdf.sap.corp:44300/sap/bc/webdyne/sap/agw_work_item?VIEW_ID=TREP&SUBVIEW_ID=...)

Figure 25: Object Spy - URI Challenging Example
As shown in the previous screenshot the Object Spy provides several Uri alternatives. However, in some situation, none of the Uris are not stable enough to be used in a test script. This is a typical use-case where regular expressions are necessary.

Our goal is to search for the link using its innerText property knowing the number is not stable overtime.

Which of the following syntax is the correct one?

- `innerText~=^All Test Configurations ([0-9]+)\$; tag=A`
- `innerText~=^All Test Configurations \(([0-9]+)\); tag=A`

As you may know, parenthesis have a specific semantic when used in a regular expression. In this example they have to be escaped (using the backslash character) because they are part of the value that is to be checked. The correct syntax of the regular expression can be verified online as show below:

![Online Tools to check Regular Expression Syntax](image-url)

Figure 26: Online Tools to check Regular Expression Syntax
6 References

6.1 Documentations

Some additional documents have to be considered to take benefit of the CBTA capabilities.

<table>
<thead>
<tr>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBTA – User Guide</td>
</tr>
<tr>
<td>CBTA – Runtime Library and Default Components</td>
</tr>
<tr>
<td>CBTA – Test Recorder</td>
</tr>
</tbody>
</table>

6.2 List of SAP Notes

The following table list the SAP Notes mentioned in this document.

<table>
<thead>
<tr>
<th>SAP Note</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2133396</td>
<td>CBTA - Test Automation Tool - Object Spy - Session not compliant with test automation requirements</td>
</tr>
<tr>
<td>1666201</td>
<td>WebCUIF - Collective Note to enable the Test Mode for CRM Web UIs</td>
</tr>
<tr>
<td>2177107</td>
<td>Test Automation of Web Dynpro scenarios - Stable ID Mode not propagated to child windows</td>
</tr>
</tbody>
</table>
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TCE - Get Technical UI Information</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Object Spy - UI Technologies</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>SAP GUI Transaction - GuiToolbarControl example</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>Object Spy - URI Example</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>SAP GUI Transaction - Embedding HTML Content</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>Test Recorder Toolbar - Attach to HTML Viewer</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>TCE - Start Web Controller Example</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>TCE - Start Web Controller Parameters</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>Object Spy - Spying HTML Content</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>Object Spy - Embedded HTML Content</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>Object Spy - HTML Content Properties</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>Object Spy - Web Sessions with Test Mode</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>Object Spy - Highlighting UI Elements</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>Object Spy - URI for a Web Dynpro UI Element</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>Object Spy - Light Speed Attributes</td>
<td>14</td>
</tr>
<tr>
<td>16</td>
<td>Object Spy - TABLE Mode</td>
<td>15</td>
</tr>
<tr>
<td>17</td>
<td>Object Spy - URI of the Parent Web Dynpro Table</td>
<td>16</td>
</tr>
<tr>
<td>18</td>
<td>Object Spy - URI Alternatives</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>Object Spy - Wiki Page Example</td>
<td>19</td>
</tr>
<tr>
<td>20</td>
<td>Object Spy - URI Alternatives</td>
<td>19</td>
</tr>
<tr>
<td>21</td>
<td>Object Spy - URI Resolution Details</td>
<td>20</td>
</tr>
<tr>
<td>22</td>
<td>Object Spy - URI Resolution Feedback</td>
<td>20</td>
</tr>
<tr>
<td>23</td>
<td>Object Spy - Link with HREF Attribute Example</td>
<td>21</td>
</tr>
<tr>
<td>24</td>
<td>Object Spy - URI Generation using a Regular Expression</td>
<td>22</td>
</tr>
<tr>
<td>25</td>
<td>Object Spy - URI Challenging Example</td>
<td>23</td>
</tr>
<tr>
<td>26</td>
<td>Online Tools to check Regular Expression Syntax</td>
<td>24</td>
</tr>
</tbody>
</table>