# Table of Contents

1 Foreword .......................................................................................................................... 5

1.1 Important Notes ............................................................................................................ 6

1.2 Supported UI Technologies ......................................................................................... 7

1.3 Test Automation Prerequisites ..................................................................................... 7

2 Terminology ..................................................................................................................... 8

2.1 Test Script ..................................................................................................................... 8

2.2 Test Composition Environment .................................................................................... 8

2.3 System Under Test (SUT) ............................................................................................ 9

2.4 Test Profile .................................................................................................................. 9

2.5 Executables .................................................................................................................. 9

2.6 Component .................................................................................................................. 9

2.7 Test Recorder .............................................................................................................. 9

2.8 Test Player .................................................................................................................. 9

3 Configuration .................................................................................................................. 10

3.1 Front-End Configuration ............................................................................................. 10

   Logs and Traces .............................................................................................................. 10

   Recording Options ........................................................................................................ 10

   Browser Setup configuration ......................................................................................... 11

   Execution Settings ....................................................................................................... 12

   Other Settings .............................................................................................................. 14

3.2 Configuration Tray ....................................................................................................... 15

3.3 Self-checks .................................................................................................................. 15

3.4 SUT Management ....................................................................................................... 16

4 Test Automation Framework .......................................................................................... 17

4.1 Solution Manager Context .......................................................................................... 17

4.2 Creating a Test Configuration ..................................................................................... 17

   Executable Selection in Solution Documentation ..................................................... 17

   Selecting CBTA as Test Automation Tool ................................................................. 18

   Assigning an Executable ............................................................................................ 19

   Selecting the Test Profile .......................................................................................... 19

   Selecting a Login Schema ......................................................................................... 21

4.3 Recording a Business Scenario .................................................................................. 23
1 Foreword

Component Based Test Automation (CBTA) is tightly integrated into SAP Solution Manager 7.2. It therefore benefits from the new Solution Administration and Solution Documentation concepts.

SAP Solution Manager provides various tools and methods for implementing and maintaining both SAP and customer solutions. The Solution Documentation model is based on a hierarchical structure, which uses a solution as the single, central point of access for all Solution Documentation content.

**Recommendation**

*As a starting point, familiarize yourself with the new terms and concepts related to process management in SAP Solution Manager 7.2.*

- See Basic Terms and Concepts on SAP Help Portal

**Solution Administration**

You manage your solutions using the Solution Administration tool - transaction code SOLADM.

[Figure 1: Solution Administration – Transaction SOLADM]
You can start the Solution Documentation via the SAP Solution Manager launch pad. You can also use transaction code SOLDOC.

![Solution Manager - Transaction SOLDOC](image)

### 1.1 Important Notes

The main SAP notes related to CBTA are listed below:

<table>
<thead>
<tr>
<th>SAP Note</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2436142</td>
<td>CBTA - Supportability and Compatibility Matrix</td>
</tr>
<tr>
<td>1763697</td>
<td>CBTA - Installation Note</td>
</tr>
<tr>
<td>1778899</td>
<td>CBTA 3.0 - Release Documentation</td>
</tr>
<tr>
<td>2640485</td>
<td>CBTA 3.0 SP10 - Release Documentation</td>
</tr>
</tbody>
</table>
1.2 Supported UI Technologies

The supported UI technologies are the one below:

<table>
<thead>
<tr>
<th>UI Technology</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP GUI</td>
<td>UI Technology used by SAP Transactions</td>
</tr>
<tr>
<td></td>
<td>SAP GUI content displayed in MS Internet Explorer</td>
</tr>
<tr>
<td>Web / Plain HTML</td>
<td>UI Technology used by non-SAP applications</td>
</tr>
<tr>
<td>WebCUIF</td>
<td>UI Framework used mainly SAP CRM Web UIs.</td>
</tr>
<tr>
<td>SAP UI5 &amp; FIORI</td>
<td>UI Framework used by SAP UI5 and Fiori apps</td>
</tr>
<tr>
<td>Unified Rendering</td>
<td>UI Framework common to SAP Applications such as:</td>
</tr>
<tr>
<td>Light Speed (LS)</td>
<td>• Web Dynpro Application (ABAP)</td>
</tr>
<tr>
<td></td>
<td>• Web Dynpro Applications (Java)</td>
</tr>
<tr>
<td></td>
<td>• SAP GUI for HTML</td>
</tr>
<tr>
<td>BSP</td>
<td>Applications based on Business Server Pages</td>
</tr>
<tr>
<td>Java Web Dynpro</td>
<td>Layer formerly used by Java Web Dynpro applications.</td>
</tr>
</tbody>
</table>

1.3 Test Automation Prerequisites

The test automation logic may need to start the application in a specific mode. The settings differ depending on the UI technology used by the application to test.

The notes describing the technical prerequisites are listed below.

<table>
<thead>
<tr>
<th>SAP Note</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2436142</td>
<td>Additional Information: Component-based Test Automation Supportability</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>
2 Terminology

2.1 Test Script

A test script is an entity persisted in the repository of the Solution Manager system. The one generated by CBTA are objects, containing:

- A list of steps to simulate user interactions
- Each step may refer to a component or another CBTA test script
- Each step may have input and output parameters.

CBTA test scripts are associated to an executable. At runtime, this information is combine with the Solution Manager Context to determine the target system and how to start the application.

2.2 Test Composition Environment

The Test Composition Environment (TCE) is the place where CBTA test scripts are created and maintained.

The TCE UI is one of the entry points to test automation activities.

You may use it to:

- Create new test scripts
- Start the CBTA Test Recorder
- Execute existing test scripts and check the execution report
- Maintain the test script

### 2.3 System Under Test (SUT)

The systems under test are those in the customer landscape that can be used for testing. Some systems are critical and, for security reasons, they should not be used to run automated test scripts.

### 2.4 Test Profile

The concept of test profile abstracts the test users that the test scripts may use to authenticate the user against the SUT.

- Test profiles are declared and maintained via the SUT Management UI.

The SUT is determined according to the *Solution Manager Context*. It is therefore important to properly define the test profiles, and maintain the corresponding user credentials, for each and every combination of logical component groups, branches and system roles.

### 2.5 Executables

An executable object represents an application entry point. The information it provides is essential and used to determine how to start the application to test.

### 2.6 Component

A component is the entity used to simulate user actions. Default components are those that SAP delivers.

For more details, refer to the documentation:

- CBTA – Runtime Library and Default Components

Additional components, like the screen components and view components, are generated dynamically while recording the business scenarios to be tested.

### 2.7 Test Recorder

CBTA test scripts can be created by recording business scenarios. CBTA includes a test recorder that collects the events thrown by the tested application. It generates test scripts by aggregating *components*.

The test recorder can be started from the TCE UI.

For more details, refer to the documentation:

- CBTA – Test Recorder

### 2.8 Test Player

The test player runs client-side. When executing a CBTA test script, the corresponding VBScript coding is built by aggregating the content of each step and sent to the client computer. A VBScript interpreter is used client-side to execute the script.
3 Configuration

3.1 Front-End Configuration

The major part of the test automation logic runs client side while recording or executing a test script but also while troubleshooting test execution issues.

Some settings are therefore configured on each machine where the CBTA front-end tool is installed. You may start the CBTA Configuration from the Windows start menu.

Logs and Traces

You may for instance, increase the log level to get additional information in the traces when an error occurs or when the tool does not behave as expected. These traces are the one that you may collect and attach to an incident before contacting the Product Support.

![CBTA Configuration](image)

Figure 4: CBTA Configuration

Recording Options

Some options may influence the default behavior of the test recorder. These settings can also be changed while recording a business scenario.
Browser Setup configuration

Since CBTA 3.0.10 it is possible to record a test into different browsers Internet Explorer, Chrome, Firefox and MS Edge. By default, only the recording in Internet Explorer 11, to enable the recording for other browsers following the procedure:

Go to browser Setup Tab, select the browser Chrome (or preferred browser) and click on Check button of Recording Setup group. Once the check done, a popup will appear to purpose to configure browser connection on local machine.

Once setup done, you need to select the default browser for recording phase, go to Recording Tab.

Only the browsers installed on machine and configured are listed in Recording Tab.
Execution Settings

The CBTA execution reports generated by a test script execution are by default displayed in the browser as an HTML content.

You may configure your machine to use a different output format:

![CBTA Configuration - Execution](image)

By default, CBTA closes all previous browser sessions opened by a test execution but not the windows opened manually by end user. If you check the following setting all browser sessions opened by CBTA and manual windows opened from browser launch by CBTA will be closed.

In CBTA it is possible to execute a test script into different browsers available on the machine. CBTA Settings only list browsers installed and supported on current machine.

![Web Browser Execution](image)

Google Chrome: minimum version 40.

Mozilla Firefox: minimum version 40.
**MS Edge**: it is only available on Windows 10 and CBTA supports only following release 10586, 14393, 15063, 16299, and 17134. It is necessary to install MS Edge Web driver to allow CBTA to communicate with Browser.

CBTA settings display the MS Edge version and help to download the correct version of Web driver.

Limitation: Object Spy not working with version 14393 due a bug into webdriver to list MS Edge sessions.

Information: The execution with Chrome, Firefox, MS Edge of web application with some IFRAME with different domains (Cross-origin resource sharing) are currently not supported, need to use Internet Explorer 11 for this scenario.
Other Settings

Some additional settings can be configured, such as the theme and possibly the accessibility mode. The location where temporary files are stored can be changed as well.

The language of the applications (CBTA Configuration, Object Spy, RTL Manager) can be changed. A list of available language can be selected, by default it will be in English. If the application is opened from solman workcenter then it will use the language set on the login page.

![Figure 6: CBTA Configuration – Other Settings](image-url)
3.2 Configuration Tray

To change some CBTA settings, it is also possible to use the CBTA configuration tray, which can be started from the start menu or the desktop icon (if these options have been selected at installation time):

Once started, a small icon will be displayed in the notification area to give some possibilities to the user, like changing the browser to use for executions or recordings, changing the execution report type, etc.

3.3 Self-checks

You may face difficulties to get your landscape properly configured for test automation use-cases. In order to make it easier to solve common issues, you may start the Self Checks application and get an overview of the settings that do not conform to test automation requirements.

As a test engineer, you start the self-checks from TCE. Open a test script and click on Goto > Self-checks.
### 3.4 SUT Management

The SUT Management application is a critical part of the test automation setup.

This application lets you extend the *system data container* to provide the additional information that the test automation process requires, such as:

- The RFC destinations to the SUT
- The *test profiles* to abstract business users
- The URL to start web applications

For More details, search for *SUT Management* on SAP Help Portal

As a test engineer, you may start it from an existing test script by clicking on *Goto > SUT Management*.
4 Test Automation Framework

The Test Automation Framework is integrated with the SAP Solution Manager system landscape. It allows creating test configurations (and their related test script) in a single step.

4.1 Solution Manager Context

The Solution Manager Context abstracts the customer landscape. This context consists of:

- A solution and its System Data Container
- A branch
- A site
- A system role

The target system can be determined by combining the information coming from the context and from the executable object which assigned to the test script.

Note that the technical infrastructure required to get access to the systems is maintained via the SUT Management UI where RFC destinations are declared and Test Profiles are defined.

4.2 Creating a Test Configuration

It is recommended that you create the test configurations directly from your solution documentation, so that the SAP Solution Manager Context gets properly defined to derive the relevant systems under test.

Alternatively, you can create your test configurations from the Test Repository - Test Configurations application. From there, you can navigate to the TCE and maintain the corresponding test script.

Executable Selection in Solution Documentation

From the Solution Documentation you may drill down to a process and create a test configuration by selecting an executable element.

![Figure 9: Solution Documentation - Executable Element](image-url)
The executable element identifies the application to test. There are several types of executable to let you test the various SAP UI technologies.

**Test Configuration Creation from an Executable Element**

The selection of the executable is implicit when the test configuration is created from the Solution Documentation – transaction code SOLDOC.

The contextual menu associated to the executable lets you create a test configuration.

![Figure 10: Solution Documentation - Test Configuration Create](image)

**Selecting CBTA as Test Automation Tool**

The creation starts in a popup window where you specify the name of the test configuration, the name of the test script and the test automation tool. Make sure to select CBTA.

![Figure 11: Selecting CBTA as Test Automation Tool](image)
Assigning an Executable

You may assign the executable later on. From the TCE UI, click Assign Executable and search for the one matching the application that is to be tested.

![Assign Executable](image1.png)

Figure 12: TCE - Assign Executable

Test Script Attributes

The Test Script Attributes section of the TCE UI let you see the logical component group and the executable of the current test script.

![Test Script Attributes](image2.png)

Figure 13: TCE - Test Script Attributes

Selecting the Test Profile

The selection of the test profile is necessary to let the tool authenticate the user automatically when recording or executing test scripts.
This test profile must have been defined beforehand. You may navigate to the SUT Management to check the configuration consistency.

The SUT Management UI starts in a new window and selects automatically the relevant information by expanding the nodes matching the Solution Manager Context and selecting the current System Role.

Press Edit to define the RFC destination (if any) and the Test Profiles.
On the right hand side, you may check the connection to the SUT and define several test profiles. User credentials have to be specified for each system role.

Figure 17: SUT Management - RFC Destination and Test Profile Definition

Selecting a Login Schema

Most of the time, the applications to be tested require to authenticate the user. The corresponding logon phase may differ depending on the actual UI technology.

- For SAP applications (hosted on an ABAP SAP system) the logon phase is performed automatically thanks to the ABAP_TAF_LOGON login schema.
- For NetWeaver applications (hosted on a Java application server) the logon phase is different and the NW_STANDARD_LOGON login schema must be used.
- For applications hosted in the SAP Cloud Platform, the Single Sign On User Authentication relies on certificates. The SAP_CLOUD_IDENTITY login schema is the one to use in that case.
Note that the Login Schema might be implicit when the Executable targets a SAP System. This is of course not the case when using an executable of type URL. In such situation, the tool cannot guess what will be the logon page. This is the reason why the Login Schema must be specified.

In the example below, because the URL points to the Fiori Launchpad, the login schema must be specified. The ABAP_TAF_LOGON schema can be used here because we know that the target system is an ABAP SAP system.

![Test Script Attributes](image)

Non-SAP applications may rely on a different authentication mechanism and use a custom logon page. In such situation, a dedicated login schema must be defined.

For more details, refer to the documentation:

CBTA – Login Schema for Web Applications
4.3 Recording a Business Scenario

Once the test configuration is created, you may switch to the edit mode and press the "Launch CBTA" button to start the CBTA front-end tool and its recorder.

![Launch CBTA from TCE](image)

This opens the Test Creation Wizard in a new window and starts the application to test. While recording, user interactions are captured and the corresponding steps are shown in a list. The test engineer may also do the following:

- define checkpoints to retrieve information from the application UI and check its consistency,
- add a step for capturing a screenshot,
- remove unnecessary steps,
- pause and restart the recording activity

For more details, refer to the documentation:

CBTA – Test Recorder

![Test Creation Wizard – Recorded Steps](image)
4.4 Maintaining a Test Script

Back in TCE, the Test Script tab shows the steps that have been recorded.

⚠️ Caution

After recording, you must press Refresh to see the newly recorded steps.

The TCE UI lets you edit the test script and tune the recorded steps. From there, you may:

- Add, remove or disable some of the steps
- Change the comments
- Adapt the default behavior on errors (if any)
- Expose the step parameters and rename their default name
- Map outputs to input parameters
4.5 TCE Concepts

Test Script Steps and Parameters

Whatever their type, test scripts shown in TCE have the following characteristics:

- A test script contains a collection of steps.
- Each step may have input and output parameters.
- A step may refer to a CBTA component or to another CBTA test script.
- Input parameters can be mapped to output parameters of the previous steps.
- Step parameters can be exposed to be seen as test script parameters.
- Test script parameters have a default value which can be replaced using data coming from the Test Data Container and Test Data Variants.

Test Script Parameters

![Test Script Diagram]

Legend
- **Test Script**
- **Step**

Inputs
- Test script input parameter (exposed)
- Step input parameter
- Step input (exposed)

Links
- Exposition / Reference
- Parameter Mapping

Outputs
- Test script output parameter (exposed)
- Step output parameter
- Step output (exposed)

Figure 22: TCE Concepts - Steps and Parameters
In our example, our test script consists of 2 steps:

- Each step exposes one input parameter. As a consequence, 2 input parameters are seen at the test script level.
- Step #1 has an output parameter mapped to one of the inputs of the step #2
- Step #2 also exposes one of its output parameter making it visible as a test script output parameter.

**CBTA Composite Test Scripts**

For a better reusability, CBTA test scripts can be reused as child test scripts of another CBTA test script. With this composition capability, each child test script is then seen as a regular step and the same logic applies here to input and output parameters. They can be exposed and mapped to pass information from the first child script to the second one.

**Composition Example:**

Composition example reusing our previous test script:

![Figure 23: CBTA Composite Test Script](image)

⚠️ **Caution**

*The composition shown here is only possible with CBTA Test Scripts.*

*When working with other test automation tools, you may do the equivalent but you will then have to create an eCATT Composite Test Script (acting as a main script) and eCATT wrappers to encapsulate the child test scripts.*
Checking and Changing Parameter Values

With CBTA test scripts, the input and output parameters of the step matches the one of the CBTA components. For more details on components, see the 

CBTA – Runtime Library and Default Components documentation.

![CBTA Test Configuration: ZCR_FIORI_LAUNCHPAD - ZCR_FIORI_LAUNCHPAD](image)

You may select one of the steps to see the corresponding input and output parameters.

In this example, the selected step uses the CBTA_WEB_CheckProperty component to check, for instance, that the value displayed is greater than 4.

![Figure 24: TCE – Test Script Steps](image)

![Figure 25: TCE - Step Parameters](image)
This type of check is typically generated by adding checkpoints while recording the scenario. For more details, refer to the documentation:

CBTA – Test Recorder

Exposing Input Parameters

In this example, some step parameters are marked as *fixed*, while some others are *exposed*.

As a rule of thumb:

- Technical parameters are marked as *Fixed*. Their value may have to be changed while developing or tuning the test script. But, once the test script is stable, they are not supposed to be changed anymore.
- Non-technical parameters (i.e. the ones the tester may have to change) are exposed by default.

![Figure 26: TCE - Exposed Step Parameters](image)

**Exposed Parameters**

Step parameters that are exposed are visible, at the test script level, as test script parameters. The *Parameters* tab lets you see them. Their default name can be renamed to improve the readability of the generated test script.

![Figure 27: TCE – Renaming Exposed Parameters](image)
Mapping Output to Input Parameters

Our CBTA_WEB_CheckProperty component does not only check the actual value (value > 4). It also collects this information and makes it reusable via its output parameter. Let’s add a new step and map the output to one of the input parameter of the new component.

![Diagram showing mapping output to input parameter](image)

Figure 28: TCE - Mapping Output to Input Parameter

The CBTA_GUI_A_ReportMessage component can be used to write additional information to the execution report. The only thing we have to do here is to map the MESSAGE input parameter to the OUTPUT parameter of the previous step. This can be done by modifying the Usage to Local and specifying the name of the output in the Ref. Parameter column.

Tokens versus Output Parameters

In the previous section, we could see that output parameter values can be reused as input of the subsequent steps. This TCE mechanism is quite important but not very flexible. With our example, the output value could be written to the execution report but without any possibility to change or format it.

Our goal here would be to provide a human-readable feed; something like:

| Date 03.03.2017 - The value was set to 5. |

Or

| On Friday, March 3, 2017 the value was set to 5. |

With CBTA test script, this can be achieved thanks to the concept of tokens.
replaces them with the value of the corresponding variable. The value is retrieved directly from the CBTA execution context.

It is important to understand that tokens are resolved client-side while executing the test script. The token syntax is quite flexible and lets you evaluate VBScript expressions or access OS Environment variables.

For more information on tokens and token resolution, see the documentation:
CBTA – Runtime Library and Default Components

There is no need to define any parameter mapping to benefit from the tokens. With this example, the values to specify in the MESSAGE input parameter are respectively:

<table>
<thead>
<tr>
<th>Date %today% - The value was set to %Output%.</th>
</tr>
</thead>
</table>

Or

<table>
<thead>
<tr>
<th>On %FormatDateTime(Date,1)% the value was set to %Output%.</th>
</tr>
</thead>
</table>

⚠️ Caution

Do not confuse tokens and output parameters. Tokens are a CBTA concept; they have a limited scope. The information can be shared between components of the current test script only.

Output parameters are a TCE concept. The information can be shared between test scripts as soon as a mapping is defined.
4.6 Test Script Composition

As already mentioned, CBTA test scripts can be reused as child test scripts.

Here is a typical example, where we have:

- a first test script to create a new contact (with a random name)
- and a second test script to search for it but using its ID

The difficulty here is that the ID of the new contact is not known at design time. The ID is generated automatically during the contact creation. Our goal is therefore to pass the information from the first test script to the second one. In other words, we need to map the output of the first test script to the second one.

The advantage here is that the child test scripts complexity is hidden. We simply reuse them as is, and their input and output parameters are seen as regular step parameters. There is no need to expose them at the composite test script level.
As explained, our first step has an input parameter providing the name of the new contact to create. Here we need to make sure we never create the same contact twice. To achieve this a random name can be built using the token capabilities that CBTA offers; the `NEW_CONTACT_NAME` value is set to:

```
Contact_%today%%random%
```

Our first step also has an output parameter named `CONTACT_ID`. In order to pass its value, to the subsequent step, a mapping must be defined. This mapping is declared by specifying the `Ref. Parameter` column. In this example, the referenced parameter name is `CONTACT_ID_SEARCH`.

![Test Script Composition – Parameter Mapping](image)

Of course, the value of the ID must be retrieved by the execution of our first test script. This is typically done using components having the ability to retrieve information from the UI controls. With this example, the ID is displayed in an input field.

![Test Script #1 - Application UI](image)
Retrieving this information is very simple. We simply need to use a “getter” component such as the CBTA_CRM_GetAttribute component shown in the screenshot below.

![CBTA Test: Z_MKT_CONTACT](image)

Figure 33: Test Script #1 - Retrieving Data from the Application UI

The execution report may show the collected information. Here is an example:

<table>
<thead>
<tr>
<th>Test Script</th>
<th>Parameter Name</th>
<th>Value</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2 - CBTA CRM I CLICKLINK</td>
<td>CRM_Link_ClickLink</td>
<td>Operation Succeeded - Label: &quot;Contact&quot;</td>
<td></td>
</tr>
<tr>
<td>1.3 - CBTA CRM IF SETVALUE</td>
<td>CRM_InputField_SetValue</td>
<td>Operation Succeeded * Last Name: &quot;Contact_06.03.2617_168743&quot;</td>
<td></td>
</tr>
<tr>
<td>1.4 - CBTA CRM PRESSKEY</td>
<td>CRM_WkControl_PressKey</td>
<td>Operation Succeeded - Label: &quot;&quot;Last Name:&quot; Key Pressed: ENTER</td>
<td></td>
</tr>
<tr>
<td>1.5 - CBTA CRM GETATTRIBUTE</td>
<td>CRM_WkControl_GetAttribute</td>
<td>Operation Succeeded - Label: &quot;ID:&quot; Attribute Name: value Attribute Value: 437415</td>
<td></td>
</tr>
<tr>
<td>1.6 - CBTA CRM BTN CLICKBUTTON</td>
<td>CRM_Button_ClickButton</td>
<td>Operation Succeeded - Label: &quot;Save&quot;</td>
<td></td>
</tr>
</tbody>
</table>

Figure 34: Test Script #1 - Execution Report
4.7 Test Script Execution

The test engineer developing test scripts can execute them directly from TCE.

![Figure 35: TCE - Execute Test Script](image)

The *SAP Solution Manager Context* provides the information required to identify the target system. You may change it to run the script against a different SUT.

![Figure 36: TCE - Test Script Execution - Start Options](image)

**Execution Reports**

The test execution runs client side and the test result is generated as an XML file which is sent back to the Solution Manager system and stored centrally into the repository.

By default, the report is shown, in the browser, as a plain HTML document. It provides information about the execution context and the overall status.
Execution Report
ZCR_CREATE_INCIDENT_SCRIPT 03.03.2017 09:58:21

Execution Settings
You may configure your machine to use a different output format:

![CBTA Configuration - Report Types](image)

Figure 37: CBTA Configuration – Report Types

⚠ Caution

The SAP Logon process must be restarted to take into account the changes made in the CBTA Configuration.
As already mentioned, the \textit{CBTA Execution Reports} are stored centrally in the repository. You can therefore change the CBTA settings to show the latest execution report using a different report type.

Here is an example with the Adobe PDF format.

![Figure 38: TCE - Test Execution - Adobe PDF Report Type](image)

Executing in Debug Mode

While developing a test script, you may have to troubleshoot test execution issues and run the test in a debug mode. This mode can be enabled from the “Start Options” screen.

When running in debug mode, the steps of the test script are shown in an additional window and you can, step by step, check the behavior at runtime.

For more information, see the documentation:
- CBTA – Test Script Debugger
4.8 Troubleshooting Failing Tests

Object Spy

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

It can be launched directly from the execution report by clicking the link shown by the step that failed.

![Image](image.png)

**Figure 41: Link to the Object Spy**

You may also start it from TCE by clicking the button “Get Technical UI Information”, which is visible when a step is selected in the Test Script tab.

Quick Repair

One of the test automation challenges is that the existing test scripts have, from time to time, to be maintained. This typically happens when a new version of the application being tested is deployed. In such situation, one of the option is to use the Object Spy and troubleshoot the test execution issue. Unfortunately, this approach can be very difficult when the test scripts have hundreds of steps.

With CBTA 3.0 SP05, you may start the recorder in different modes instead. These mode allows you to maintain the existing test scripts by re-recording only a subset of the scenario. The newly created steps can be added at the end of the existing test script or inserted after a particular step.

![Image](image.png)

**Documentation**

The quick repair capabilities are documented in the document:

CBTA – Test Recorder
Recording Options

Figure 42: Test Recorder - Quick Repair Options
5 Appendix

5.1 Unsupported UI Technologies

Web applications can be built using various UI technologies. Some are not supported, and are documented as known limitations. Some are generic enough to be automated as plain HTML pages.

<table>
<thead>
<tr>
<th>Non-SAP UI Technology</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adobe Flash</td>
<td>Not Supported</td>
</tr>
<tr>
<td>Java Applets</td>
<td>Not Supported</td>
</tr>
</tbody>
</table>

5.2 Known Issues

You must be aware that CBTA cannot automate 100% of your scenarios. The tool still has some limitations. For instance, drag and drop capabilities are not recorded.

The notes describing the known issues are listed below.

<table>
<thead>
<tr>
<th>SAP Note</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1835958</td>
<td>Known Issues</td>
</tr>
</tbody>
</table>
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 – Runtime Library &amp; Default Components</td>
</tr>
<tr>
<td>CBTA – Runtime Library Manager</td>
</tr>
<tr>
<td>CBTA – Custom Code Patterns</td>
</tr>
<tr>
<td>CBTA – Query API</td>
</tr>
<tr>
<td>CBTA – Test Recorder</td>
</tr>
<tr>
<td>CBTA – Object Spy</td>
</tr>
<tr>
<td>CBTA – Login Schema for Web Applications</td>
</tr>
<tr>
<td>CBTA – Test Script Debugger</td>
</tr>
</tbody>
</table>

6.2 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>2436142</td>
<td>Supportability and Compatibility Matrix</td>
</tr>
<tr>
<td>1763697</td>
<td>Installation Note</td>
</tr>
<tr>
<td>1778899</td>
<td>CBTA 3.0 - Release Documentation</td>
</tr>
<tr>
<td>1835958</td>
<td>Known Issues</td>
</tr>
<tr>
<td>2029868</td>
<td>Runtime Library – Fixes and Improvements</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>
7 Table of Figures

Figure 1: Solution Administration – Transaction SOLADM ........................................................................................................ 5
Figure 2: Solution Manager - Transaction SOLDOC.................................................................................................................. 6
Figure 3: Test Repository Tile ................................................................................................................................................ 8
Figure 4: CBTA Configuration................................................................................................................................................ 10
Figure 5: CBTA Configuration - Execution.............................................................................................................................. 12
Figure 6: CBTA Configuration – Other Settings ..................................................................................................................... 14
Figure 7: TCE - Go to Self-checks ......................................................................................................................................... 16
Figure 8: TCE – Go to SUT Management ............................................................................................................................... 16
Figure 9: Solution Documentation - Executable Element........................................................................................................ 17
Figure 10: Solution Documentation - Test Configuration Create .......................................................................................... 18
Figure 11: Selecting CBTA as Test Automation Tool ............................................................................................................ 18
Figure 12: TCE - Assign Executable ..................................................................................................................................... 19
Figure 13: TCE - Test Script Attributes ................................................................................................................................. 19
Figure 14: TCE - Selecting a Test Profile ............................................................................................................................... 20
Figure 15: TCE – Go to SUT Management............................................................................................................................... 20
Figure 16: SUT Management – SDC and System Role Selection ............................................................................................ 20
Figure 17: SUT Management - RFC Destination and Test Profile Definition ........................................................................ 21
Figure 18: TCE - Selecting a Login Schema.......................................................................................................................... 22
Figure 19: Launch CBTA from TCE...................................................................................................................................... 23
Figure 20: Test Creation Wizard – Recorded Steps................................................................................................................ 23
Figure 21: TCE - Test Script Steps ......................................................................................................................................... 24
Figure 22: TCE Concepts - Steps and Parameters..................................................................................................................... 25
Figure 23: CBTA Composite Test Script............................................................................................................................... 26
Figure 24: TCE – Test Script Steps ....................................................................................................................................... 27
Figure 25: TCE - Step Parameters........................................................................................................................................ 27
Figure 26: TCE - Exposed Step Parameters ............................................................................................................................ 28
Figure 27: TCE – Renaming Exposed Parameters .................................................................................................................. 28
Figure 28: TCE - Mapping Output to Input Parameter .......................................................................................................... 29
Figure 29: Test Composition Overview ................................................................................................................................. 31
Figure 30: Test Script Composition Example........................................................................................................................ 31
Figure 31: Test Script Composition – Parameter Mapping .................................................................................................... 32
Figure 32: Test Script #1 - Application UI.............................................................................................................................. 32
Figure 33: Test Script #1 - Retrieving Data from the Application UI ......................................................................................... 33
Figure 34: Test Script #1 - Execution Report ......................................................................................................................... 33
Figure 35: TCE - Execute Test Script................................................................................................................................... 34
Figure 36: TCE - Test Script Execution - Start Options .......................................................................................................... 34
Figure 37: CBTA Configuration – Report Types ....................................................................................................................... 35
Figure 38: TCE - Test Execution - Adobe PDF Report Type .................................................................................................... 36
Figure 39: TCE - Start Options - Debug Mode ......................................................................................................................... 37
Figure 40: TCE Executing in Debug Mode .............................................................................................................................. 37
Figure 41: Link to the Object Spy........................................................................................................................................... 38
Figure 42: Test Recorder - Quick Repair Options................................................................................................................ 39