SAP Solution Manager 7.2 – Test Suite

How-To Guide

Test Automation Framework and integration of SAP and Partner Test Automation Products

Public

February 2018
## TABLE OF CONTENTS

1 INTRODUCTION .................................................................................................................. 4

2 PREREQUISITES AND SETUP OF TEST AUTOMATION FRAMEWORK .................................. 6

2.1 Solution Manager Setup ....................................................................................................... 6

2.2 Solution Documentation ....................................................................................................... 6

2.2.1 System Landscape Documentation .................................................................................... 6

2.2.2 Business Process Documentation ..................................................................................... 6

2.2.3 Solution Manager Context ............................................................................................... 7

2.3 Setup of Test Automation Framework .................................................................................. 8

2.3.1 General Configuration Steps ......................................................................................... 8

2.3.1.1 Test Automation Preparation ...................................................................................... 8

2.3.2 Installation of third party test tool on client PC ................................................................. 9

2.3.3 Solution Manager Launch Pad ......................................................................................... 10

2.4 Authorization ....................................................................................................................... 10

3 DEFINITION OF AUTOMATED TEST CASES .................................................................... 11

3.1 Create a new Test Configuration ....................................................................................... 12

3.1.1 Delete a Test Configuration ............................................................................................ 17

3.2 Test Configuration: Components ...................................................................................... 18

3.2.1 Test Script ...................................................................................................................... 18

3.2.1.1 Recording of a test script .......................................................................................... 19

3.2.1.2 Example: Record a test script for Sales order creation (SAP ERP) with Microfocus UFT .... 20

3.2.2 SUT Management .......................................................................................................... 25

3.2.2.1 Concept .................................................................................................................... 25

3.2.2.2 Maintenance of SUT .................................................................................................. 26

3.2.2.3 Use of SDC in the Test Configuration ....................................................................... 26

3.2.3 Test Data Container ....................................................................................................... 27

3.2.3.1 How to define a TDC ................................................................................................ 28

3.2.3.2 Import parameter definition from an external script .................................................. 32

3.2.3.3 Assign a TDC in the Test Configuration ................................................................... 32

3.3 Parameterization of test cases ......................................................................................... 33

3.3.1 Parameter definition and handover from test tool .......................................................... 33

3.3.2 Parameterization using Variants only ............................................................................ 34

3.3.3 Parameterization using Test Data Container .................................................................. 37

4 TEST PLANNING OF AUTOMATED TESTS .................................................................... 43

4.1 Prerequisites for executing automated tests .................................................................... 43

4.1.1 Test Plan .......................................................................................................................... 43

4.1.2 Test Package & Assignment of Testers ......................................................................... 44

4.2 Completeness and Gap Reports ....................................................................................... 46

4.2.1 Test Case Coverage for Business Processes ................................................................. 46
### 4.2.2 Check assignment of Test Cases to Test Plans or Test Packages ........................................ 48
### 4.2.3 Inconsistency Check for Test Plans .................................................................................. 50
## 5 EXECUTION OF AUTOMATED TEST CASES ......................................................................... 54
### 5.1 The Tester Worklist ........................................................................................................ 54
### 5.2 Execution of automated tests from the Tester Worklist .................................................. 55
#### 5.2.1 Single Test Execution .................................................................................................. 55
#### 5.2.2 Collective execution of test cases ............................................................................... 56
### 5.3 Handover of System Data for test execution .................................................................... 58
### 5.4 Start Options for Automated Tests .................................................................................. 59
### 5.5 Scheduling of Automated Tests for unattended execution on a remote machine .......... 61
#### 5.5.1 Prerequisites .............................................................................................................. 62
#### 5.5.2 How to schedule the unattended execution ............................................................. 63
### 5.6 Access to Test Execution Logs ......................................................................................... 67
### 5.7 Notification emails .......................................................................................................... 69
#### 5.7.1 Register users for receiving notification emails when test executions fail .............. 70
#### 5.7.2 Change templates for notification emails .................................................................... 71
## 6 BUSINESS PROCESS CHANGE ANALYZER .......................................................................... 75
### 6.1 Generation a TBOM (Technical Bill of Material) .......................................................... 75
#### 6.1.1 Automatic recording of a TBOM ................................................................................. 76
#### 6.1.2 Prerequisites and additional information on BPCA ................................................ 78
### 7 REPORTING ......................................................................................................................... 79
#### 7.1 Status Reporting ............................................................................................................. 79
#### 7.1.1 Check Status for current test phase ........................................................................... 79
#### 7.2 Status and Progress Analytics ....................................................................................... 82
#### 7.2.1 Status Analytics ......................................................................................................... 83
#### 7.2.2 Progress Analytics ..................................................................................................... 84
#### 7.2.3 Defect Analytics ......................................................................................................... 85
#### 7.2.4 Test Effort Analytics .................................................................................................. 86
## 8 REPAIR OF DAMAGED TEST CASES ................................................................................ 88
### 8.1 Workflow to report a Damaged Test Case ................................................................. 88
#### 8.1.1 Prerequisites ............................................................................................................ 88
#### 8.1.2 Create a Test Case Error Message ............................................................................ 88
### 8.2 Repair environment for Test Engineers .......................................................................... 89
#### 8.2.1 Damaged Test Case Worklist .................................................................................. 90
#### 8.2.1.1 Prerequisites for handling Test Case Error Messages ...................................... 91
#### 8.2.2 Customizing of the transaction type for Test Case Errors ..................................... 92
#### 8.2.2.1 Status Customizing ............................................................................................... 94
#### 8.2.2.2 Category Customizing .......................................................................................... 95
### 9 APPENDIX ............................................................................................................................ 98
#### 9.1 Further documentation .................................................................................................. 98
1 INTRODUCTION

The Test Automation Framework provided with SAP Solution Manager 7.2 allows to design and execute automated test cases with certified external tools, directly in the business process context. It is integrated with the various test management capabilities supporting the whole test cycle:

SAP Solution Manager – Test Suite

Capabilities

The Test Automation Framework allows you to create and execute automated test cases. In this way it complements SAP Solution Manager test automation capability CBTA and facilitates functions such as creating test configurations, test data containers and system data containers. For this it makes use of the system data already available in the system landscape documentation in Solution Manager.

The various steps within the Test Management cycle are complemented as follows:

• Test Preparation:
  o Set up automated tests in SAP Solution Manager Solution Documentation.
  o Create Test Configurations using an external test tool.
  o Create a Test Configuration and a test script in one single step.
  o When test cases are executed, RFC destinations are derived automatically from SAP Solution Manager system landscape, depending on the system role the user chooses.
  o The Test Automation Framework supports the usage of Test Data containers, which store required test data and which can be reused in different test cases, thus saving time through reduced maintenance effort.

• Test Plan Management: Change the system role without changing the system data containers. The RFC destinations are derived automatically when the test cases are executed.

• Tester Worklist (execution):
- Execute automated test cases
- Assign custom generic attributes to classify failures
- Create Test Case Error Messages for follow-up by a responsible test engineer

- Damaged Test Case Worklist: Exchange information about the process of repairing damaged test cases. Also the Damaged Test Case Worklist serves as a central repair environment for Test Engineers, providing direct access to all relevant information and features for repairing test cases.

- Analytics: Different reports for status, progress and message reporting as well as gap and consistency reports as well as direct access to test logs of third-party automated test tools.

- Administration:
  - Register 3rd party test tools
  - Define generic attributes of test cases
  - Register users for receiving notification e-mails when test executions fail
  - Edit templates for notification e-mails
2 PREREQUISITES AND SETUP OF TEST AUTOMATION FRAMEWORK

2.1 Solution Manager Setup

- The content described in this document require SAP Solution Manager 7.2 SP3.
- Initial- and basic configuration has been performed with transaction SOLMAN_SETUP.

2.2 Solution Documentation

As a prerequisite to do Test Management with SAP Solution Manager it is required to have at least a basic documentation of all systems relevant for testing (System Landscape Documentation) as well as all processes relevant for testing (Business Process Documentation).

2.2.1 System Landscape Documentation

The following prerequisites need to be fulfilled:

- Solution Manager local System Landscape Directory is set up and contains technical system information of your managed system landscape
- Managed systems considered for testing are available in Solution Manager Landscape Management (transaction LMDB)
- Logical Component Groups are in place and managed systems are assigned to according System Roles
- RFC destinations from Solution Manager to managed system and vice versa have been created

2.2.2 Business Process Documentation

In Solution Manager test cases are assigned to process steps or processes. As a prerequisite it is therefore required to have all relevant processes documented in a Solution. The following prerequisites need to be fulfilled:

- A Solution Manager Solution has been setup (transaction SOLADM - Solution Administration) with the corresponding System Landscape
The Solution contains Solution Documentation, which describes all relevant scenarios, processes and process steps.

Each process step has assigned the relevant transaction (SAP GUI transaction, WebDynpro Application, CRM WebClient Application, URL, FIORI, etc.).

2.2.3 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.

2.3 Setup of Test Automation Framework

The Test Automation Framework provides the possibility to seamlessly integrate a test automation application of choice with SAP Solution Manager. Besides SAP CBTA (which can be used for automated testing of SAP based applications only) there are various external test tools from Independent Software Vendors (ISV) which have been certified.

If you have purchased a certified third party test tool the following steps are required to setup the integration with SAP Solution Manager.

2.3.1 General Configuration Steps

2.3.1.1 Test Automation Preparation

Within SAP Solution Manager transaction SOLMAN_SETUP several guided procedures are provided leading through all required CBTA configuration steps.

- Goto: Scenarios → Test Suite → Test Automation Preparation

When executing automated tests using eCATT, CBTA or any external test tool the system makes use of eCATT functionality. Therefore the following prerequisites need to be fulfilled in the Solution Manager system:

- Generate ECATT User
- Enable Scripting in the System Under Test
• Enable SAP GUI Scripting on the SAP GUI Frontend

Follow the instructions with a click on Display in the Documentation column. A click on the Navigation column leads you to the correct place of performance.

All activities in Step 2 (Special Configuration) are optional, for one of the following reasons:
• SAP delivers predefined customizing (e.g. for e-mail notifications): the activity is then only relevant if you want to define your own settings,
• Configuration is only needed if you want to use the feature (e.g. use external test tools).

You have completed the configuration scenario of the Test Automation Preparation. In this final step check the overall status for all previous steps.

Note:
If you want to use CBTA as a test automation tool in future, please continue with the next Guided Procedure Component-based Test Automation. For other 3rd party tools, no further configuration needs to be done.

2.3.2 Installation of third party test tool on client PC

Please follow the installation instructions provided by the software vendor.
All test tools require a client-based installation. 
Note: The test tool needs to be installed on each machine on which front-end based automated tests shall be executed. 
Some test tools can be installed on a Terminal Server. Please check the installation and licensing guidelines provided by the software vendor.

### 2.3.3 Solution Manager Launch Pad

The Launchpad group Test Suite is the central access point for all the information and functions you need to determine the scope of testing required for cross-system processes, manage the tests centrally, and execute the tests.

For fulfilling the prerequisites of using Test Management within SAP Solution Manager, you need the Launchpad group Project and Process Management as well.

You can add or hide Launchpad groups and Tiles via 'Personalize Home Page'.

![Launchpad group Test Suite](image)

**How to access it:**

**Transaction SM_WORKCENTER**
The Solution Manager Launch Pad opens in a new browser session. Browse catalog “Test Suite”.

**Recommendation:**
We of course do not expect your users to log on to the Solution Manager by SAP GUI only to start SM_WORKCENTER: distribute the resulting URL to your users and have them bookmark it.

### 2.4 Authorization

Please refer to the Security Guide for the SAP Solution Manager:

http://service.sap.com/instguides ↦ SAP Components ↦ SAP Solution Manager ↦ newest release ↦ Security Guide SAP Solution Manager (e.g. chapter "End-User Roles for Test Management")
DEFINITION OF AUTOMATED TEST CASES

An automated Test Case is actually a **Test Configuration** of reusable objects.

A Test Configuration comprises the following object types:

- **The Test Script** is an entity persisted in the repository of the Solution Manager system and represents the steps of a business process which is performed by the automated test case, for example:
  - A sales order is entered in the CRM system.
  - The order is passed to the ERP system where production is scheduled.
  - The ERP system passes information to the SCM system where a table is updated.

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

- **The Test Data Container** is a reusable data object with a set of user-defined parameters. It contains multiple combinations of test data which is retrieved during execution of an automated test. It is maintained independently from the test script and therefore can be reused in different tests.

- **The System Data Container** defines the systems on which these steps are performed. A system is defined as follows:
  - Operational function (CRM, ERP system, for example)
  - Technical role (system roles such as development, test, evaluation, production system, for example)

  When executing a test case the system role can be switched (from development system to test system, for example) without having to adjust any system parameters or without the need to adapt the test script.

Within the Test Automation Framework, the execution of an automated test case always means the execution of a Test Configuration.
3.1 Create a new 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.

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.
Selecting the 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.

Assign 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.
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.

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.

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.

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.

Note: For more details, refer to the documentation: Login Schema for Web Applications

### 3.1.1 Delete a Test Configuration

Note: The deletion of Test Configurations is not supported directly from Solution Documentation (transaction: SOLDOC) for technical reasons. Removing a Test Configuration from the “Elements of…” area does not delete the Test Configuration, it just removes the selected Test Configuration assignment from the process/process step.

In order to delete a Test Configuration, proceed as follows:

1. Open Solution Manager Launchpad
2. Click on tile Test Repository – Test Configuration (Launchpad group: Test Suite)
3. Search for the Test Configuration to be deleted
4. Hit the button “Delete”
5. Select an option in the popup and hit OK.

3.2 Test Configuration: Components

In Solution Manager an automated test case is represented by a Test Configuration which consists of a Test Script, a System Data Container and a Test Data Container.

3.2.1 Test Script

A test script performs the execution of a process in one or more applications in an automated way. The test script is recorded using SAP CBTA or eCATT (for SAPGUI based applications only) or an external test tool. In a second step the test data used during recording of the test script can be replaced by parameters which allows to run the test script with different variants of test data.

When starting the recording of a test script from the Solution Manager (e.g. from transaction SOLDOC) the Test Automation Framework allows to run the external test tool "in integrated mode", which means it automatically opens the external test tool and hands over information (e.g. System Data Container, Test Data Container) to the external tool. Once the test script has been recorded with the test tool, it is stored in Solution Manager and linked to the Test Configuration. The test script can also be reused in different Test Configurations. If test data was replaced by parameters in the external tool, the defined parameters are also automatically transferred to the Test Configuration where it can be used for defining different variants of test data.
3.2.1.1 Recording of a test script

In order to profit from the integration provided by the Test Automation Framework, a test script is created directly from the Test Configuration.

Prerequisite: You have created a Test Configuration as described in chapter “Create a new Test Configuration”.

Note: For illustration purposes the following example is described using Microfocus Unified Functional Testing as external test tool. For details on how to use other external test tools please refer to the documentation of the test tool provider.

- Initially, create the Test Configuration like described in chapter 3.1 Create new Test Configuration
- Click the Launch <test tool> (e.g. HP UFT) button to launch the external test tool:

  ![External Test Tool Launch Demonstration](image)

  ➔ The external test tool is automatically launched and the recording of the script can be started.

Note:
While the external test tool is open in integrated mode (launched from Solution Manager) the SolMan sessions are not accessible and kept in the background. To go back to a SolMan session, either (save and) close the test tool or logon to the SolMan system via a new session.

Once UFT has been launched in integrated mode, the initial screen is displayed and UFT is ready to record a new script:
The following describes an example how to record a test script for the process “Create Incident” (SAP Fiori).

3.2.1.2 Example: Record a test script for Sales order creation (SAP ERP) with Microfocus UFT

- UFT, click the Record button
  ➔ The Record and Run Settings Popup appears
- For recording a SAP GUI based process select option “Open the following SAP Gui client when record or run session begins”
- Select the target system for recording and executing the test script from the dropdown list box Server description, in this example it's ERP system CSP client 004
- Check mark “Enable auto-logon”
- Enter a User and a Password for the recording and the execution
- Enter the Target Client
- Enter the logon Language
- Check mark “Remember password”
Once all logon data is provided in the Record and Run Settings hit button Apply and OK.

The recording of the process is started. Recording mode is indicated by a red blinking “Recording” sign in the lower right corner of UFT as well as with the scripting indicator in the SAP sessions.

Note: During the recording, all activities/clicks on the screen are captured by UFT, therefore try to focus exactly on the necessary steps for execution the process and avoid any additional clicks, data entries or keystrokes. This is to assure that the script can be executed without issues later.

1. UTP automatically launches system under test, in this example S4H (ERP) and opens a session with the provided logon data
2. Open the OKCode field (click on the triangle), enter transaction code VA01 and click the Enter button.
3. On screen “Create Sales Order: Initial Screen” enter the following data:
   - Order Type = OR
   - Sales Organization = 1000
   - Distribution Channel = 10
   - Division = 00
4. Hit Enter to continue
5. On screen “Create Standard Order: Overview” enter the following data:
   - Sold-To-Party = 1000
   - Ship-To-Party = 1000
   - Material = P-100
   - Order Quantity = 1

6. Hit Enter
7. Save the Order by clicking button “Save”
8. Confirmation message “Standard Order XXXXX has been saved” appears in the lower left corner.
9. Leave the system by clicking button “Exit” twice, confirm the Logoff popup with “Yes”.
10. Back in UFT click button “Stop” to finish the recording:

![Screenshot of UFT interface with Stop button highlighted.]

11. UFT shows the recorded steps.
12. Click button “Save” and then Close UFT:

![Screenshot of UFT interface with Save button highlighted.]

→ The test script is automatically transferred and saved to Solution Manager.

13. Back in Solution Manager, “Save” the Test Configuration:
The Test Configuration has now been created and assigned to the Business Process and can be executed.

**Adhoc Execution of a Test Configuration**

Right after recording the script with UFT and saving it to Solution Manager the test configuration can be executed with the default data which was entered during the recording.

Note: This kind of execution is usually done by the Test Engineer only. For the execution of automated test cases within a test cycle, the execution needs to take place using a Test Plan and Test Package with assignment to testers.

In Process Management, do a right mouse-click on the relevant Test Case and click *Execute*:

Other option: From the Test Configuration screen, hit the *Execute* button:
3.2.2 SUT Management

The SUT Management application is a critical part of the test automation setup. This application lets you extend the system data container (SDC) 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

Note:
In Solution Manager 7.2 the System Data Container is generated and updated automatically by the Test Suite, on the basis of the documented System Landscape of the dedicated solution.

3.2.2.1 Concept

Instead of having the target system “hard-coded” in the test script, the use of a SDC allows the flexible execution of one and the same test script in different target systems. When creating a solution, RFC destinations to the systems under test are retrieved automatically from SDC in the Test Automation Framework. For executing a test case in different target systems the system role can be changed and the RFC destination is automatically adapted accordingly without additional effort.

The systems are represented by logical component groups. Relevant parameters of the respective system landscape (target systems and RFC destinations) are retrieved automatically from the Solution Manager solution which is assigned to the test configuration on each and every change. When you execute a test case you can switch the system role (from development system to test system, for example) without having to adjust any system parameters.
3.2.2.2 Maintenance of SUT

You can access the SUT management as follows:

- In the SAP Solution Manager launchpad, choose Administration → Test Suite → Test Automation Framework → System under Test Management
- In the Test Composition Environment, choose Goto → SUT Management from the drop down list

For SUT management in CBTA, see Creating Test Profiles and Logon Data for SUTs.

For eCATT and third-party test tools:

1. Select the system data container, logical component group, logical component, and system role.
2. Select the eCATT and Third Party Tools tab in the SAP ABAP Backend tab.
3. Under Test Profile, define the RFC destination to be used by tests.

3.2.2.3 Use of SDC in the Test Configuration

A System Data Container has been assigned automatically, when a Test Configuration is created. To check which system is used, follow the next steps:

1. Open a Test Configuration, e.g. via Process Management:

![Diagram](image)

2. Take a look at the Technical Data and the selected system role:
Switch system role

The target system shown in the Technical Data area depend on the system role which is currently maintained by the individual user.

To switch to a different system role, click on on the upper right corner and choose the relevant system role:

3.2.3 Test Data Container

A test data container is a persistent data object with a set of parameters that belong together from a business point of view. They can be maintained independently from the test script.

The parameters are user-defined and have a default value. These values can have variants and are assigned to the test configuration.

In automated test cases, the bulk of the test data is normally stored separate from the test scripts in test data containers. The main reasons for this are reusability and maintainability. Test data containers and a test script are brought together in the test configuration to create an executable automated test case.
The simplest use of test data containers is to create a separate one for each test script. However, this does not provide the advantage of reuse. A more effective way to manage test data is to create a single test data container for a whole application or sub-application. By storing all of the test data in one container, it is easier to keep it consistent. In contrast to using a single test data container for all parameters, parameters can also be distributed over several test data containers. For example, if you have a large number of scripts, each with many parameters, you may find that using a single test data container is no longer a practical option. In this case, you could split the parameters into logical groups, each in its own test data container.

As with other eCATT objects, the test data container has mandatory attributes (title, package, person responsible, and application component) as well as attributes containing administrative information. Test data containers consist of parameters and variants. The parameters describe the interface of the container and the variants store the data. Each variant contains a field for each parameter. If no value has been entered in a field, the value specified in the ECATTDEFAULT is the value of the field.

3.2.3.1 How to define a TDC

Example 1: Define a simple TDC with data for single fields for VA01 “Create Sales Order”

1. Open transaction SECATT (Extended Computer Aided Test Tool)
2. Select option “Test Data”, enter an ID for the new TDC Example: ZTDC_VA01_SALES_ORDER_SIMPLE and click Create Object:

![Extended Computer Aided Test Tool: Initial Screen](image)

3. Enter the following data on the Attributes tab:

   o Title
   o Application Component
   o System Data Container + Target System
     → Assigning a SDC and a target system to the TDC allows to reference to the data definition in the target system when defining the parameters in a later step. This assures data consistency and a better quality of the test data (automatic plausibility check)
   o Optional: Search terms. This allows easier search for certain TDC
4. On the **Parameters** tab, define all parameters to be used for testing:

**Parameter ID:** Use e.g. I/E to distinguish between Import and Export parameters  
**Description:** Enter a description for each parameter (automatically retrieved if reference to target system is used)  
**Value:** optionally maintain a default value. This value will not be taken into account when executing tests with values from the Variants tab. The values are shown in the ECATTDEFAULT variant.  
**Reference:** DDIC reference in the target system. When a Target System is maintained on the Attributes tab of the TDC, the system automatically retrieves the field’s description and data definition (type, length) from the target system.

5. On the **Variants** tab, add additional Variants with different combinations of valid test data:
Tip: Before entering a set of test data, make sure the data is valid and executes correctly at least two times in the target system.

Example 2: Define a complex TDC with data for multiple line items in VA01 “Create Sales Order”

1. Create a TDC as described in Example 1, e.g. ZTDC_VA01_SALES_ORDER_COMPLEX
2. On the Parameters tab, define all parameters to be used for testing.
3. Add an additional parameter I_STRUCTURE and reference it to the DDIC structure RV45A[]. The relevant table/structure can be found by positioning the cursor in a field of the table you want fill with test data > click F1 > Technical information.  
   Note: Make sure you add the string “[]” to the Parameter Reference, otherwise it is not possible to define multiple line items within a variant.
4. Double-click on the parameter I_STRUCTURE to display the fields of the structure in the sub screen below:

   ![Change Test Data Container: ZTDC_VA01_SALES_ORDER_COMPLEX](image)

   5. On tab Variants define a variant and maintain the values for the single fields.
6. In order to maintain the values for the line items attached to this variant, double-click on the field for I_STRUCTURE (field is either empty or shows "<INITIAL> or "<VALUE>");

7. Add line items by clicking on button "Insert" or "Append"

8. Maintain the values for each line item. In this example the Material and the Quantity:

9. Hit Enter.
   Additional option: you can double click on the left lower subscreen (entry 1 of I_STRUCTURE) to change the data entry form on the right lower subscreen (switch the format for data entry):

10. Add a second Variant and maintain different data for it (different material and quantity):
3.2.3.2  **Import parameter definition from an external script**

When creating a new TDC it is also possible to auto-generate the parameters from the variables which were transmitted from the external test tool to the Solution Manager to the Variants tab on the Test Configuration. Steps:

- From a Test Configuration (Test Data tab) create a new TDC
- Enter a Title and an Application Component
- From the menu bar select Edit > Import Parameters
- Indicate the external test script from which you want to copy the parameter definitions
- Click button “Get Parameter”
- Select the Parameters you want to import into the TDC
- Click the Attach button in the center of the split screen
- Click the Copy button to exit the Import Parameters screen.
- The parameter IDs have been copied into the TDC

*Note:* This procedure only copies the parameter IDs, the complete parameter definition (data reference, target system etc.) is not copied and needs to be maintained manually.

3.2.3.3  **Assign a TDC in the Test Configuration**

One or more Test Data Container can be assigned to a Test Configuration on the Test Data tab:
How to use the Test Data Container for executing a Test Case with various different variants of test data, see chapter Parameterization.

3.3 Parameterization of test cases

Instead of executing automated tests with always the same test data (field values) as used during the recording of the script, these static values can be replaced by parameters which are populated at runtime of the execution. This allows a much more flexible use of existing test cases and a much more extensive test of business processes.

3.3.1 Parameter definition and handover from test tool

The definition of parameters is handled in the external test tool. After recording the script, the field values used during the recording are replaced by parameters, e.g. replacement of Sold-To Party “1000” with parameter “I_SOLD_TO_PARTY”. The procedure for parameterization differs depending on the test tool. Some test tools use an automated approach and automatically provide a parameter ID for each field value. For details regarding the parameterization in the external test tools, please refer to the documentation of the individual test tool providers.

After completing the parameterization in the external test tool, when saving the test script to SAP, all defined parameters are automatically transferred to the Test Configuration in Solution Manager. The parameters can then be connected to Test Data Containers which provide multiple variants of test data which are handed over to the external test tool at the time of execution.

In the example below, the following field values have been replaced by parameters in the external test tool (example with UFT):
Sales Document Type = I_ORDER_TYPE
Sales Organization = I_SALES_ORG
Distribution Channel = I_DISTRIBUTION_CHANNEL
Division = I_DIVISION
Sold-to party = I_SOLD_TO_PARTY
Ship-to party = I_SHIP_TO_PARTY
Material Number = I_MATERIAL
Order Quantity = I_QUANTITY

The parameters which have been defined in the external test tool are automatically transferred to the Test Configuration into the Test Data tab:

3.3.2 Parameterization using Variants only

Note: This method is not recommended for the provisioning of test data for the test execution. But it can be helpful during the design phase of an automated test, e.g. for ad hoc testing of the test script with a few variants.
Use the Expert Mode via Menu-Button Goto → Expert Mode

In order to add additional Variants with test data on the Variants tab, click button “Append Variant” and maintain additional test values:

Make sure to checkmark the Variants relevant for execution in the column “Execute”.

The Test Configuration can now be executed with the different variants. The execution can be started either directly from the Test Configuration using the button Execute Test Configuration or from the Process Management (make sure the field Test Data Variant is maintained, see field help for options):
On the Start Options screen, select Error Behavior = **Termination, Continue with Next Variant**, in order to execute all variants, even if one of them fails:

The execution of the test configuration will be performed for each of the selected Variants. After completion, the result is displayed (SAP internal log):
3.3.3 Parameterization using Test Data Container

For the storage and maintenance of test data for automated tests it is recommended to use one or more Test Data Containers (TDC). The test data in the TDC can be linked to a Test Configuration by defining multiple variants which contain a reference to the single values in the TDC.

Prerequisite:

- You have created a Test Data Container which contains test data variants relevant for the business process to be tested.
- Parameters have been defined in the external test tool and have been transferred to the Parameters tab of the Test Configuration:
1. In the Test Configuration, go to tab Test Data and add a Test Data Container by clicking on button "Add":

2. Go to tab "Test Data" and click on button "Test Data Assignment Wizard":

---

**Maintain External Element**

**HP UNIFIED FUNCTIONAL TESTING (UFT) Test Configuration**

**Test Data Container(s)**

<table>
<thead>
<tr>
<th>Add</th>
<th>Remove</th>
<th>Test Data Container</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---
3. In the Test Data Assignment Wizard goto Step Assign Parameters; the screen is split into two parts. On the left side the content of the Test Configuration is displayed. On the right side the Parameters of the Test Data Container are displayed:

Use Assign by Name and Data Type to assign the TDC Parameters to the Test Configuration Parameters automatically. If the system does not contain any suggestions, please select a line in each table and use button Assign Manually.
4. Now, go to the next step **Assign Variants** and select the variants of the TDC which you want to take over to the Test Configuration and hit button **Result – Test Configuration**:

![Image of Test Data Assignment Wizard]

5. Finally, check the right assignment of variants and click at the bottom of the screen to button **Save Test Data assignment to Test Configuration**.

![Image of Test Data Assignment Wizard]

**Tip:** It is strongly recommended to name the parameters in the TDC in the same way as in the test script. If this is the case, the variants from the TDC can be mapped easily to the parameters defined in the Test Configuration using the “Assign by Name and Data type” functionality. Otherwise parameter names need to be mapped individually using the “Assign manually” option, which requires more effort and which can be quite complex if many parameters need to be mapped.

6. The selected variants from the TDC is copied to the Test Configuration.
The Test Configuration can now be executed with the different variants getting the values from the TDC. The execution can be started either directly from the Test Configuration using the button "Execute Test Configuration" or from the Process Management (make sure the field "Test Data Variant" is maintained, see field help for options):

On the Start Options screen, select Error Behavior = Termination, Continue with Next Variant, in order to execute all variants, even if one of them fails:
The execution of the test configuration will be performed for each of the selected Variants. After completion, the result is displayed (SAP internal log):
4 TEST PLANNING OF AUTOMATED TESTS

The test planning consists of the identification of the required test scope and definition of test plans and test packages as well as the assignment of test packages to testers. The Test Automation Framework supports this process by providing several reports for coverage and consistency checks.

The assigned test packages appear in the Tester Worklist of the Tester and can be executed directly from there or scheduled for execution at a specific time and even on remote machines.

4.1 Prerequisites for executing automated tests

Automated tests are handled the same way as manual test cases. In order to provide automated test cases to testers it is required to create a test plan, a test package and assign this package to one or more testers.

4.1.1 Test Plan

For creating a Test Plan for automated test, the following steps are required:

- In the Launchpad Group Test Suite select Tile Test Plan Management
- Click button Test Plan > Create:

- Maintain the General Data and Settings and press Save.
• Switch to Tab Test Case Selection and press ; the Process Structure will be loaded. Select the automated test cases relevant for your test plan (e.g. by using a filter) and press button .

The Test Plan will be generated.

4.1.2 Test Package & Assignment of Testers

For creating a Test Package and assignment of testers, the following steps are required:

• Switch to tab Test Packages and click button Test Package > Create:
• Fill out (at least) all mandatory fields for tabs General Data and Settings
• Switch to tab Test Case Selection and select the automated test cases relevant for this test package and press button Save:

• Switch to tab Testers and click button "Assign Tester":


Search and select the User names you want to assign to the test package.
The Test Package is now displayed in the Tester Worklist of the assigned Testers and ready for execution or scheduling.

4.2 Completeness and Gap Reports

The Test Automation Framework supports the test planning with useful reports for validating the complete coverage and the consistency of test plans.

4.2.1 Test Case Coverage for Business Processes

Goal:
Check to what extent business processes are covered by test cases in order to identify potential gaps in the test scope.

Benefit:
Ensure transparency during test scope definition and test planning to gain reliable test results.

How to generate the report:

1. Call up Solution Manager Launchpad and choose Catalog Test Suite and tile Test Suite Analytics.
2. Open the bar *Completeness and Gap Reports* and click on *Test Cases*

![Completeness and Gap Reports](image)

3. Change Layout to *Test Case Assignment* via button.

![Test Case Assignment](image)

4. Use the Test Case Type filter on the left and select *Non-Existent* to see only Processes and Process Steps, where no Test Case is existing. Take care of the path above the table: only Processes and Process Steps in that area will be shown!
This allows you to see at a glance what still needs to be done to complete the test scope definition.

4.2.2 Check assignment of Test Cases to Test Plans or Test Packages

**Goal:**
Find out whether there are test cases which are not included in test plans or test packages.

**Benefit:**
Ensure that specified Test Cases are part of the relevant Test Cycle.

**How to generate the report:**
To create a negative list of test cases that are not assigned to a test plan or test package, proceed as follows:

1. Call up Solution Manager Launchpad and choose Catalog Test Suite and tile Test Suite Analytics.
2. Open the bar Completeness and Gap Reports and click on Test Case not included in Test Plan.
3. In the search area, select at least the Solution, Branch and Scope. Other parameters can be used for having an more detailed view (e.g. only new Test Cases).

4. At the bottom of the Search Criteria, choose Search. The results will be shown below the Scope section.
5. The Scope area provides detailed information about the selected Solution and Branch, plus the amount of Test Cases in total and not included in Test Plan or Test Packages.
6. Select the topmost element in the Process Structure and choose Expand subtree (5). A list of test cases without assignment to test plans or test packages is displayed.

A remark in the Missing in column indicates what kind of assignment is still missing for the relevant test case.

4.2.3 Inconsistency Check for Test Plans

Goal:
Determine whether the business process structure and/or test case descriptions has been changed after the relevant test plan was generated. Identify test plans that are affected by changes in the business process hierarchy and test documents.

**Benefit:**
Ensure that test execution is based on the most up-to-date process and test descriptions.

**How to generate the report:**
To perform an inconsistency check for test plans proceed as follows:

1. Call up Solution Manager Launchpad and choose Catalog Test Suite and tile Test Suite Analytics.
2. Open the bar Completeness and Gap Reports and click on Test Plans – Nodes and Test Case Changes
3. In the Search Criteria area, choose at least the Solution, Branch and the kind of change that you want to look for. Click on Search.
   The Inconsistent Test Plans are shown below the search area, where you can see at a glance, which test plans within the selected Solution are inconsistent.
4. Click on a test plan ID that is marked as inconsistent. The test plan maintenance screen opens.

5. In the tab Test Case Selection, click on [Show Change]. If this button is disabled, change to Edit-Mode via click on [Edit].

All changed elements are shown. Click on button [Go To Next Change] and the next changed item within the business process structure is highlighted. An icon is displayed in column Changed next to the changed structure element to indicate the type of change (e.g. Structure element was added to the Solution).

6. Select the checkbox if you want to include the changed item into the test scope.

7. Repeat the previous steps as often as required.

8. Choose [Save] to save the test plan so that it includes the changed items.
From the Test Plans – Nodes and Test Case Changes screen, you can jump via Goto > Test Packages Analysis or Test Cases Analysis or Defect Analysis to find Defects, which might be caused by a wrong test case version or process change.
5 EXECUTION OF AUTOMATED TEST CASES

5.1 The Tester Worklist

The Tester Worklist in SAP Solution Manager is the central place for the execution of all test cases within a test cycle, no matter if manual or automated test cases. A tester can execute all test cases which are assigned to his Test Package. The Tester Worklist provides an immediate overview of the status of Test Packages and Test cases and serves as a central place to enter problem messages for failed test cases. In addition, automated test cases can be scheduled for execution and test case errors can be reported.

The Tester Worklist can be opened by launching the Solution Manager Launchpad and selecting tile “My Tasks – Tester Worklist”:

For the handling of automated test cases, the following functionality is integrated into the Tester Worklist:

1. Execution of automated test cases
   - single test case execution
   - collective execution of all automated test cases in a test package
2. Reporting of Test Case Errors for repair by a Test Engineer
3. Scheduling of automated test cases
4. Direct access to execution logs
5. Overview on related Error Messages and Support Messages (Test Case Errors)
5.2 Execution of automated tests from the Tester Worklist

Prerequisites: In order to execute automated test cases from the Tester Worklist, a user needs at least the standard authorization roles for Testers (see appendix).

5.2.1 Single Test Execution

Steps for execution of a single automated test case:

1. In the Tester Worklist select a Test Package.
2. In the “Assigned Test Cases” area, select an automated test case (Type Test Configuration) and click button “Run”:

3. The “Start Options” screen comes up (for details regarding the different start options refer to chapter “Start Options for Automated Tests”). Select the appropriate start options.
4. Click button “Execute” (F8).
5. The external test tool is automatically launched and the automated test execution takes place.
6. After the execution has finished the log of the automated test case is displayed. Check the log and close it.
7. Refresh the Tester Worklist to update the status of the test case:

If the option “Copy Status to TWB” was selected on the start options screen (default setting) the status is automatically changed in the Tester Worklist. Otherwise, maintain the status manually by clicking on the status icon.

8. If the test execution failed create message:
   - In case of an application error, create an Error message. Click on button “Defect” and “Create Defect”
   - In case of a Test Case defect create a Test Case Error Message. Click on the button “Damaged Test Case”. See chapter “Repair of Damaged Test Cases/Report a Test Case Error” for a detailed description.
5.2.2 Collective execution of test cases

The collective execution of automated test cases allows to execute multiple test cases of a test package at once.

Steps for collective execution of multiple automated test cases:

1. In the Tester Worklist select a Test Package.
2. Click button Automatic Test

3. The "Automated Test - Immediate Execution" screen comes up.

4. Select all test cases to be executed in the collective mode and click the Execute Test button.

5. The "Start Options" screen comes up (for details regarding the different start options refer to chapter "Start Options for Automated Tests"). Select the appropriate start options:
9. Click button **Execute (F8)**.

10. The external test tool is automatically launched and the automated test cases are executed one after the other.

11. After the execution has finished the SAP internal log is displayed. Check the log an close it.

12. Refresh the Tester Worklist to update the status of the test case:

   ![Refresh](image)

   If the option “Copy Status to TWB” was selected on the start options screen (default setting) the status is automatically changed in the Tester Worklist. Otherwise, maintain the status manually by clicking on the status icon.

13. If the test execution failed create message:

   - In case of an application error, create an **Error message**. Click on button “Defect” and “Create Defect”
   - In case of a Test Case defect create a **Test Case Error Message**: Click on the button “Damaged Test Case”. See chapter “Repair of Damaged Test Cases/Report a Test Case Error” for a detailed description.
5.2.3   Handover of System Data for test execution

The system in which the automated test is executed is determined by the System Role defined in the Test Plan:

The system role can be changed as long as there was no Test Package assigned!

During runtime of the test case execution the connection to the respective target system is automatically determined through the System Data Container (SDC) assigned to the Test Configuration:
On the Test Configuration screen click on button Goto → System Data Container to open it. (Alternatively you can display a SDC via transaction SECATT).

The Logical Component Group contains the assignment of a specific System/Client to a System Role:

![Logical Component Group](image)

Double-click on the Logical Component Group to display or maintain the system assignments:

![Logical Component Group Editor](image)

### 5.2.4 Start Options for Automated Tests
The start options available on the Start Options screen depend on the context from where the execution is triggered and the test tool:

- Execution from the Tester Worklist
- Execution from Test Configuration Maintenance Screen
- Execution of scheduled automated tests

The most commonly used start options are the following:

**Error Behavior:**
You can specify how eCATT reacts to an error in a test. You can choose from the following:

- Termination. Execution continues with the next variant.
- Termination. Execution continues with the next test configuration.
- Termination of the start process. Execution does not continue.
- No termination. Execution continues with the next script command.

**Debugging Mode:**
You can specify if and when eCATT executes in the eCATT debugger. You can choose from the following:

- Normal Breakpoint Handling, Stop at BREAK
- Execution with Immediate Debugging
- Stop When an Error Occurs
- Ignore Breakpoints (Default for Collective Execution)

**Execution Control:**
When you select Execution Control, a dialog is available during replay in which you can pause or stop the script, or jump into the debugger. Any such user interaction will be recorded in the log.

**System Data:**
When starting a test, you can choose to use a system data container other than the one specified in the test configuration or test script.
You can also choose a different target system instead of the default target system. The default target system is:
- The target system of the test configuration when executing a test configuration. The maintenance system of the test script is not relevant.

**Logs:**
If you want the log to be displayed automatically at the end of a test, select Display Log.
If you want the archive flag to be set for the generated log, select Archive.

**RFC:**
You can specify that RFC connections be closed after a script or variant has been executed.
You can specify whether synchronous or asynchronous RFC should be used for remote access.

**Copy Status to TWB (Test Workbench):**
This setting automatically updates the status of the test case in the Test Workbench (traffic light). This setting is set by default.

**Activate TBOM recording:**
This setting triggers the TBOM recording used for the Business Process Change Analyzer.
Note: As a prerequisite the test case must have been assigned for TBOM recording in (please refer to chapter "Automatic recording of a TBOM").

5.3 **Scheduling of Automated Tests for unattended execution on a remote machine**

The Test Automation Framework provides functionality which allows to schedule the execution of automated test for a specific time, e.g. at night time (so called "light-out tests") and on remote machines (either physical PCs or virtual machines).
Steps for the scheduling and executing automated tests on a remote PC:

1. The Test Engineer/Coordinator schedules a job for execution of a test package in the Solution Manager on his own local PC.
2. He/she then logs on to the remote PC (or virtual machine) and registers and activates a Solution Manager session which listens periodically if a scheduled job needs to be executed.
3. Once the job execution has started, the automated tests contained in the test package are executed.
4. The external test tool executes the scripts and automatically logs on the relevant system under test.
5. When finished, the test results are returned and stored in Solution Manager and the Test Engineer/Coordinator can follow-up with an analysis of the test results and initiate further activities (e.g. create support messages or report a test case error) if needed.

5.3.1 Prerequisites

The following prerequisites need to be fulfilled for scheduling the execution of unattended tests on a remote machine:

- One or more physical clients or a Terminal Server/Citrix client are available
- The following software must be installed on the remote machines:
  - SAPGUI. Note: All Systems under test need to be maintained in the SAP Logon so they can be called by the external test tool. Enable scripting in the SAPGUI settings.
  - The external Test Tool.
- The machines need to be accessible remotely. Adjust the system settings of the remote machine if needed – after consultation with your IT security department. The Remote Settings can be change via
Start > Control Panel > System > Remote Settings > Remote Desktop: Click an option and specify who can connect, if needed.

- You need **access rights** (user and password) to logon to the remote machine.
- You need to know the **Computer name** of the remote machine on which the execution shall take place. The computer name of a windows computer can be found e.g. via Start > Control Panel > System.
- The scheduling of the job and the registration of the session on the remote machine need to be performed using the **same User-ID**.

### 5.3.2 How to schedule the unattended execution

1. In the Tester Worklist, select a test package and click button **Automatic Test → Schedule Execution**:

   ![Screen Shot](image)

2. The Schedule Execution screen is displayed. Enter a Job Name and select **Execution in Foreground**.
3. Enter a Start Time, e.g. a Date/Time and click **Save**.
4. A confirmation message is displayed:

5. Go back to the Tester Worklist and do a Refresh. The scheduled test package is marked with the Scheduled icon.

6. Logon to the remote machine on which the test cases shall be executed. This can be done e.g. using Windows program “Remote Desktop Connection”. Under the newest Windows versions this program can be found via Start > All Programs/Apps > Accessories > Remote Desktop Connection.
7. Enter the Computer name of the remote machine and click Connect:

8. Logon to the remote machine:

9. On the remote machine, start SAP Logon and logon to the same Solution Manager system on which you scheduled the job for the automated execution:
Important: The logon and the following registration needs to be performed with the *SAME* user which was used for scheduling the job. The User-ID is the sole connection between the scheduled job and the session on the remote machine.

10. Start transaction STPFE (Foreground Scheduler):

11. In the Foreground Scheduler click button *Register*:

12. In the Foreground Scheduler click button *Register*:
13. A system confirmation message is displayed:

- This SolMan session is now registered and active and checks every 60 seconds if a job needs to be executed which was scheduled by the same user.
- This round-trip from the remote session to the SolMan server keeps the session "alive". Therefore the session does not time-out after the usual period of inactivity.

14. Remote execution
Once the start time has been reached the registered session on the remote PC executes the scheduled test cases: The external test tool is automatically launched and the individual tests and variants are automatically executed on the remote PC.

15. After the execution has finished, refresh the Tester Worklist. The test package is indicated with an icon confirming that the job was [finished].

5.4 Execution Log

Execution logs can be viewed from different locations in Solution Manager.

a) Directly after the execution of an automated test case
- In the Start Options screen for the test case execution mark the Option "Log Display":

Commented [PC2]: Screenshot of icon “Job finished”
This option will bring up the execution log right after the execution of the automated test case.

b) SAP Solution Manager Launch Pad > Test Suite > Test Execution > Log

- In catalog Test Suite go to tile “Test Execution Log”:

It is possible to search for logs by various search criteria.

c) In Test Composition Environment

- Open a Test Configuration which was executed at least once and click button “DispLog”
- It is possible to open the Solution Manager Log (Composite log) or the Test Tool Log (log display of external test tool):
d) SECATT > Log selection

- Open transaction SECATT
- Click button “Logs”
- Enter selection criteria
- Hit Execute
- The result list displays all relevant logs with various data like status, start date & time, etc.

5.5 Notification emails

Testers can receive notification emails about the execution of test cases and the test results.

Notification e-mails can be sent in two cases:
- A failure occurs during the execution of a test case (only the first error triggers a notification email).
- The execution is finished.

To receive notification emails, the user must be registered. To register for receiving notification emails means that one of the following roles in the testing process must be assigned to the user:
- Tester
- Test Plan owner
• Test Package owner

Typically, the role “Tester” is used. You can use the additional roles “Test Plan owner” or “Test Package owner” if the testing process in your organization requires to do so. During development, for example, it can be useful that a user can appear in different roles: as a tester in test plan 1 and test package owner in test plan 2.

Note: The roles “Test Package owner” and “Test Plan owner” are attributes of the respective objects (Test Package or Test Plan, respectively). There is no relation to SAP authorization roles.

Prerequisites:
• You have authorization as an administrator.
• A business partner with a valid e-mail address is assigned to the user.

5.5.1 Register users for receiving notification emails when test executions fail

The sending of notification emails is setup individually for each user by maintaining a certain user parameter (AGS_SMT_NOTIFICATION) in the user data (SU01).

1. In the Solution Manager system call transaction SU01.
2. Select the user for which you want to activate notification emails.
3. Open the user data in change mode
4. Go to the Parameters tab.
5. Add Parameter AGS_SMT_NOTIFICATION. Enter parameter values depending on the role of the user in the test management. The attribute value has three digit positions.
   • If the user is a Tester, enter an X on the first position. The Tester is notified each time the user executes the test.
   • If the user is the Test Plan owner, enter a blank on the first position and an X on the second position. The Test Plan owner is notified each time someone executes a test from a test plan which is assigned to the Test Plan owner.
   • If the user is the Test Package owner, enter blanks on the first and second position and an X on the third position. The Test Package owner is notified each time someone executes a test from a test package which is assigned to the Test Package owner.
   ➔ If you want to be notified in every case, enter X on all three positions.

Example (for notification in all cases):

```plaintext
Display Users

<table>
<thead>
<tr>
<th>User</th>
<th>Status</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGS_SMT_NOTIFICATION</td>
<td>X00</td>
</tr>
</tbody>
</table>
6. Save your entries.

5.5.2 Change templates for notification emails

The templates used for the test management notification emails are based on SAP Smartforms technology. The following standard templates (Smartforms) are delivery by SAP:

"Test Execution Completed" (Smartform ID AGS_SMT_NOTIFY_EXECUTED)

"Test Failure Occurred" (Smartform ID AGS_SMT_NOTIFY_FAILED)

In order to adapt the notification templates to your needs, first copy the standard smartforms to the customer namespace and then change the assignment in customizing.
Prerequisites: You need authorization to execute transaction SMARTFORMS.

1. In Solution Manager open transaction SMARTFORMS
2. Search for one of the above standard templates and copy it to the customer namespace (Z*).
3. Adapt the smartform to your needs (please refer to the Smartforms documentation for more details regarding this technology).
4. Replace the smartforms assignment in customizing: Open the SAP Solution Manager Launch Pad > go to tab Test Suite Administration > open tab Preparation and click on “Email Notification Settings” (optional path via Customizing: SPRO > SAP Solution Manager > Capabilities (Optional) > Test Suite > Test Suite for SAP Solution Manager > Preparation > Setup > Standard Configuration > Email Notifications for Test Executions > Assign Email Forms for Notifications):
5.6 Access to Test Execution Logs

Up to SAP Solution Manager 7.2 SPS06, the Test Suite provides tile Test Execution – Logs, which provides a central access to all test execution logs of automated tests.

With SAP Solution Manager 7.2 SPS07, a new application will be provided to access and analyze test execution logs of automated tests. The application can be reached via tile Automated Tests – Results.

Via selection parameters, the user can influence which logs shall be displayed. The available logs are shown in an overview with columns that can be personalized by the user.

After line selection, the application shows details about the log, such as involved Test Configurations, test data and allows access to 3rd party logs of integrated partner test automation tools.
6 BUSINESS PROCESS CHANGE ANALYZER

Business Process Change Analyzer is an application which helps in executing a change impact analysis and allows to do a risk based test planning and execution. It is part of the end to end Integration Testing standard for SAP Solutions. The below picture shows the different phases of Business Process Change Analyzer:

Within the Test Automation Framework, BPCA is used for risk-based test scope optimization as well as for change impact analysis during the repair process of damaged test cases resulting in an accelerated maintenance of damaged test cases.

The information provided in this document focuses on BPCA in the context of the Test Automation Framework, especially how to make use of automated test cases for recording of TBOMs (Technical Bill Of Material). By recording TBOMs when executing automated tests significant time savings can be realized compared to manual recording of TBOMs.

6.1 Generation a TBOM (Technical Bill of Material)

A prerequisite for doing a Business Process Change Analysis is the availability of TBOMs for process steps. TBOMs are collected for each executable entity associated with various business process steps in the Solution Documentation. While executing these business process steps, a trace is running in the background which collects information of all SAP objects touched during this execution. This is a mandatory documentation step which is needed for all critical business process steps for evaluating them against any changes using BPCA.

TBOMs can be generated manually (while a user executes the process step) or automatically during execution of a test case related to the process step. The automatic recording of TBOMs is available with the introduction of the Test Automation Framework with SAP Solution Manager.

Also with Solution Manager 7.2 it is possible to create TBOMs for business processes which span across multiple systems: This feature will eliminate the limitation of not being able to create TBOMs when business process steps are executed on more than one system.
6.1.1 Automatic recording of a TBOM

For automatic recording of a TBOM using an automated test case, the following steps are required (example for transaction ME21N (Create Purchase Order):

Activate a test case for automatic TBOM recording:

1. Open the SAP Solution Manager Launch Pad and go to catalog Test Suite.
2. Go to tile Test Suite – Test Preparation.
3. Select the relevant Solution/ Branch and open the Solution Documentation via the link.
4. Navigate to the business process step in the process hierarchy.
5. Make sure an automated test case is assigned which covers the transaction for which the TBOM needs to be created.
6. Select the transaction ME21N (example)
7. Click on tab “Elements of ‘Create Purchase Order’”
8. In the “Elements of <transaction>” screen, do a right mouse click and choose Assign for TBOM Generation.
9. In the screen “Assign Test Configuration for Automatic TBOM Generation” all automated test cases from the same process level are listed.
10. Add a flag in the “Assigned” box for each test case you would like to enable for automated TBOM recording:

![Assign Test Configuration for Automatic TBOM Generation](image)

**Execute a test case with automatic TBOM recording:**

Note: The execution of an automated test case with TBOM recording is only possible within a test package. It is therefore required that a test plan and a test package have been created and assigned to a tester. From the Tester Worklist, the execution with TBOM recording can be triggered:

1. Open SAP Solution Manager Launch Pad and goto catalog Test Suite.
2. Go to tile My Tasks – Tester Worklist.
3. Select the relevant test package and test case which shall be executed
4. Click on “Run”
5. On the Start options screen, mark the option “Activate TBOM Recording”

![Start options screen](image)

6. Start the execution of the test case.
7. The TBOM is automatically created in the background.

You can verify the recording of the TBOM by checking the execution log or directly in the Solution Documentation > Transactions > Elements of <transaction>. 

---

77
6.1.2 Prerequisites and additional information on BPCA

For detailed information regarding the BPCA concept, prerequisites, setup and execution please see HowTo-Guide for BPCA, available in the SAP Service Marketplace: https://help.sap.com/viewer/fbc7b5ec5094fa0b6a2eb966160008f/7.2.06/en-US/e7ae657ed0291149adaca05adccce33abb.html
7  REPORTING

Regarding consistency and gap reports for Test Plans please refer to chapter “Test Planning of automated tests” > Coverage & Consistency checks for Test Plans.

7.1  Status Reporting

7.1.1  Check Status for current test phase

Goal:
Check the status for the current test phase.

Benefit:
The Multiple Test Plan Status Overview within SAP Solution Manager offers an efficient and flexible way of getting a snapshot of current test activities and related messages.

How to generate the report:
To create a list of test plans, which are created for a solution, proceed as follows:

1. Call up the SAP Solution Manager Launch Pad.
2. From the Test Suite catalog, choose the Test Suite Analytics tile.
3. Choose Test Execution Analytics and Multiple Test Plan Status Details.
   The view that opens contains an overview of all test plans and the status of the corresponding test cases on an aggregated level.
To evaluate the test plans on a more granular level, various tools are available, which will be explained in more detail below.

**Note:**
The Test Suite Analytics area within the Test Suite Catalog is the central place to access most of the test reporting tools. The SAP Solution Manager Launch Pad provides a high degree of flexibility when it comes to personalization and the definition of own queries.
4. In the contester Status area, search for the desired test plan and highlight the appropriate row. If required, use the filter functionality to find your test plan.

5. **Choose Goto > Test Plan – Test Package Analysis.**

   ![Test Plan - Test Package Analysis](image)

   This takes you to the **Test Plan – Test Package Analysis** screen.

   ![Test Case Analysis](image)

   This view provides you with high-level status information on the results of the selected test plan and its associated test packages. It also indicates whether problem messages have been created for the test phase.

   **Further Options - Detailed Status Analysis:**

6. To obtain more detailed information place the mouse pointer on either the test plan (top level node) or a test package in the structure.

7. **Choose Goto > Test Case Analysis**
This opens the Test Case Analysis screen for the selected test plan/test package.

The Test Case Analysis screen provides you with all kinds of test-related data (such as test status, result documentation, availability of messages and notes, etc.) and allows you to drill down to individual test cases.

Note: We just show here an example of the reporting opportunities for the Test Execution Analytics. There are some more reportings available, which can be used for the different needs. Please refer to the following page to get more information: https://help.sap.com/viewer/fbc7b5ecf5094fe0b6a2eb96616008f/7.2.06/en-US/4cb41b5684b77c65e10000000a44147b.html

7.2 Status and Progress Analytics

You can analyze test management using the BI reporting analyses which allow you to access various graphic overviews and extensive filter capabilities. You also have access to the functions of the BI system. These tools are particularly useful for quality managers and test organizers.
7.2.1 Status Analytics

Goal:
Display an overview of the status of selected test plans/test packages/test cases.

Benefit:
Efficient and flexible way to visualize the test status on a day-to-day basis.

How to generate the report:
1. Call up SAP Solution Manager Launch Pad.
2. From the Test Suite catalog, choose tile Test Suite - Analytics.
3. From the selection, choose Status and Progress Analytics > Test Plan – Status Analytics or Test Package – Status Analytics or Test Case – Status Analytics.
4. In the search area, select the values for that you want to perform the status reporting.

The BI report for the current status of the selected test plan is displayed in the same window under Status – Test Plan Chart.
Using drilldown and filter functionalities, you can adjust the view on the data to obtain different levels of granularity. In addition, detailed information is displayed in a tabular overview at the bottom of the screen.

### 7.2.2 Progress Analytics

**Goal:**
Display the progress of the test case status over a certain period to support project leads and test coordinators in detecting potential delays.

**Benefit:**
Efficient and flexible way to visualize the test progress on a day-to-day basis.

**How to generate the report:**

1. Call up SAP Solution Manager Launch Pad.
2. From the Test Suite catalog, choose tile Test Suite - Analytics.
3. From the selection, choose Status and Progress Analytics > Progress Analytics.
4. In the search area, select the values for that you want to perform the progress reporting.
The BI report for the test progress of the selected test plan is displayed in the same window.

Using drilldown and filter functionalities, you can adjust the view on the data to obtain different levels of granularity.
In addition, detailed information is displayed in a tabular overview at the bottom of the screen.

7.2.3 Defect Analytics

Goal:
Display an overview of the number and status of test-related defects for a selected test plan.
**Benefit:**
Efficient and flexible way to analyze message-related data.

**How to generate the report:**

1. Call up SAP Solution Manager Launch Pad.
2. From the Test Suite catalog, choose tile Test Suite - Analytics.
3. From the selection, choose Status and Progress Analytics > Defect Analytics – Status or Priority or Operation.
4. In the search area, select the values for that you want to perform the defect reporting.

The BI report for test-related defects is displayed in the same window.

You get an overview of the number and processing status of incidents at a specified time. By default, the system uses the date of the last data extraction as a filter.

### 7.2.4 Test Effort Analytics

**Goal:**
Display an overview of the test effort and show if the planned effort is sufficient or whether it will be exceeded.

**Benefit:**
The test effort report allows you to analyze the ratio between planned effort, actual effort, and expected total effort. It can support project leads and test coordinators in identifying potential resource bottlenecks.
How to generate the report:

1. Call up SAP Solution Manager Launch Pad.
2. From the Test Suite catalog, choose tile Test Suite - Analytics.
3. From the selection, choose Status and Progress Analytics > Test Effort Analytics.
4. In the search area, select the values for that you want to perform the defect reporting.

5. The BI report for test-effort is displayed in a new window.

   The test effort report indicates the progress of the test status and visualizes the evolution of the test effort. Apart from that, the report provides a graphic representation of effort that is above and below plan. The test effort report also allows you to compare planned effort, actual effort, and expected total effort. To detect deviations between actual effort and expected effort, proceed as follows:
8 REPAIR OF DAMAGED TEST CASES

The Test Automation Framework supports the repair process for damaged test cases by using the IT Service Management functionalities with an own transaction type SMDT – Test Case Error.

8.1 Workflow to report a Damaged Test Case

The failure of an automated test case execution can have various reasons. In case of an application error, the Tester/QA Expert creates a support message. If the failure is rather related to a problem within the test configuration itself, a Test Case Error Message is created. The Test Case Error Message is automatically send to the responsible department (e.g. Test Engineer) who analyses and repairs the test case and initiates a re-test by the Tester.

8.1.1 Prerequisites

- Role SAP_STWB_WITC_CREATE (Role for Work Item Test Case (WITC) Creator) needs to be assigned to the user.
- In the Tester Worklist of the Test Suite, you have performed a test case which failed. The status shows red (Error).

8.1.2 Create a Test Case Error Message

1. In the Tester Worklist, select a failed test case.
2. Click button “Damaged Test Case”.

![My Tasks - Tester Worklist](image)
8.2 Repair environment for Test Engineers

Once a Test Case Error Message has been created by the Tester, the indicated Test Engineer automatically receives the message in its Damaged Test Case Worklist.

As the Repair environment is now part of Standard IT Service Management, you will have several possibilities to have such a worklist available:

1. Home > My Messages – Assigned to Me
2. Worklist > Business Transactions
3. Saved Search (IT Service Management > Incidents > Enter Search Criteria > Save)

![Saved Search Image]

8.2.1 Damaged Test Case Worklist

The Damaged Test Case Worklist supports the administration of the repair process of damaged test cases.

![Damaged Test Case Worklist Image]

Test Engineers and QA Experts use the Damaged Test Case Worklist for monitoring the processing status of Test Case Error Messages. The following actions are possible directly from the Damaged Test Case Worklist:
1. **Run Test Configuration**: Directly executes the Test Configuration (opens start options screen)
2. **Display Test Case**: Directly opens the Test Configuration in display mode
3. **Distribute**: Shows a list of all test plans, where the damaged test case is assigned. The test case can be marked as damaged automatically for this test plans as well, without an additional execution.
4. **Test Script > Display**: Opens the Test Script in display mode in the external test tool
5. **Test Script > Change**: Opens the Test Script in edit mode in the external test tool and allows immediate repair or adaptation of the test script
6. **Execution Log > Original**: Shows the initial execution log from the failed test case execution
7. **Execution Log > Latest**: Shows the latest execution of the test case.
8. **Goto > Usage Location**: Directly opens the location (process step) in the corresponding Solution Manager solution (transaction SOLDOC) and show the attributes of the Test Configuration
9. **Goto > Original Test Status**: Shows the test execution status of the initial failed test execution
10. **Goto > Latest Test Status**: Shows the test execution status of the latest test execution
11. **Goto > Test Data Variant**: Opens the Test Data screen of the corresponding Test Configuration
12. Click on `<Damage ID>`: Open the IT Service Management UI; in the Assignment Block for Related Test Case you will find all the root of this incident message

### 8.2.1.1 Prerequisites for handling Test Case Error Messages

- The following authorization roles exist and need to be assigned to the respective users:
  - SAP_STWB_WITC_ADMIN: Role for Work Item Test Case (WITC) Administrator
  - SAP_STWB_WITC_CREATE: Role for Work Item Test Case (WITC) Creator
  - SAP_STWB_WITC_DISP: Role for Work Item Test Case (WITC) Display
  - SAP_STWB_WITC_EXE: Role for Work Item Test Case (WITC) Processor
The Test Engineer has assigned all relevant roles needed for analyzing, executing and maintaining automated test cases.

8.2.2 Customizing of the transaction type for Test Case Errors

The Damaged Test Case Work Item is based on a CRM business transaction type. The default transaction type delivered by SAP and maintained for the creation of Damaged Test Case Work Items is SMDT “Test Case Error”.

The assignment of the transaction type can be changed via SPRO > SAP Solution Manager Implementation Guide > SAP Solution Manager > Capabilities (Optional) > Test Suite > Test Suite for SAP Solution Manager > Preparation > Setup > Incidents > Assign Business Transaction Types and Templates:

![Diagram showing the steps to customize transaction types](image1)

The transaction type relevant for Damaged Test Cases is maintained for Application "Test Case Error".

![Customized Transaction Types for Test Suite](image2)

You can either use the standard transaction type SMDT "Test Case Error" or you customize your own transaction type by copying transaction type SMDT to the customer namespace and adapt it to your needs,
e.g. by defining your own status profile, partner profile, text profile, categorization and more. Transaction type customizing can be done via SPRO > SAP Solution Manager Implementation Guide > SAP Solution Manager > Capabilities (Optional) > IT Service Management > Transactions.

In order to copy the standard transaction type SMDT (as delivered by SAP) to the customer name space proceed as follows:

1. SPRO > SAP Solution Manager Implementation Guide > SAP Solution Manager > Capabilities (Optional) > IT Service Management > Transactions > Define Transaction Types:
   - Search for Transaction Type SMDT and copy it to the customer name space, e.g. ZSMD

2. SPRO > SAP Solution Manager Implementation Guide > SAP Solution Manager > Capabilities (Optional) > IT Service Management > Transactions > Specify Several Transaction Types:
   - a) Specify Transaction Type
     - Copy the standard entry for transaction type SMDT and adapt it to your new transaction type. Increase the Seq.No. by 1:
   - b) Classify Transaction Type
     - Copy the standard entry for transaction type SMDT and adapt it to your new transaction type.

3. Assign the newly created Transaction Type in the Central Test Workbench Settings as described above (replace SMDT with your own one, e.g. ZSMD):
Please refer to the CRM transaction type customizing documentation for further details.

8.2.2.1 Status Customizing

In order to adapt the Status profile to your needs, proceed as follows:

1. Copy the standard status profile SMDT_STD to the customer name space:
   
   [SPRO > SAP Solution Manager Implementation Guide > SAP Solution Manager > Capabilities (Optional) > IT Service Management > Status Profile > Define Status Profile for User Status]

   ➔ Search for standard status profile SMDT_STD and copy it to the customer name space, e.g. ZSMD_STATUS:

   ➔ Adapt the new status profile to your needs. (See customizing documentation or online documentation for further details.)

2. Assign the newly created status profile to your transaction type:
   
   [SPRO > SAP Solution Manager Implementation Guide > SAP Solution Manager > Capabilities (Optional) > IT Service Management > Transactions > Define Transaction Types]

   ➔ Open transaction type ZSMD and replace the standard status profile with the new one.
8.2.2.2 Category Customizing

The Category of a Test Case Error message allows to classify the reason for a damage in a structured way. The so-called multi-level categorization can contain up to 4 Levels:

- It categorizes the message in a structured and standardized way
- It can be used for the determination of responsibilities and the dispatching of messages
- It is important for later reporting (e.g. what are the most common reasons for damaged test cases?)
SAP does not deliver a standard Categorization Schema for Test Case Error Messages. Customers can setup their own Categorization Schema.

How to create a Categorization Schema

Instructions on how to create a Categorization Schema for Test Case Error Messages can be found in the IMG:

SPRO > SAP Solution Manager Implementation Guide > SAP Solution Manager > Capabilities (Optional) > Change Control Management > Transactions > Multilevel Categorization > Assign Transaction Types to Catalog Categories (see documentation for all assigned customizing steps).

1. Map Transaction Types to Catalog Categories

Under Transaction Types to Catalog Categories:
If you have created your own transaction type for Damaged Test Cases (e.g. ZSMD) copy the entry for standard transaction type SMDT for your new transaction type. Make sure the Catalog Category is “Defect Locations/Object Parts”:

2. Create a Categorization Schema

The creation, activation and maintenance of Categorization Schemes are done in the Category Modeler. There are two different ways to access the Category Modeler.

a) Via BSP Application

In the favorites of your Easy Access Menu, add an entry of type SAP CRM People Centric UI Application, Application = CRMM_ERM_CAT, description = Category Modeler.

Prerequisite: The service “CRM_BSP_FRAME” needs to be activated via transaction SICF.
b) Via IT Service Management

The Category Modeler can also be accessed via the CRM WebUI, Business Role “SOLMANPRO – Solution Manager ITSM” > Service Operations > Categorization Schemas

Create a new Categorization Schema.
Indicate Application "Service Request/Incident" with Parameter "Transaction Type/Catalog category" and the value: <Description of Transaction Type for Test Case Error Messages> and <Defect Locations/Object Parts>.
Define your multi-level categorization entries.
Save and activate your new Categorization Schema.

For additional information regarding the use of the Category Modeler, see also online help:
9 APPENDIX

9.1 Further documentation

For further help on Solution Manager and Test Management please refer to the following links (some of the links are accessible only to SAP Customers and Partners)

- **SAP Standard Help - SAP Solution Manager → Test Suite**
  - [https://help.sap.com/viewer/c41364787a54db59d18cb074ce3dafd/7.2.06/de-DE](https://help.sap.com/viewer/c41364787a54db59d18cb074ce3dafd/7.2.06/de-DE)

- **Ramp-up knowledge transfer - SAP Solution Manager 7.2 - EHP1: Build, Test and Deploy with SAP Solution Manager.**

- **SAP Solution Manager**

- **SAP Solution Manager e-learning material**

- **SAP Solution Manager Media Center**
  - [https://wpb101101.hana.ondemand.com/wpb/wa/wa/~tag/published/index.html?library=library.txt&show=group!GR_267560D84F4CD84#show=group!GR_267560D84F4CD84&library=library.txt](https://wpb101101.hana.ondemand.com/wpb/wa/wa/~tag/published/index.html?library=library.txt&show=group!GR_267560D84F4CD84#show=group!GR_267560D84F4CD84&library=library.txt)

- **Focused Solutions circle partner:**
  - [https://support.sap.com/solution-manager/focused/focused-solutions-circle.html](https://support.sap.com/solution-manager/focused/focused-solutions-circle.html)