Testimony - Administrators Guide

2.50 — Last update: 6 September 2022

Basis Technologies

Table of Contents

1. Introduction **3**

2. Product Overview **4**

2.1. Testimony Process 5

2.2. Technical Overview 6

3. Administrators Guide **8**

3.1. Security 9

3.1.1. Overview 10

3.1.2. SAP Roles 11

3.1.3. Testimony User Roles 12

3.1.4. Action Management 14

3.1.5. UI Profiles 16

3.1.6. Securing Recordings and Results 18

3.2. Test Plan and System Connections 19

3.2.1. Creating a Test Plan 20

3.2.2. System Setup 22

3.2.3. Authorisations 32

3.2.4. Managing Test Plans 33

3.3. Version check 35

3.4. Licence key check 36

3.5. Filters, Suppressions & Plan Configuration 37

3.5.1. Exclusions 39

3.5.2. Defect Suppression 43

3.5.2.1. Step-level suppression 44

3.5.2.1.1. Example of a step-level suppression 45

3.5.2.1.2. Creating a step-level suppression 46

3.5.2.2. Script-level suppression 50

3.5.2.2.1. Example of a script-level suppression 51

3.5.2.2.2. Creating a script-level suppression 52

3.5.3. Sampling 54

3.5.4. User Preference and Plan Configuration 56

3.5.4.1. Default Validation Methods 61

3.5.4.2. Object Level Validation Method 62

3.6. Notifications 63

3.6.1. Quick Notifications Configuration 64

3.6.2. Detailed Notification Configuration 66

3.6.3. Batch Job for Notifications 71

3.7. Performing a Recording 72

3.7.1. Recording Preparation 73

3.7.1.1. Settings for DB writes 80

3.7.1.2. Reviewing shared memory parameters ... 83
3.7.1.3. Changing optional/mandatory enhancements ... 86
3.7.1.4. Setting up file system monitoring ... 87
3.7.1.5. Preparing the backup ... 89
3.7.2. Start Recording ... 91
3.7.3. Monitoring a Recording ... 94
3.7.3.1. Shared Memory Explorer ... 96
3.7.4. Stop Recording ... 98
3.7.5. Aborting a Recording ... 100
3.7.5.1. Manually Aborting a Recording ... 102
3.8. Review Recordings/Transfer to Repository ... 104
3.8.1. Check for application server files ... 107
3.8.2. The Recording Performance Analysis ... 108
3.9. Building the Execution Queue ... 115
3.9.1. Run the playback simulation ... 117
3.9.2. Run the screen size analysis ... 119
3.10. Analyzing Coverage ... 121
3.10.1. Usage Retrieval ... 123
3.10.2. Prioritization ... 126
3.10.3. Coverage Analysis ... 131
3.11. Performing a Playback ... 134
3.11.1. Preparing Playback ... 135
3.11.1.1. System Refresh Steps ... 136
3.11.1.2. Predictive Difference Analyzer ... 142
3.11.1.2.1. Supported Objects and Check Types ... 143
3.11.1.2.2. Preparation Phase ... 145
3.11.1.2.3. Comparison Phase ... 147
3.11.1.2.4. Results Phase ... 148
3.11.1.2.4.1. Remediation of Test Scripts ... 150
3.11.1.3. Batch processes on the central system ... 152
3.11.1.4. Activate Playback Enhancements ... 154
3.11.1.5. Execute Check Steps ... 156
3.11.1.6. Reset Target Date and Time ... 157
3.11.1.6.1. Checking time travel on SAP ... 159
3.11.1.7. Start the Bots ... 160
3.11.2. Start Playback ... 162
3.11.3. Monitor Playback ... 163
3.11.4. Stop Playback ... 164
3.12. Double Playback ... 165
3.12.1. Double playback Preparation ... 167
3.12.2. Causes of noise during playbacks ... 168
3.13. Review Playback Results ... 171

3.13.1. Playback Overview 172
3.13.2. Script statuses and execution results 174
3.13.2.1. Status: Complete 176
3.13.2.2. Status: Error 177
3.13.2.3. Status: Partially complete 178
3.13.2.4. Status: Suppressed 179
3.13.2.5. Status: Cancelled 180
3.13.3. Playback duration analysis 184
3.13.4. Root Cause Analysis 186
3.13.4.1. Setup of RCA 187
3.13.4.2. Link Creation 188
3.13.4.3. Build Lists 189
3.13.4.4. RCA Results 190
3.14. Useful Programs and Reports 191

4. Glossary 193

5. Software Support 196
5.1. Support from Basis Technologies 197

1. Introduction

Welcome to the Testimony Administrators Guide, a detailed overview of the product and processes to enable your organization to record and playback a successful automated regression test. The Administrator role within Testimony allows the user to configure and execute all functions of the tool. Because of this, the role is typically only given to a few select team members.

In addition to this document, you'll also find the Quick Start Guide which is a high level overview of the key functions and the Testers' Guide which details the defect management aspect of Testimony

Support is also available from Basis Technologies if required via the contact details outlined at the end of this guide.

2. Product Overview

Testimony is a one of a kind, digital regression test generation and execution product developed by Basis Technologies.

It is installed on and operates within the SAP ABAP stack and is particularly used to reduce the duration and cost of regression testing SAP systems.

It does this by eliminating effort in:

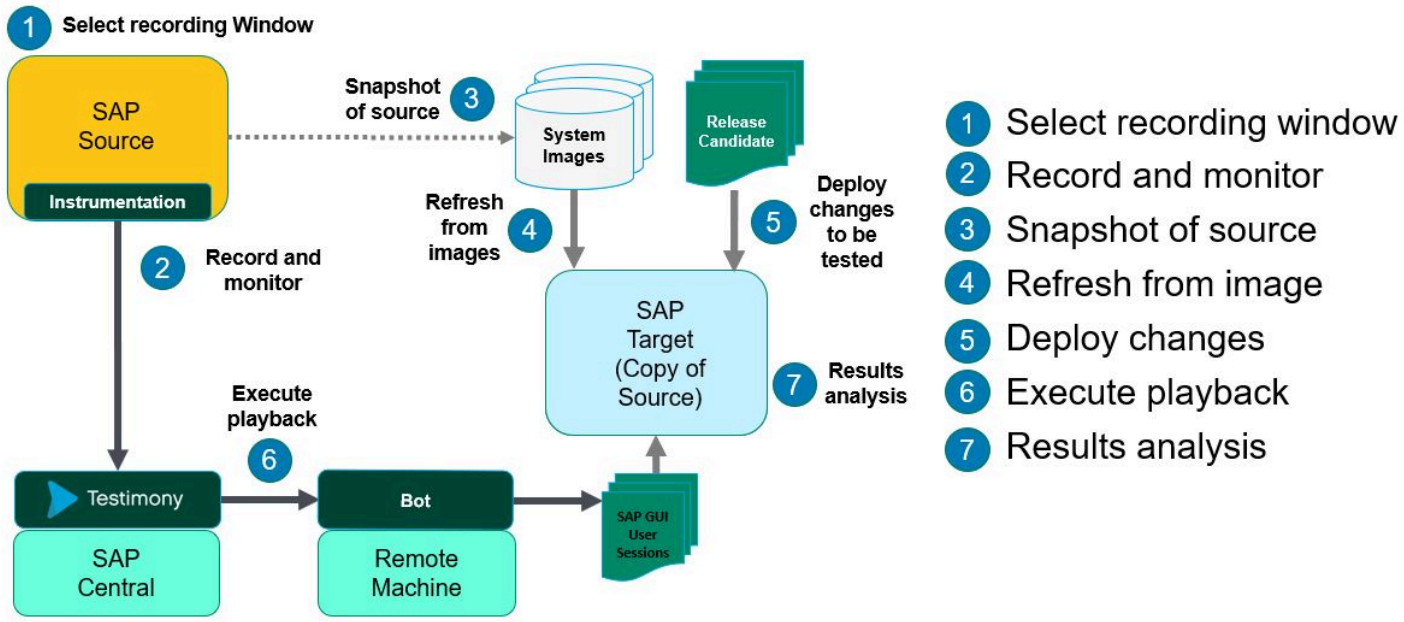
- Regression test coverage analysis and planning
- Regression test script development
- Regression test execution (including initial defect analysis)
- Test user maintenance
- Test data creation

It uniquely generates regression test scripts that reflect how your business teams really use their SAP applications on a day to day, month on month, and year on year basis.

This product comes to the fore when managing the following risk scenarios, when you cannot afford to break the as-is production service:

- SAP Upgrades and Support Packs
- SAP Cloud re-platforming
- System migrations
- Kernel upgrades
- Application of ongoing SAP security patches
- Regular production support and maintenance
- Project releases

2.1. Testimony Process



2.2. Technical Overview

The following steps provide an overview at a more technical level of how the Testimony product operates.

Installation Process

Testimony is installed as a third-party add-on solution to a central ABAP system (typically your SAP Solution Manager system). Transports are provided, which include code and enhancements that must be installed on the recording (Source) systems and the playback (Target) systems. Please see the link here to the installation guide.

Test Plan Configuration

All Testimony processes are executed within the organization of a “Test Plan”. These represent the regression test cycle(s) that are currently being performed for the necessary scenario (e.g. HANA DB migration, AWS re-platforming, technical upgrade etc). Within the central Testimony system, you must configure a test plan and provide some basic information relating to it (e.g. which the system(s) involved for recording and playback, which users are involved from a testing perspective). RFC destinations must be setup to allow Testimony to trigger both the recording and playback processes.

Recording Setup

A setup activity is then run to prepare the system to be recorded. After this point, the system is ready to enable you to activate the “recording” function in Testimony. Please note that until you perform this activation, there is zero impact on production service.

Recording Activation

When you are ready to perform a regression test cycle, the Testimony Administrator is able to activate the recording process. This will record all technical operations upon the SAP source system including activities such as SAP GUI dialog transactions and RFC / BAPI calls. Immediately after activating the recording, a backup of the source (SAP production) system should be taken. This will be used as the basis for the creation of the playback system. It is typically envisaged that a period of approximately 24 hours of operational activity would be recorded. At the end of the 24 hour recording period, the Testimony Administrator deactivates the recording process. Due to the unique manner in which Testimony operates, the impact on the production system is near zero (in terms of response times and system performance).

Coverage Analysis

The recording process has in effect automatically generated an entire test script library covering percentage of the customers actual SAP system usage. After the recording process is finished, the testing team can understand the level of coverage they have with the recording based upon the usage information of what the customer actually uses over the past M months (where M is typically 3 months). If key transactions are missing (e.g. month end processes) then the coverage analysis within Testimony can provide this information. In this case, you can either test these processes outside of Testimony or look to include the

periods when these processes run in your next Testimony recording.

Playback Process

Once the recording and coverage process are run, then it is possible to execute the playback process. The playback (target) system that was built from the production database backup must now be started. The Testimony playback agent must be setup on the target system and “bots” run upon supporting infrastructure including Windows based machines. The bots simulate the actions of users as well as other external systems during the playback. Changes that are being made must be deployed to the target system (e.g. upgrade process, project release transports deployed, database migrated). Once the landscape is ready, the playback process can begin. This is simply a process of starting the “execution queue” and allowing it to execute (technically). If issues arise, notifications can be sent to appropriate test team members. The queue can be restarted if problems are detected.

Results Review

After the playback process has completed, the test team can review the results. They are specifically looking for issues that have arisen during the playback where the “expected output” differs from the “actual output”. Any discrepancies identified can be automatically generated as defects. The testing and development team can review these defects (once created) and provide a resolution. Once the bulk of defects are resolved, a new test cycle can be created and the system copy image restored. The playback process can be repeated to ensure that defects identified have been correctly resolved.

3. Administrators Guide

The quick start guide is to be used after installation and includes the key basic steps required to create a recording and a playback. Please check the installation guide here if that process is not complete. It details the process steps required to allow you to use Testimony for a Regression test cycle. It assumes a reasonable level of knowledge of the Testimony architecture, as well as some knowledge of SAP systems architecture and SAP Basis.

Each section in the Quick Start Guide covers a separate step of the overall process. The steps should be followed in the order in which they are described.

- Test Plan and System Connections
  A test plan must first be created and configured prior to any recordings being performed or playbacks.

- Performing a Recording
  This is the process for activating and monitoring the recording on the source systems.

- Review Recordings/Send to Repository
  Once recording is complete, you should review the recordings that have been captured and move them to the Testimony script repository.

- Performing a Playback
  This is the process to prepare for, execute and monitor the playback of recorded activities on the target systems.

- Review Playback Results
  The end result of the process which highlights discrepancies between what was recorded from the live system and what potential issues have been identified in the test system that is played back into.

3.1. Security

The following sections describe the various security considerations within Testimony.

- Overview
  Provides an overview of the various security concepts.

- SAP Roles
  Details setup of standard SAP roles provided with Testimony for core user setup.

- Testimony User Roles
  Details the internal role concept within Testimony for users operating the product.

- Action Management
  Describes the configuration screens available for customizing security of actions and users.

- UI Profiles
  Describes how to modify which menu items and functions are viewable to Testimony users.

- Securing Recordings and Results
  Details the manner in which recordings of live systems and associated playbacks can be locked down.

3.1.1. Overview

Security for Testimony has four basic layers: SAP Roles, Testimony User Roles, Action Management, and UI Profiles. This document will discuss how security is assigned and controlled for the Testimony application.

Administrators are automatically granted full access to all functionality within the Testimony application. SAP-level access for all users and Testimony-level access for non-administrator users access are further discussed within this document.

3.1.2. SAP Roles

Testimony is delivered with the following SAP authorization roles:

Role Name	Description
/BTI/AUT_BATCH_ADMIN	Background Processing Administrator Role for the Default Batch User during playback
/BTI/AUT_BOT_RFC	Authorizations for the remote bot RFC user
/BTI/AUT_BOT_SIMULATION	Authorizations for the BOT Simulation report
/BTI/AUT_CENTRAL_ADMINISTRATOR	Testimony Central System Administrator role
/BTI/AUT_CTS_ADMIN	Testimony transport admin role for root cause analysis
/BTI/AUT_INBOUND_RFC	Authorizations for Inbound RFC users within the target system during playback
/BTI/AUT_NCO_METADATA_ACCESS	Role to access NCo metadata
/BTI/AUT_SOURCE_RFC	RFC authorizations for the source system RFC user for recording
/BTI/AUT_TARGET_RFC	RFC authorizations for the target system RFC user for playback

! Customer security policy may require a custom role be created for assignment to users within SAP. If so, these custom roles must contain all of the included authorization objects and values of the delivered roles or Testimony may not operate as expected.

* It is recommended to regenerate these roles when they are imported into the system for the first time

Testimony users will also need to be granted access to transaction code /BTI/AUT to access Testimony on the Central system. This is done through the standard SAP security processes and is not covered in this document.

All other access to Testimony application functions and tools is handled within the Testimony application as detailed in this Administrators Guide in the following sections:
Testimony User Roles
Action Management
UI Profiles

3.1.3. Testimony User Roles

> Note the user roles are assigned and maintained in Testimony rather than with standard SAP Roles, however, users will require the transaction /BTI/AUT in their standard SAP authorisations to access Testimony.

Within Testimony, users are considered to be **administrators** or **non-administrators**.

Administrators have full access to all Testimony functions whereas non-administrators only have access to certain functions.

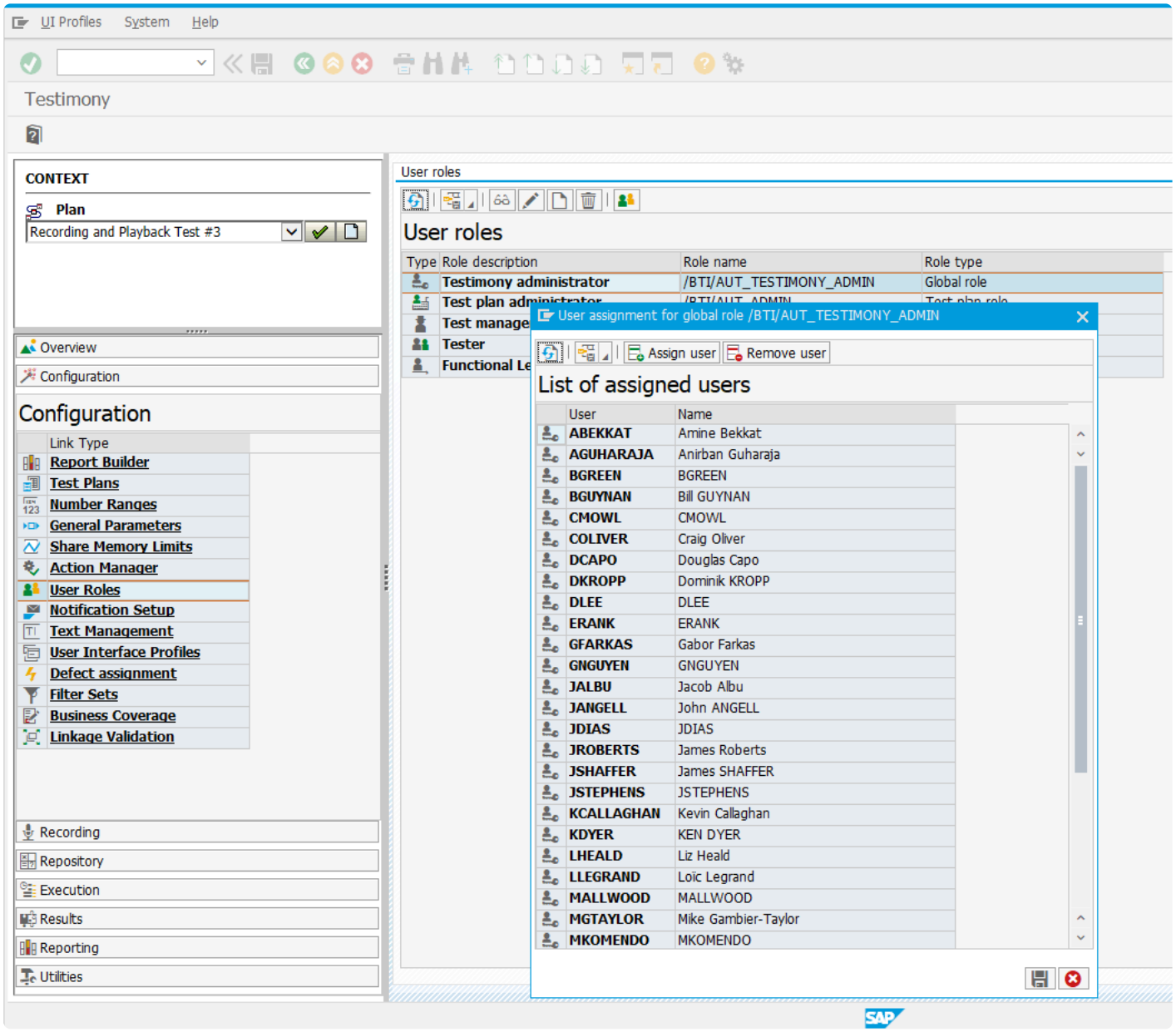
Authorisation for Testimony users requires one of the following Testimony roles to be assigned:

Role Name	Description
Testimony Administrator	This is a Global role that contains all authorizations to execute and run all steps in Testimony and should be assigned to all Admin users.
Test Plan Administrator	This role contains all authorizations at the Test Plan level to execute and run all steps for the test plan in Testimony and should be assigned to all Test Plan Admins.
Test Manager	This role contains all authorizations required to execute and run test management all steps in Testimony and should be assigned to all Test Managers.
Tester	This role contains all authorizations required for test execution steps in Testimony and should be assigned to all Testers.

The non-administrator roles must be assigned at the Test Plan level. For details on how to assign users to each plan, see Authorizations in the Test Plan section of this guide.

The actions inlcuded in Testimony roles can be adjusted via the Action Manager. Check that section for details.

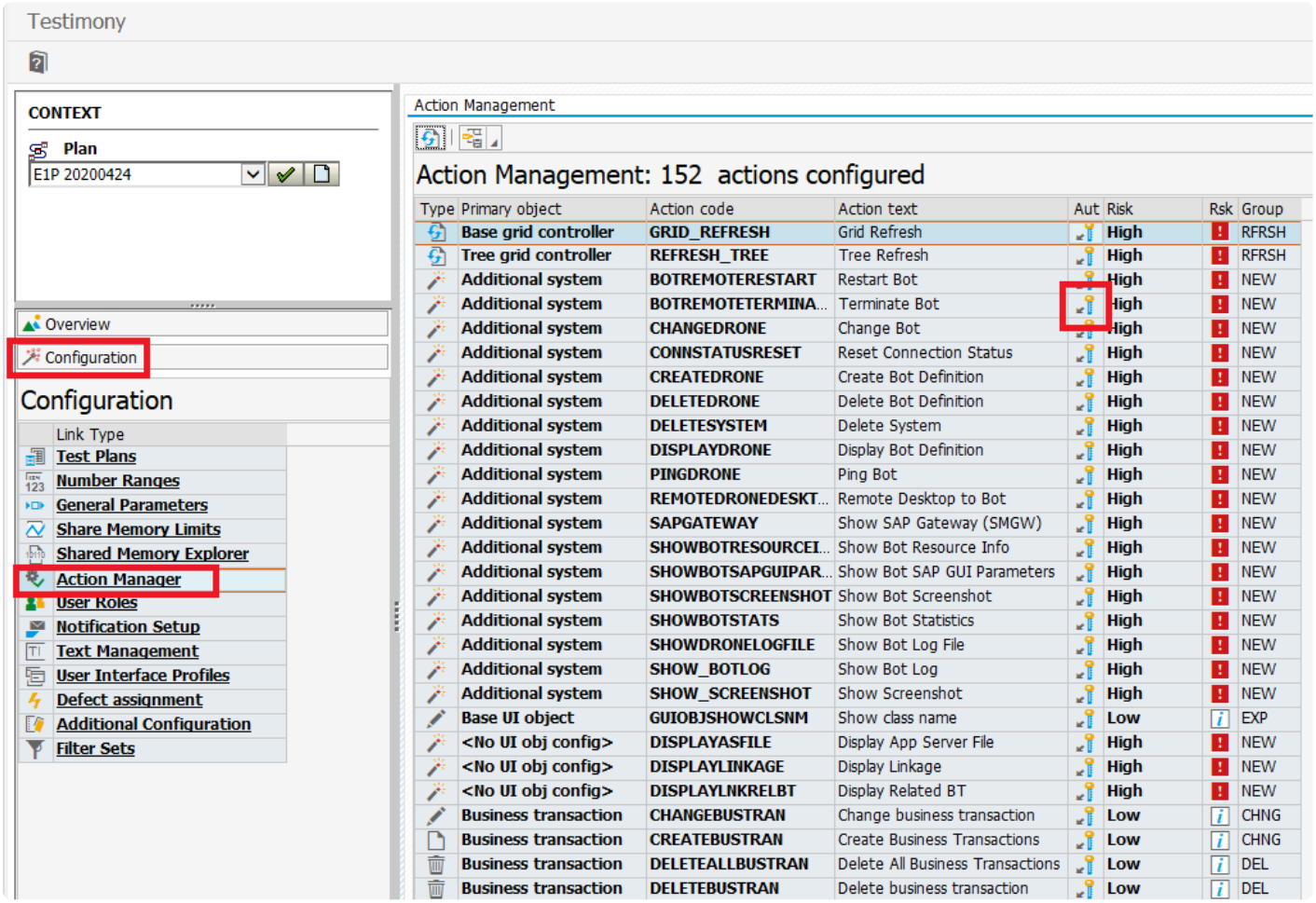
Administrator users are assigned in Testimony by navigating to **Configuration>User Roles**, clicking the  button and then the ‘Assign User’ button. Then add a new global administrator to the list and save.



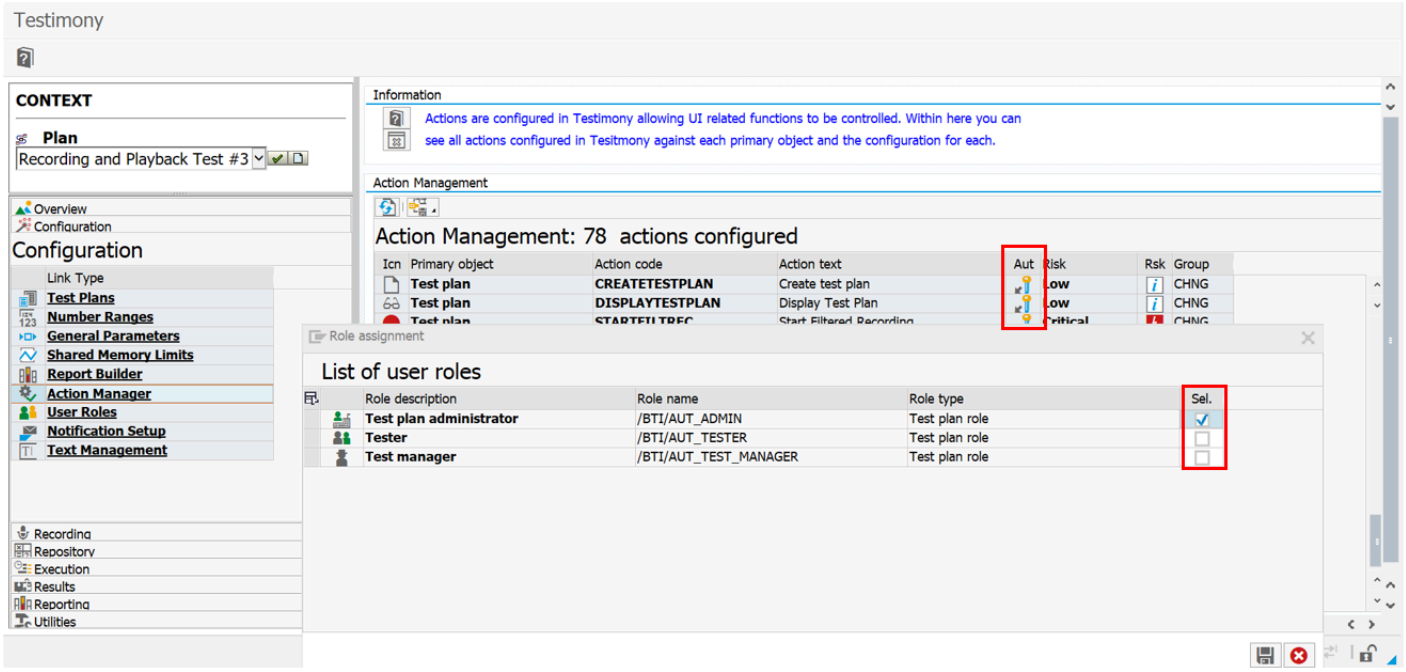
The Admin user being assigned here must already exist in the Central system and should be assigned SAP role /BTI/AUT_CENTRAL_ADMINISTRATOR through the standard SAP security process.

3.1.4. Action Management

Access for each role to the individual actions/activities within Testimony are assigned within the Action Manager. In **Configuration>Action Manager** you can see all authorizations within Testimony, information about each action, and assign them to the appropriate roles by clicking in the icon in the “Aut” column.



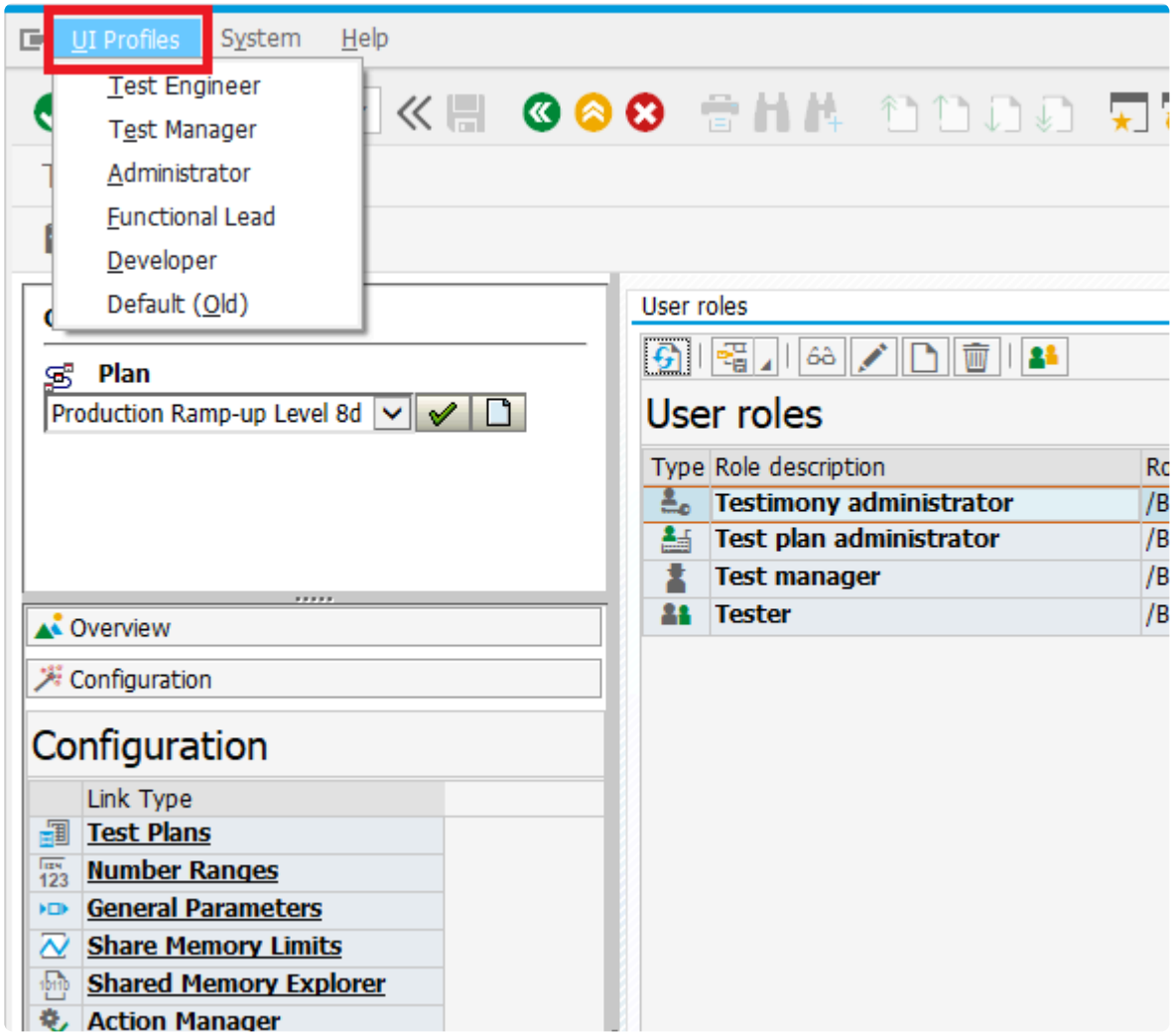
Within each of these Action Management Objects, Testimony Administrators can restrict role assignment to the list of user roles and which user roles can execute those Action Management Objects with the select indicator set to active or inactive as below:


Testimony
CONTEXT
Plan
Recording and Playback Test #3
Overview
Configuration
Configuration
Link Type
Test Plans
Number Ranges
General Parameters
Shared Memory Limits
Report Builder
Action Manager
User Roles
Notification Setup
Text Management
Recording
Repository
Execution
Results
Reporting
Utilities
Information
Actions are configured in Testimony allowing UI related functions to be controlled. Within here you can
see all actions configured in Tesitmony against each primary object and the configuration for each.
Action Management
Action Management: 78 actions configured
Icn Primary object Action code Action text Aut Risk Rsk Group
Test plan CREATETESTPLAN Create test plan Low CHNG
Test plan DISPLAYTESTPLAN Display Test Plan Low CHNG
Role assignment
List of user roles
Role description Role name Role type Sel.
Test plan administrator /BTI/AUT_ADMIN Test plan role
Tester /BTI/AUT_TESTER Test plan role
Test manager /BTI/AUT_TEST_MANAGER Test plan role

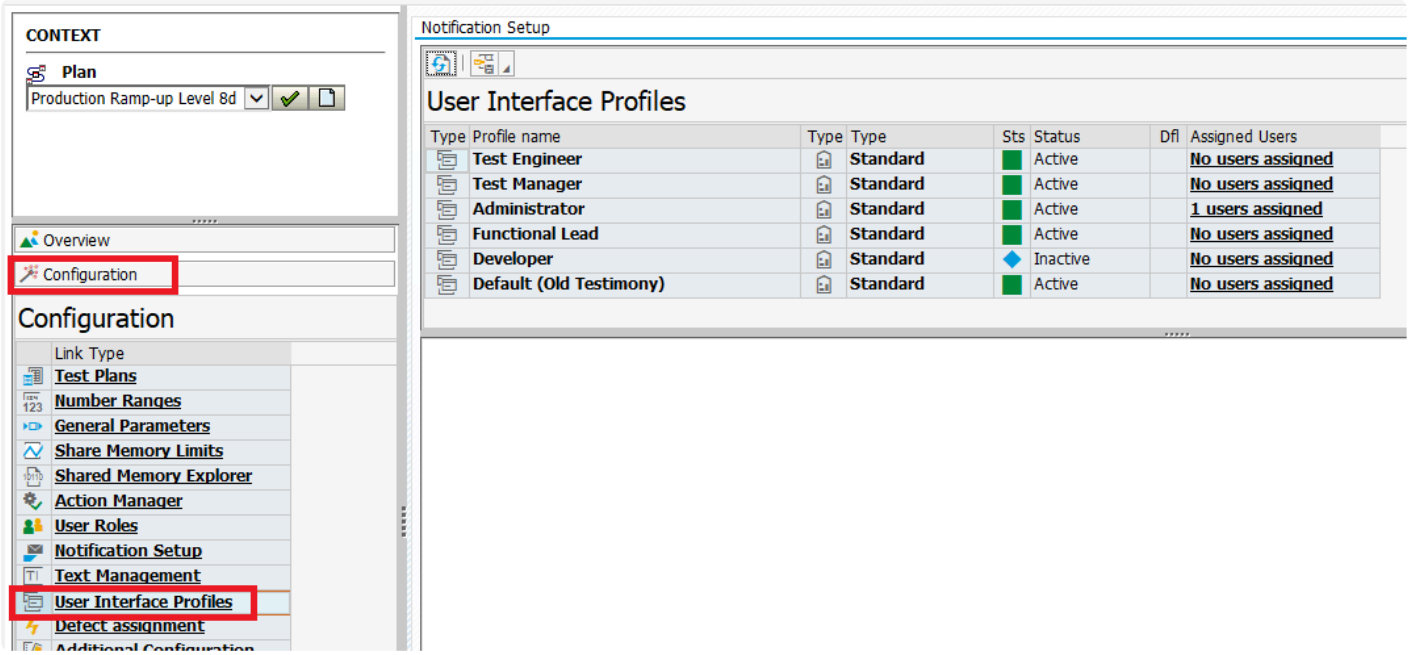
3.1.5. UI Profiles

The UI Profile assigned to a user determines what menu items and functions that users will see within the Testimony tool. The Testimony Admin by default will see the Administrator profile but can choose which UI Profile to view. All other users are either assigned one by default or manually by an administrator.

A Testimony Admin will see the UI Profile menu item and can change their assigne UI Profile at any time.

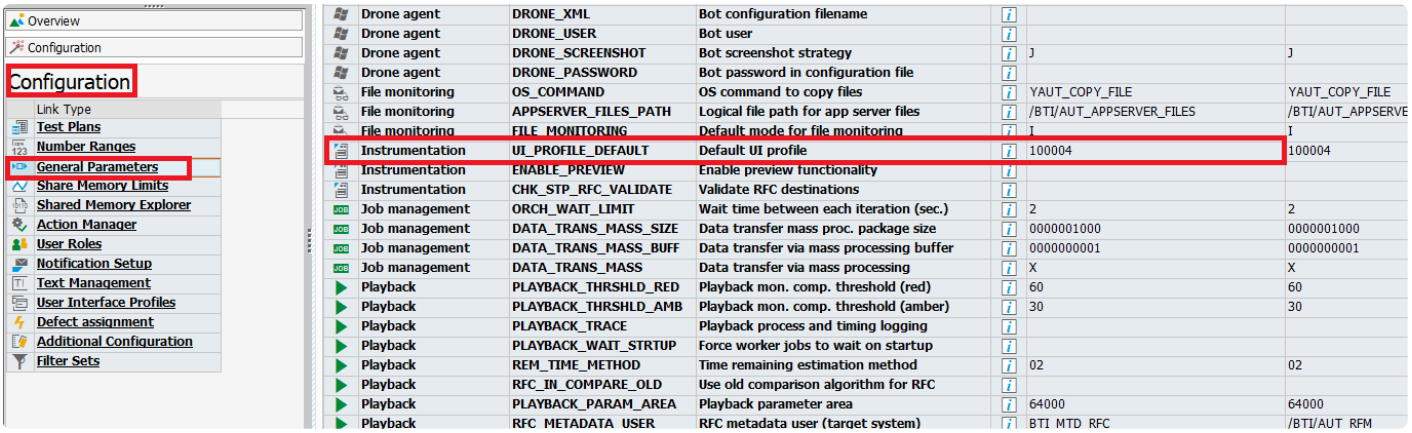


The UI Profiles can be found in the Configuration menu under User Interface Profiles



From here the Administrator can assign users to the appropriate UI Profile by clicking on the “Assigned Users” field.

The UI Profile that is assigned to new users by default can be set or changed in **Configuration>General Parameters**, and the numeric value updated to the UI Profile ID desired, which can be found in the UI Profile details.



3.1.6. Securing Recordings and Results

It is also possible to secure access to potentailly sensitive information by enforcing an authority check for the Testimony user attemtping to view test results. This can be achieved in the General parameters section for the object “Authority check for data display”. If this is enabled, or set to “X”, then Testimony will perform an authority check for the current user to validate authorization in the Source or Target system, depending on whether they’re looking at recordings or results.

The items affected by this authorization check are:

- Recording steps (business transactions)
- Recording inputs
- Recording outputs
- Playback steps
- Playback inputs
- Playback expected outputs
- Playback actual outputs
- Playback screen-shots

If the flag is set, an authority check against either the Source or Target system against the current user record. If the user has the appropriate access in the associated system then access is granted to perform the action. Otherwise access to the information is denied. This configuration setting is global and applies to all Test Plans if enabled.

3.2. Test Plan and System Connections

Overview

The Test Plan in Testimony is the “container” for all testing activities that take place within Testimony. Each Test Plan can be configured to represent a regression test cycle such as a Monthly Release, Project or SAP Upgrade undergoing testing. You can have as many Test Plans setup in the system as required. Typically you will create a new test plan for each recording, although you can repeat multiple playbacks within the same test plan.

Within each Test Plan, you define:

- A plan name and description, as well as some attributes about the scenario (e.g., Upgrade Testing) and whether or not the test plan is currently active.
- The systems that are participating in the testing phase. Each test plan must have at least one source system (in which the recording takes place) and one target system (into which the recording is played back). Multiple recording and playback “pairs” are also possible.
- The mapping of systems, which tells Testimony that a recording executed on System A will be played back in System B.
- The users who will be involved in the testing, and their roles.

Existing test plan definitions can be copied, bringing across all of the information (including systems, system mappings and user roles) into the new test plan. This makes the creation of multiple test plans for the same system pairings much easier.

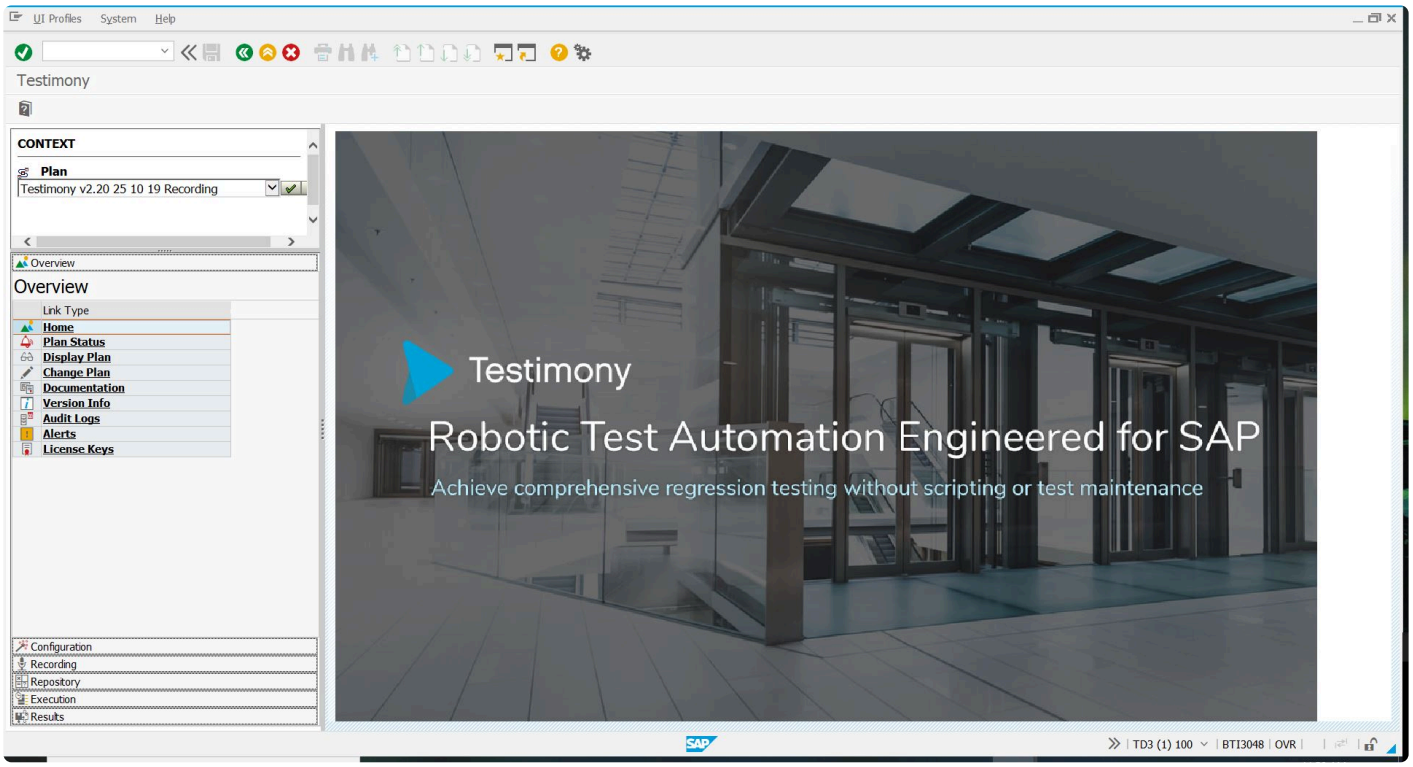
Process Steps

- Creating a Test Plan
- System Setup
- Authorisations
- Managing Test Plans

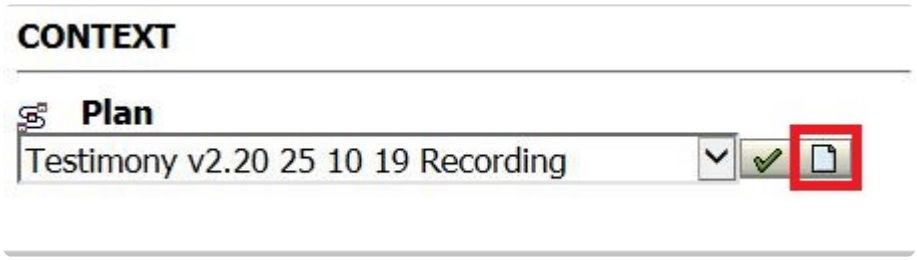
3.2.1. Creating a Test Plan

Access Testimony

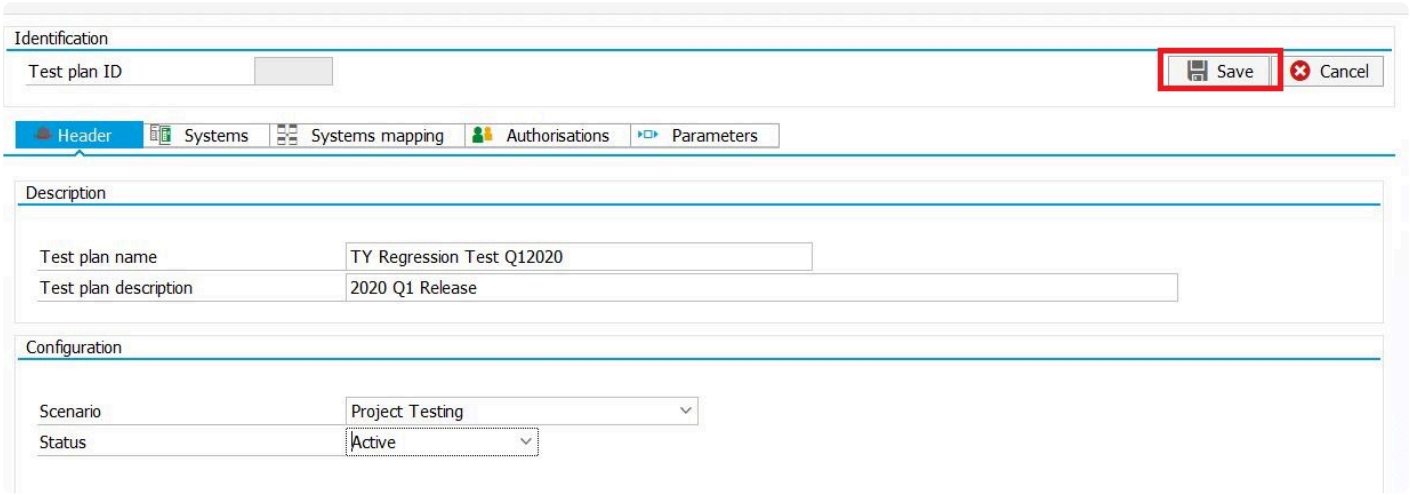
Once logged into the Testimony central system, enter transaction **/n/bti/aut**. This will display the Testimony main menu.



To create a new blank plan, select the create button



You will be presented with a screen where you enter a test plan name and a description, specify the scenario and mark the plan as active. Once complete click Save

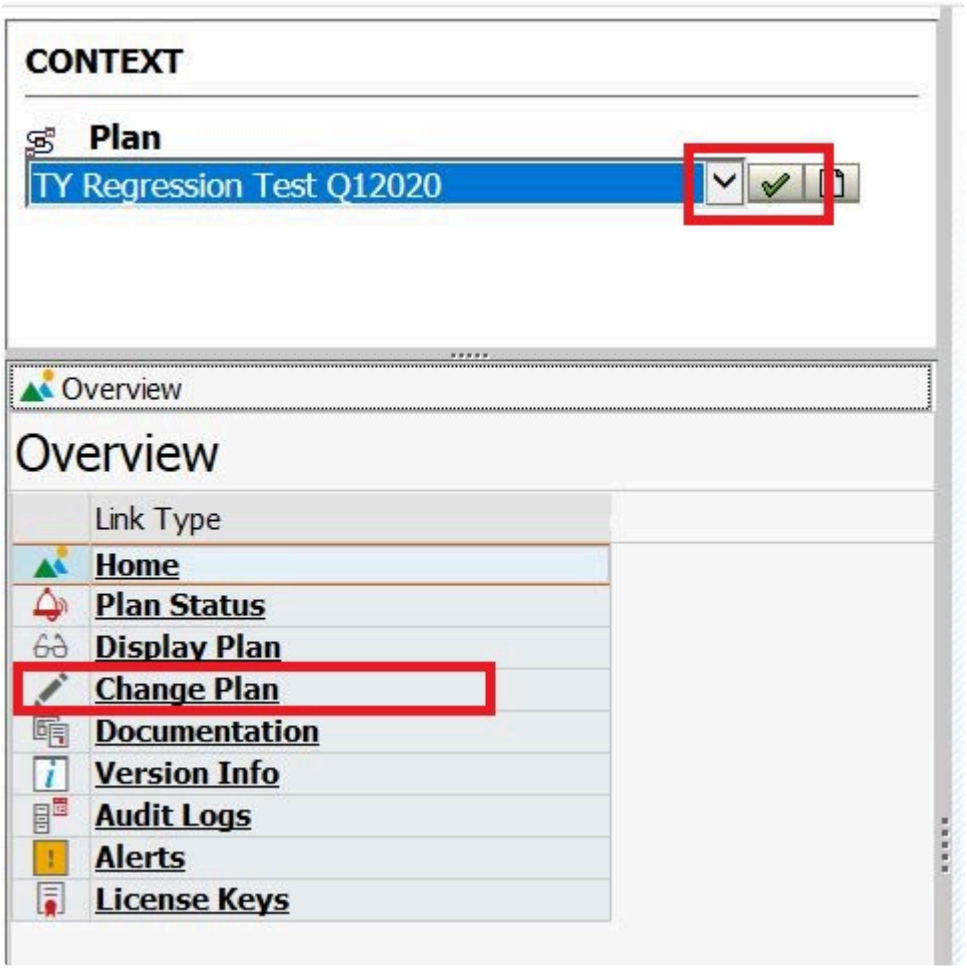


Once saved, you'll see the following status. From here, you will update the plan with the relevant system information

3.2.2. System Setup

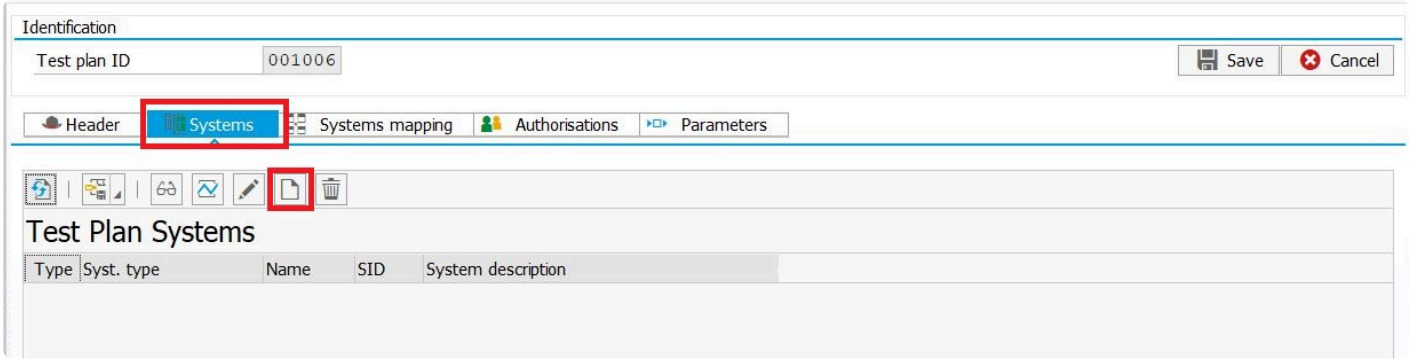
Select Test Plan to change

Use the drop down menu to select your newly created Test Plan and click the Tick. Now you can click Change Plan



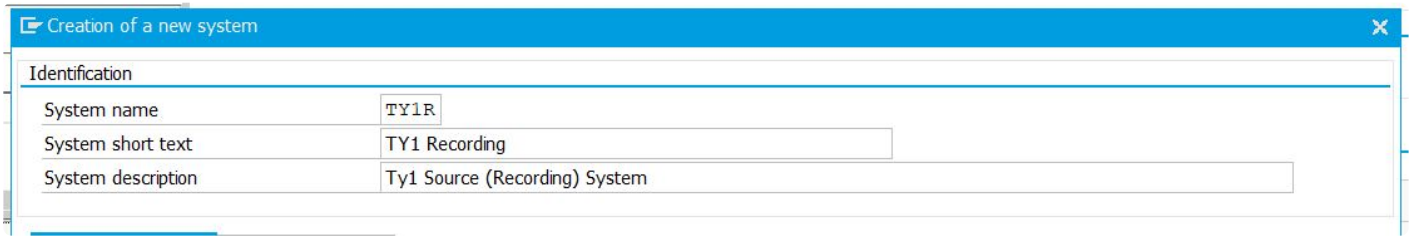
Systems

In the main window, you will see the Test Plan header information. Select the Systems tab and click the Create button. It’s good practice to start by creating your Source (Recording) system



In the Identification section, enter the required information as follows:

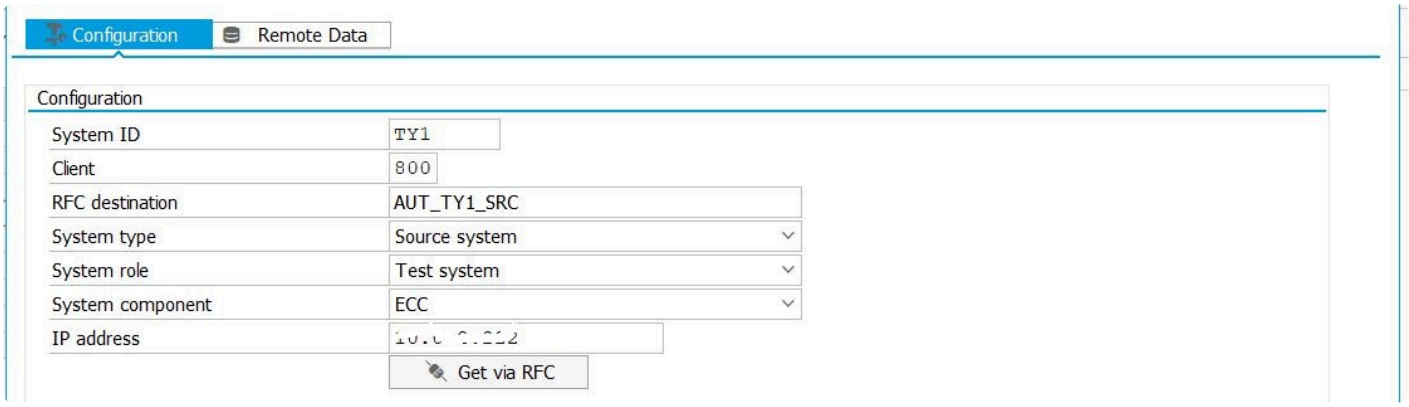
- System name: A 4-character internal identification for the system. Note that this is different than the standard 3-character SAP SID but it is suggested you use your SAP SID along with “R” for your Source (Recording) system and “P” for your Target (Playback) system. The 4-character identifier allows you to specify the same SAP system for both recordings and playback (for example, when performing initial smoke testing of Testimony).
- A short-text description for the system.
- A longer system description



In the Configuration section, enter the initial technical information to allow the Testimony central system to connect to this remote system.

For a source system (i.e., a system that will be recorded), only the following information is required:

- System ID: The SAP SID
- Client: The client of the remote system to which the central system should connect
- RFC destination: the RFC destination to the remote system. This should already have been set up as part of the initial installation and configuration phase.
- System type: whether the system is a source (recording) system or a target (playback) system.
- System role: whether the system is a production system or a test system
- System component: the SAP system component (ECC, BW, etc.)
- IP Address: you can get the IP address of the system by clicking on the “Get via RFC” button. This will connect to the remote system using the RFC destination that you specified during the technical setup.

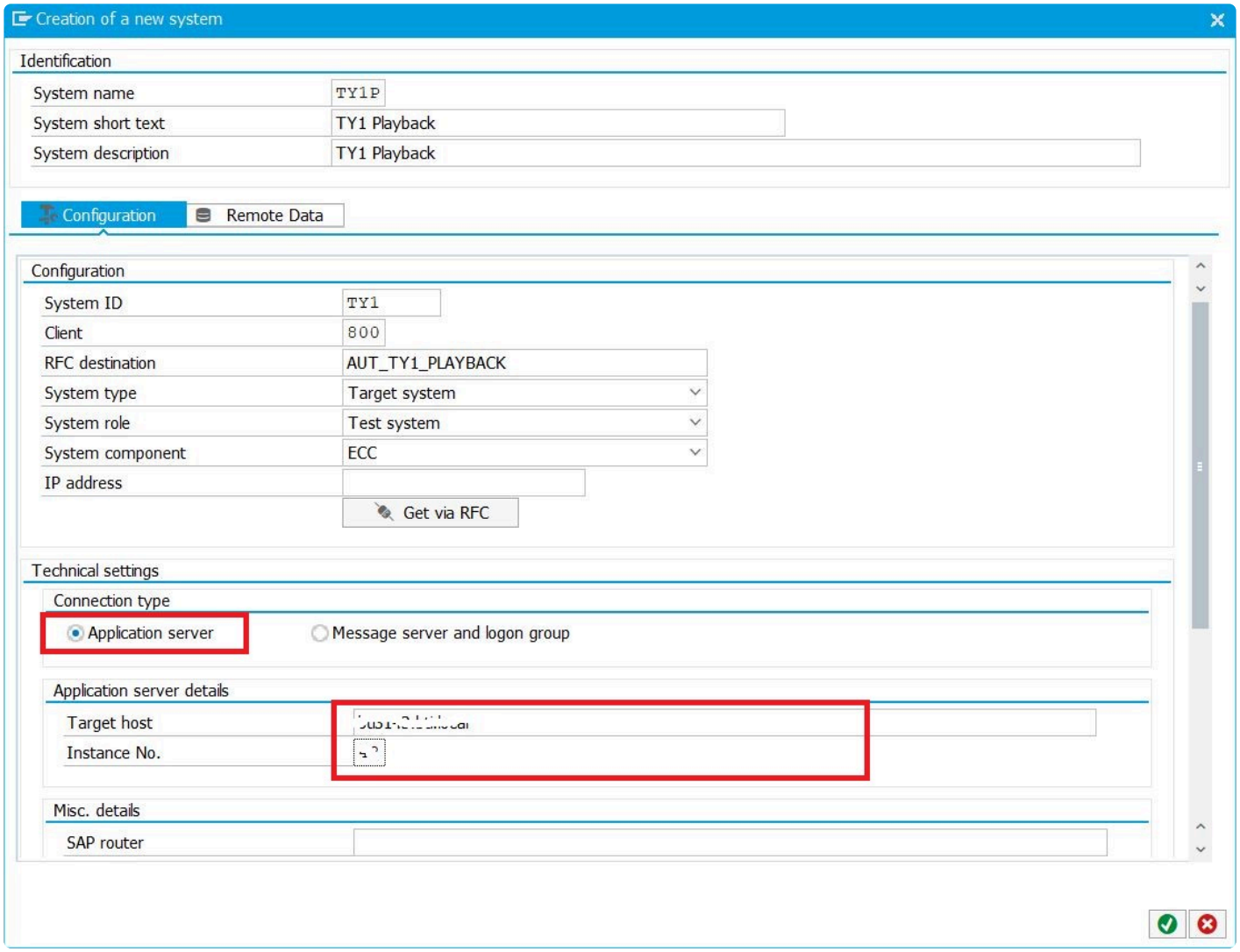


For a target system (i.e., a system into which your recordings will be played back) the following additional information is needed so that the playback bots know how to connect:

- Connection type: Whether the bots will connect directly to an application server or will use logon load balancing

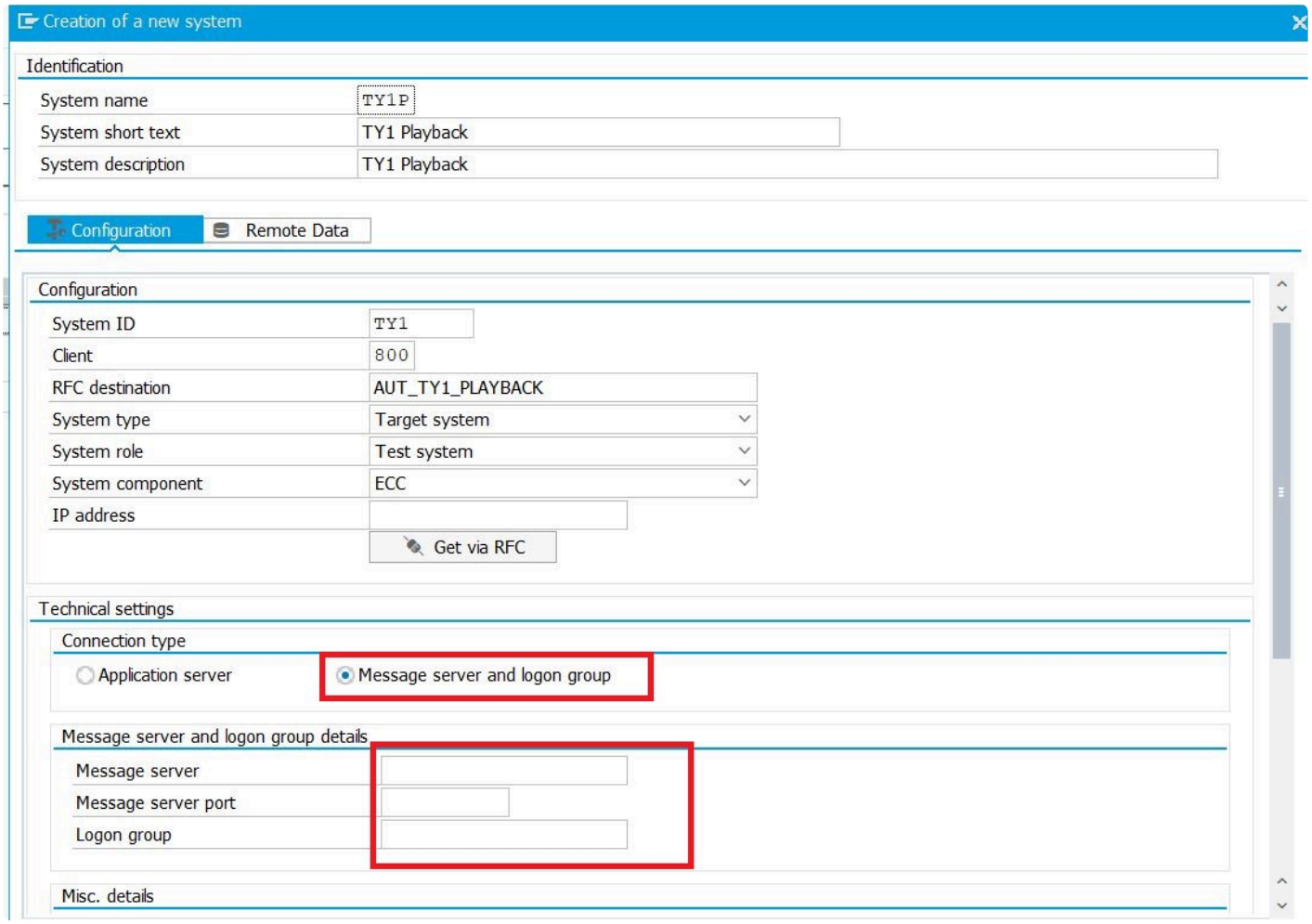
If the connection type is “Application server”:

- Target host: The host name of the application server
- Instance No.: The SAP instance number



If the connection type is “Message server and logon group”:

- Message server: The host name of the SAP message server
- Message server port: The port number (usually 36nn) that the message server listens on
- Logon group: The logon group (from SMLG) that the bots should use to connect to the system



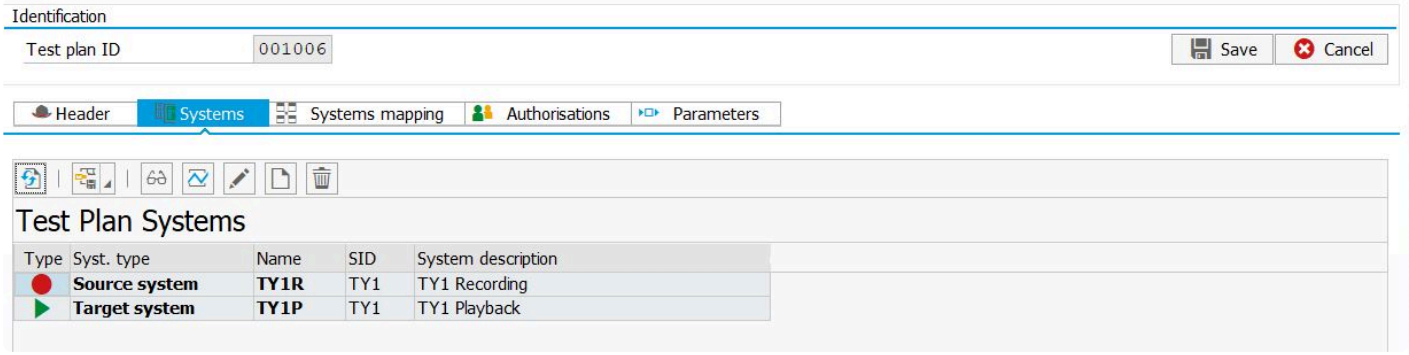
In the Misc. details section enter any (optional) configuration:

- SAP router: If you are using a SAP router to secure access to the playback system, enter the SAP router string here
- Gateway host: Enter the SAP gateway host
- Gateway service: Enter the SAP gateway service id

Once complete click the Tick to save your configuration. You'll receive the following status message. At this point you should also save your plan

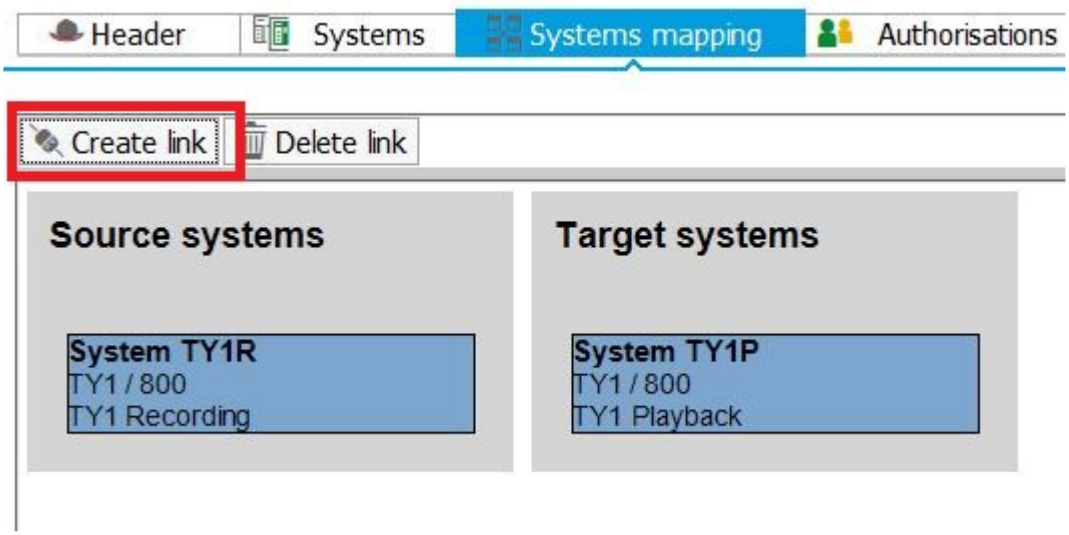
Repeat the same steps to configure additional systems. You will minimally need a Source and Target system defined.

You should now have at least 2 sytems defined in your plan

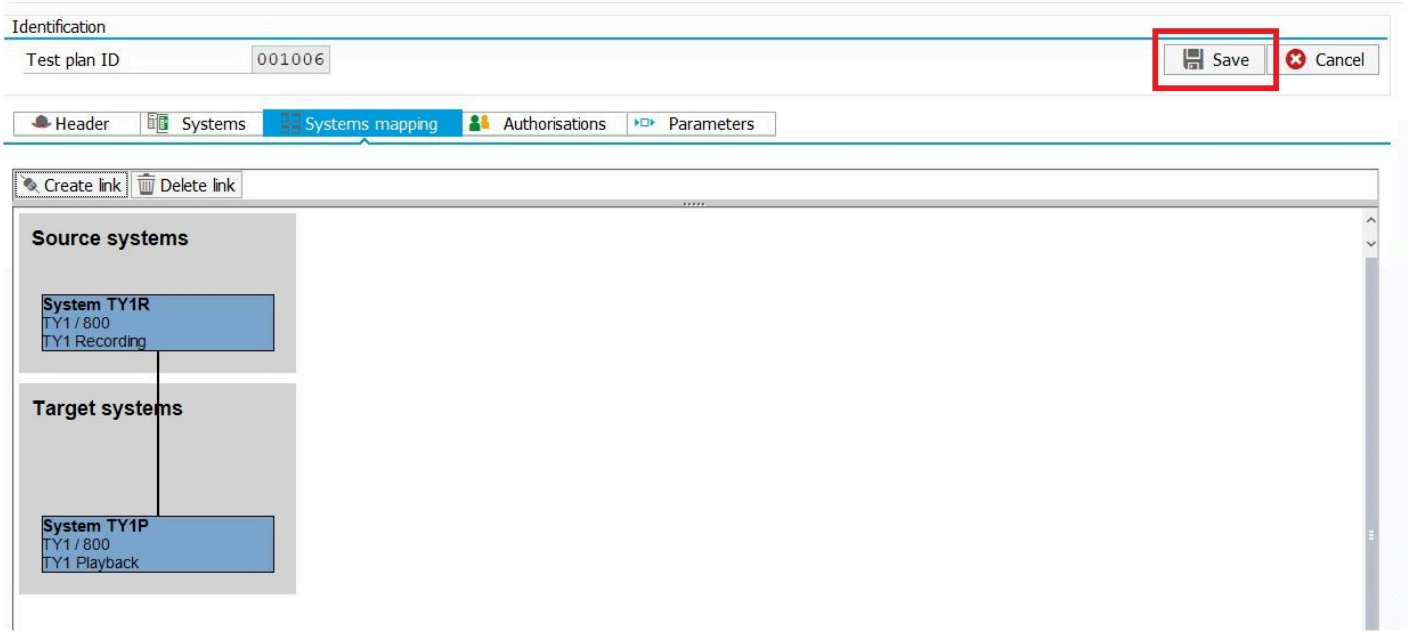


Systems Mapping

It is now necessary to map the systems so that Testimony knows which recording system equates to which playback system. This is a required step, even if you only have two systems in the test plan. In the test plan screen, go to the Systems mapping tab. Here you will see the systems you have defined. Click on the Create Link button



At this point, your cursor will change to a Pencil. Draw a line between the two systems you want to link and you get the following screen where you can click the Save button to save your System Mapping.



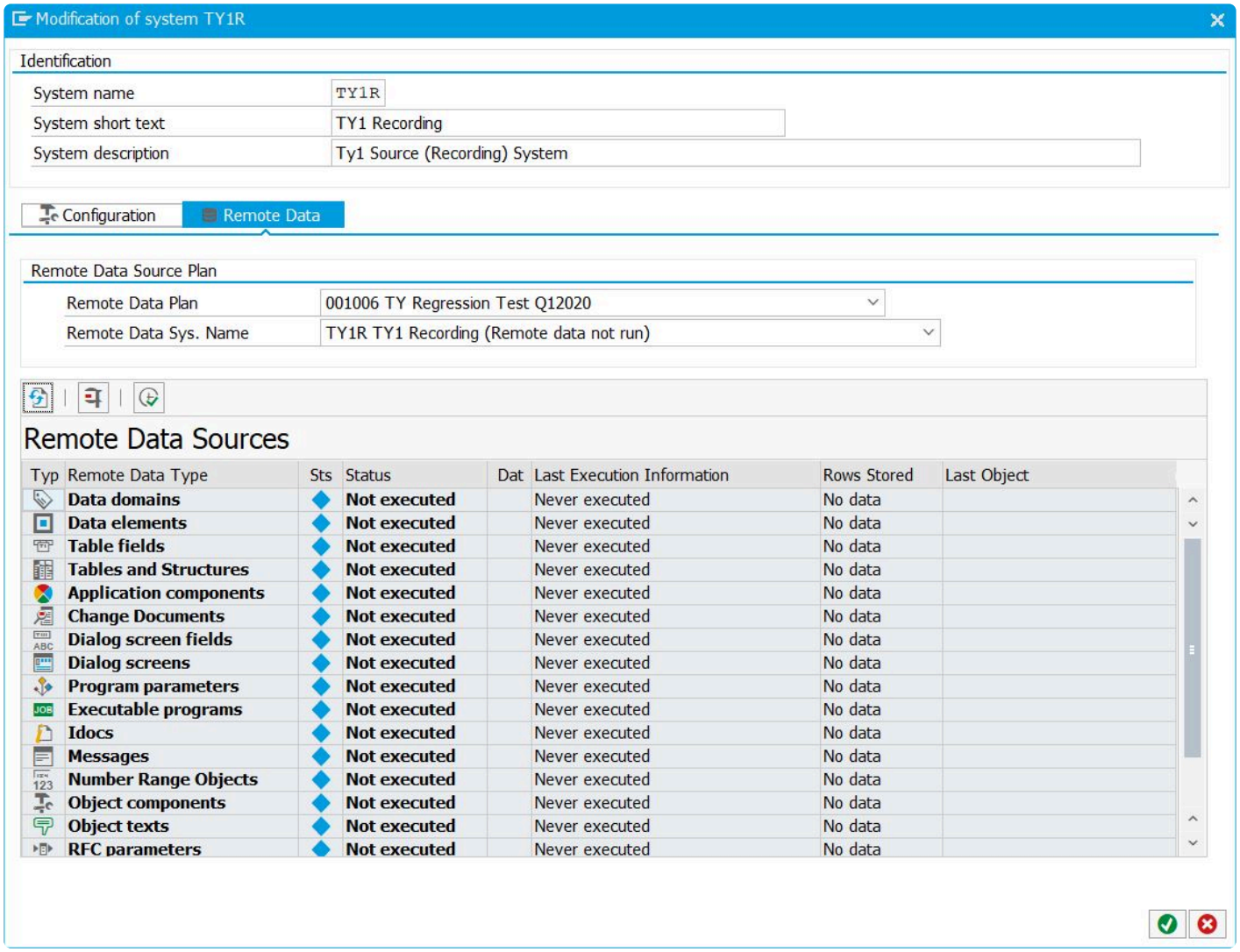
Remote data

The retrieval of remote data from the source system enables Testimony to perform a number of important functions. Firstly, it allows text descriptions of objects (e.g., transaction codes, screens, etc.) to be displayed, making the job of interpreting the output from Testimony much easier. It also allows Testimony to gather certain technical information (e.g., screen field definitions) which are an important part of Testimony's "learning" process.

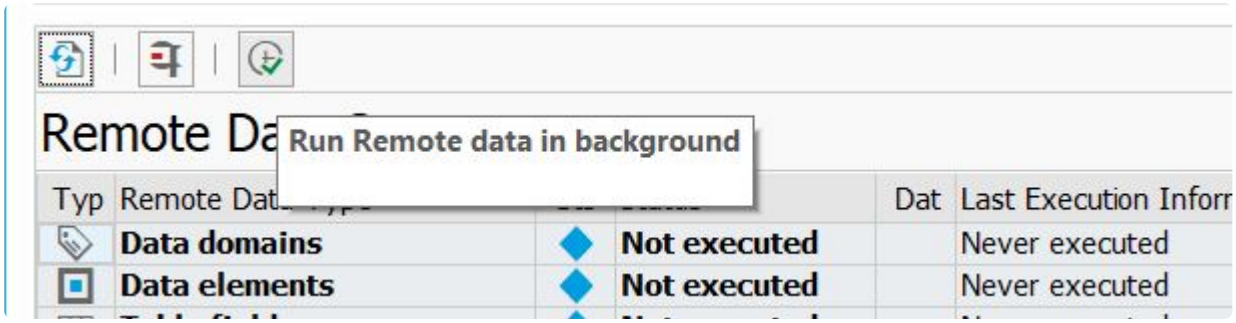
Remote data extraction only needs to be run in one system for each SAP component type (e.g., once in ECC, once in SRM, etc.). Other systems of the same component type can share this information. Basis Technologies recommends that this data be extracted from a test system to avoid putting any additional overhead on production.

The remote data extraction needs to be run when a new system landscape is first added to Testimony, and then when there is a major upgrade or release. For example, if you are applying support packages then a further extraction will be required so that Testimony has the most up-to-date definitions of the objects that it uses. We also recommend that, in any case, the extraction is run at least every 6 months so that Testimony can be kept up-to-date.

To extract the remote data, within the Change Plan screens, navigate to your source system and go to the Remote Data tab: You will see all of the remote data types that Testimony needs to extract, and also that the remote data extraction has not yet been executed.

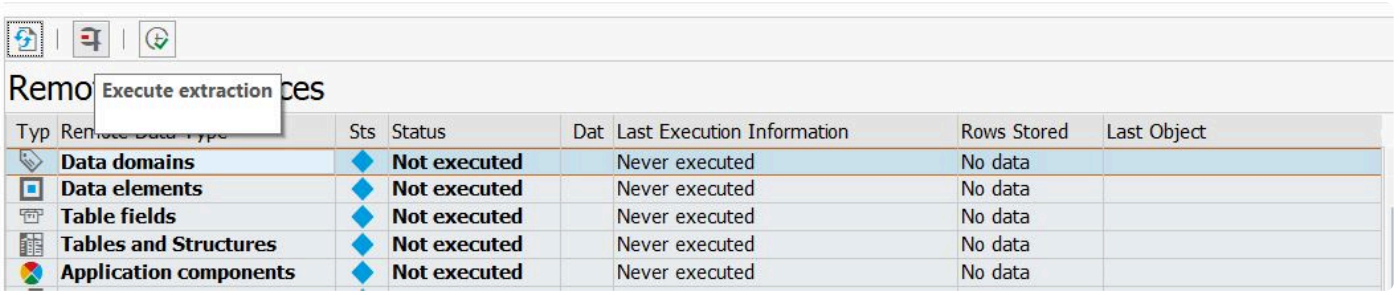


You can start the remote data extraction by clicking on the “Run remote data in background” button.



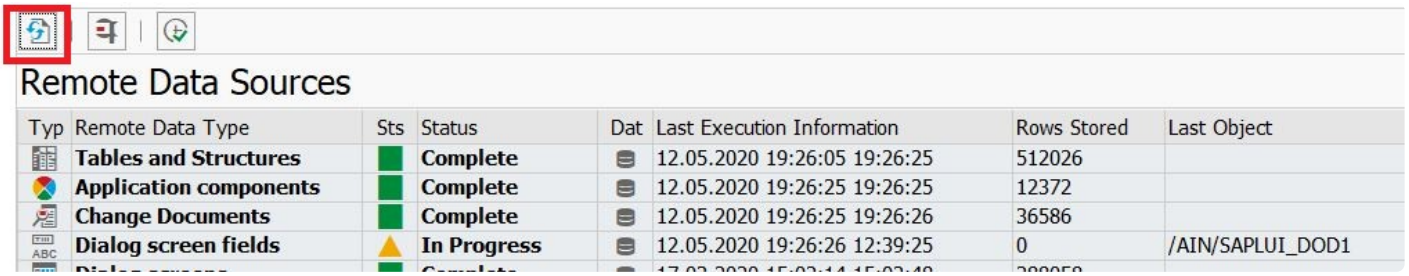
This will start a background job in the Central System which submits requests via RFC to the source system to gather the required data.

Alternatively, you can extract the data for individual components by selecting the component and clicking on the “Execute extraction” button.



Note that some of the data extraction steps (in particular, the extractions of Dialog screen fields, Table fields and Program parameters) can take a very long time.

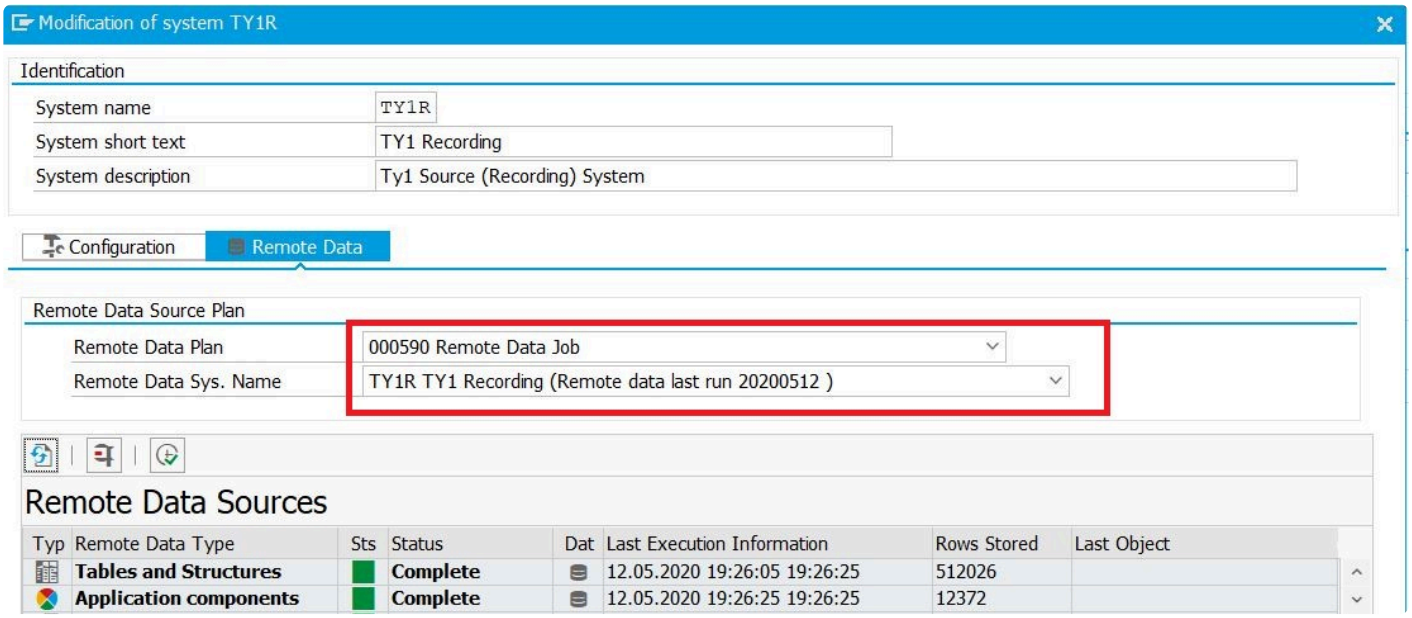
If submitting the extraction in the background, you can check on the status of the extraction process from within this screen by using the Refresh button.



Note that extracting all of the remote data from a system can take 18-24 hours.

If you have already extracted the remote data for one system, then you can re-use this data, avoiding the need to re-run this process. To do this, go into Change Plan and change your source system, then navigate to the Remote Data tab.

You will see that the Remote Data Plan and Remote Data Sys. Name fields are editable via drop-downs. Select the plan whose remote data you wish to use, and the system field will be automatically updated, as will the extraction statuses of the remote data objects.



Click on the tick to confirm your selection, and then save the plan.

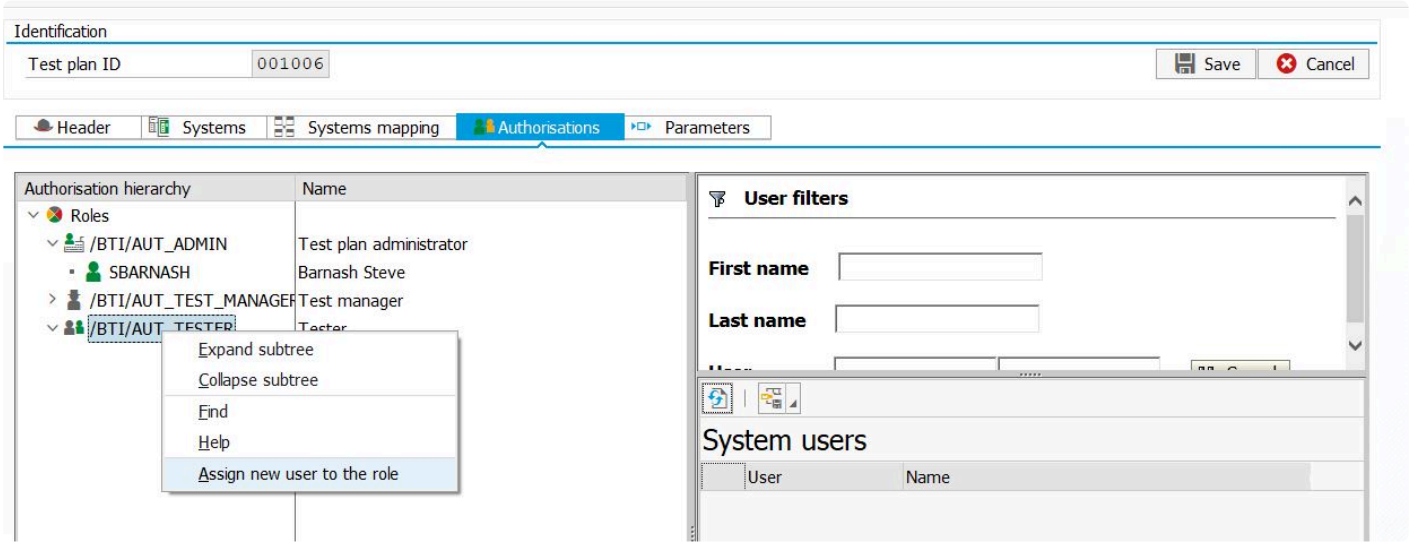
3.2.3. Authorisations

Testimony comes with three internal roles:

- /BTI/AUT_ADMIN: The test plan administrator
- /BTI/AUT_TEST_MANAGER: The test manager
- /BTI/AUT_TESTER: A tester role

Each test plan must have at least one administrator and by default the user who creates a test plan is automatically assigned to the test plan administrator role.

As an optional step, you can add users into your test plan that belong to one or more **“User Roles”**. A role represents a set of actions that users in that role are allowed to perform (e.g. activate a recording or view scripts). In this screen, you can control which users belong to which roles for the current test plan. To assign a user a role on a plan, right mouse click on the role and then search for the user and select them. Once you save the plan, the user is assigned to the role.

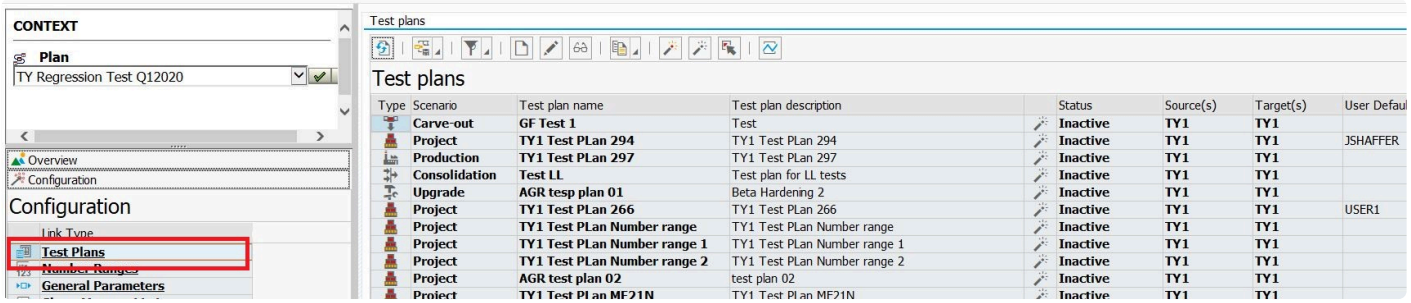


3.2.4. Managing Test Plans

As well as maintaining Test Plans from the Overview -> Change Plan screen (as previously shown) it is also possible to maintain them without first having to select the individual plan.

Navigate to Test Plans item from the Configuration section. You will see a list of all Test Plans defined in the system. From this screen you can perform the following actions to your Test Plans:

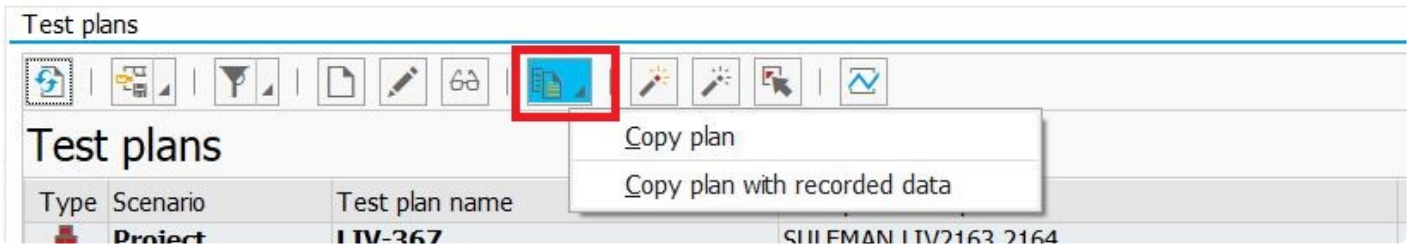
- Create Plan
- Display Plan
- Change Plan
- Copy Plan
- Activate Plan
- Deactive Plan
- Analyse Plan



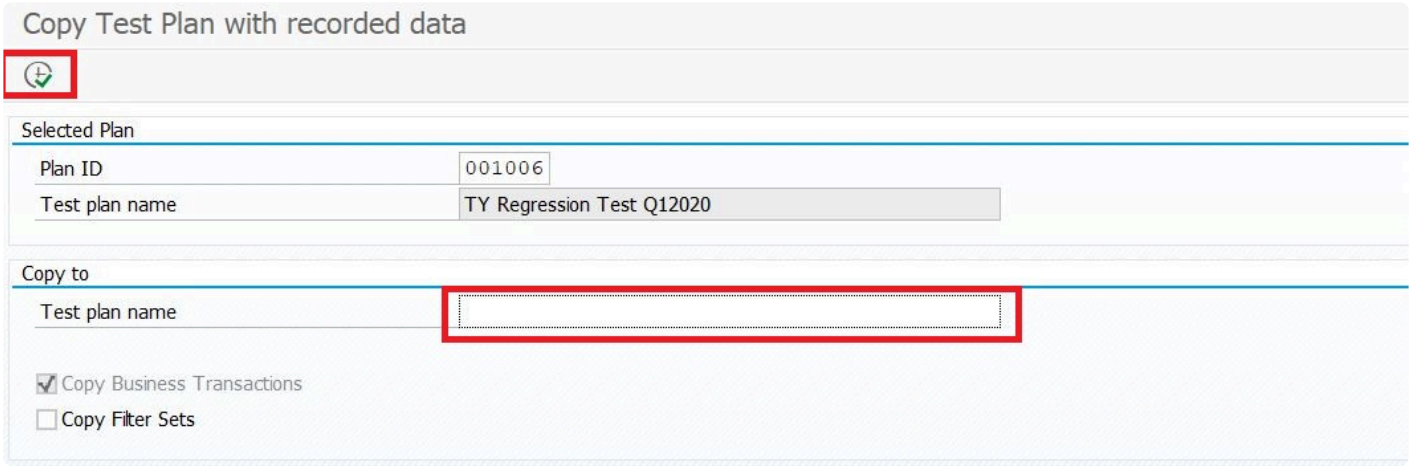
Copy a Test Plan

If you have a test plan that you wish to duplicate, then the **Copy Test Plan** function allows you to do this, copying across not only the test plan header information but also the system assignments and mappings and the user assignments. This is a great time saver to creating new Test Plans

Simply select the Test Plan you want to copy and click the Copy Test Plan button. You have the choice to create the copy with a recorded data or not.

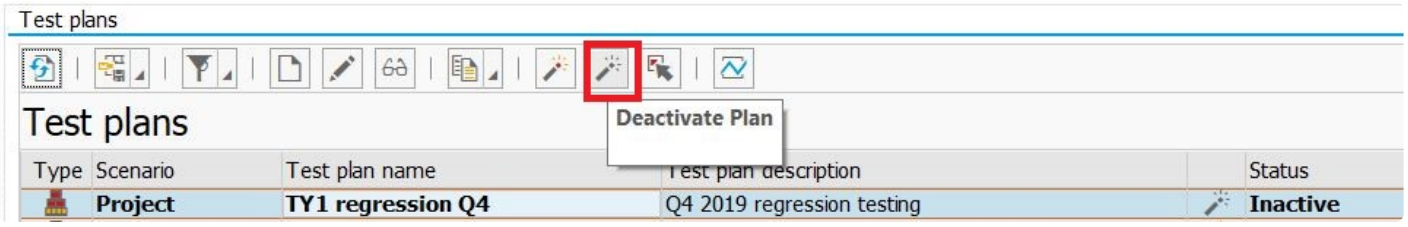


If you selected to copy with Recorded Data, you'll then need to give the new copied project a name and optionally select copying Filter Sets. Once these fields are complete, click the Execute button.



Deactivate Test Plan

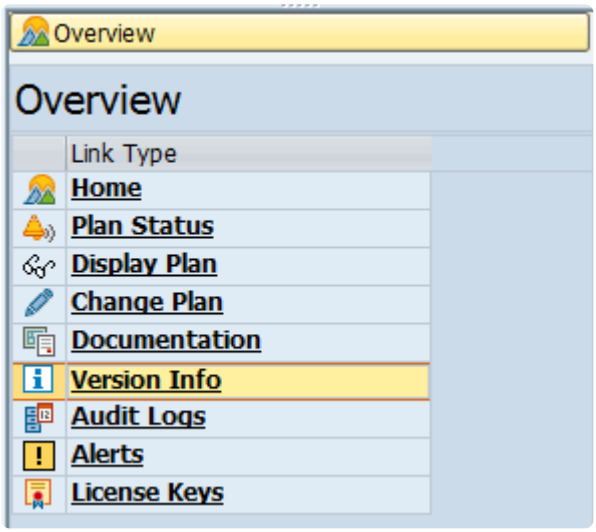
By default, new test plans are created as active plans, meaning that they are available for selection in the drop-down list in the plan context menu. If you wish to remove a plan from this list (for example, if it is an old plan that should no longer be used) then you can deactivate from the configuration screen. Simply select the plan and click on the Deactivate Plan button and the plan's status will change to inactive.



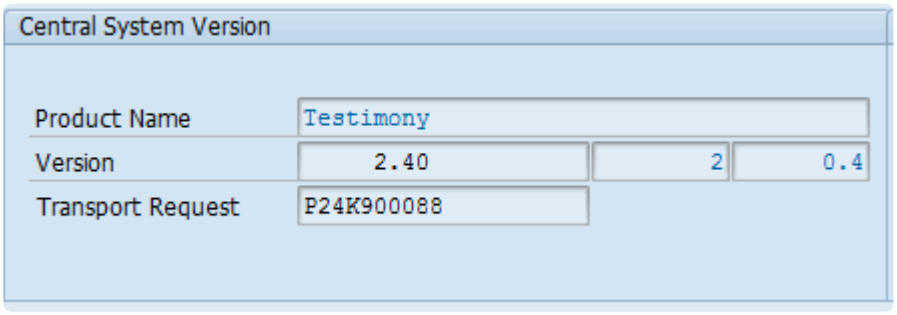
Likewise, you can reactivate an Inactive Test Plan in order to make it visible in the drop-down list by selecting the Test Plan and clicking on the Active Test Plan button

3.3. Version check

To check your current Testimony version on the Central System, go to the Testimony transaction and choose Overview —> Version Info:



This will show the current version, as well as the base version transport that was applied on the Central System:



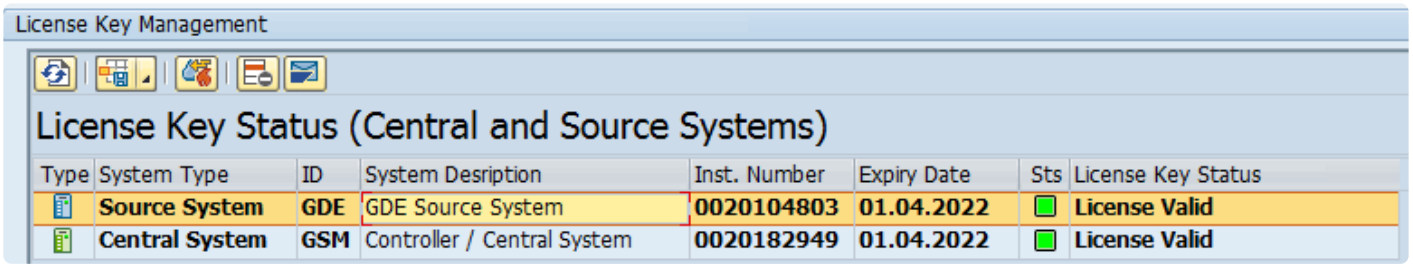
To check which Cumulative Patch level your Central System is on, go to the import overview for the Central System in STSM and display all Testimony transports. (These begin with P24K*.)

Number	Request	RC	Owner	Short Text
5	P24K900088	⚠	COLIVER	Basis Technologies: Testimony (Version: 2.40)
6	P24K900110	⚠	GFARKAS	Basis Technologies: Testimony 2.40.5 Patch
7	P24K900122	⚠	GFARKAS	Basis Technologies: Testimony 2.40.6 Patch
8	P24K900126	⚠	GFARKAS	Basis Technologies: Testimony 2.40.9 Patch
9	P24K900137	⚠	GFARKAS	Basis Technologies: Testimony 2.40.15 Patch

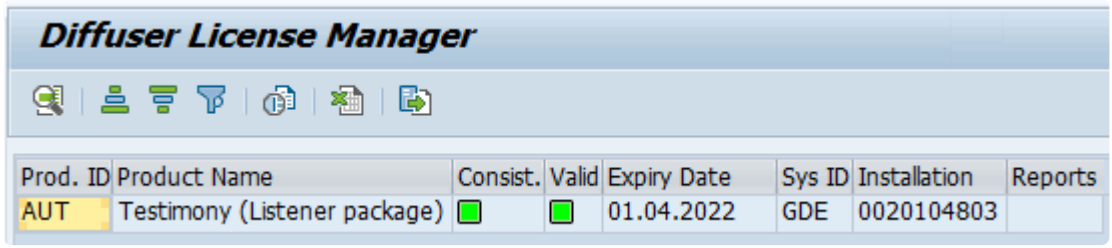
You can check the source and target system version and cumulative patch level by looking at the STMS import history, again looking for P24K* transports.

3.4. Licence key check

Testimony licences are installed using the Testimony transaction /BTI/AUT. You can go to Overview —> Licence Keys to view the status of your licences. Before starting a recording it is important that valid licences exist for *both* the Central and Source systems:



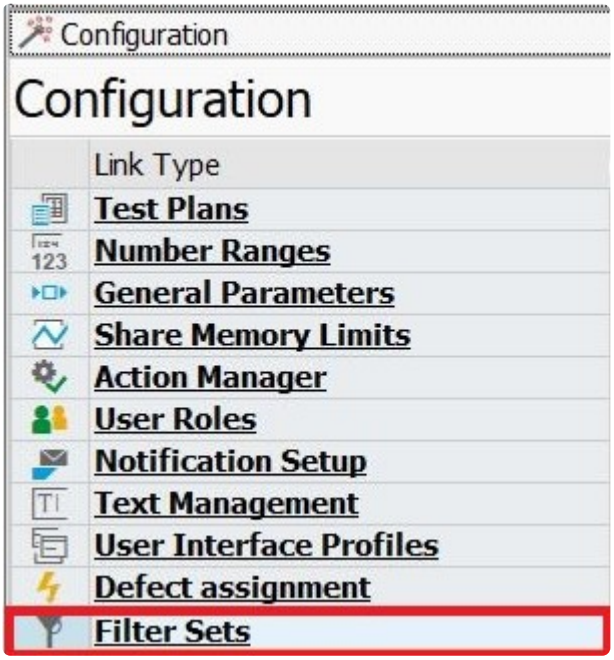
In addition to Testimony licences, Diffuser licences must also be installed on both the Central and Source systems. You can check the status of Diffuser licences by logging on to each system and calling transaction /BTR/LICENSE and using the “Check installed keys” function. Diffuser licence keys for Testimony are listed as “Testimony (Listener package)”:



3.5. Filters, Suppressions & Plan Configuration

Overview

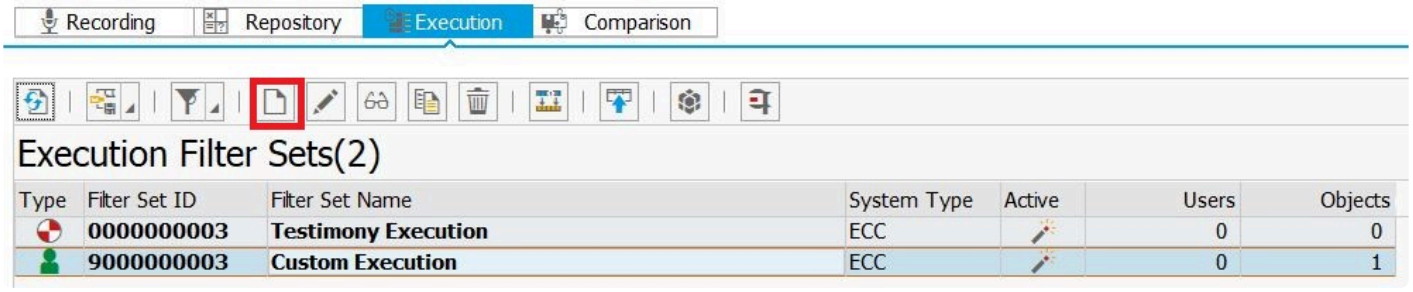
Filter sets are the way to configure Testimony to limit the selection of what is recorded and played back for a test and/or to suppress error messages for certain objects. There are 4 points (Process Steps) where this can occur: Recording, Transfer to repository, Transfer to Execution & Comparison (Playback). You can exclude and/or include RFC/Dialogue/Batch jobs at every point. Filter Sets can be set up for specific Test Plans or to used across all Test Plans. They are accessible from the Configuration menu as shown below.



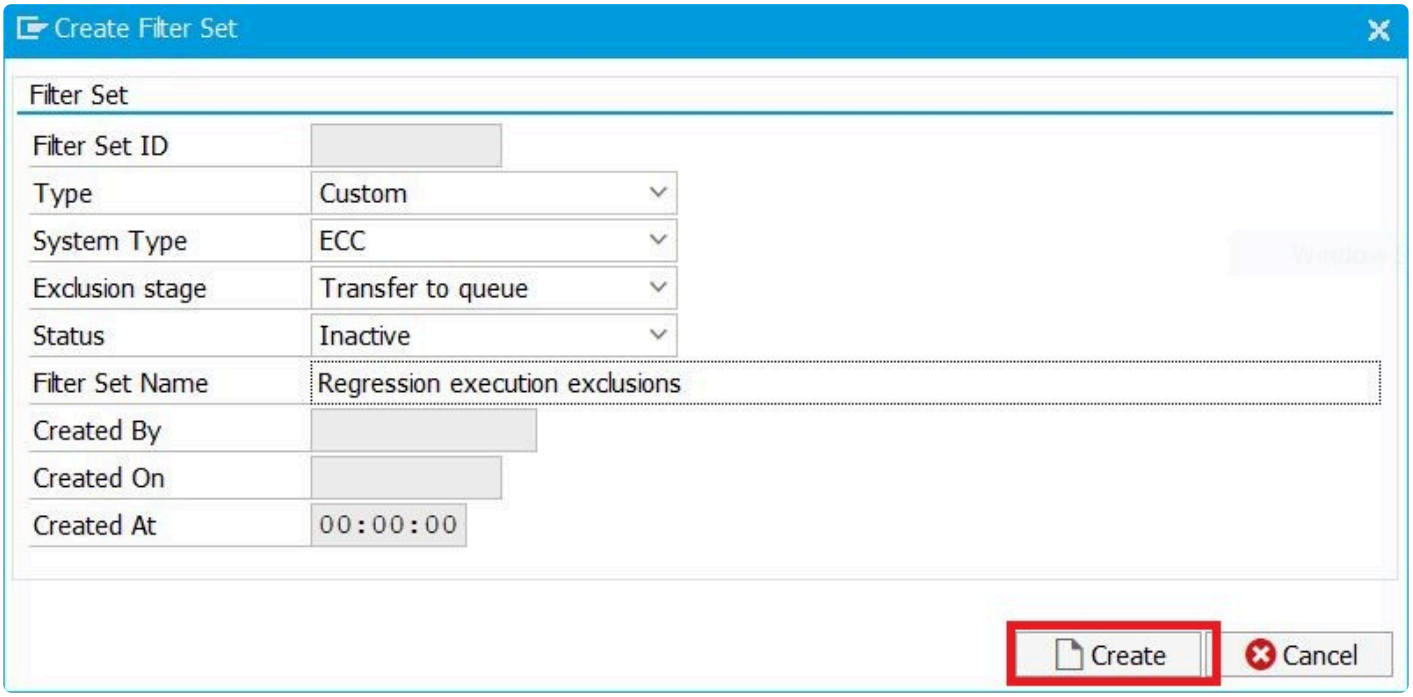
By default, there will always be a pre-configured (Default) Filter Set for each Process Step (Recording, Repository, Execution and Comparison). They are used to exclude activities that are running in your Source system, but are not required for Testimony to accurately play back in your Target System. During your installation and initial configuration, your Basis Technologies consultant may also add exclusions specific to your environment.

> NOTE: Since Filter Sets can be Active or Inactive, you can easily set up multiple custom Filter Sets that you Activate only when required for a particular test.

To create a new Filter Set, click on the Process Step tab (in this case, “Execution”) and then on the create button



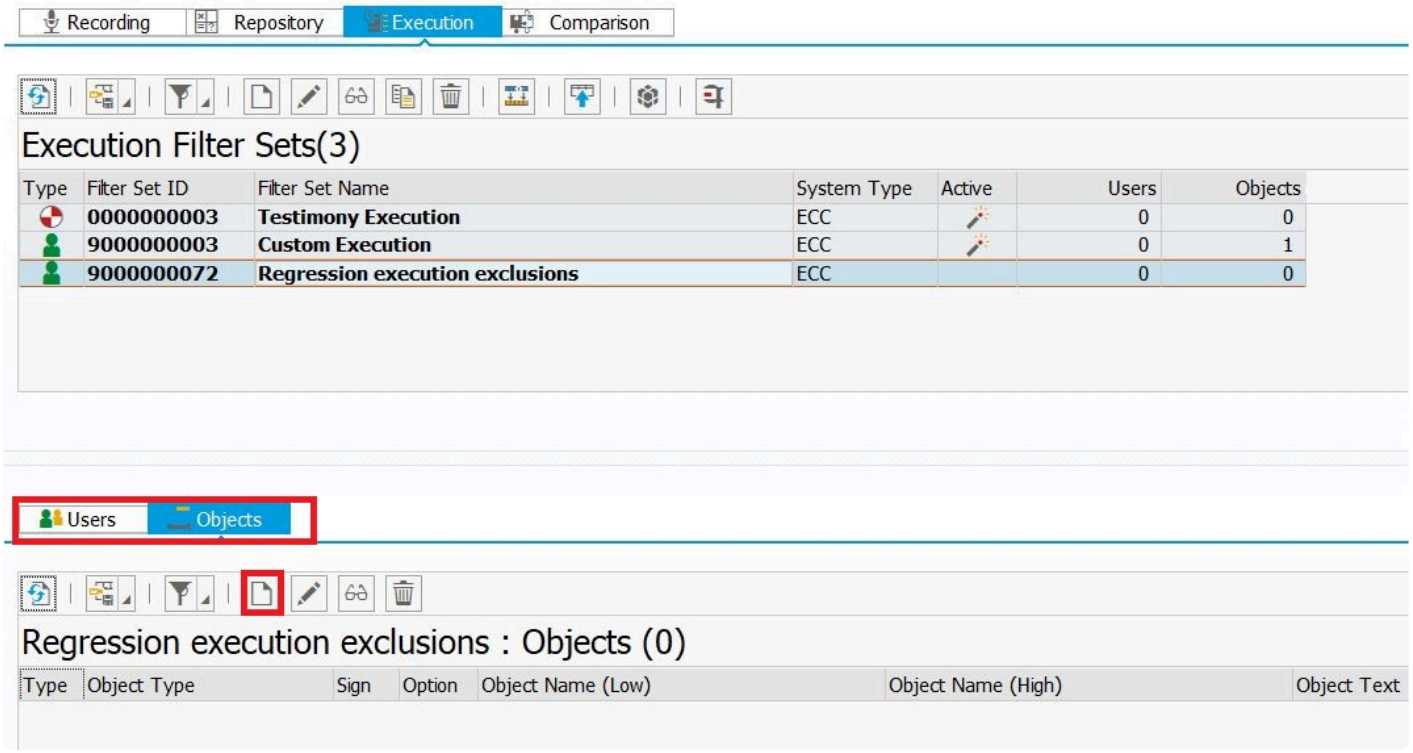
The follow window will appear. Fill in the required info. The “Type” field will determine if this filter set will be Custom (You’ll need to give it a name) or a plan specific Filter Set (you’ll need to give it the Test Plan number and the name will be generated). By default, your new Filter Set will be Inactive, so be sure to Activate it when you want to use it. The Exclusion Stage field will default to the tab you were in when you clicked the Create button. You can change it if your intent was for a different stage.



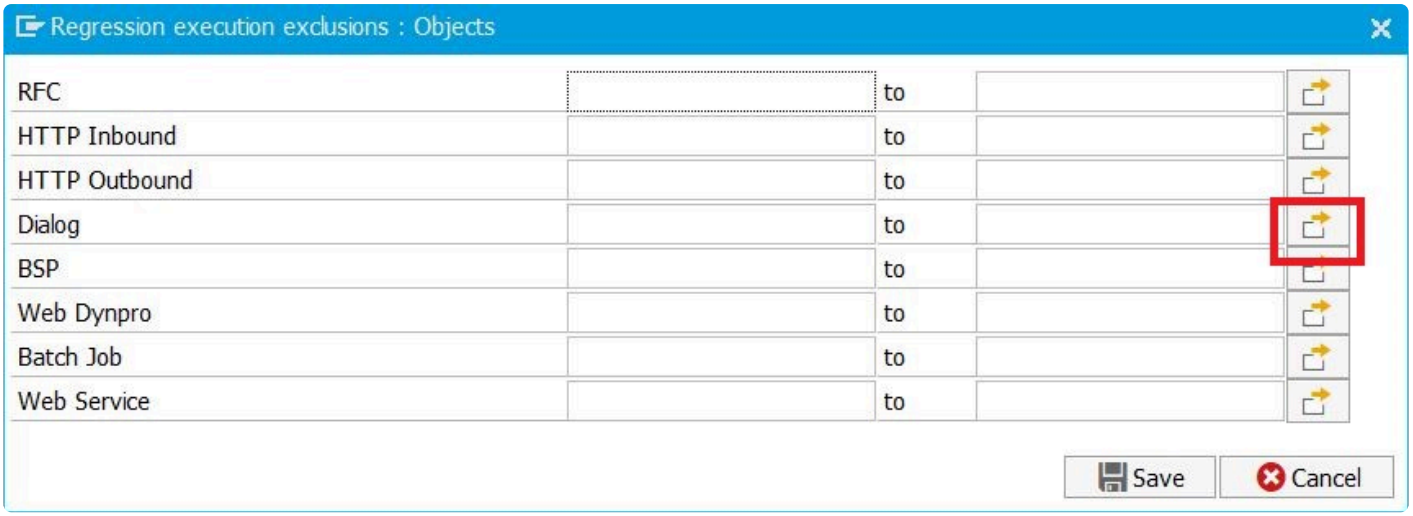
Once you click the create button, you’ll see the new Filter Set in the list where you can now add Exclusion and Suppression objects.

3.5.1. Exclusions

To exclude objects, double click on a Filter Set and select the tab for the exclusion type (in this case, Objects was selected). You'll then click the "Create Filter Values" button seen below where you get a new window to create the rules. **Note that you will not be able to update a default Filter Set identified with the icon**

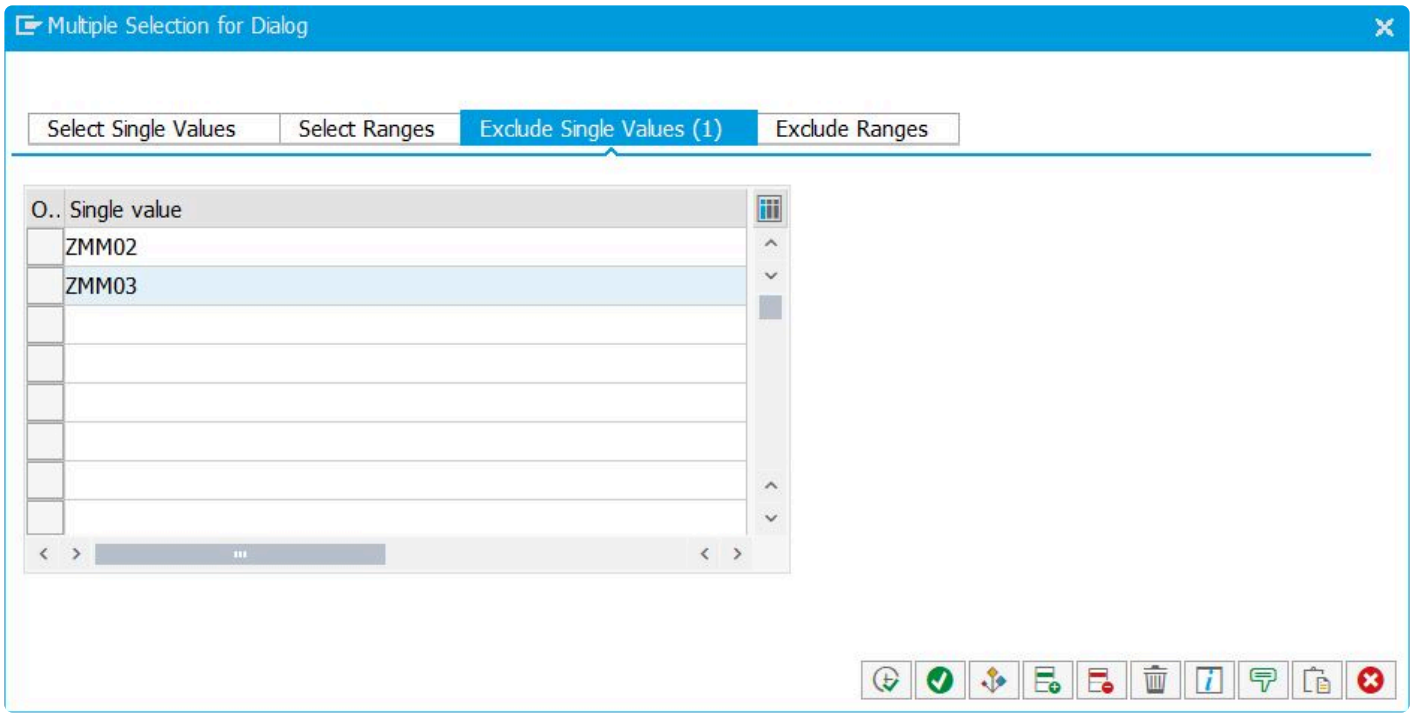


The following window is where you'll create a rule. You'll need to click on one of the selection icons (Dialog in the example below) which will take you to a Multiple Selection window.

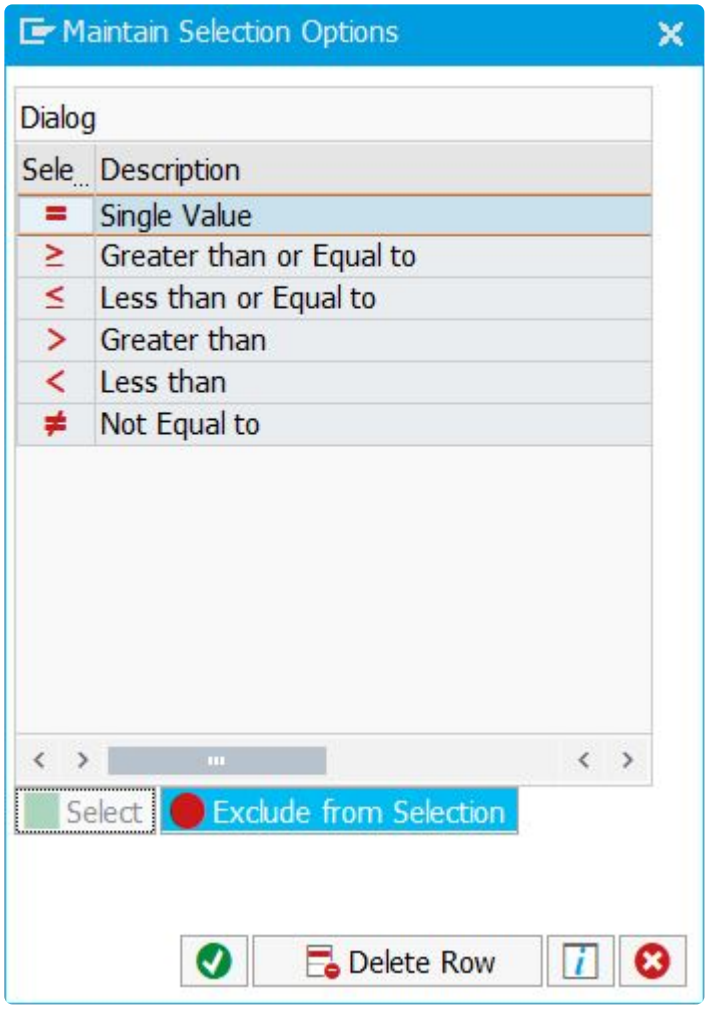


Once in the Multiple Selecction window, you can choose Single Values or Ranges to either select or

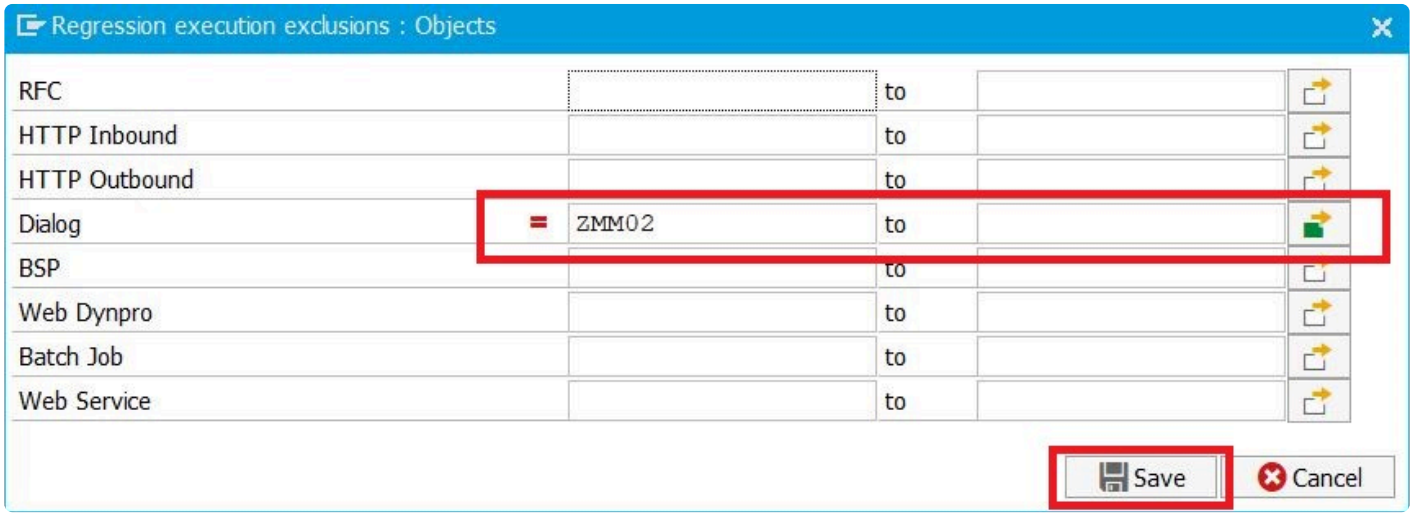
exclude. **It is far more common to Exclude so the examples below will reflect this scenario.** Clicking on the “Exclude Single Values” tab will allow you to enter 1 or more objects to exclude. In the example below, transactions “ZMM02” and “ZMM03” were added.



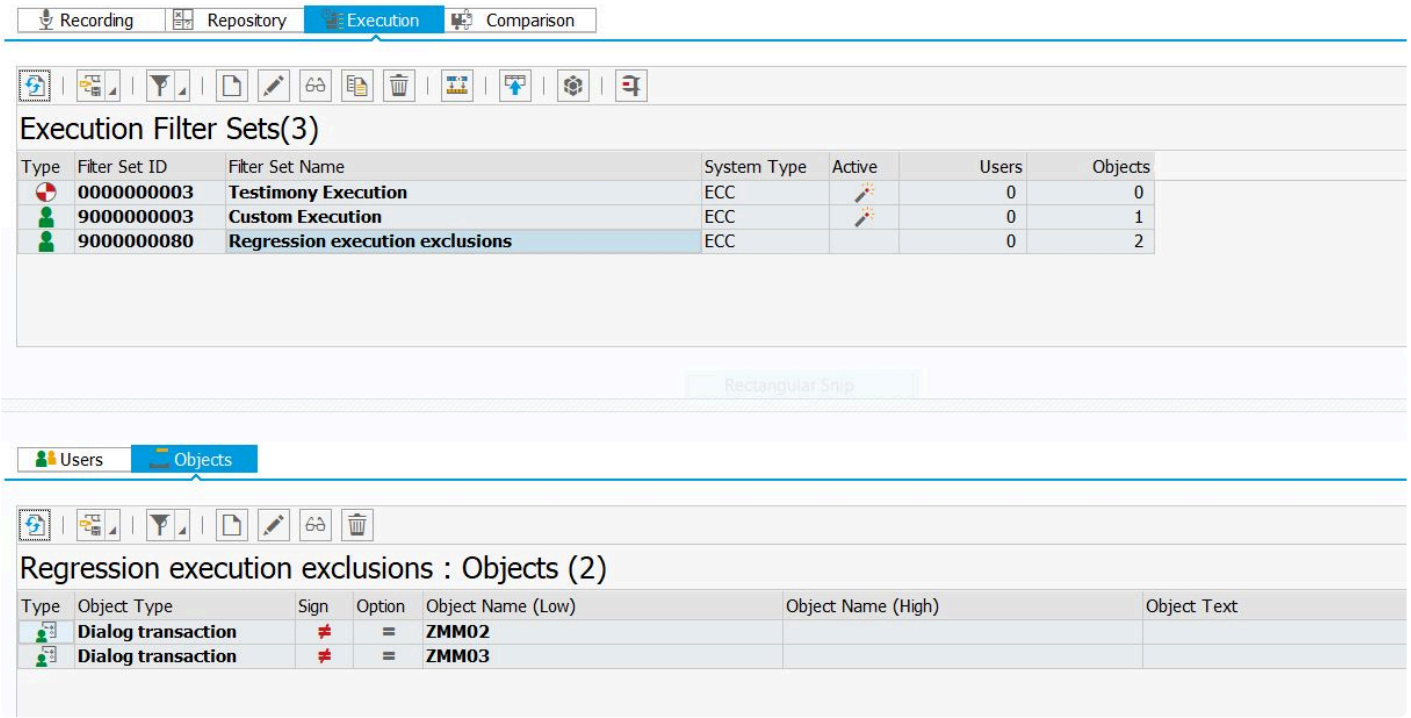
You can also double click on a row which will open the following window to change the boolean operator from the defaul “=” to another operator.



Once all selections are entered, click the Copy button  which will take you back to the prior screen where you will save your selections. Notice how the Icon changes to reflect multiple values.



You'll now see your excluded objects back on the Filter Sets screen. **Be sure to active the filter set once you intend to use it.**



NOTE: When setting up “Recording” exclusions for Batch Jobs, you will need to use the Program Name of the first step in the batch job. When setting up “Repository” exclusions for Batch Jobs you will need to used the Job Name.

3.5.2. Defect Suppression

As you use Testimony over time, you can “teach” it to suppress certain failures that are either expected or unavoidable, given the way that Testimony works (data-related defects for example) or the set-up of your environment (e.g., bot-related defects) By suppressing these failures, you can ensure that defects are not raised for them, meaning that the effort required for defect analysis will decrease and become more focussed over time.

There are two types of failure suppression in Testimony: step-level suppression, and script-level suppression. These are discussed below.

3.5.2.1. Step-level suppression

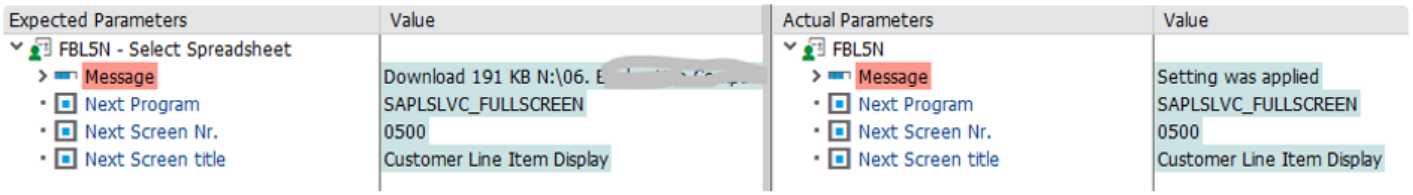
In some cases it may be possible to suppress a failure at the step level and carry on with the rest of the script. When you switch on step-level suppression for a particular failure, Testimony will mark the step as successful and attempt to continue with the rest of the script.

There are two steps involved in deciding whether or not a failure is suitable for step-level suppression. Firstly, of course, you need to have determined that this is a failure that should be ignored in subsequent Testimony playbacks. The examples given above should help you to determine this, based on some common types of defects.

The second step is to determine whether or not it would be possible for the script to continue, given the defect that has been raised.

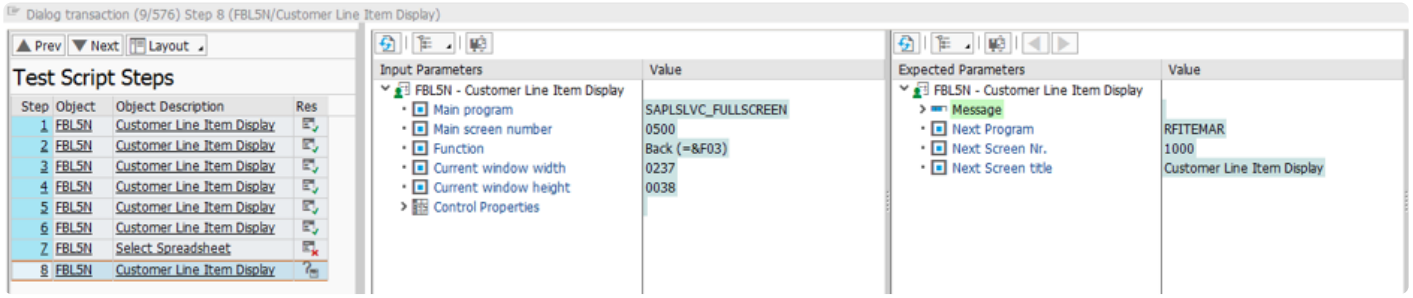
3.5.2.1.1. Example of a step-level suppression

If we take another look at one of the defects we saw earlier relating to a file download failure, we can see that we had a message in the recording (“Download 191 KB …”) and a different message (“Setting was applied”) in the playback.



Note here that the Next Program and Next Screen Nr. are identical in the expected and actual parameters. This is important, as if we are on a different screen in the playback then it will not be possible to continue with the rest of the script.

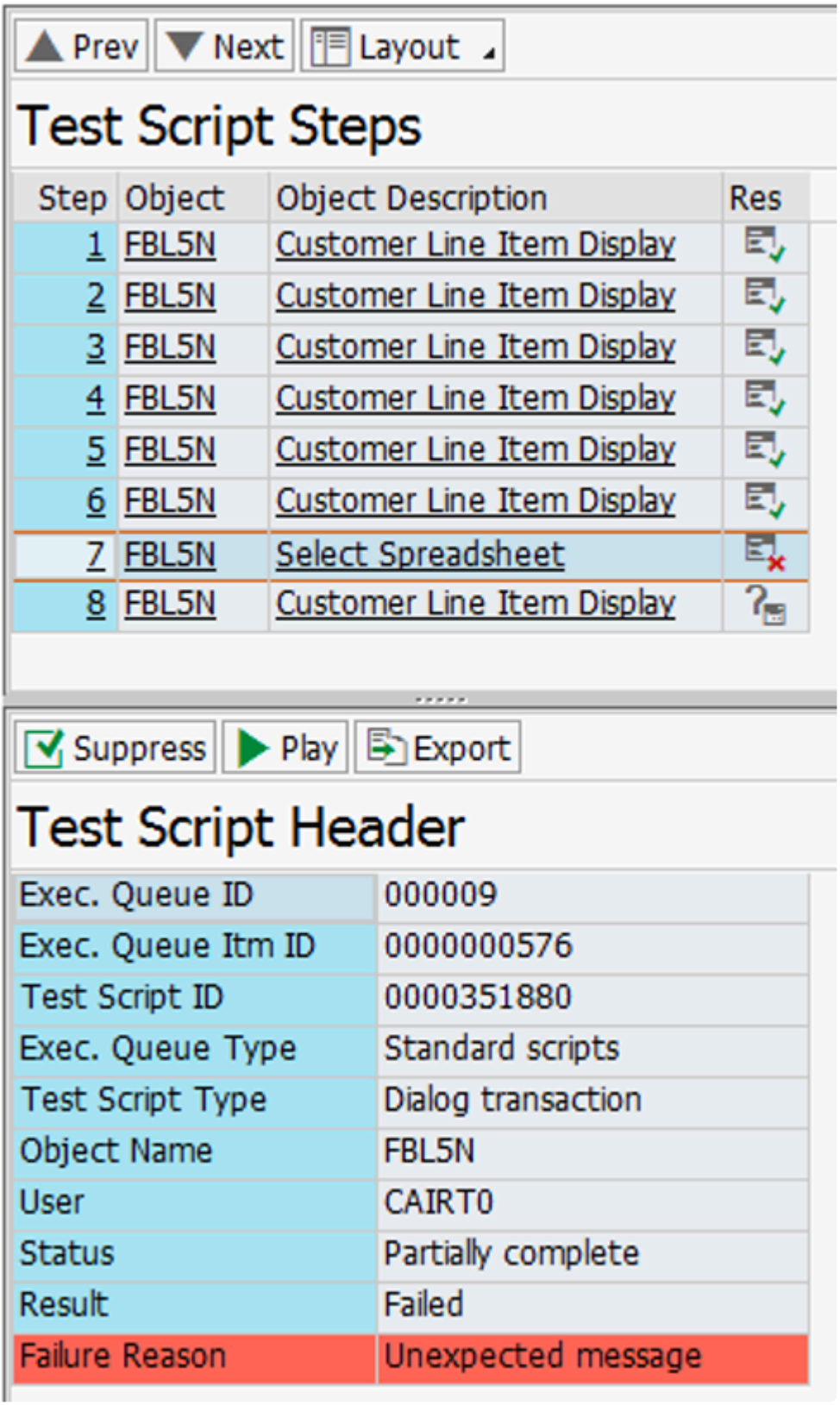
We now need to take a look at the function that would have been executed next in the script. This can be done by selecting the step after the one that failed.



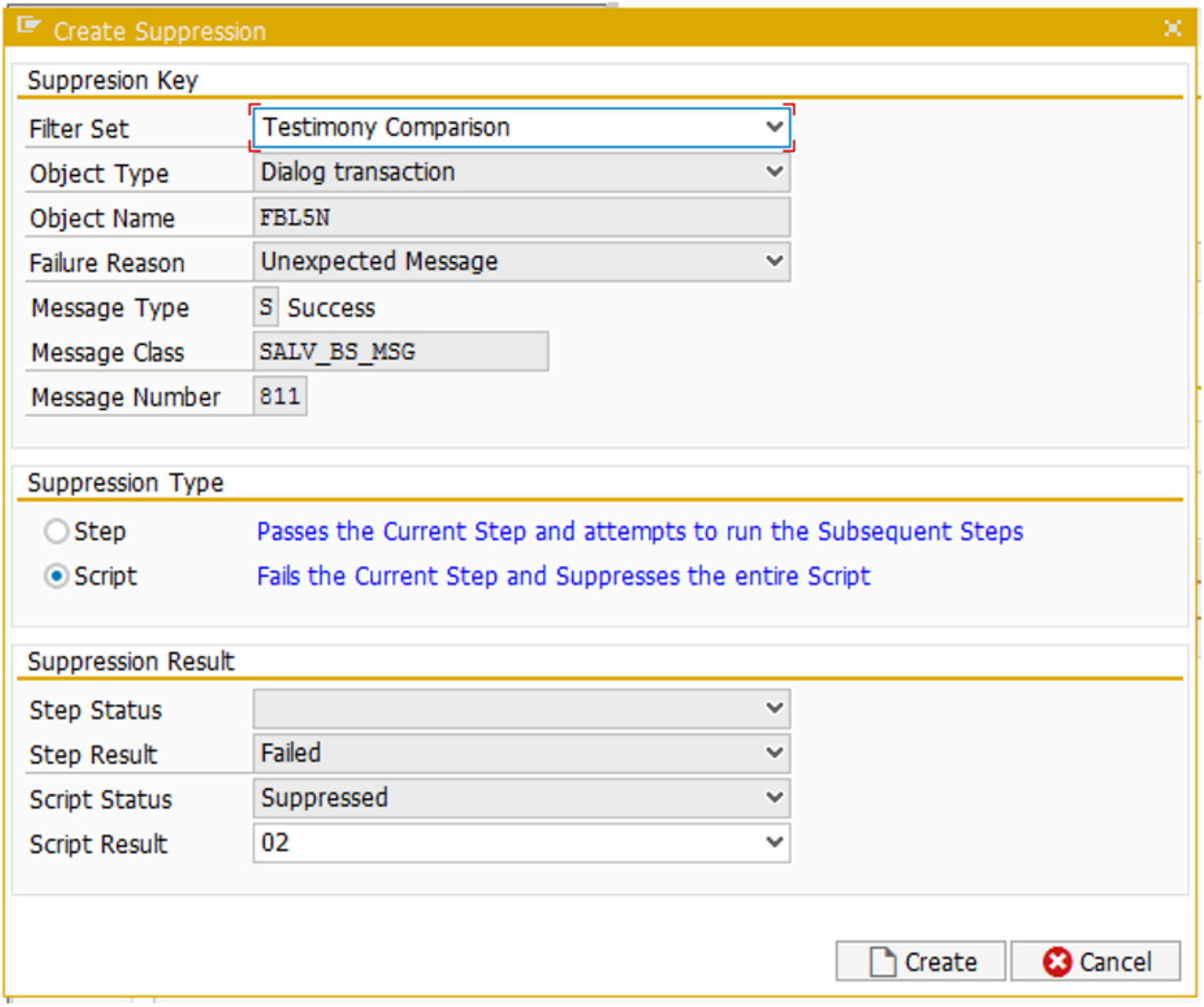
We can see here that the user pressed the back button (Function Back (=&F03)). So in this case, since we have an identical screen on the step that failed, and the function being executed in the next step is a standard function, we can deduce that if the failed step (step 8) was suppressed by Testimony, then the rest of the script would be able to continue. This failure is therefore suitable for step-level suppression.

3.5.2.1.2. Creating a step-level suppression

You can create a step-level suppression from within the Investigate Screen by selecting the failed step and then clicking on the Suppress button.



The following screen is then displayed.

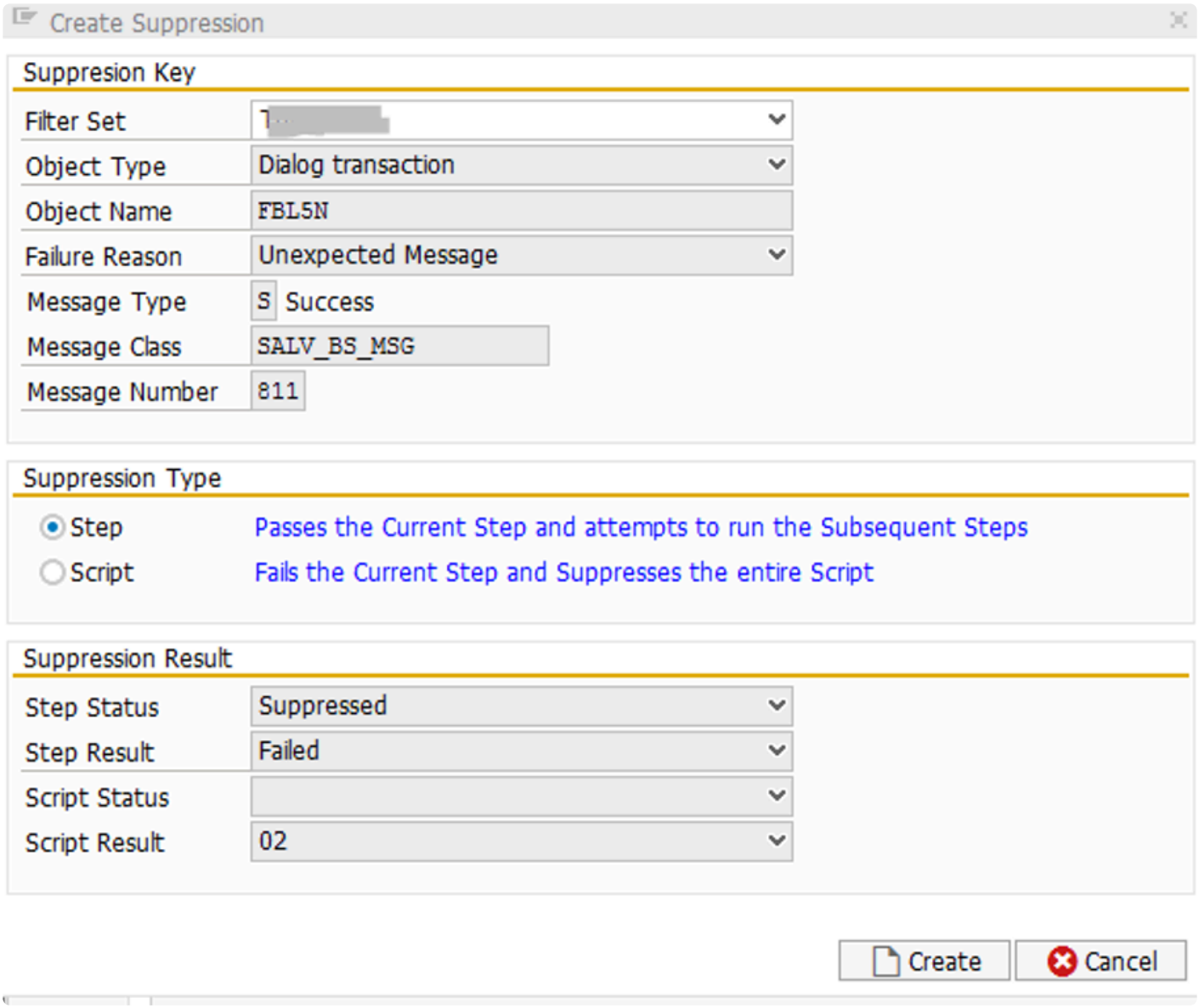


The Suppression Key contains information on the failure to be suppressed, including the object name, error type and the message to be suppressed.

Firstly, select your custom filter set from the first drop-down. By default, the option to create a script-level suppression is selected. You will therefore need to select the Step suppression type.

> ! Don't use the default Testimony Comparison filter set, as this will be overwritten when you come to upgrade Testimony.

Your screen should now look like this.



Click on Create, and the suppression will be created. In future playbacks, if Testimony encounters this same error again it will mark the step as passed as attempt to execute the rest of the script.

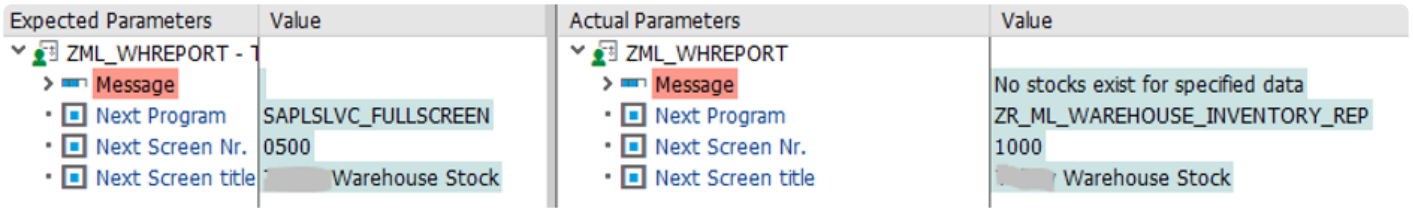
3.5.2.2. Script-level suppression

Where there is an error you want to suppress, but it is not suitable for step-level suppression, then a script-level suppression is possible.

With a script-level suppression, the failed step is still marked as failed, and the script is terminated, but no defect is raised. It is also possible to determine how this is reported in the overall playback statistics.

3.5.2.2.1. Example of a script-level suppression

In the below defect, we can see that we have a data-related defect we received a “No stocks exist…” message on the selection screen.

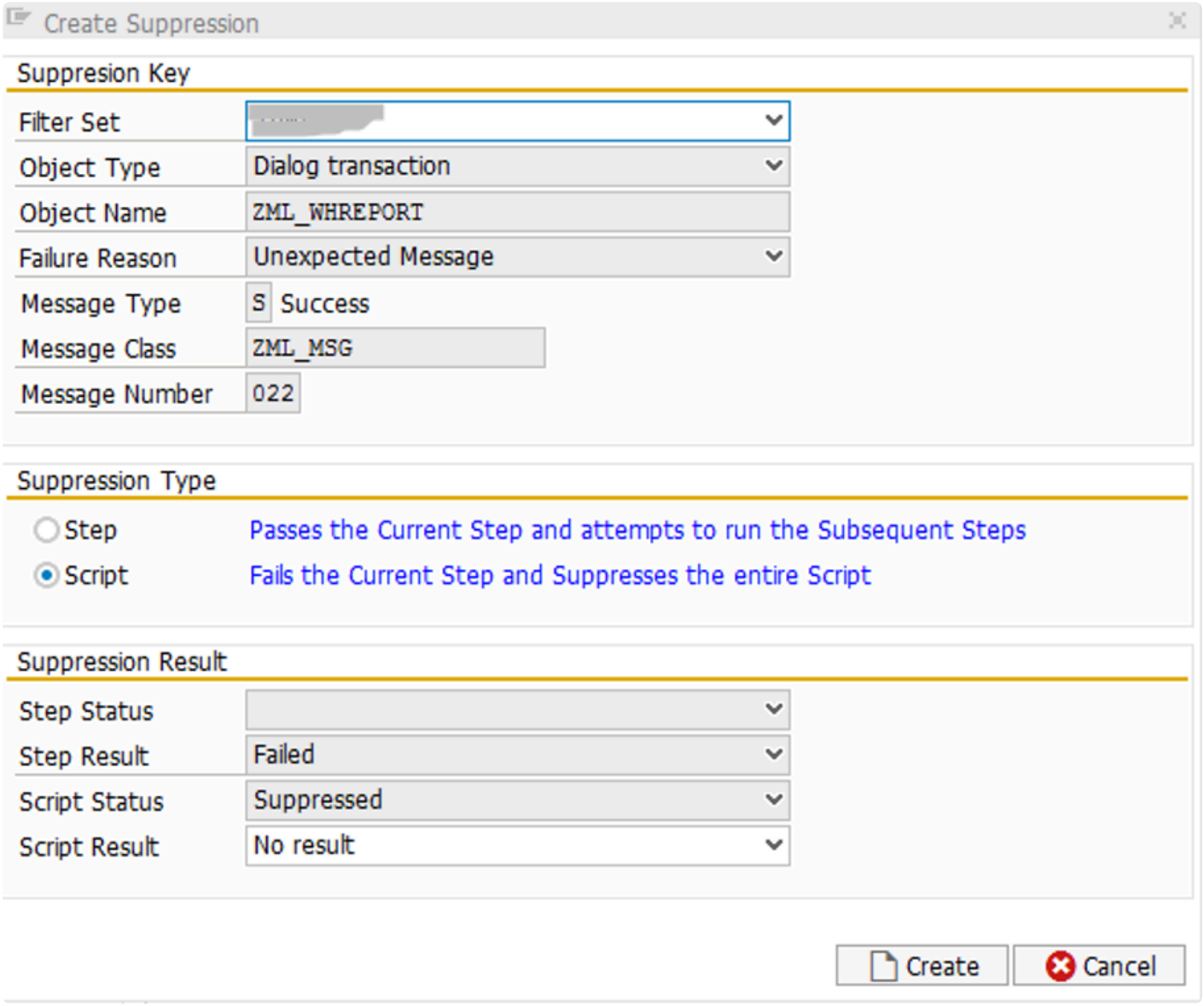


In this case, notice that the screens are different in the expected and actual parameters. In the recording, the user was taken to a list of stock, whereas in the playback the bot remained on the selection screen. This failure, therefore, is not suitable for a step-level suppression.

However, since we know that this is a data-related defect caused by either sequencing or the failure of a previous transaction, we want to suppress the creation of a defect, which we can do at the script level.

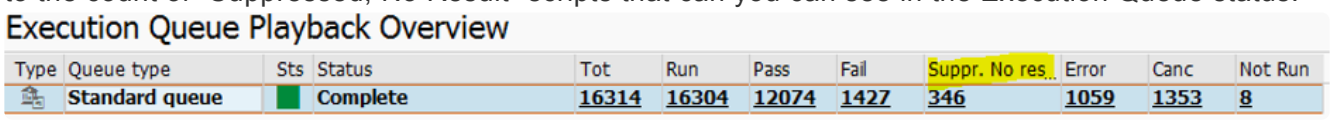
3.5.2.2.2. Creating a script-level suppression

As before, we can create a script-level suppression by clicking on the Suppress button on the Investigate Screen. We again get a pop-up to enter the suppression details, and this time we select our custom filter set and keep the Script suppression type radio button selected.



In the Script Result drop-down, there are two possibilities for determining what the final status of the script should be.

- No result: this excludes this script from the calculation of passed or failed scripts. Instead, it is added to the count of “Suppressed, No Result” scripts that can you can see in the Execution Queue status:



- Passed: this passes the script, adding it to the total of passed scripts for the playback.

In general, it is best to select No Result, as this gives a more accurate representation of the status of the playback. We only want genuinely passed scripts to marked as such.

3.5.3. Sampling

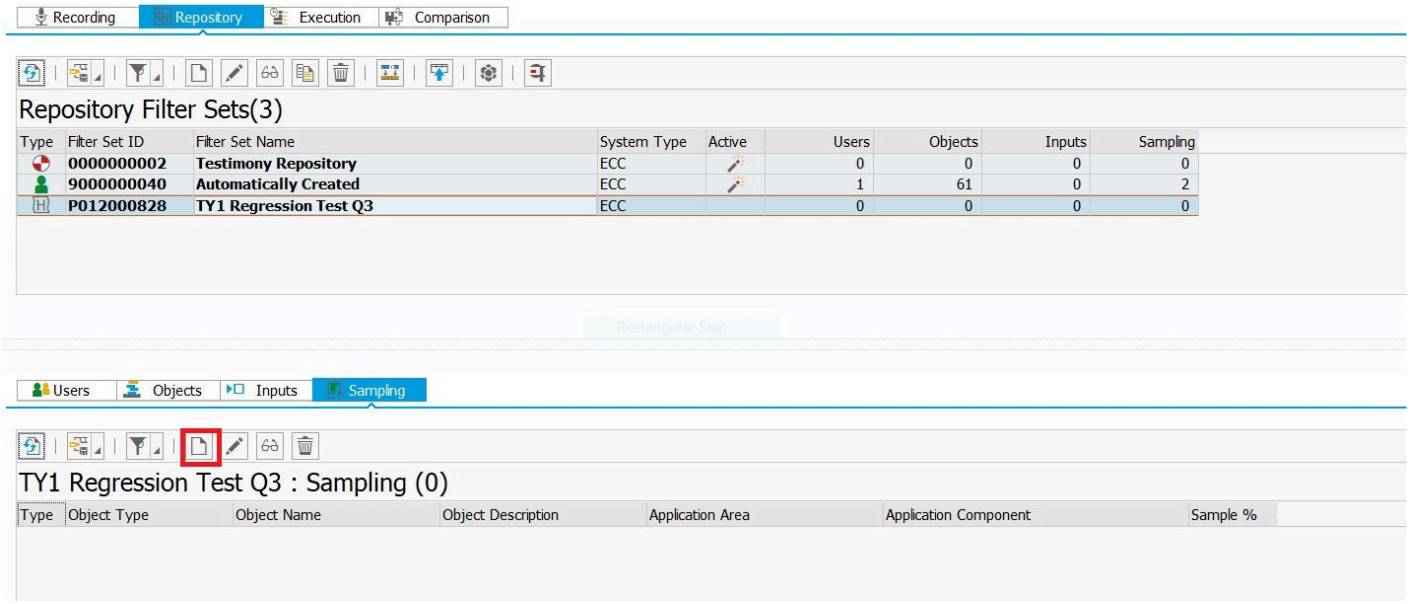
Sampling is part of Filter Set functionality and is only applicable at the “Transfer to Repository” stage. The idea behind it is as follows :

If you recorded 1000 VA03 transactions, you can play a sample of x% to help reduce playback times. Sampling works for:

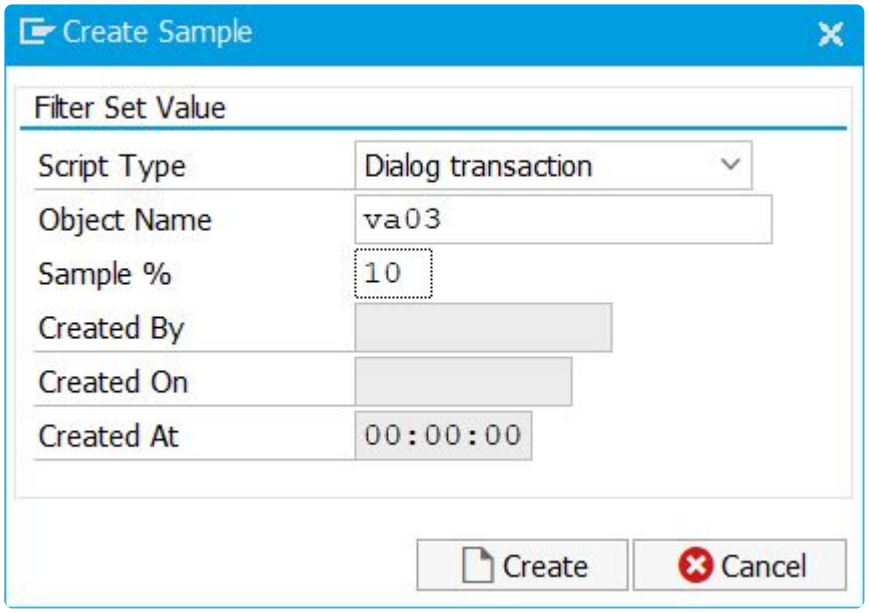
- Dialogue transactions
- Batch Jobs
- RFC’s

Within Filter Sets

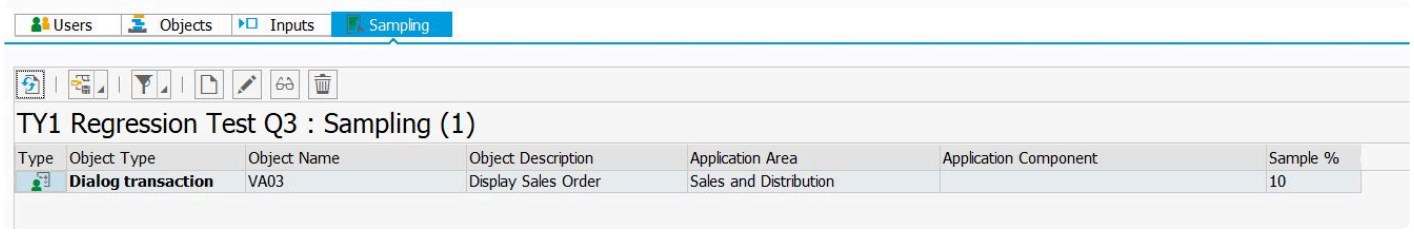
To Create a sampling you need a “Repository” Filter Set already created. Go to the “Sampling” tab and click on the “Create Sample” button



The following will will appear where you enter the relevant info about your sample. In the example below, we will only be executing the Dialog Transaction VA03 10% of the actual recorded volume.

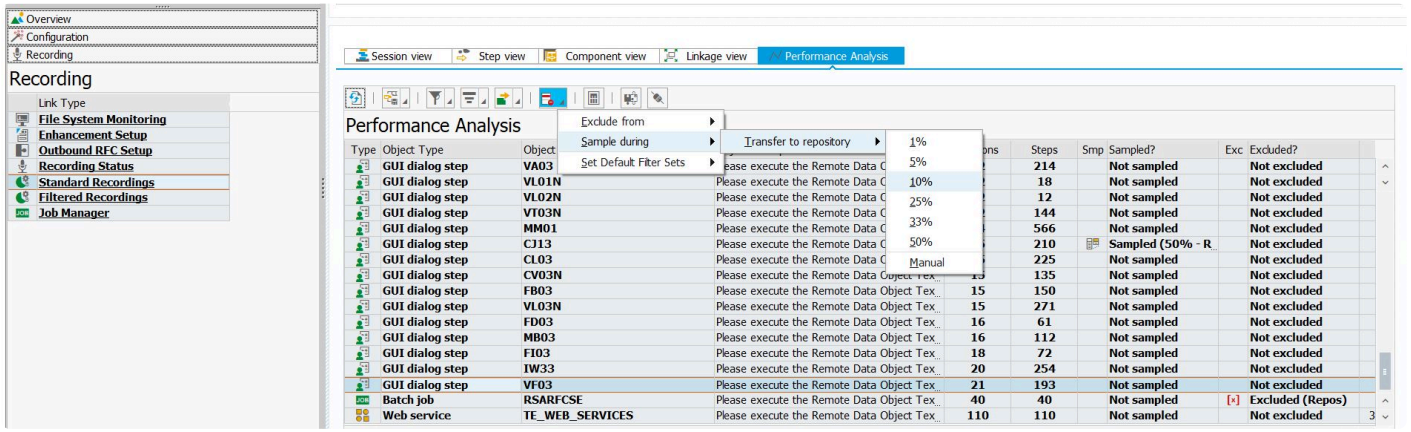


Once you click the “Create button, you see your new Sampling listed under the “Sample” Tab.



After Recording and before “Transfer to Repository”

You can also create a Sampling Filter after you’ve creaded a recording and before you transfer it to the repository. From the “Performance eAnalysis” tab, find the transaction you want to sample. Once highlighed, click the Exclude/Sample button and select the rate of your sampling. In the example below, Dialog step VF03 would be sampled at a rate of 10%.



3.5.4. User Preference and Plan Configuration

> ! Important: All user preferences are configured in the Central system only. The two important tables are **/BTI/AUT_C_USPA** and **/BTI/AUT_C_USPR**.

User Preferences

The following user preference configuration has been added or amended to the configuration tables in order to more accurately capture this data during the recording and restore during the playback. Please review these entries and check if any custom configuration entries should be added (for example for Y/Z transactions that store and use user-specific settings).

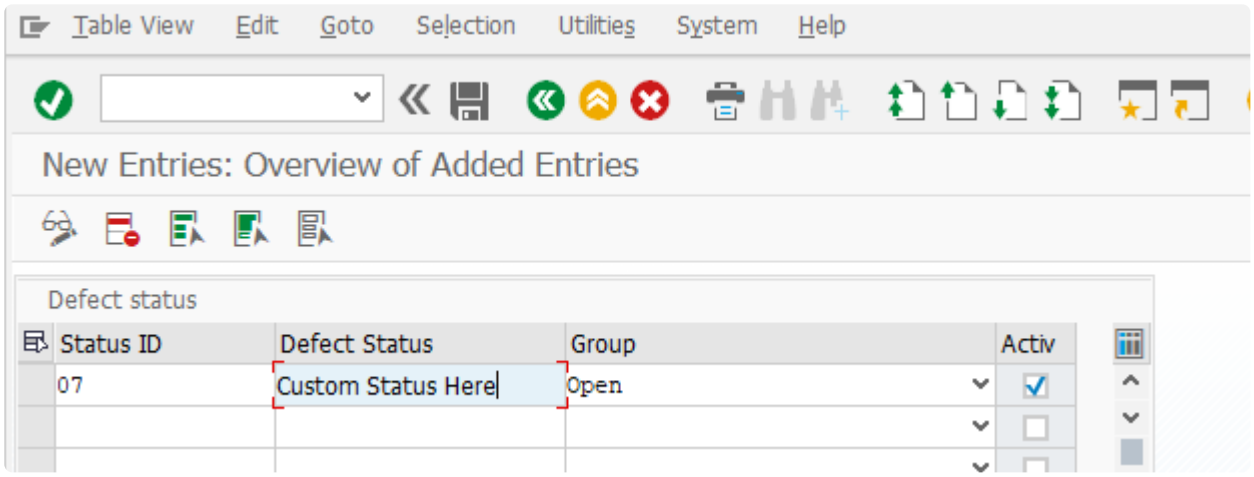
Transaction	Configuration Table
MB51	ESDUS
ME21N	ESDUS
ME22N	ESDUS
ME23N	ESDUS
ME29N	ESDUS
ME51N	ESDUS
ME52N	ESDUS
ME53N	ESDUS
MEPO	ESDUS
MIGO	ESDUS
MIRO	ESDUS
PA30	LTDX
PPO1	LTDX
PPO2	LTDX
PPO3	LTDX
PPO4	LTDX
PPO5	LTDX
PPO6	LTDX

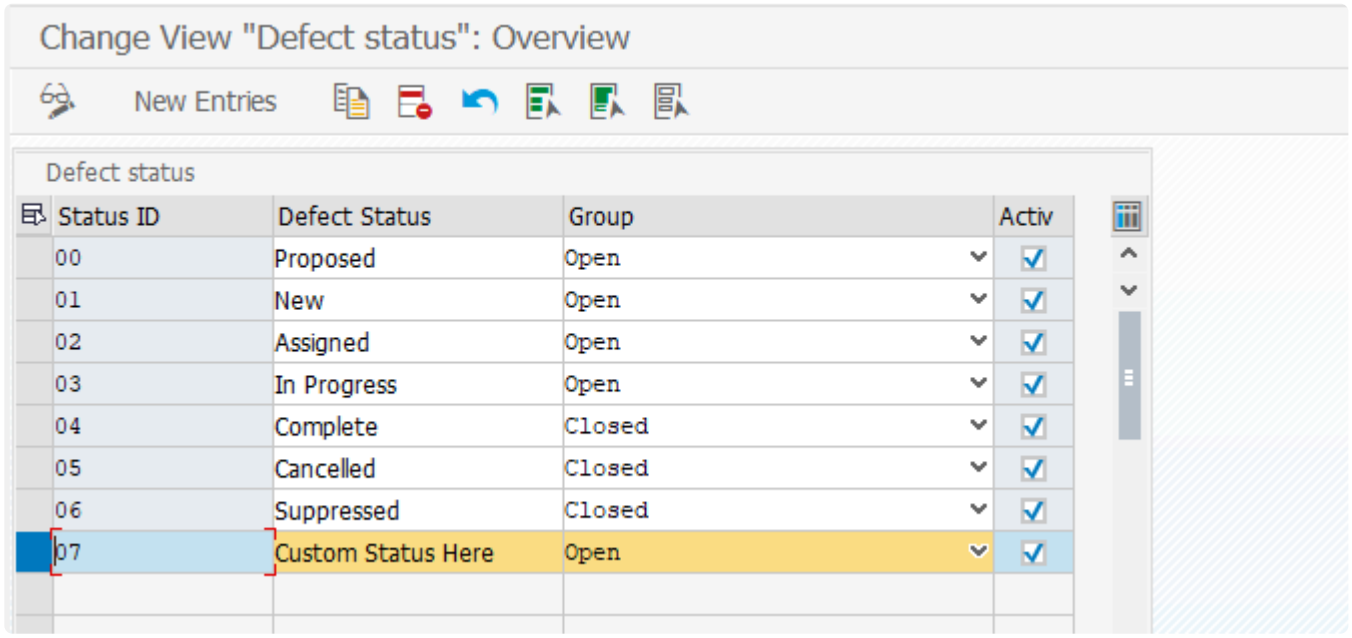
PPOC	LTDX
PPOCA	LTDX
PPOCE	LTDX
PPOCW	LTDX
PPOC_OLD	LTDX
PPOM	LTDX
PPOMA	LTDX
PPOME	LTDX
PPOMW	LTDX
PPOM_OLD	LTDX
PPOS	LTDX
PPOSA	LTDX
PPOSE	LTDX
PPOSW	LTDX
PPOS_OLD	LTDX

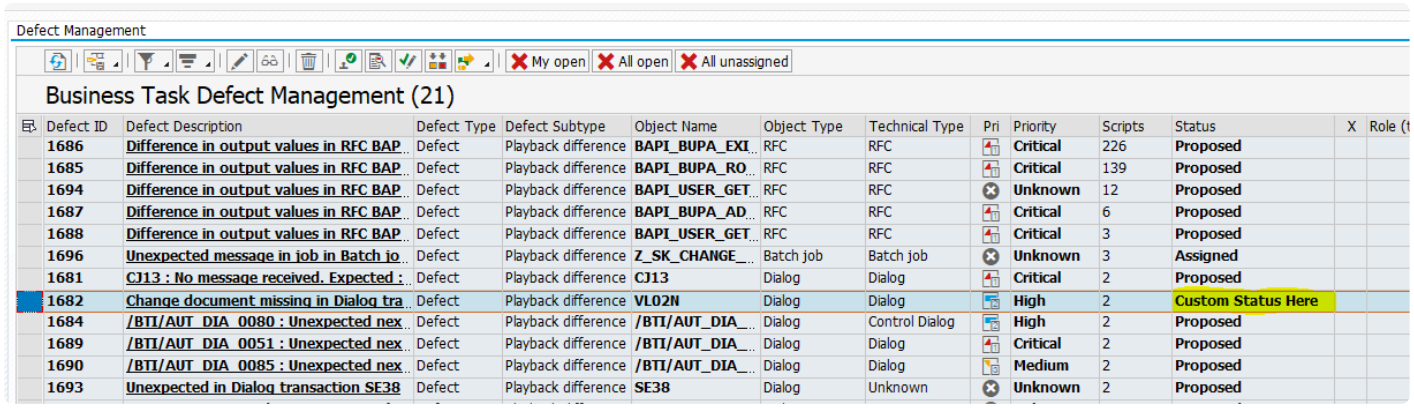
Defect Configuration

Define your own Defect Status names.
Configurable table via SM30: /BTI/AUT_C_BTSG (the corresponding text mapping table is /BTI/AUT_C_BTST, this however does not need to be touched)
All entries are modifiable and deletable except for “Proposed”, “New”, “Assigned” and “Suppressed”
ACTIV flag decides whether the status is in use or not – Deleting from this table is not needed, however allowed (except for “Proposed”, “New”, “Assigned” and “Suppressed”)

This can also be used to define subtypes

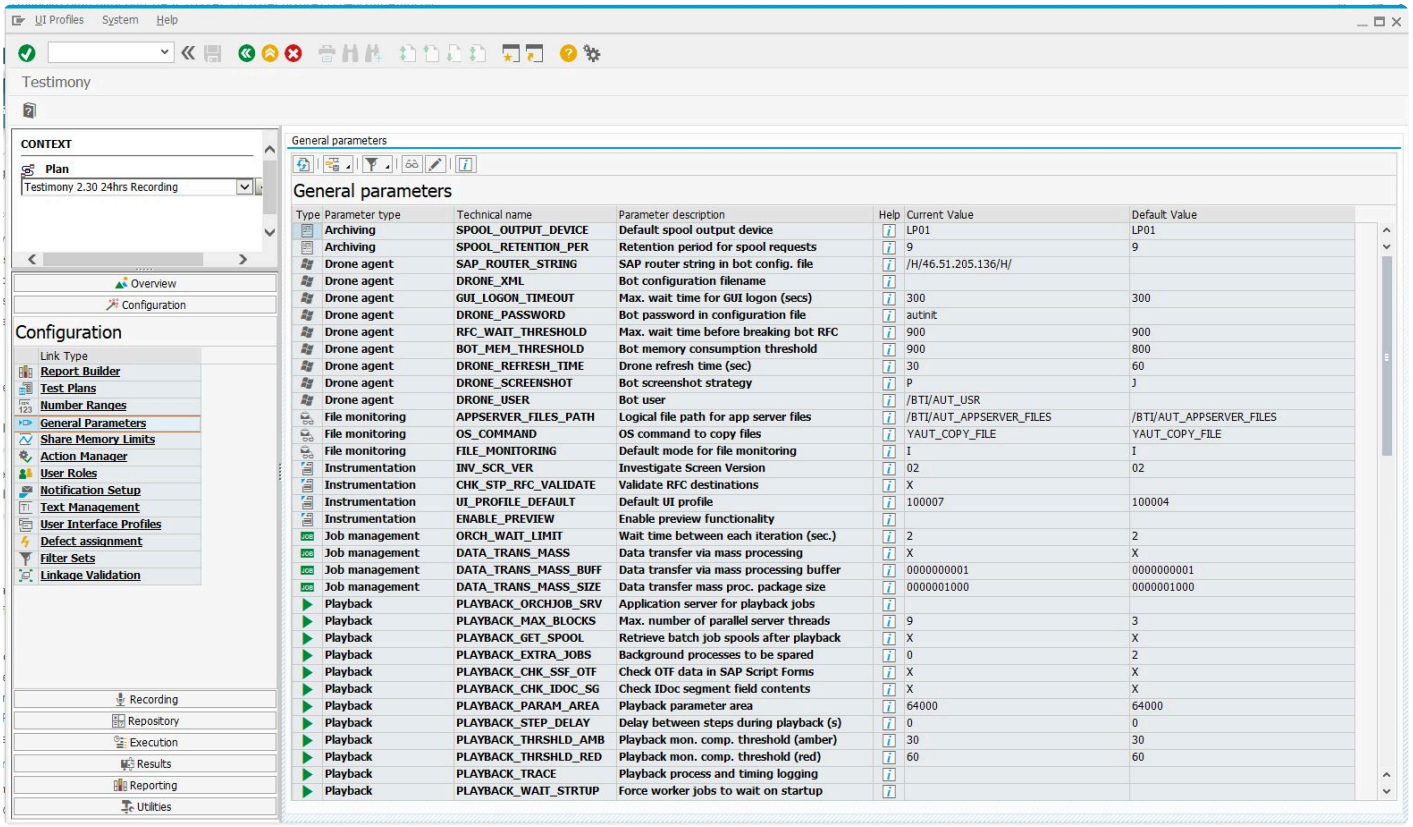






Plan Configuration

Prior to playback it is important to recheck the Plan's configuration within General Parameters



Linkage Validation Configuration

A major issue with validation of linkages for batch jobs in particular was that is often be slight discrepancies in timing and data during the playback. This means that during the recording, a batch job (for example) may produce 1000 outbound IDoc's, but during the playback it may produce something slightly different (e.g. 1001 or 999). The chances of it producing precisely what was captured in the recording, especially for recurring batch jobs, is low. Testimony is sensitive to these differences so almost always flag these batch jobs up with a failure, and creates a defect for investigation.

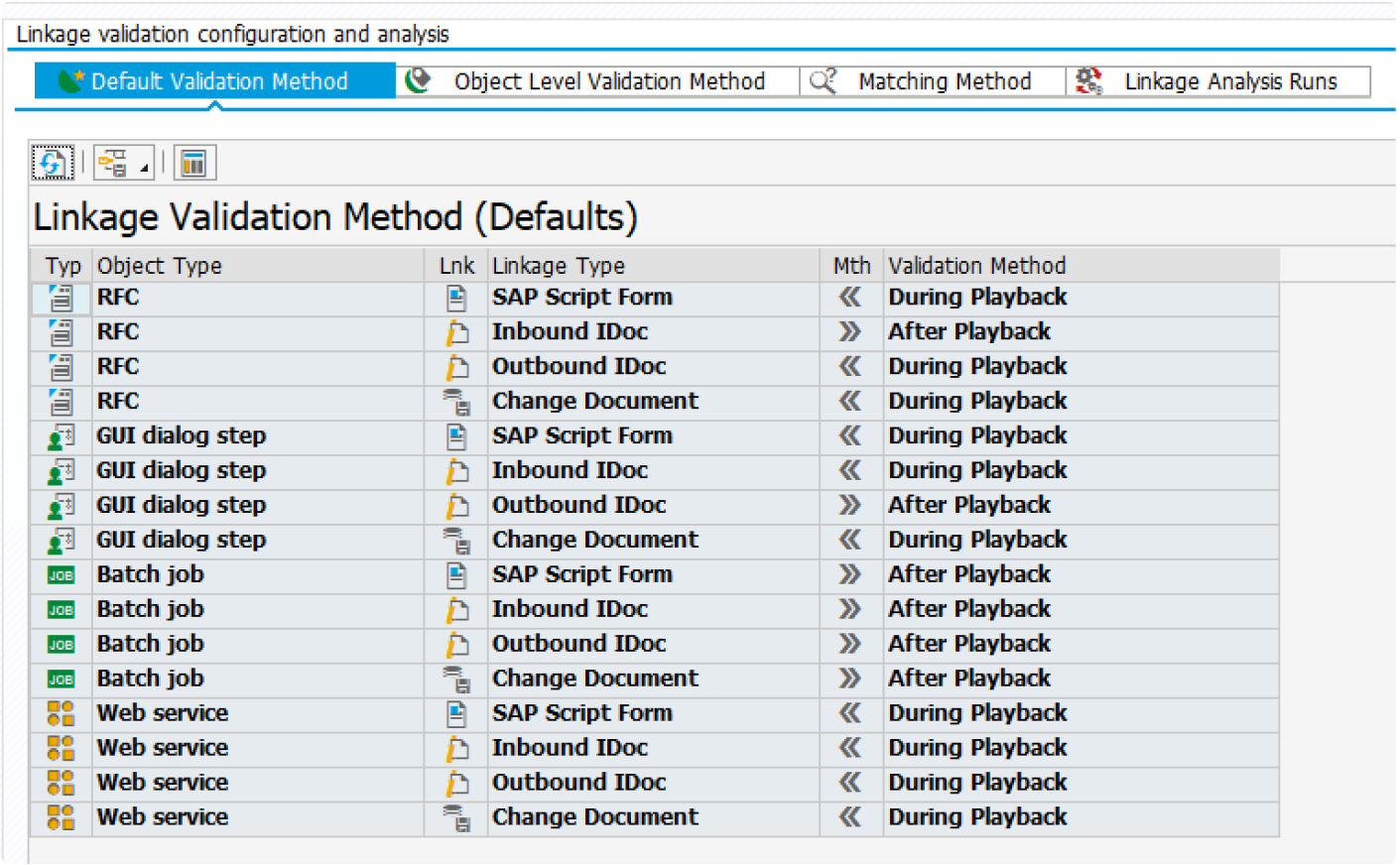
To mitigate this problem Testimony allows you are able to mark a particular object type (e.g. batch job) along with the corresponding linkage type (e.g. outbound IDoc's) so that instead of validating the linkage

during the playback, it is instead done **after the playback**.

> In addition, you can specify a particular object type (e.g. dialog transaction VA02 and linkage type change documents) if you require this capability to be more focused.

3.5.4.1. Default Validation Methods

Default configuration is delivered for the way in which linkages are validated. This is by object type and by linkage type. The default configuration is the following:



Typ	Object Type	Lnk	Linkage Type	Mth	Validation Method
	RFC		SAP Script Form	«	During Playback
	RFC		Inbound IDoc	»	After Playback
	RFC		Outbound IDoc	«	During Playback
	RFC		Change Document	«	During Playback
	GUI dialog step		SAP Script Form	«	During Playback
	GUI dialog step		Inbound IDoc	«	During Playback
	GUI dialog step		Outbound IDoc	»	After Playback
	GUI dialog step		Change Document	«	During Playback
JOB	Batch job		SAP Script Form	»	After Playback
JOB	Batch job		Inbound IDoc	»	After Playback
JOB	Batch job		Outbound IDoc	»	After Playback
JOB	Batch job		Change Document	»	After Playback
	Web service		SAP Script Form	«	During Playback
	Web service		Inbound IDoc	«	During Playback
	Web service		Outbound IDoc	«	During Playback
	Web service		Change Document	«	During Playback

Default Linkage Validation Methods

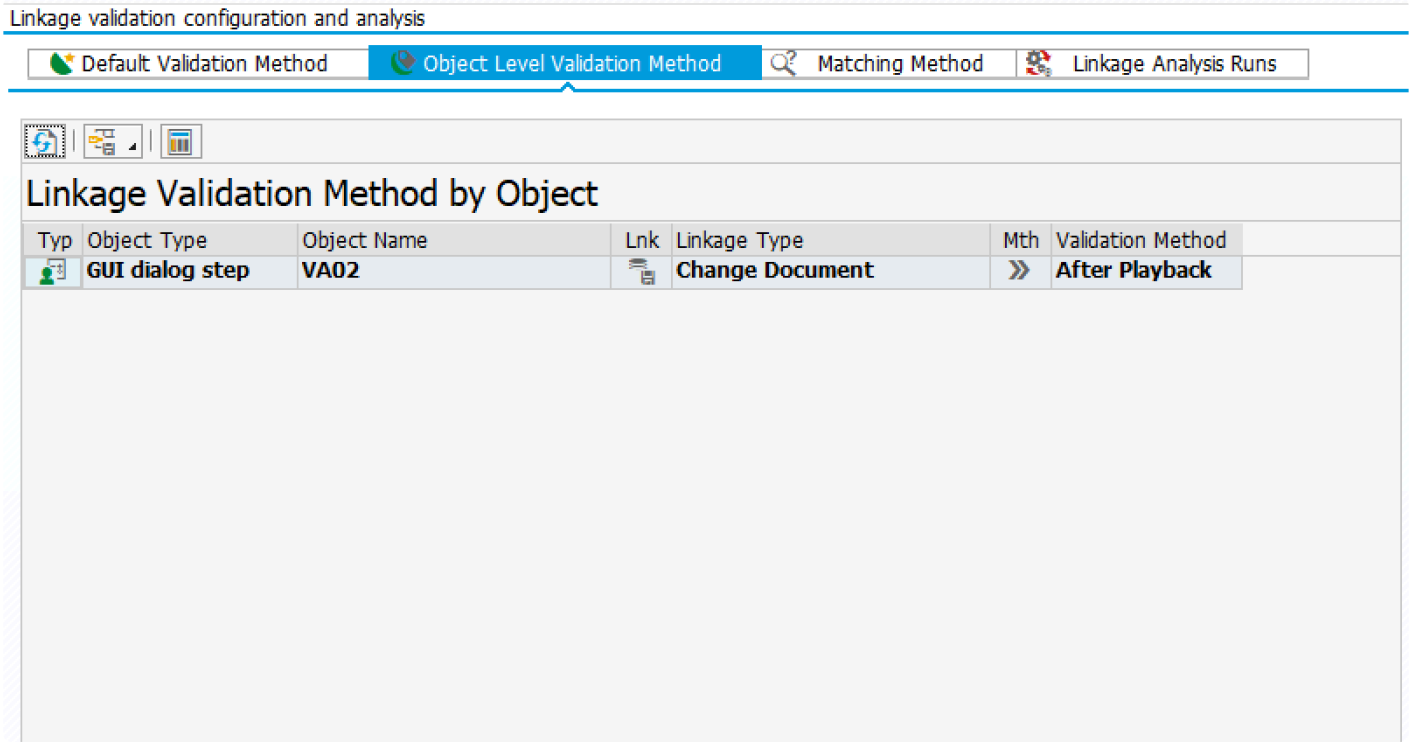
As can be seen, the default validation methods are delivered as default for batch jobs only. You can override these defaults with your own settings as per your requirements. There are really only two settings here – either the linkages of that particular type are validated **during the playback** or they are done **after the playback**.

Batch jobs are the most common scenario that requires validation of linkage types after the playback rather than during it.

3.5.4.2. Object Level Validation Method

It is possible to configure object level validation methods which will **over-ride** the default configuration in the previous section. You simply add the entry via the configuration section with the following key fields:

- Object Type
- Object Name
- Linkage Type
- Validation Method



Object level validation methods

In the example from the screen-shot above, this means that even though the default validation method for dialog transactions and change documents is “during playback”, because VA02 has been entered here it means that this will over-ride the default configuration for dialog transactions and change documents.

> Please note that this works in the opposite direction. If the default validation method is to perform it after playback (for example for batch jobs), then you can specify an entry above which over-rides this and you can ensure that object performs that type of validation during the playback instead.

3.6. Notifications

Overview

It is possible to configure Testimony so that certain activities or events automatically notify one or more users as a one-off notification or on a regular basis during a particular activity.

Process Steps

To setup the notifications three steps are required:

1. It is required that the central system has SAPconnect set up (see SAP transaction SCOT). This requires a Basis Administrator to set up if this is not already available. The RSCONN01 program also needs to be running regularly as a batch job.
2. Testimony has to be configured to determine which notifications are sent and to which users/roles. Instructions on quickly configuring notifications are here
3. The program /BTI/AUT_NOTIFICATION_ENGINE needs to be set up as a batch to run at regular intervals. The instructions on this program are here

3.6.1. Quick Notifications Configuration

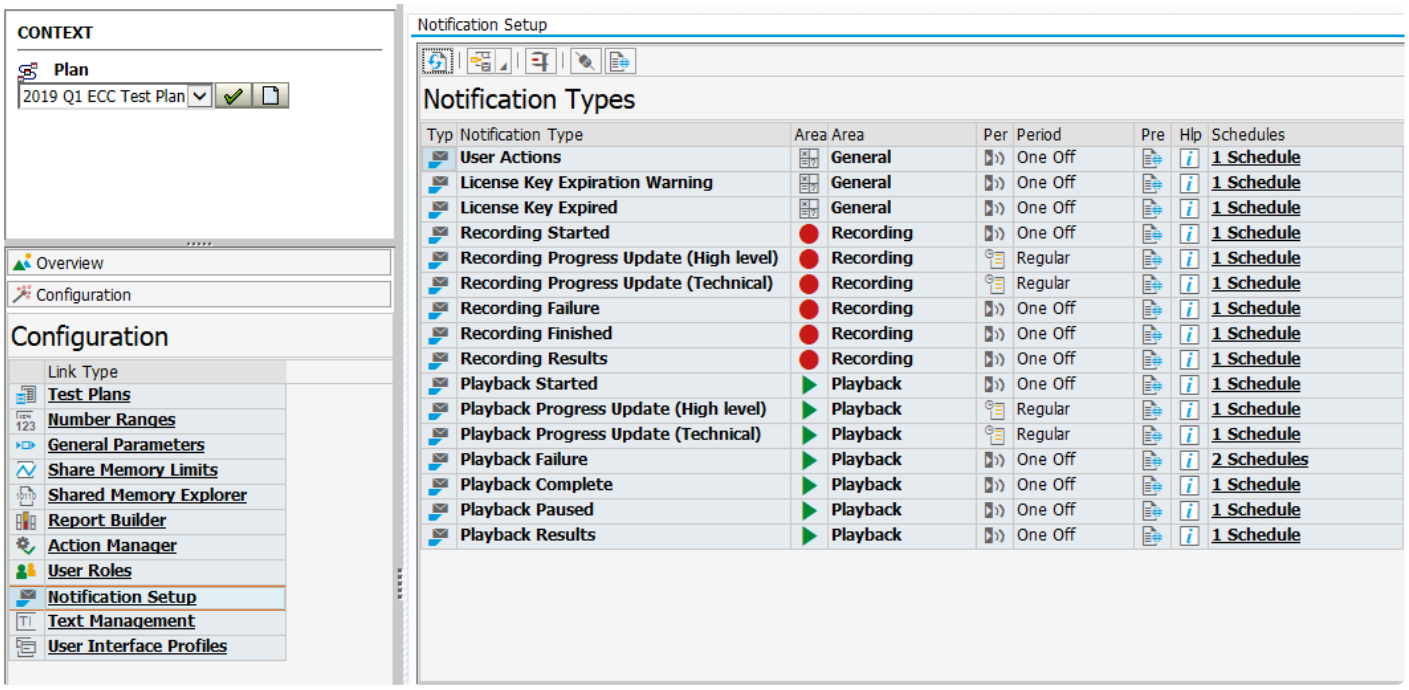
Overview

This section provides a quick guide to getting notifications configured.

Testimony provides a variety of activities or events that can be configured to automatically notify one or more users as a one-off notification or on a regular basis. The process of determining which activities trigger a notification, to which users and roles, and for which plans is detailed here.

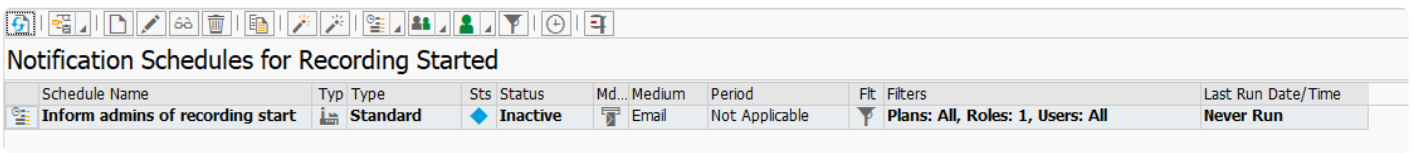
Accessing Notification Setup

As a Testimony Administrator the **Notification Setup** should appear under the **Configuration** tray as per the screenshot below.



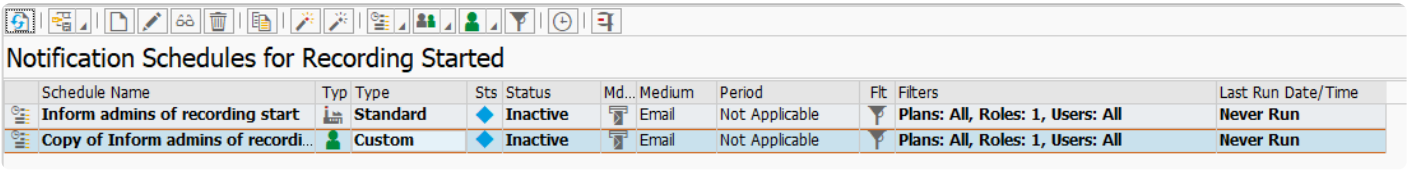
Setting up a Notification

To view a **Notification Schedule** double click on the **Notification Type** that you want to set up and the details of the **Notification Schedule** appear as below.



The simplest way to configure a notification is to copy the standard **Notification Schedule**, by highlighting

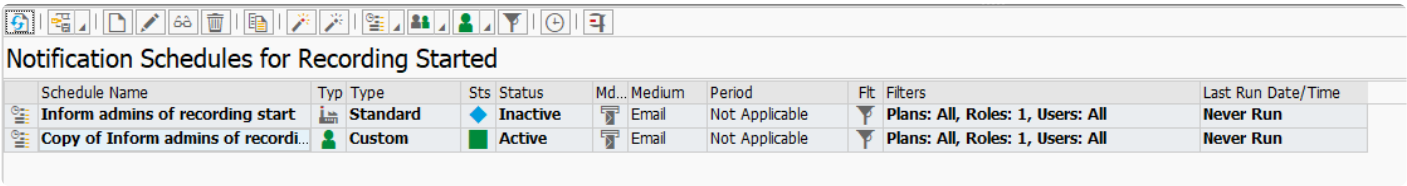
the standard and pressing the copy button  . A new **Notification Schedule** will then appear with a type of **Custom** as below. For full details on all of the functions available for notifications please see the Detailed Notification Configuration section.



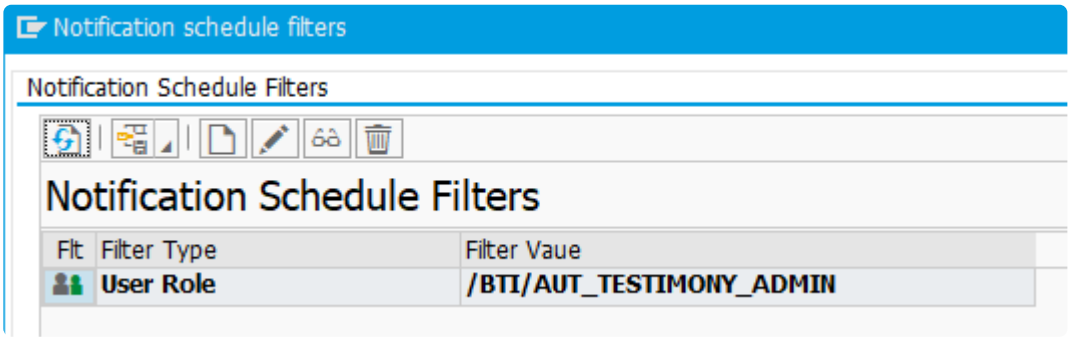
Next, ensure that the **Notification Schedule** is active. To do this select the new custom **Notification Schedule** that you just created and press the activate button  . Pressing the refresh button  will now show the custom **Notification Schedule** to be active as below.

> Note if a schedule is inactive it won't be triggered and no notifications will be sent out.

If you already have the program /BTI/AUT_NOTIFICATION_ENGINE running as a batch job then when this action is triggered at the next point the batch job runs the notifications will be sent out. If you don't have the batch job set up you should set this up as per the instructions here..



To check to whom the notification will be sent select the new custom **Notification Schedule** and press the filter button  . The screen below shows that for this schedule any user with the Testimony Administration role will receive this notification.



3.6.2. Detailed Notification Configuration

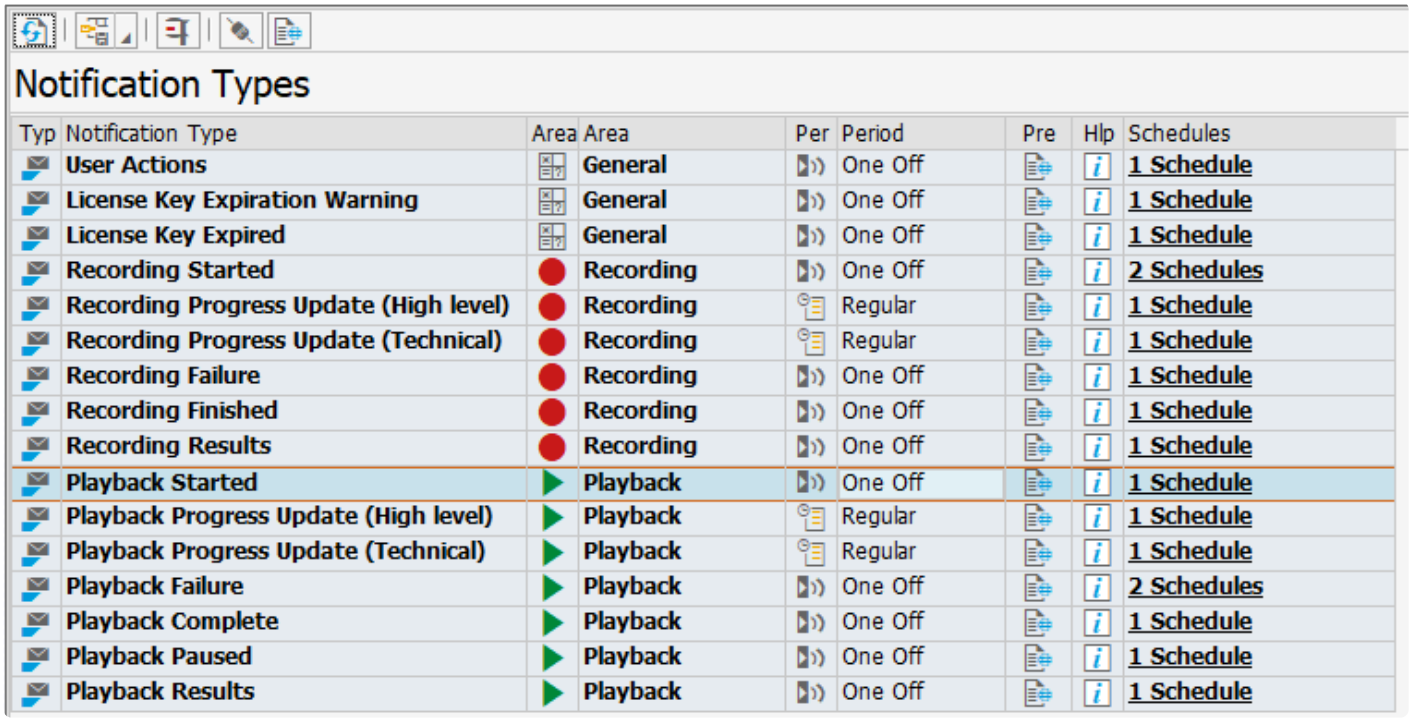
Overview

This section provides a more detailed guide to the functionality available for notifications.

Notification Types

The **Notification Types** are displayed as below. Each one has the following fields:

- **Area** is the part of Testimony that this **Notification Type** relates to.
- **Period** can be a one off notification sent each time that action is triggered or regular notifications occurring throughout the recording or playback at periods determined by the schedule.
- Clicking on **Pre** shows a preview of the notification that is sent out.
- Clicking on the information button under **Help** provides details on each **Notification Type** if further clarity is required on when it might get triggered.
- **Schedules** shows how many schedules have been set up for that **Notification Type**. Note that each **Notification Type** will have a standard **Notification Schedule** provided as a template to copy. A double click displays the schedules set up under this type.



Notification Types

Typ	Notification Type	Area	Area	Per	Period	Pre	Hlp	Schedules
	User Actions		General		One Off			1 Schedule
	License Key Expiration Warning		General		One Off			1 Schedule
	License Key Expired		General		One Off			1 Schedule
	Recording Started		Recording		One Off			2 Schedules
	Recording Progress Update (High level)		Recording		Regular			1 Schedule
	Recording Progress Update (Technical)		Recording		Regular			1 Schedule
	Recording Failure		Recording		One Off			1 Schedule
	Recording Finished		Recording		One Off			1 Schedule
	Recording Results		Recording		One Off			1 Schedule
	Playback Started		Playback		One Off			1 Schedule
	Playback Progress Update (High level)		Playback		Regular			1 Schedule
	Playback Progress Update (Technical)		Playback		Regular			1 Schedule
	Playback Failure		Playback		One Off			2 Schedules
	Playback Complete		Playback		One Off			1 Schedule
	Playback Paused		Playback		One Off			1 Schedule
	Playback Results		Playback		One Off			1 Schedule

The Notification setup has the following actions:

Refresh
This allows a simple refresh of the grid to update any changes.

Grid Tools 
The grid tools enable you to change the layout of any grid.

Run Notification Job 
On selecting one of the notification types this button can be pressed to call the program /BTI/AUT_NOTIFICATION_ENGINE and as a test trigger the one off sending of the notification to SAP connect.

SAPconnect 
Directs you to the SAPconnect transaction SOSA which displays requests that are being sent, or have already been sent by using SAPconnect, according to the selection criteria

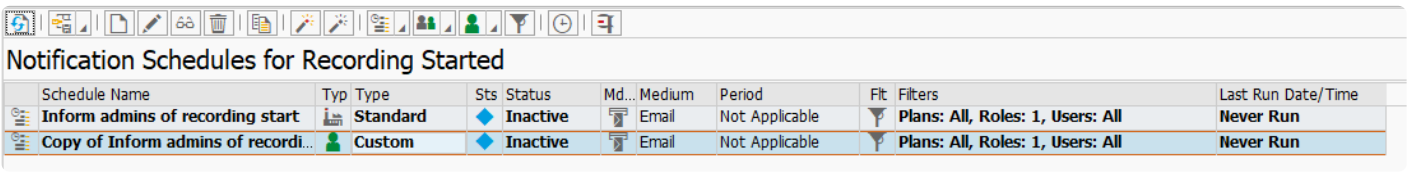
Web Repository 
Directs you to the SAP standard transaction SMW0 to view the templates for the notifications

Note that double clicking a **Notification Type** will display the **Notification Schedules** with further details as below:

Notification Schedules

The **Notification Schedules** are displayed as below. Each one has the following fields:

- **Type** can be standard as supplied with Testimony or custom which is a customised version. The standard version can't be used so you will need create your own custom versions to operate with Testimony.
- **Status** can either be inactive which means that the notifications won't be sent or active which means is should be operating. Note that you can't activate the standard template supplied: it is provided as a template to copy.
- **Medium** is how the notifications are supplied. Only email notifications are available at present.
- **Period** is only shown for regular Notification Types to show how frequently these will be sent while the activity is running.
- **Filters** shows the groups that will receive this notification and this can be restricted by plans, roles and users.
- **Last Run Date/Time** is simply the last time that the notification was run.


Notification Schedules for Recording Started

Schedule Name	Typ	Type	Sts	Status	Md...	Medium	Period	Flt	Filters	Last Run Date/Time
Inform admins of recording start		Standard		Inactive		Email	Not Applicable		Plans: All, Roles: 1, Users: All	Never Run
Copy of Inform admins of recordi...		Custom		Inactive		Email	Not Applicable		Plans: All, Roles: 1, Users: All	Never Run

The Notification Schedule has the following actions:

Refresh 

This allows a simple refresh of the grid to update any changes.

Grid Tools 
The grid tools enable you to change the layout of any grid.

Create 
Enables the user to create a new Notification Schedule based on the Notification Type. Only custom notification schedules can be created.

Change 
Enables the user to change the selected Notification Schedule. The schedule's name, period and whether or not it is active can be changed.

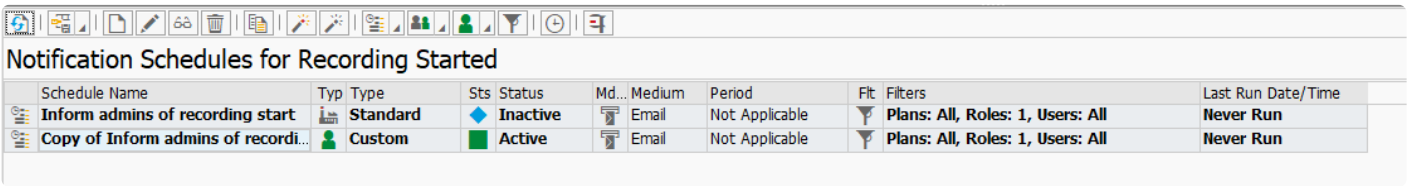
Display 
Simply displays the selected Notification Schedule.

Delete 
Simply deletes the selected Notification Schedule. Note: don't delete the standard one as then it won't be available to copy in the future.

Copy 
The recommended method to create a new schedule is to select the standard template provided and press the copy button.

Activate 

Select a custom **Notification Schedule** press the activate button then press the refresh button  . The custom **Notification Schedule** should now be active as below. If a schedule is inactive it won't be triggered and none of its notifications will be sent out.

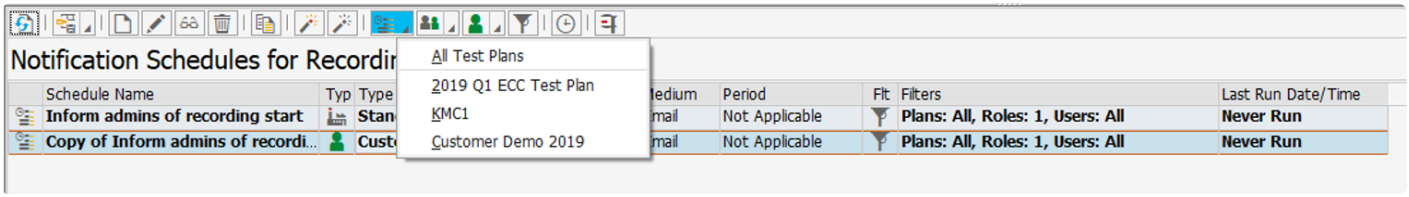


Deactivate 

Select a custom **Notification Schedule** press the deactivate button then press the refresh button  . The custom **Notification Schedule** should now be inactive. No further notifications will now be sent out for this schedule.

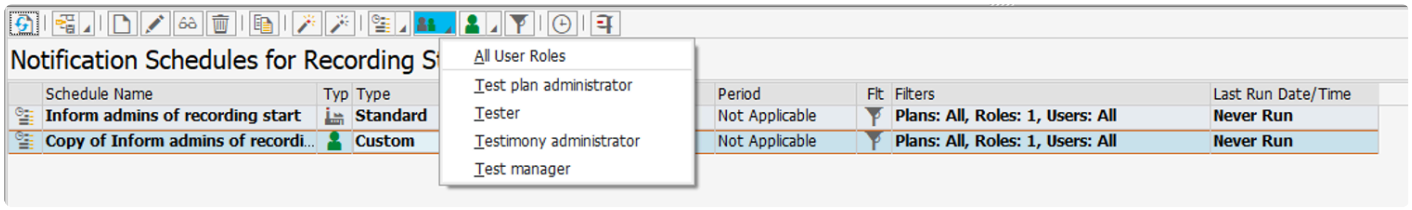
Test Plans 

Notifications can be configured just for one Test Plan or all plans. Pressing the right hand side of button reveals the options to select all Test Plans or select an individual Test Plan as below. Note it is possible to select more than one test plan if required.



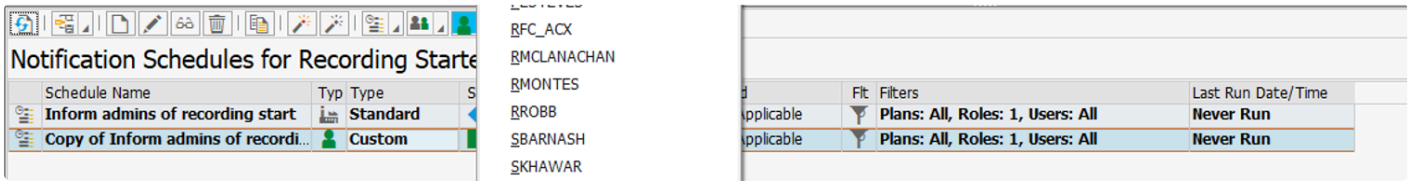
Roles 

Notifications can be configured to be sent to certain user roles. Pressing the right hand side of button reveals the roles that can be selected as below. Note it is possible to select more than one role if required.



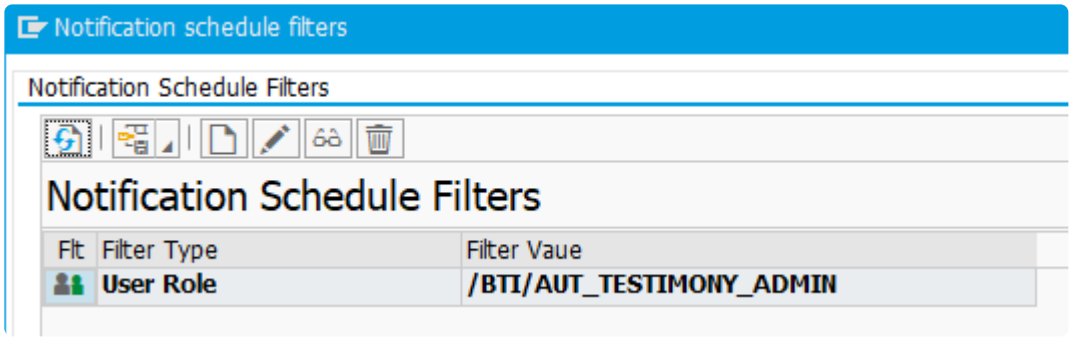
Users 

Notifications can be configured to be sent to specific users. Pressing the right hand side of button reveals all the users that can be selected as below. Note it is possible to select more than one user if required.



Filters 

Filters can be used to check who the notification will be sent to. Select a custom **Notification Schedule** and press the filter button  . The example below shows that any user with the Testimony Administration role will receive this notification. Using the delete option here is the best way to remove part of a filter. Options are available here to create or change filters, however, it is recommended that the **Test Plan**, **Roles** and **User** options are used to do this as above.



Set Last Run Date/Time

The ability to set the last run date and time if required.

Run Notification Job

On selecting a notification schedule this button can be pressed to call the program /BTI/AUT_NOTIFICATION_ENGINE and as a test trigger the one off sending of the notification to SAP connect.

3.6.3. Batch Job for Notifications

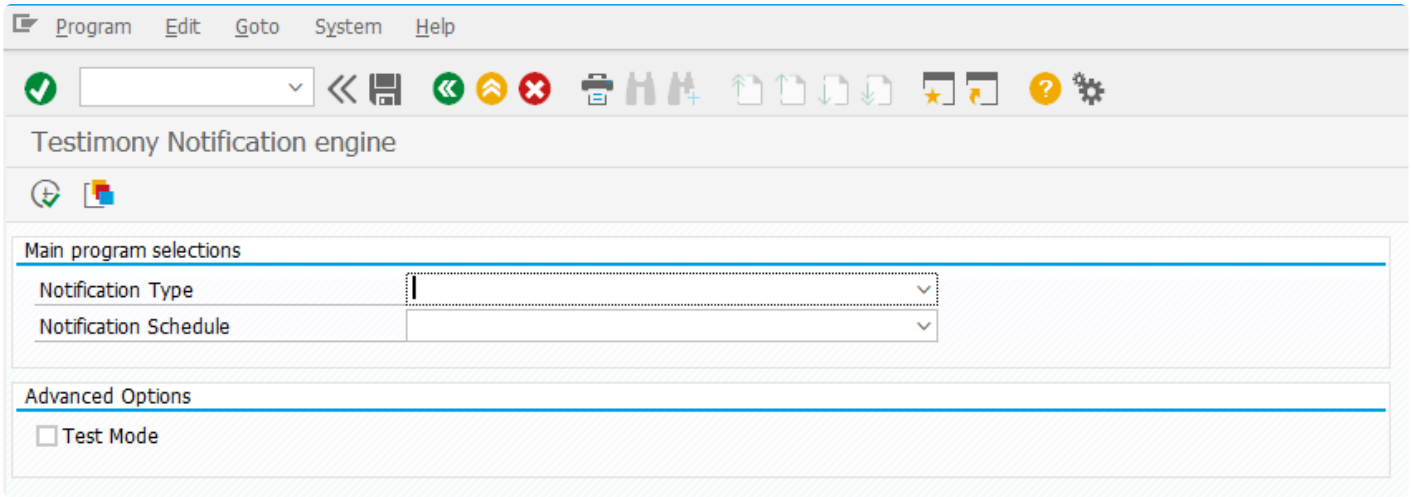
Overview

To enable the notifications to be sent out regularly the program /BTI/AUT_NOTIFICATION_ENGINE must be set to run as a batch job at regular intervals. When determining the periodicity of the batch job, consider the following.

- The minimum “delay” you want between a one-off event (e.g,, a playback finishing) occurring and a notification being received
- The minimum period defined for a recurring notification (e.g., how frequently recording status updates are sent)
- The periodicity of the batch job that runs SAPCONN01 to send emails using SAPConnect

Setting up the Batch Job

Use SM36 to set up the program /BTI/AUT_NOTIFICATION_ENGINE as a SAP standard batch job. It is recommended that you don’t select any **Notifications Types** or **Notification Schedules** so you don’t need a variant, however, it is possible to have more frequent runs for certain selections if you want a more complex setup.



3.7. Performing a Recording

Overview

The Recording phase can occur once a Test Plan has been created with the associated Source and Target systems identified. It’s important to have an understanding of the anticipated activity load on the source system to help guide the amount of timefor which the recording should be active.

Process Steps

- Recording Preparation
- Start Recording
- Monitor Recording
- Stop Recording

3.7.1. Recording Preparation

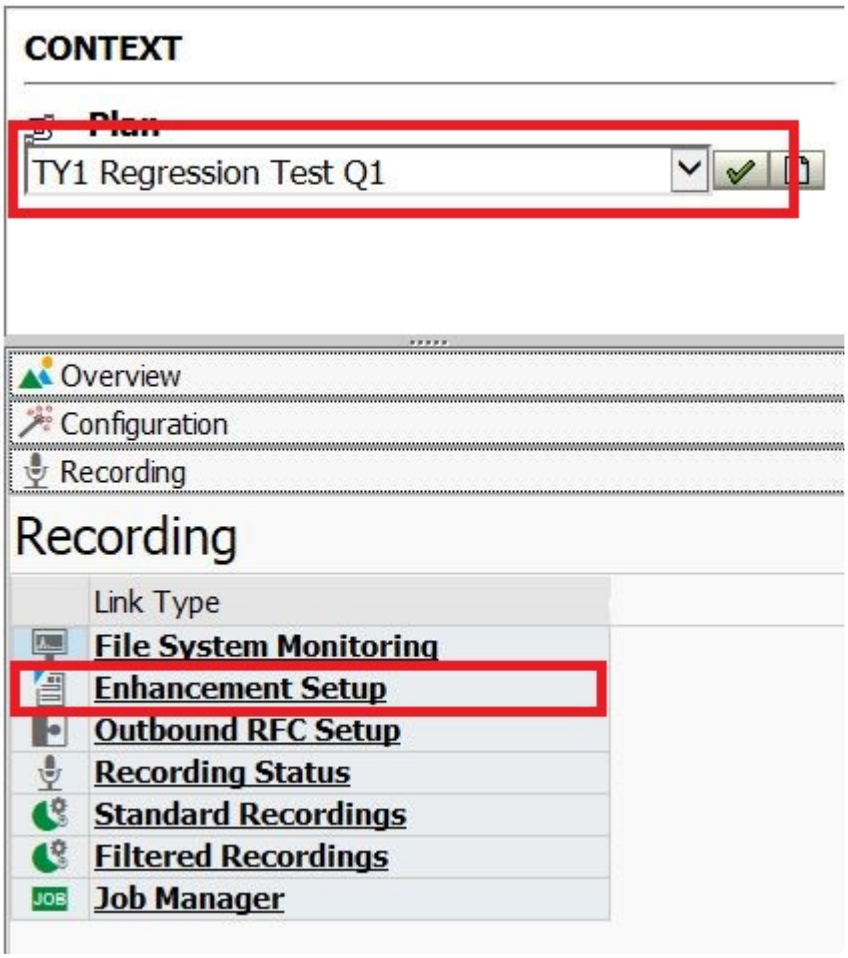
Overview

When you are ready to run a test phase leveraging the recording / learning capability in Testimony, you need to first prepare the source systems.

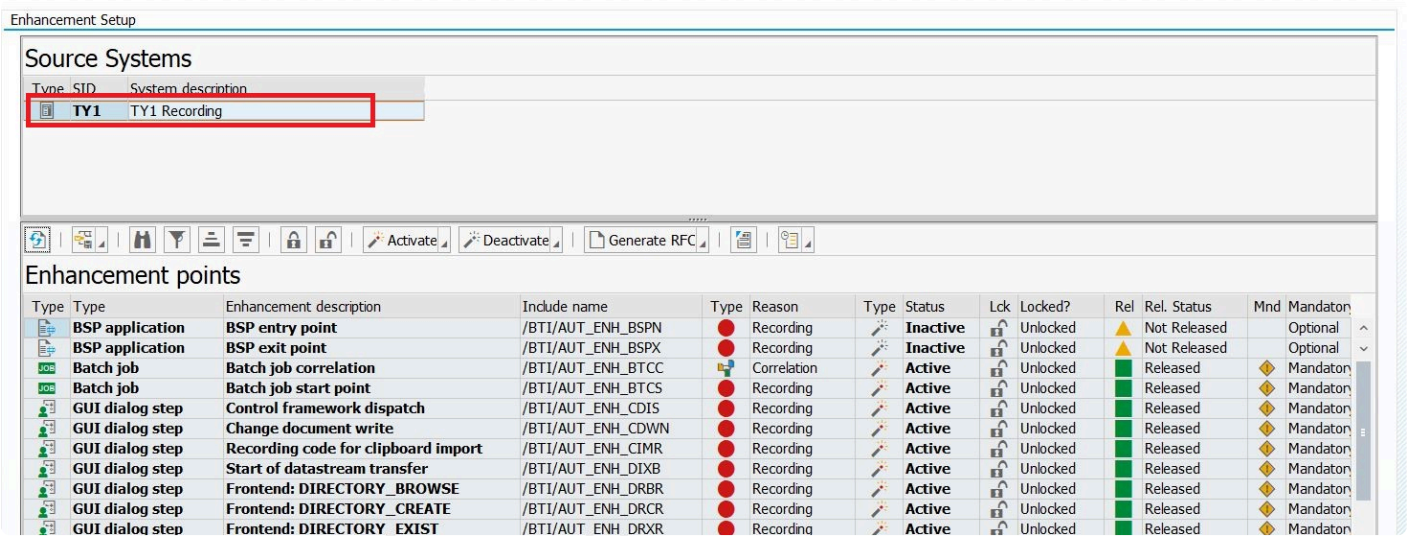
Process Steps

1. Select Test Plan to use
2. Generate Inbound RFC Capture – Prepares for inbound RFC calls to be captured. This is an optional step an only needed once per recording system and only if you will be recording RFC calls.
3. Activate enhancements – Prepares the system for capture of all interactions.
4. Execute Check Steps – Validation that the Source system is ready for recording

1. Select Test Plan from the drop down and click on the Tick. From there, you'll need to click on the Enhancement Setup option within the Recording Section



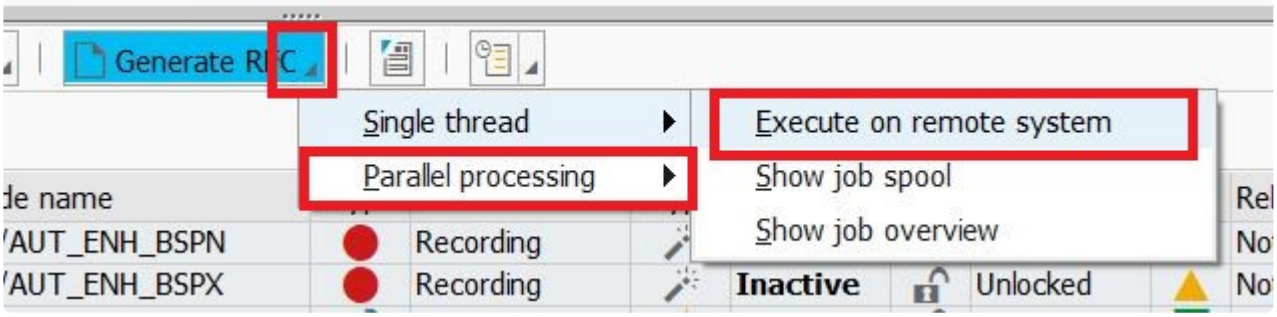
On the right panel, you'll see your defined Source System(s). Double click on the system to show the Enhancement Points screen below



2. Generating the RFC Wrapper Programs – OPTIONAL

If you are planning on recording RFCs, then you will need to generate the RFC wrapper programs. These are small pieces of code that are executed before and after each RFC function module is called in order to capture the required information for the playback.

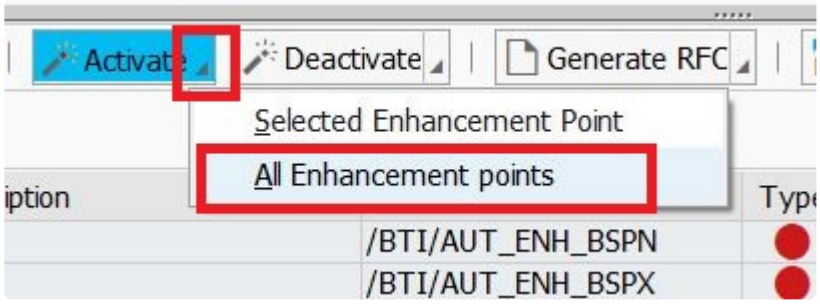
In the Enhancement Points section, click the button Generate RFC and select Parallel Processing -> Execute in Background.



This will submit several background jobs in the source system which analyse each RFC-enabled function module and create a wrapper program for it. This can take an hour or more to run, so execute this step well in advance of the time at which you want to start your recording.

3. Activating recording enhancements

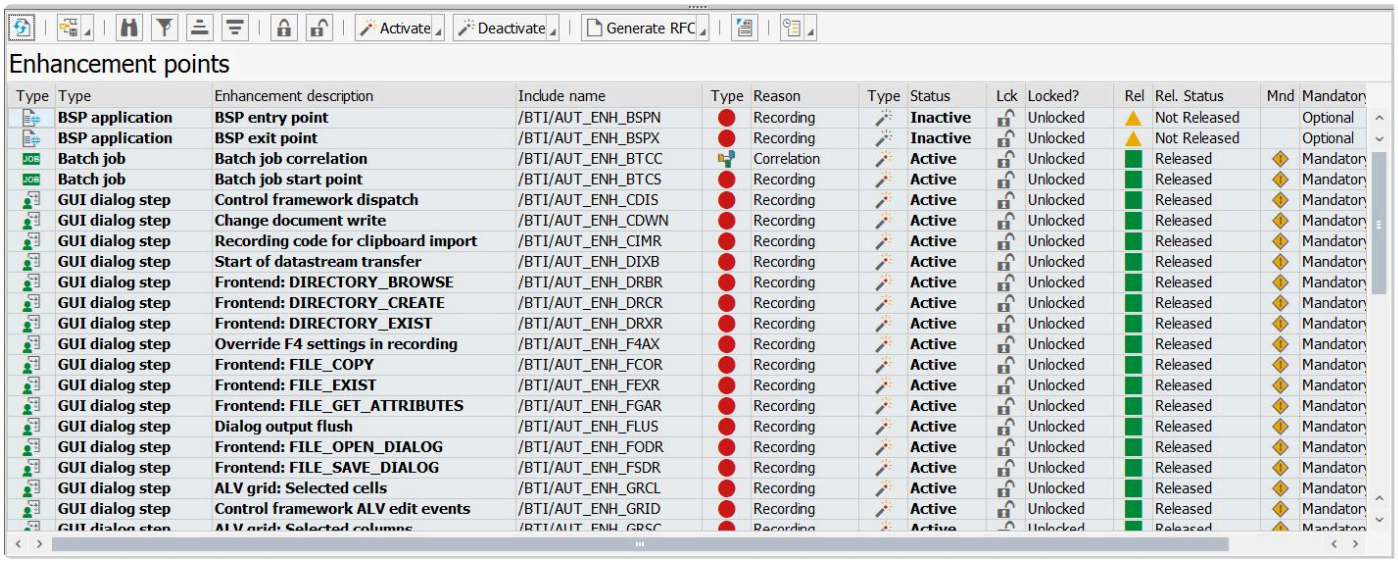
Testimony enhancements need to be activated in the source (recording) system before a recording can take place, but after the optional Generate RFC step is complete. If you are not already there, navigate to the Enhancement Setup section and click on the Source system. You should see a list of enhancement points and their current status (which should be Inactive). To activate the enhancements, click the Activate button and then choose “All Enhancement points”



You will be shown a pop-up giving information on the number of enhancements to be activated. Click on Proceed and after a few seconds the enhancements will be activated.

Note: The number of enhancements will depend on those marked as Mandatory in the Enhancement Configuration

Note that some enhancement points (with Locked and/or Unreleased statuses) will not be activated.



Type	Type	Enhancement description	Include name	Type	Reason	Type	Status	Lck	Locked?	Rel	Rel. Status	Mnd	Mandaton
	BSP application	**BSP entry point**	/BTI/AUT_ENH_BSPN		Recording		**Inactive**		Unlocked		Not Released		Optional
	BSP application	**BSP exit point**	/BTI/AUT_ENH_BSPX		Recording		**Inactive**		Unlocked		Not Released		Optional
	Batch job	**Batch job correlation**	/BTI/AUT_ENH_BTCC		Correlation		**Active**		Unlocked		Released		Mandaton
	Batch job	**Batch job start point**	/BTI/AUT_ENH_BTCS		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Control framework dispatch**	/BTI/AUT_ENH_CDIS		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Change document write**	/BTI/AUT_ENH_CDWN		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Recording code for clipboard import**	/BTI/AUT_ENH_CIMR		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Start of datastream transfer**	/BTI/AUT_ENH_DIXB		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Frontend: DIRECTORY_BROWSE**	/BTI/AUT_ENH_DRBR		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Frontend: DIRECTORY_CREATE**	/BTI/AUT_ENH_DRCR		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Frontend: DIRECTORY_EXIST**	/BTI/AUT_ENH_DRXR		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Override F4 settings in recording**	/BTI/AUT_ENH_F4AX		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Frontend: FILE_COPY**	/BTI/AUT_ENH_FCOR		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Frontend: FILE_EXIST**	/BTI/AUT_ENH_FEXR		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Frontend: FILE_GET_ATTRIBUTES**	/BTI/AUT_ENH_FGAR		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Dialog output flush**	/BTI/AUT_ENH_FLUS		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Frontend: FILE_OPEN_DIALOG**	/BTI/AUT_ENH_FODR		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Frontend: FILE_SAVE_DIALOG**	/BTI/AUT_ENH_FSDR		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**ALV grid: Selected cells**	/BTI/AUT_ENH_GRCL		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**Control framework ALV edit events**	/BTI/AUT_ENH_GRID		Recording		**Active**		Unlocked		Released		Mandaton
	GUI dialog step	**ALV grid: Selected columns**	/BTI/AUT_ENH_GRSC		Recording		**Active**		Unlocked		Released		Mandaton

When activating the recording enhancements for the first time after the transports are imported, you may see this error:

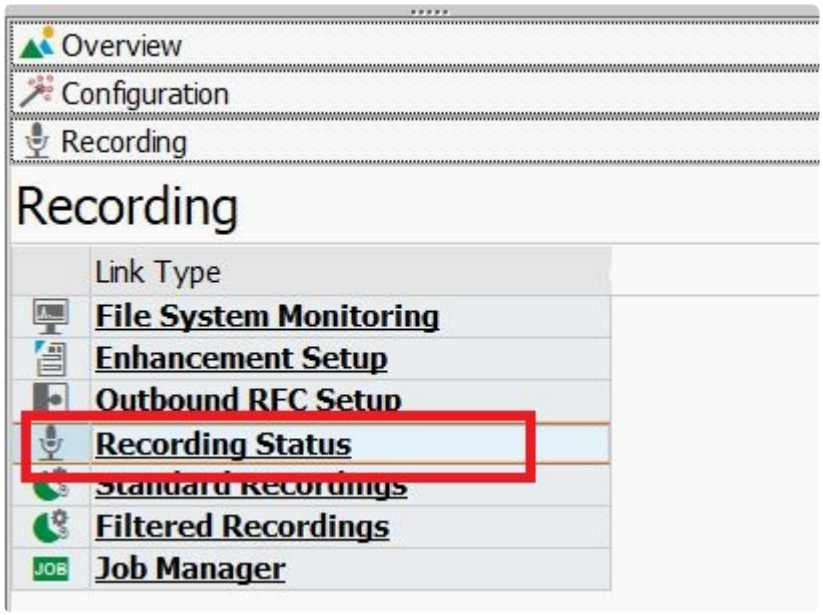
"Failed to activate the enhancement point : Include /BTI/AUT_ENH_GRZC cannot be found..."

If you encounter this error, follow the below steps to correct it.

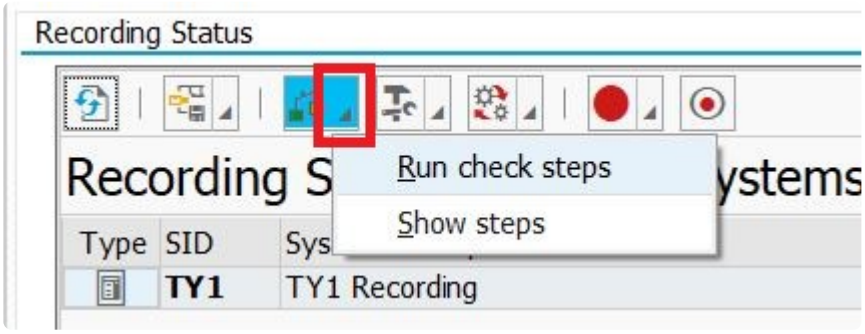
- In Testimony, deactivate all the recording enhancements
- In the Source System
  - Go to transaction SE24
  - Display class CL_SALV_GRZ_CONTROLLER
  - Go to -> Sections -> Public section
  - Utilities -> Update navigation index
- Exit Testimony
- Open Testimony
- Activate the recording enhancements

4. Execute Check Steps

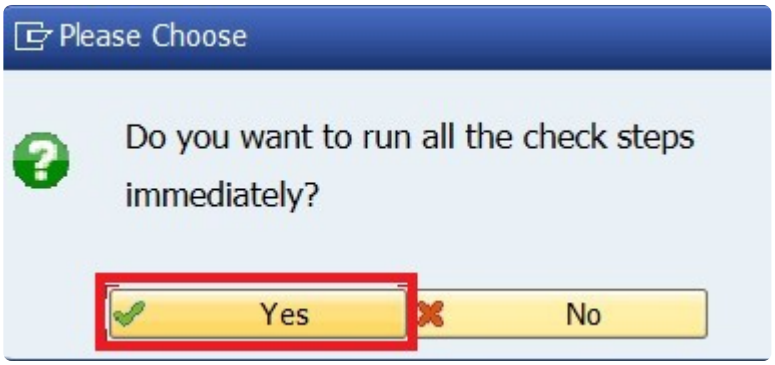
Once the enhancements are activated, you can execute the recording Check Steps
Select Recording -> Recording Status from the left-hand navigation pane.



The Recording Status screen will appear on the right. Select the Source System and Click on the Check Steps button and choose “Run check steps”

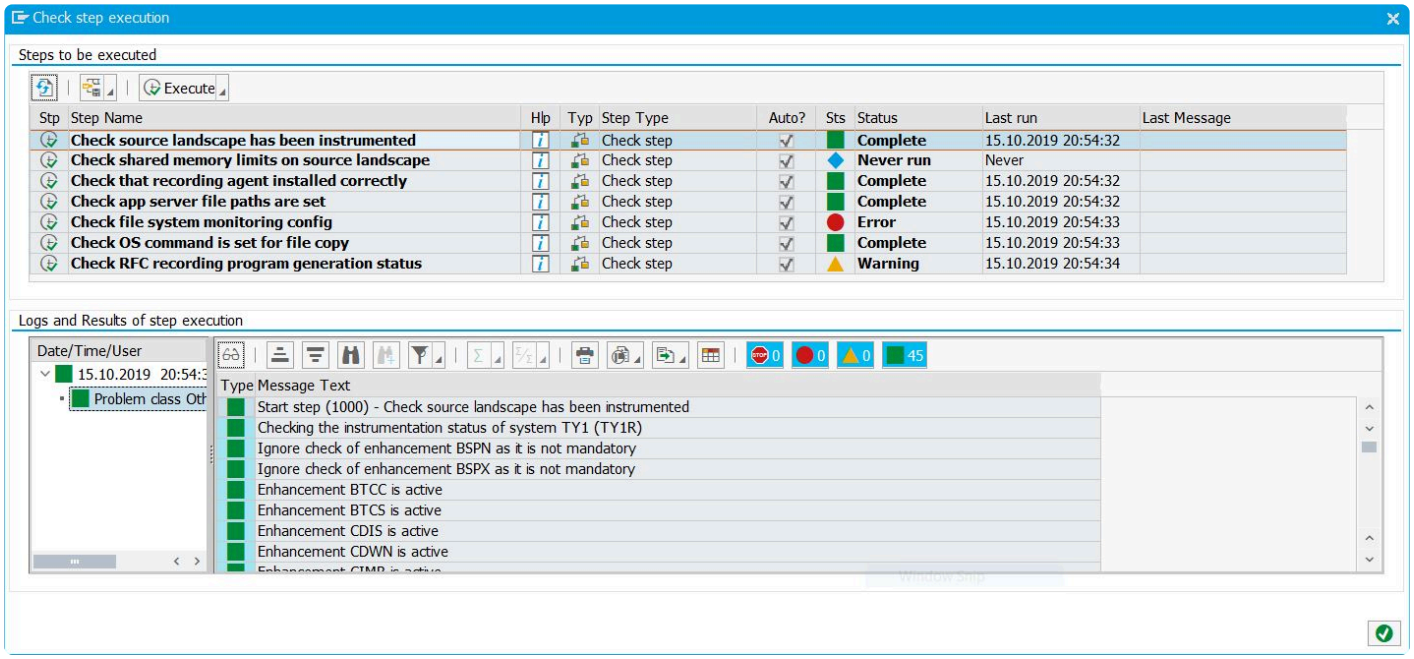


Choose Yes in the following pop-up.

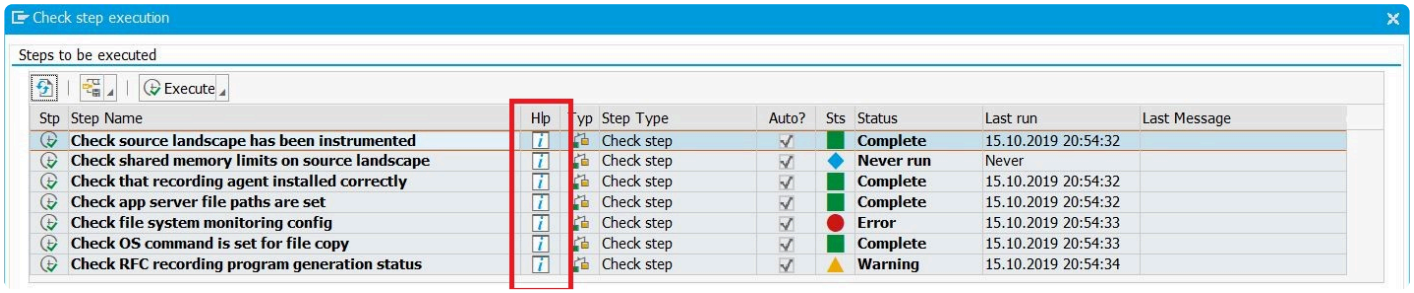


The check steps will run, and results will be returned in a pop-up screen, as below.

To help troubleshoot errors there are logs providing information on the error, double click on the line in error and then double click the log on the bottom left. The example below shows the error received if you do not active the Ehnancement Points in the Source system.



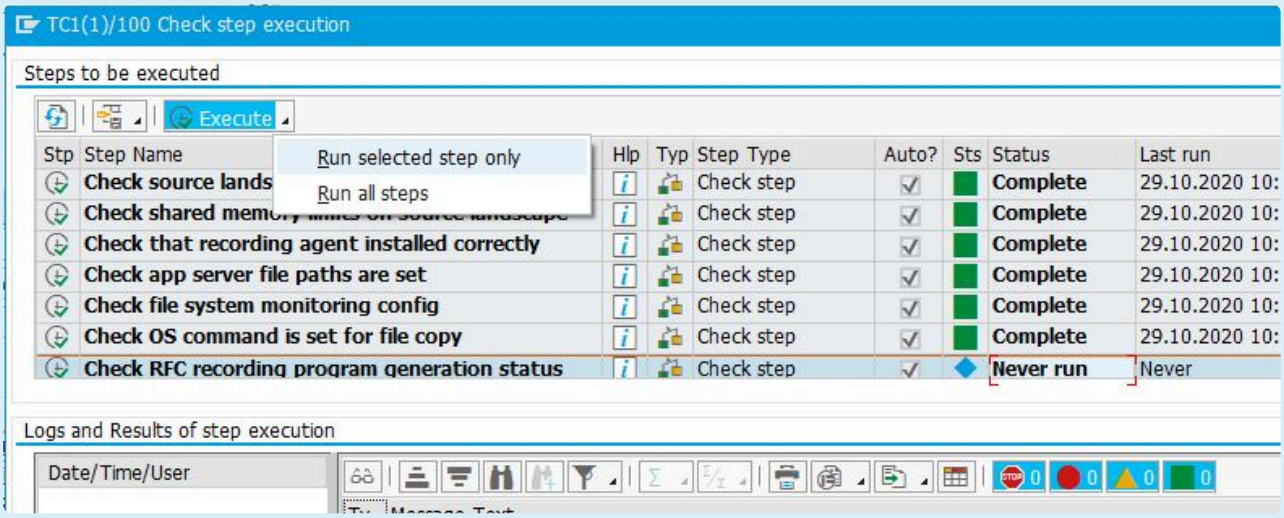
There is also help available to guide you in resolving any Check Step erros. Simply click on the blue help icon on the row in question.



You can also use the buttons below to hide the Information, Warning or Error messages if there are a large volume of messages to check.

Once you have finished analysing errors, close the pop-up screen and you can repeat the Check Steps as needed.

From the Check Step list's pop-up window it is possible to run individual Check Steps


TC1(1)/100 Check step execution
Steps to be executed
Execute
Run selected step only
Run all steps
Stp Step Name Hlp Typ Step Type Auto? Sts Status Last run
Check source lands Check step Complete 29.10.2020 10:
Check shared memory limits on source landscape Check step Complete 29.10.2020 10:
Check that recording agent installed correctly Check step Complete 29.10.2020 10:
Check app server file paths are set Check step Complete 29.10.2020 10:
Check file system monitoring config Check step Complete 29.10.2020 10:
Check OS command is set for file copy Check step Complete 29.10.2020 10:
Check RFC recording program generation status Check step Never run Never
Logs and Results of step execution
Date/Time/User

3.7.1.1. Settings for DB writes

How Testimony stores recorded data

When a recording is running, Testimony is capturing information about dialog steps, batch jobs, etc. The data Testimony captures includes inputs, outputs and technical information required to enable Testimony to playback. In order to avoid (especially online) performance degradation while a recording is running, this data is initially stored in shared memory on the application server. (Writes to shared memory typically take 1-3 microseconds, whereas writes to the DB take 7-10 milliseconds.) A Testimony batch job (which is automatically started when you start the recording) then periodically checks each application server and writes the data saved in memory to Testimony tables on the database.

Bypassing the shared memory

There may occasionally be situations where, in between one execution of the batch job and the next, the shared memory area into which Testimony stores its data fills up. Since this memory area is part of shared memory (and so is used by other SAP processes), if Testimony detects that this has happened it bypasses the shared memory and, until the memory is cleared down by the next execution of the batch job, writes directly to the database. However, while this is happening there may be a small performance impact on transactions that are being recorded (as the DB write time is effectively added to the overall response time of the transactions). In order to ensure that this situation does not persist for too long, Testimony will automatically abort the recording if the number of direct DB writes (i.e., writes of Testimony data that bypass the buffer), exceeds a certain number within one minute. This value is configurable via the General Parameter ALLOWED_DB_WRITES.

General parameters

Type	Parameter type	Technical name	Parameter description	Help	Current Value	Default Value
▶	Playback	FAILED_SCRIPT_ACTION	Action when a script fails		02	01
▶	Playback	FORCE_GUI_WIN_SIZE	Force SAP GUI main window size		X	X
▶	Playback	UPD_TASK_TIMEOUT	Max. time to wait for upd. tasks (secs)		60	60
▶	Playback	ACC_FACTOR_REF_TIME	Acceleration factor refresh time		300	300
▶	Playback	LOCK_ATTEMPTS	Lock Virtualisation Attempts		10	10
▶	Playback	LOCK_DEPNCY_CANCEL	Cancel item with Lock dependency fails		1	1
▶	Playback	LOCK_WAIT	Lock Virtualisation Wait Time		2	2
▶	Playback	SHORT_DUMP	Short Dump Time Difference		600	600
▶	Playback	SEQ_STRATEGY	Sequencing strategy		02	02
●	Recording	STAD_SUBPERIOD	STAD data period breakdown (secs)		60	60
●	Recording	STAD_PERIOD	STAD data recording period (mins)		60	60
●	Recording	STAD_ACTIVE	STAD data recording activation		X	X
●	Recording	RFC_DATA_SIZE_LIMIT	Inbound/Outbound RFC data size limit(MB)		50	50
●	Recording	RFCTAB_CMP_THRESHOLD	RFC TABLES param. compression threshold		5000	5000
●	Recording	REMOTE_USER_CHECK	Remote user check during transfer to rep			
●	Recording	RELDATA_WAIT	Rel. data periodic save wait (m/s)		500	500
●	Recording	RELDATA_LOCKS	Related data read lock on saves		X	X
●	Recording	RECORD_WAIT_STRTUP	Force recording job to wait on startup			
●	Recording	RECORDING_TRACE	Recording process and timing logging			
●	Recording	OUTBOUND_RFC	Activate Outbound RFC functionality			
●	Recording	ORFC_USER_PASSWORD	Outbound RFC default password for user		**********	**********
●	Recording	ORFC_USER_GROUP	Outbound RFC default user group			
●	Recording	ORFC_DB_SELECTBYPASS	Outbound RFC DB Select bypass			
●	Recording	OPMODE_PERIOD	Operation mode recording period (mins)		60	60
●	Recording	OPMODE_ACTIVE	Operation mode rec/playback activation			
●	Recording	IGNORE_US_WO_LOGON	Ignore user sessions without Log-on			X
●	Recording	IGNORE_FAULTY_SCRIPT	Ignore faulty scripts (transfer to repo)			
●	Recording	IDOC_MAX_TTL_SEG	Maximum IDoc segments to capture		1000	1000
●	Recording	IDOC_MAX_REC_SEG	IDoc segment threshold for SHM storage		100	100
●	Recording	FIORI_BETA	Enable Fiori beta functionality			
●	Recording	CUSTOM_STR_TO_TAB_FM	Use Custom String to Table FM?		X	X
●	Recording	CREATE_START_TR	Derive start. tr. scripts (ignore rec.)			
●	Recording	BUTID_SAVE	Business Transaction Save Logic		DEFAULT	DEFAULT
●	Recording	ALLOWED_DB_WRITES	Allowed DB writes (/minute) before abort		10000	1000

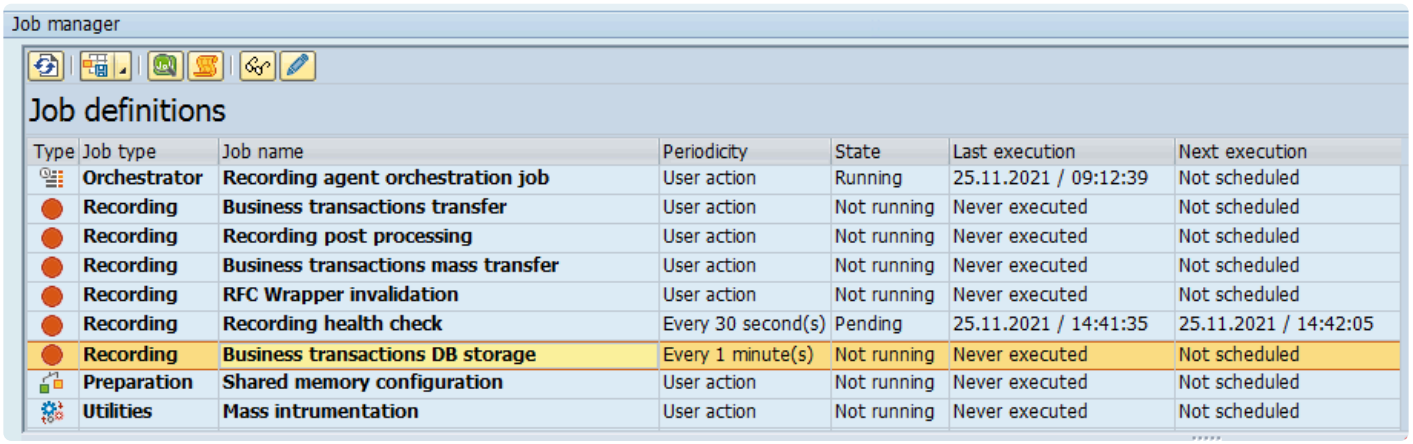
In the image above (accessed in the Testimony transaction by choosing Configuration —> General Parameters) you can see that the value for ALLOWED_DB_WRITES has been changed to 10,000 from its default of 1,000.

Clearing the shared memory via the batch job

As mentioned above, a batch job is automatically started when you begin a recording to save the data stored in shared memory to the DB. When it does this, it also deletes the data from the shared memory area in order to free up memory.

By default, the batch job checks each application server every minute. It is often useful – especially in busy systems – to change the frequency of this job to run continuously. When this is done, it will constantly monitor the shared memory on each application server (using a round-robin approach so it saves the data for each application server in turn).

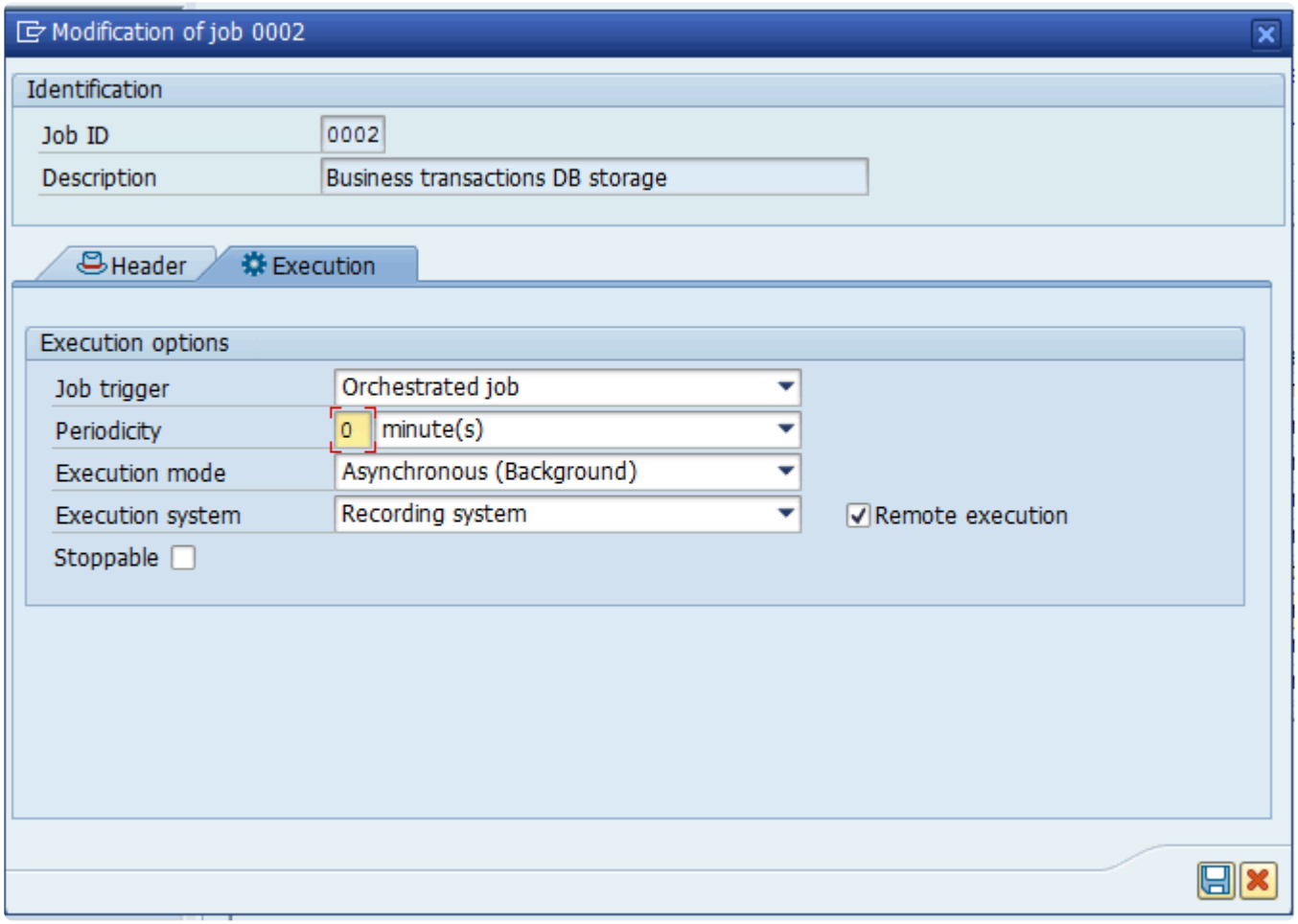
To change the frequency of the batch job go to Recording —> Job Manager in the Testimony transaction on the Central System.



Type	Job type	Job name	Periodicity	State	Last execution	Next execution
	Orchestrator	Recording agent orchestration job	User action	Running	25.11.2021 / 09:12:39	Not scheduled
	Recording	Business transactions transfer	User action	Not running	Never executed	Not scheduled
	Recording	Recording post processing	User action	Not running	Never executed	Not scheduled
	Recording	Business transactions mass transfer	User action	Not running	Never executed	Not scheduled
	Recording	RFC Wrapper invalidation	User action	Not running	Never executed	Not scheduled
	Recording	Recording health check	Every 30 second(s)	Pending	25.11.2021 / 14:41:35	25.11.2021 / 14:42:05
	Recording	Business transactions DB storage	Every 1 minute(s)	Not running	Never executed	Not scheduled
	Preparation	Shared memory configuration	User action	Not running	Never executed	Not scheduled
	Utilities	Mass intrumentation	User action	Not running	Never executed	Not scheduled

Here you can see that the “Business transactions DB storage” job has the default frequency of “Every 1 minute(s)”.

To change this, highlight the row and click on the change button. Then go to the Execution tab and delete the number 1 from the Periodicity field.



Click on Save and the periodicity of the job will be changed so that it constantly monitors and clears down the shared memory.

3.7.1.2. Reviewing shared memory parameters

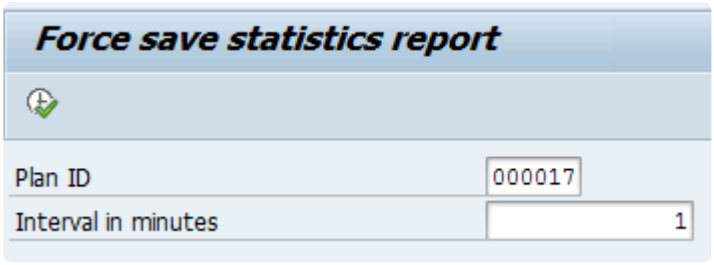
Overview

During the installation and initial configuration of Testimony, you will have set instance profile parameters on the source system. Once you have performed at least one large recordings in your production system, it is useful to review these settings to see if they need to be increased.

The Force Save Statistics Report

The program /BTI/AUT_FORCE_SAVED_STAT allows you to review a previous recording to see if and when the recording was forced to write directly to the DB rather than to shared memory.

Execute the program in SE38/SA38 on the Central System.



In Plan ID, enter the test plan ID for the recording you are analysing. (You can get this information by displaying the test plan in Configuration —> Test Plans.)
In the “Interval in minutes” field, enter a value of 1. This will show a minute-by-minute breakdown the recording’s forced DB writes.
When you execute the report, you will see output similar to this:

Force save statistics report

Period	Start Date	Time (UTC)	End Date	Time (UTC)	Count
1.183	20.09.2021	07:53:19	20.09.2021	07:54:20	0
1.184	20.09.2021	07:54:20	20.09.2021	07:55:20	0
1.185	20.09.2021	07:55:20	20.09.2021	07:56:21	0
1.186	20.09.2021	07:56:21	20.09.2021	07:57:21	960
1.187	20.09.2021	07:57:21	20.09.2021	07:58:22	795
1.188	20.09.2021	07:58:22	20.09.2021	07:59:22	0
1.189	20.09.2021	07:59:22	20.09.2021	08:00:23	0
1.190	20.09.2021	08:00:23	20.09.2021	08:01:23	0
1.191	20.09.2021	08:01:23	20.09.2021	08:02:24	0
1.192	20.09.2021	08:02:24	20.09.2021	08:03:24	0
1.193	20.09.2021	08:03:24	20.09.2021	08:04:25	0
1.194	20.09.2021	08:04:25	20.09.2021	08:05:25	0
1.195	20.09.2021	08:05:25	20.09.2021	08:06:26	0
1.196	20.09.2021	08:06:26	20.09.2021	08:07:26	0
1.197	20.09.2021	08:07:26	20.09.2021	08:08:27	0
1.198	20.09.2021	08:08:27	20.09.2021	08:09:27	0
1.199	20.09.2021	08:09:27	20.09.2021	08:10:28	0
1.200	20.09.2021	08:10:28	20.09.2021	08:11:28	0
1.201	20.09.2021	08:11:28	20.09.2021	08:12:29	39
1.202	20.09.2021	08:12:29	20.09.2021	08:13:29	97
1.203	20.09.2021	08:13:29	20.09.2021	08:14:30	0
1.204	20.09.2021	08:14:30	20.09.2021	08:15:30	3
1.205	20.09.2021	08:15:30	20.09.2021	08:16:31	103
1.206	20.09.2021	08:16:31	20.09.2021	08:17:31	0
1.207	20.09.2021	08:17:31	20.09.2021	08:18:32	0
1.208	20.09.2021	08:18:32	20.09.2021	08:19:32	24
1.209	20.09.2021	08:19:32	20.09.2021	08:20:33	0
1.210	20.09.2021	08:20:33	20.09.2021	08:21:33	0
1.211	20.09.2021	08:21:33	20.09.2021	08:22:34	0
1.212	20.09.2021	08:22:34	20.09.2021	08:23:34	1.044
1.213	20.09.2021	08:23:34	20.09.2021	08:24:35	0
1.214	20.09.2021	08:24:35	20.09.2021	08:25:36	0
1.215	20.09.2021	08:25:36	20.09.2021	08:26:36	2.053
1.216	20.09.2021	08:26:36	20.09.2021	08:27:37	2.869
1.217	20.09.2021	08:27:37	20.09.2021	08:28:37	514
1.218	20.09.2021	08:28:37	20.09.2021	08:29:38	0
1.219	20.09.2021	08:29:38	20.09.2021	08:30:38	8
1.220	20.09.2021	08:30:38	20.09.2021	08:31:39	1.036
1.221	20.09.2021	08:31:39	20.09.2021	08:32:39	0
1.222	20.09.2021	08:32:39	20.09.2021	08:33:40	0
1.223	20.09.2021	08:33:40	20.09.2021	08:34:40	502
1.224	20.09.2021	08:34:40	20.09.2021	08:35:41	551
1.225	20.09.2021	08:35:41	20.09.2021	08:36:41	625
1.226	20.09.2021	08:36:41	20.09.2021	08:37:42	320
1.227	20.09.2021	08:37:42	20.09.2021	08:38:42	2
1.228	20.09.2021	08:38:42	20.09.2021	08:39:43	608
1.229	20.09.2021	08:39:43	20.09.2021	08:40:43	2.587
1.230	20.09.2021	08:40:43	20.09.2021	08:40:53	347

In the image above, you can see that there were several one-minute periods where the number of forced DB writes exceeded the default ALLOWED_DB_WRITES threshold of 1,000. In the case of this recording, the threshold had been set higher so the recording did not abort. However, it does show that there were frequent periods when the shared memory was full, so increasing the relevant shared memory parameters is advisable.

3.7.1.3. Changing optional/mandatory enhancements

Introduction

When Testimony is first installed, it comes with a set of recording enhancements which must be activated before starting a recording. While many of these are delivered as mandatory enhancements, some are delivered as optional enhancements because they won't be relevant for all customers.

Enhancement points

Type	Type	Enhancement description	Include name	Type	Reason	Type	Status	Lck	Locked?	Rel	Rel. Status	Mnd	Mandatory?
	GUI dialog step	**List command**	ZBTI_AUT_ENH_LCOM		Recording		**Inactive**		Unlocked		Released		Mandatory
	GUI dialog step	**Enqueue / Dequeue 01**	ZBTI_AUT_ENH_LK1R		Recording		**Inactive**		Unlocked		Released		Optional
	GUI dialog step	**Enqueue / Dequeue 02**	ZBTI_AUT_ENH_LK2R		Recording		**Inactive**		Unlocked		Released		Optional
	GUI logon	**SAP GUI logon**	ZBTI_AUT_ENH_LOGO		Recording		**Inactive**		Unlocked		Released		Mandatory
	GUI dialog step	**List PBO**	ZBTI_AUT_ENH_LOUT		Recording		**Inactive**		Unlocked		Released		Mandatory
	GUI dialog step	**Recording code for NUMBER_GET...**	ZBTI_AUT_ENH_NGER		Recording		**Inactive**		Unlocked		Released		Mandatory

If there are areas of functionality that are important to you which are covered by optional enhancements, then rather than have to individually activate them you can change the configuration so that these are marked as mandatory.

Updating enhancement configuration

Enhancement configuration is updated via table /BTI/AUT_C_ENHT, which can be edited using transaction SM30. In the table, optional enhancements have a blank entry in the "Mand" Column. Change this entry to Mandatory for the enhancements you want to change.

Enhancements configuration

Enhan. ID	Rel Sts	Mand	Main pr
LCOM	Released	Mandatory	SAPLSY
LK1P	Released		SAPLSE
LK1R	Released	Mandatory	SAPLSE
LK2P	Released		SAPLSE
LK2R	Released	Mandatory	SAPLSE
LOGO	Released	Mandatory	SAPLSU

> ! Make sure that you are changing the setting for the correct enhancement. Pay particular attention to the type of enhancement (Recording or Playback) and the name of the enhancement.

3.7.1.4. Setting up file system monitoring

Introduction

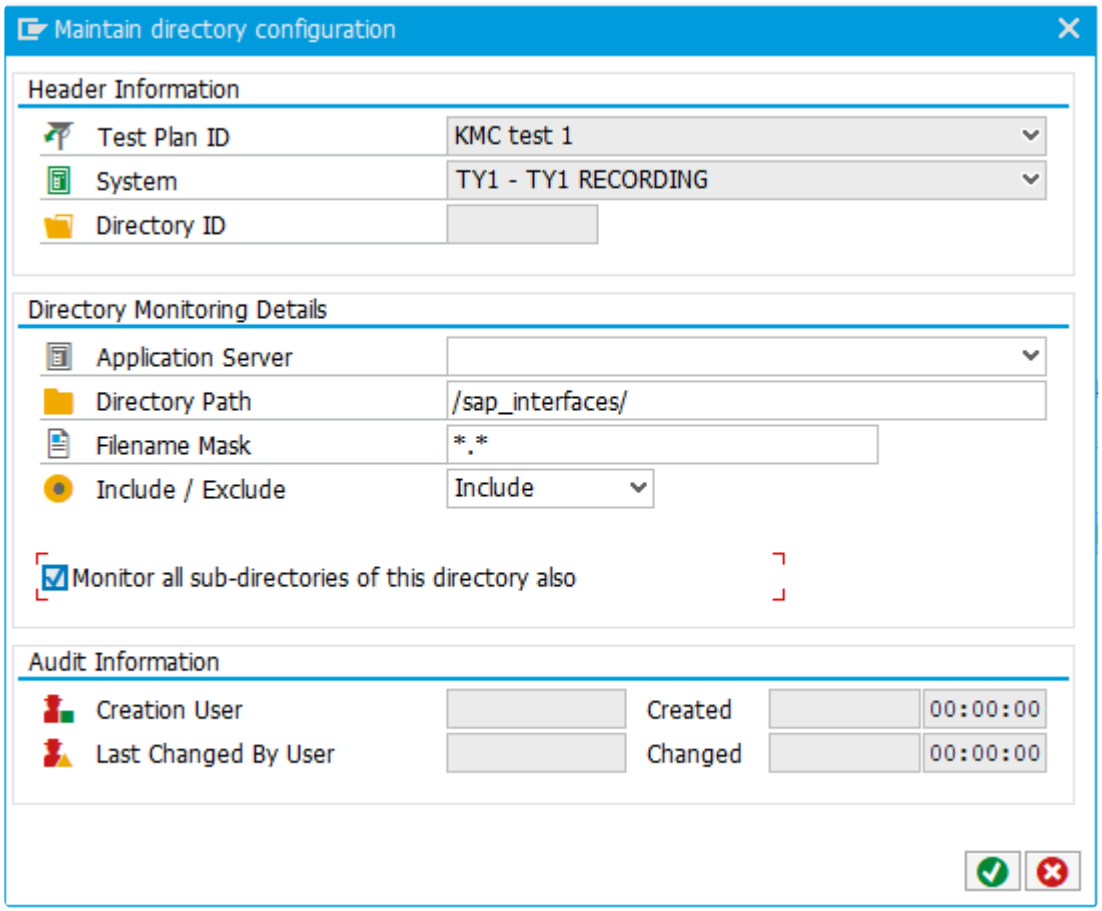
Many processes in SAP – especially batch processes – read from external filesystems. For example, you may have an external sales system in which sales orders are created. This external system exports new sales orders to a file which is then uploaded to SAP for the creation of sales order documents. It is obviously important that Testimony is able to capture these file accesses – and the files themselves – so that these processes can be played back.

File system monitoring configuration

In order for Testimony to be able to capture the files, we need to tell Testimony which filesystem directories to monitor for file access. When Testimony detects that a batch job or other process has accessed a file in the monitored directories, it will copy that file to its own directory. This copied file is later copied to the Central System for use in the playback.

> The method and directory locations for the file copy are configured during the installation of Testimony.

To add directories to the File System Monitoring configuration, go to Recording —> File System Monitoring, click on the Configuration tab, double-click on your source system and then click on Create.



Enter the information for your interfaces directories.

- If you have the same directory structure mounted on all application servers, then you can leave the Application Server field blank. This tells Testimony to monitor all application servers
- If you have only specific filenames that you want to monitor, then enter details in the Filename Mask field, otherwise, to monitor all files, leave this field to the default
- If you want Testimony to monitor all subdirectories of the main interfaces directory, then select the “Monitor all sub-directories of this directory also” checkbox

Click on the tick, and the monitored directory will appear in the Configured directories list.

3.7.1.5. Preparing the backup

Introduction

One of the most important parts of the recording/playback cycle is ensuring that you have consistent data between the system being recorded and the system that you will be playing back into. For this reason, one requirement of recording for playback on another system is that you take an online backup of the source system. This will then be restored to build the target system (using standard SAP system copy tools). However, different database and backup tools and technologies have different ways of creating a consistent online backup. It is important that you understand how your particular DB and backup tools work, so that the backup can be taken at the appropriate time, and also so that the restore from the backup can be done in a way that makes a consistent dataset for starting the playback. In particular, it is important that you understand whether the point of consistency is the start of the backup, the end of the backup, or must be defined to a point just after the end of the backup.

Online backup data consistency

There are, in general, three different approaches taken by different databases to achieving a consistent online backup (i.e., a backup which can be restored whilst leaving no half-finished database transactions).

Snapshots

Snapshot tools use disk storage technologies to take a virtual snapshot of the database at a particular point in time. Because they are using functionality directly within the storage, they are very fast at taking the backup. When restoring from a snapshot to a target system, once the copying of the backed up data is done, the database will automatically roll back any DB transactions that were open at the time of the snapshot. This means that, for snapshots, the point of consistency is the ***start*** of the backup.

Online consistent backups

Some DB backup options (for example Oracle's Online Consistent Backup option) will change the way the database operates whilst the backup is in progress. During normal operation, the database writes the after images of any updated DB blocks to a set of files (in Oracle, these are the redo logs). The aim of these files is to allow transactions to be rolled forward following a restore from an online backup. During an online consistent backup, however, the DB will write both the before *and* the after images of the updated blocks to these logs. Once an online consistent backup has been restored, the database can roll back any updates that took place while the backup was in progress. This means that, for online consistent backups, the point of consistency is the ***start*** of the backup.

Other online backups

In most other cases, when restoring from an online backup it is also necessary to apply a set of redo logs (using the example of Oracle) once the backup has been restored. By applying all of the logs that were generated whilst the backup was running, you are able to ensure that all database updates which took place during the backup are replicated on the target system and that the database is consistent. This "point in

time" recovery means that, for most online backups, the point of consistency is either the ***end*** of the backup, or a point ***just after*** the end of the backup.

Coordinating the backup with the start of the recording

Depending on the type of online backup you are using and the point of consistency, you need to schedule the backup and the start of the recording at different relative times. The key to ensuring that you have a consistent dataset between the recording and the start of the playback is that the recording **must** be active at the point of consistency of the backup.

Where the backup's point of consistency is the **start** of the backup, then you must switch on the recording **first** and then start the backup.
Where the backup's point of consistency is the **end** of the backup, then you can start the backup first, but you **must** start the recording before the backup finishes.
If you are using a point-in-time recovery to apply logs to a point after the end of the backup, then the recording **must** be running at the time to which you are going to roll forward after the restore.

3.7.2. Start Recording

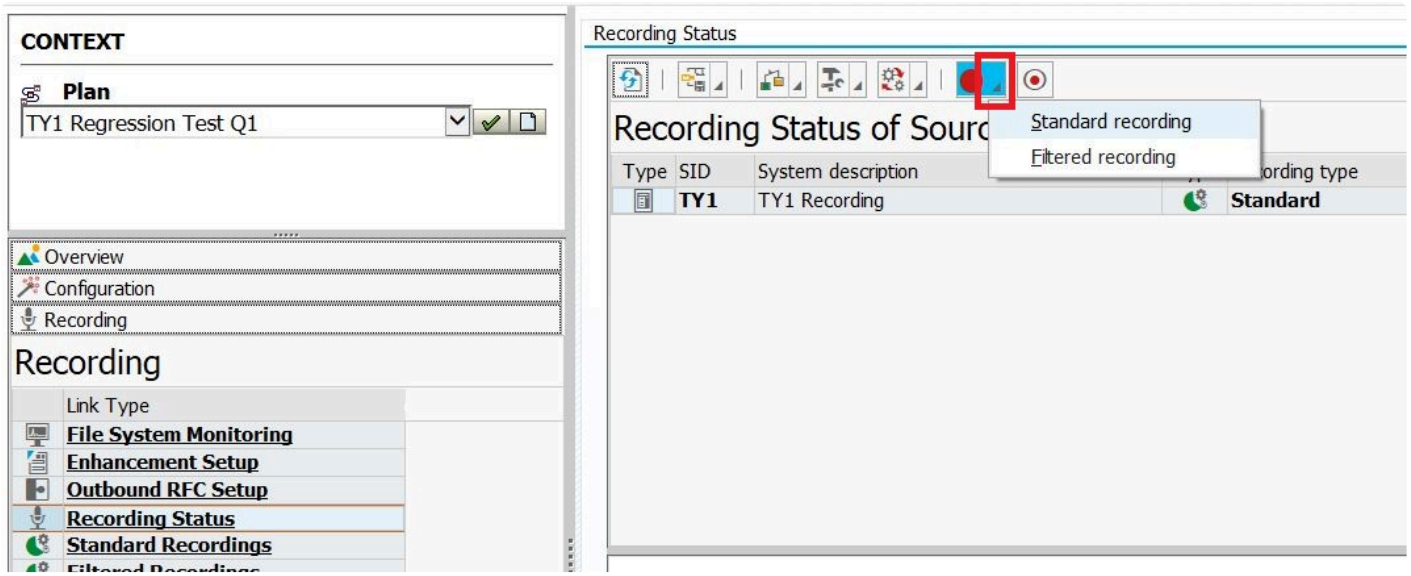
Overview

There are two types of recordings you can perform with Testimony. The most common and the one that will be shown below is a **Standard Recording**. A Standard Recording will record all activity in the Source system until the recording is stopped. The second type of recording is a **Filtered Recording** and is only needed when you want to limit your recording to particular users, transactions or activity types.

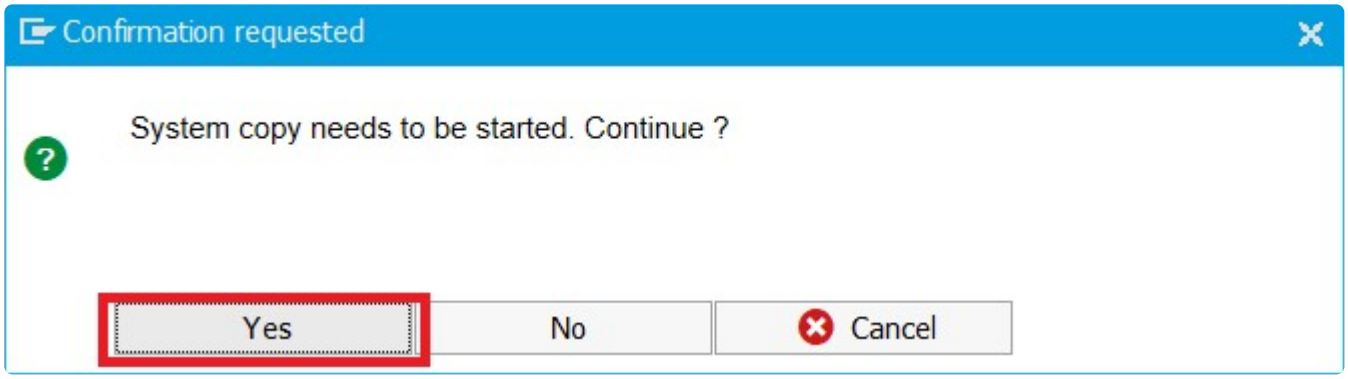
> It's a best practice to time the beginning of your system backup (or snapshot) to be slightly after starting the recording for use in a point-in-time restore of the Playback system. Be sure to coordinate your activities with the technical resources required to create your backup.

Process Steps

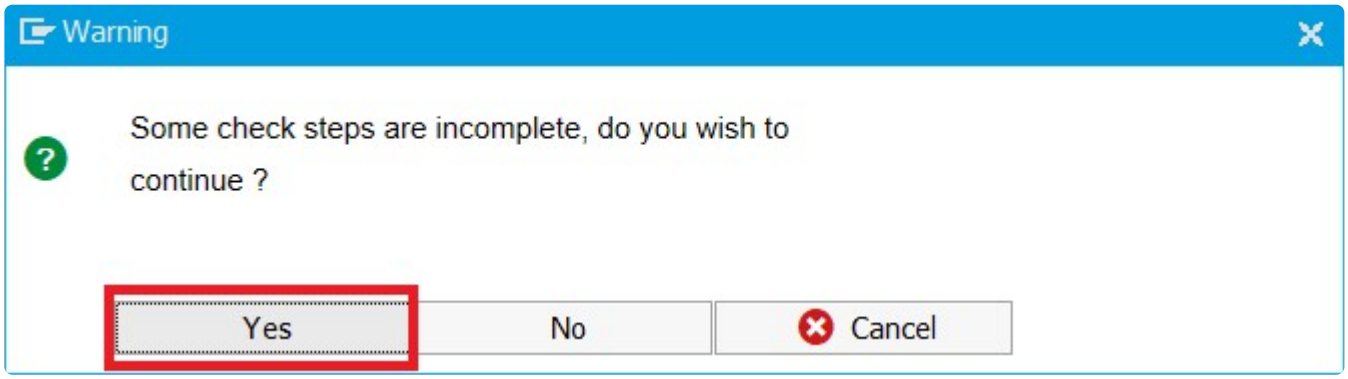
To start a Standard Recording, navigate to the Recording Status option on the Context Menu and then click on the Start Recording button and choose "Standard recording"



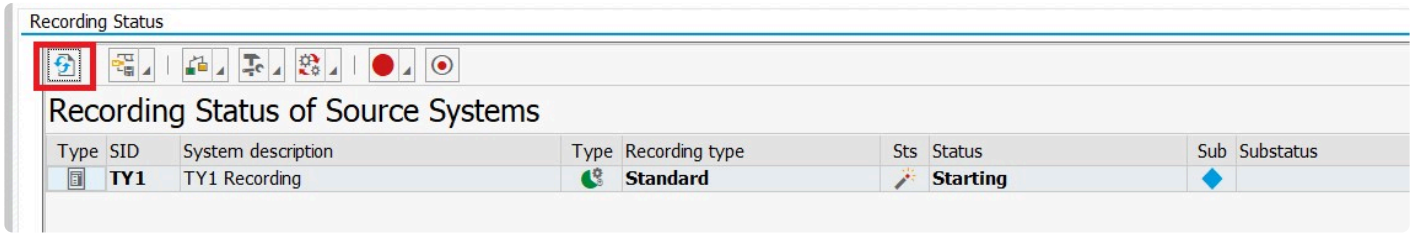
Confirm that you wish to continue at the next pop-up by clicking Yes. (This is just to remind you to initiate the backup from which the playback system will be built.)



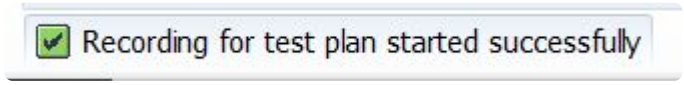
If not all of the check steps completed successfully you will get this pop-up. Click Yes if you've analysed the checks.



At this point, the recording should begin. To check, click on the refresh button until the status shows as Running.



The SAP status bar will also show the following message



Preparation Error

At the start of the recording process, Testimony will run "Preparation Steps". If any of these steps fail, then you will see the recording go into a "Preparation Error" status. In this case, you can check the results of the preparation steps by clicking on the Preparation steps button and choosing "Show steps". This will allow you to navigate to the logs for each step and analyse the cause of the problem.

Once you have analysed and resolved any preparation errors, you can restart the playback following the process above.

3.7.3. Monitoring a Recording

Overview

During the recording phase, which may last for any length of time, it is important whilst you become accustomed to Testimony that you monitor the production systems that are being recorded.

There are various standard SAP tools that can be used to monitor the SAP system for anomalies. Most of these transactions would be run directly in the system to be monitored. These include:

(1) **Monitor Short-dumps (ST22)** – Testimony has been designed to ensure that it has no impact on production. However, in the early days of use of Testimony, you might like to ensure that users and batch jobs are not being impacted by monitoring for any short-dumps. Run transaction ST22 in the systems being recorded and look for any short-dumps directly specifying that Testimony related objects are involved (i.e. /BTI/AUT* related objects). You might like to check this more regularly after recording is turned on (e.g. every 5 minutes), then reduce the frequency as the recording progresses (e.g. every 1 hour after the first hour of recording is complete).

(2) **Monitor Long Running Processes (SM66)** – We recommend the use of SM66 (rather than SM50) since it is across all application servers of the system you are monitoring. In this transaction, you are looking for long running background processes that would usually be fast, this is to check for a batch job that has become long running.

(3) **Monitoring Testimony jobs (/BTI/AUT)** – The main job that runs on the source system is /BTI/AUT_SAVE_BTRAN_PERIODIC. Every minute, it saves data out of the shared memory and into the database and will run for the full length of the recording. The central system also checks this job is running to update the central system and ensure that the RFC connection to the source system is still operating.

(4) **Monitor Shared Memory** – Testimony achieves what it does with near zero impact on production by using shared memory to save the recording before the job above runs to move this to the database. Shared memory is allocated per application server on the system being recorded and is set via the profile parameter **rsdb/esm/buffersize_kb**. As a part of the installation there will be recommendations on the amount of shared memory that will be required.

To monitor shared memory in the Central System the Shared Memory Explorer (more details in the link) should be used. It enables a view of all the app servers on one system as well as the objects contained in each app server's shared memory, it details the memory availability and usage as well as the maximum objects available and their usage.

If the RFC user to the source system does not have the correct authorisations the Shared Memory Explorer view can be restricted. In this case you can use the SAP standard transaction ST02 on the source system to monitor each application server. This transaction only shows you the shared memory of the application server you are currently logged onto. In order to check other application servers, you should use transaction SM51 in order to jump between the application servers and then once switched, run transaction ST02.

You should be checking the row that says “**Exp./ Imp. SHM**”. The column “Alloc. KB” shows how much has been allocated for that particular application server. This should be reflected in the profile parameter setting in transaction RZ11. The current level of shared memory (the amount of free space) is then seen in the following column “**Freesp. KB**”. Other processes in SAP can use up the shared memory (not just Testimony), so this means you’ll see this value typically lower than what was allocated in the profile parameter.

The key task is to ensure that this value does not get close to zero (0KB) if the shared memory limit is reached or no shared memory is left then Testimony will abort the recording.

(5) **System Messages (SM21)** – You can check various system messages via transactions in SM21 whilst the recording is running. Short-dumps would be reflected in here. If there looks to be any anomalies that you do not expect, then investigate appropriately and if required, deactivate the recording.

If any significant impacts are detected (e.g. from a performance or impact perspective), you can deactivate the recording immediately.

(6) **Recording Table Entries** – In the system being recorded (Source), new entries for captured steps will be written into table /BTI/AUT_BUTR at least every 60 seconds as they are flushed from the shared memory. Depending on activity, this may be more frequent. To check that entries are still being written, go into transaction SE16 on the Source system and click ENTER. In the screen that follows, click on the “Nimber of Entries” button and verify that the row count is increasing.

3.7.3.1. Shared Memory Explorer

Overview

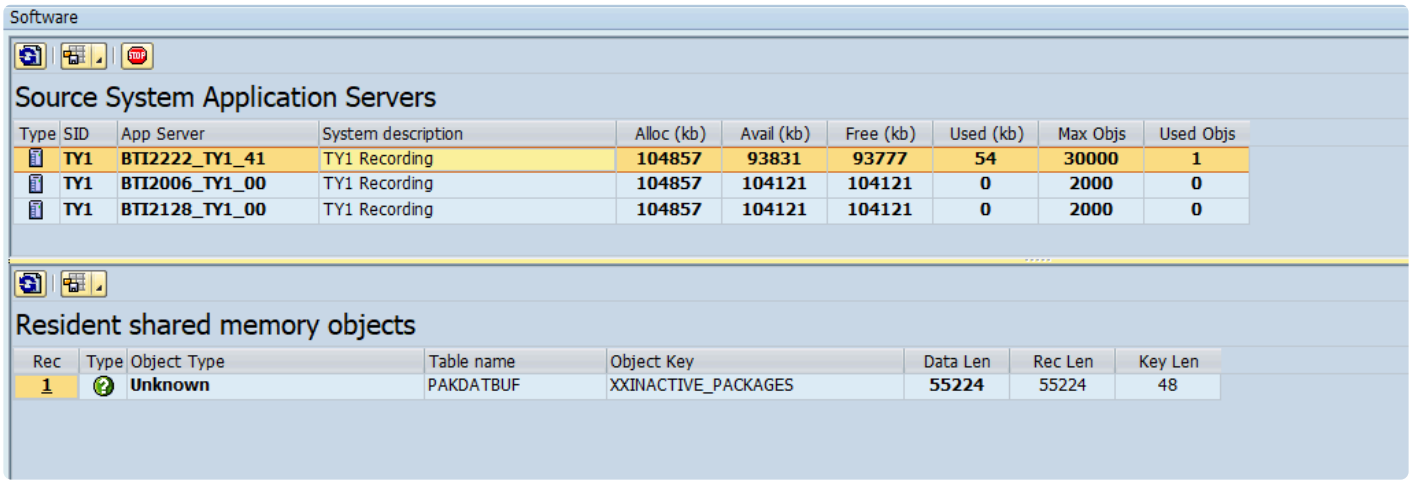
Shared memory is used by Testimony to achieve near zero impact upon the production systems during the recording phase. Therefore, it is important to be able to monitor the shared memory in the system being recorded, This can be achieved in the source system via ST02, however, access to the source system is not always possible. The Shared Memory Explorer solves this issue. The Shared Memory Explorer has the added advantage of being able to see all app servers on the one system as well as the objects contained in each app server's shared memory, It details the memory availability and usage as well as the maximum objects available and their usage. Shared memory parameters would have been checked and adjusted as a part of the installation of Testimony. The recommended settings for the shared memory parameters are here. These might vary based on the usage of the source system.

Monitoring

The Shared Memory Explorer is found under the 'Configuration' tray if using the Administrators UI profile. This is the recommended method for monitoring the shared memory of a system being recorded, primarily because it offers one view of all the application servers on a source system. The usage of shared memory is expected to rise and then fall back every minute as the data is saved to the database. Memory climbing without being cleared out every minute should be investigated firstly before checking the recording job /BTI/AUT_SAVE_BTRAN_PERIODIC is running on the source system and if shared memory continues to climb consider stopping the recording.

Testimony has a fail safe and if the shared memory is exhausted, Testimony will start to save directly to the database. However, the general parameter ALLOWED_DB_WRITES is used to determine how much data can be written directly to the database before the recording aborts. The recommendation for this parameter is 1000 database writes per minute before aborting.

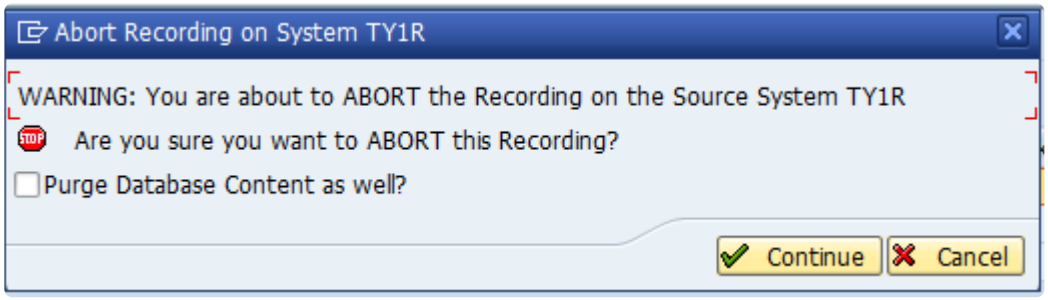
Shared memory is displayed as below. To view the objects on each app server, double click the row. The used objects should not get to within 10% of the max objects and the free memory should not get to below 1000 kb. If they do, you should consider stopping the recording and reviewing if there are large objects you should be excluding or reviewing or if the shared memory settings should be boosted.



Stopping a Recording

> Note you should use the Recording Status tab to stop a recording the stop option here is for emergency use or for the cleanup of shared memory.

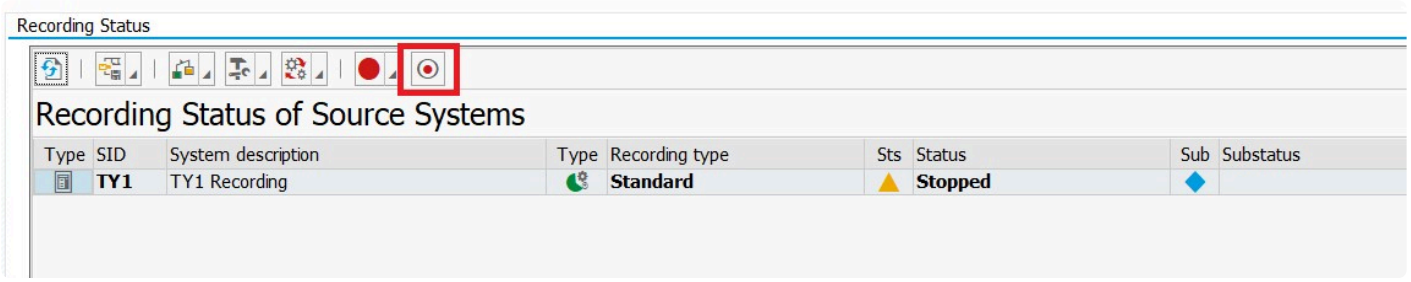
This is an emergency stop, but it also requires the RFC to be operational between the central and source systems. The stop button can be used to abort the recording. Selecting the purge database checkbox option will remove all of the recorded data from the BTI tables and clean up any data in the shared memory of the source system. Note that you won't be able to use the data recorded for a playback if you purge the system of BTI data.



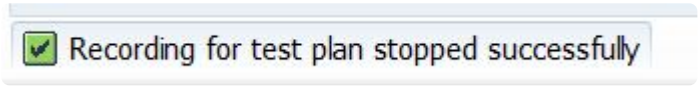
3.7.4. Stop Recording

Once you have executed the recording for the time period you planned, stop the recording as follows.

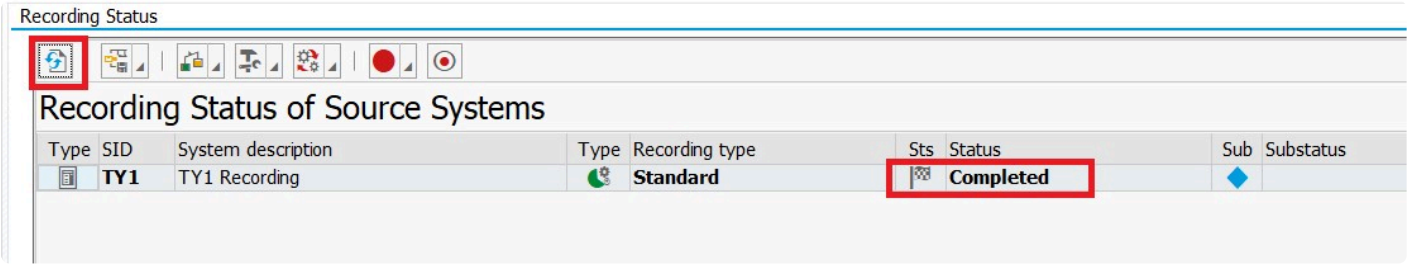
Navigate to Recording -> Recording Status. Double-click on the Recording Status row for your system and you should see your recording running in the Recording history pane.



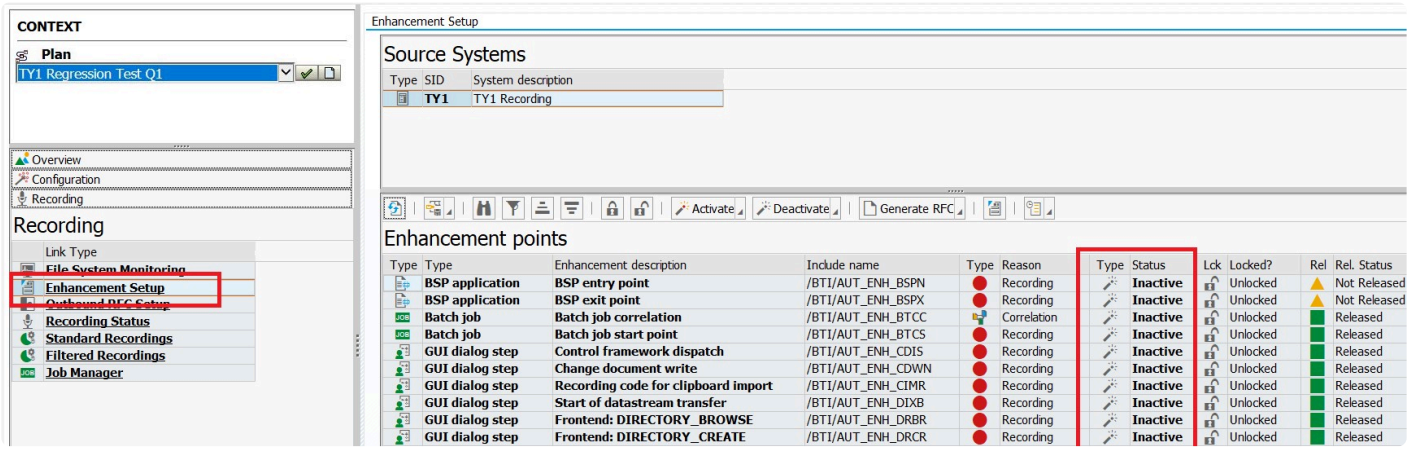
The Recording status should change to **Stopped** and the SAP status bar will also show the following message



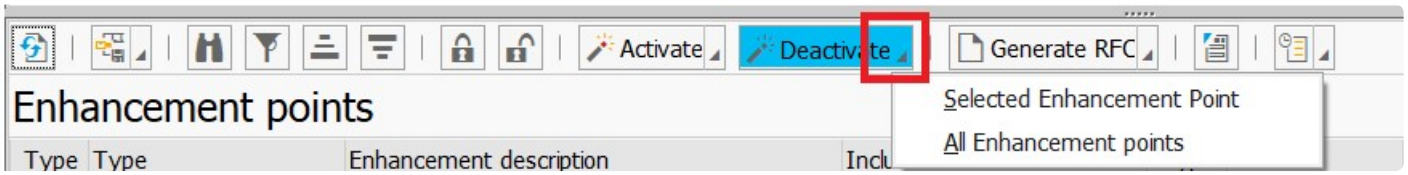
Press the refresh button, and you'll see the status move from Stopped, to Post-Processing, to Data Transfer, and finally to **Completed**. If you executed a long recording in a busy system, then the data transfer step can take some time.



One of the Post-Processing steps is to deactivate the Recording Enhancements on the Source system, but it's always good practice to doublecheck. Navigate to the Enhancement Setup option within Recording and review the Status. You may need to use the refresh button to get the most-up-to-date status of the enhancements.



If any are still Active, click on the Deactive button to turn them off. You can also make use of the transaction /BTI/AUT_DEACT_ENH to deactivate the enhancements.



3.7.5. Aborting a Recording

Overview

When a recording needs to be stopped immediately the abort option can be used. Whilst used infrequently this can be an option to use if you have a concern about the recording on the source system. This option means the recording is not stopped in the usual manner and may require more effort if you want to use this recording to playback.

There are three ways a recording can be aborted:

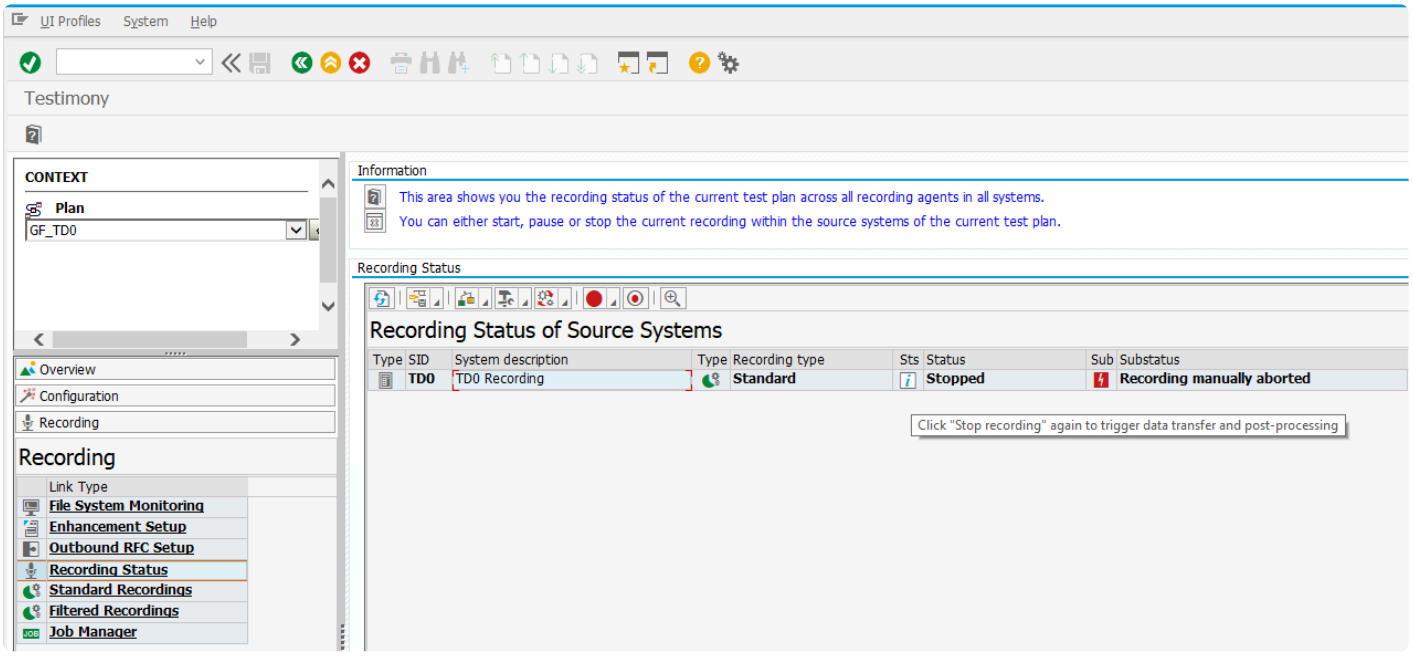
1. Manually via Testimony on the central system using the abort button in the Shared Memory Explorer, more details here.
2. Manually directly on the source system via the transaction /BTI/AUT_ABORT_REC or program /BTI/AUT_ABORT_REC in SE38. This option is helpful if there is an issue with the RFC connection between the central and source system, more details here.
3. Testimony ca also abort automatically when the recording has completely filled the shared memory on an application server and writes directly to the DB more times than the general parameter ALLOWED_DB_WRITES per minutes (normally set as 1000).

All aborts ensure the following:

- The recording switch is turned off on each application server
- The shared memory is cleared down on each application server
- The status on the central system is updated as aborted.

Note the status of the recording appears as below after an abort then to retrieve the data and run the full post processing steps you should use the stop button.

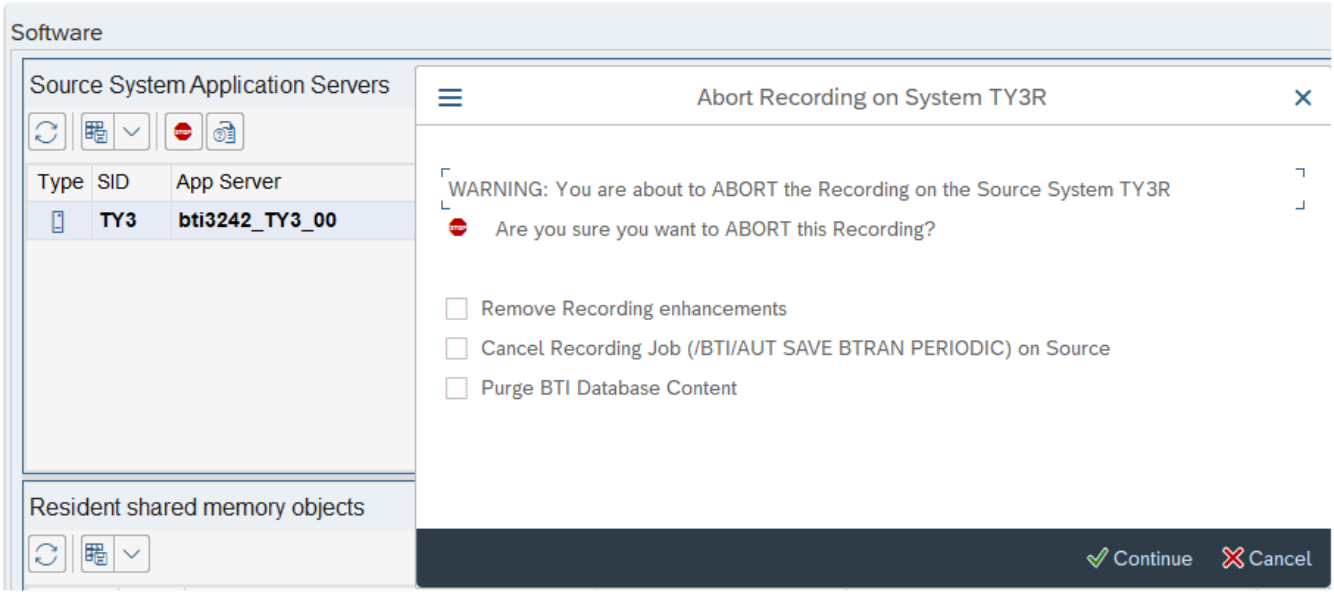
The button  can also be used to analyse how many direct DB writes were performed to enable you to tune to prevent an automatic abort on exceeding the general parameter ALLOWED_DB_WRITES per minutes (normally set as 1000) for your next recording.


UI Profiles
System
Help
Testimony
CONTEXT
Plan
GF_TD0
Overview
Configuration
Recording
Recording
Link Type
File System Monitoring
Enhancement Setup
Outbound RFC Setup
Recording Status
Standard Recordings
Filtered Recordings
Job Manager
Information
This area shows you the recording status of the current test plan across all recording agents in all systems.
You can either start, pause or stop the current recording within the source systems of the current test plan.
Recording Status
Recording Status of Source Systems
Type
SID
System description
Type
Recording type
Sts
Status
Sub
Substatus
TD0
TD0 Recording
Standard
Stopped
Recording manually aborted
Click "Stop recording" again to trigger data transfer and post-processing

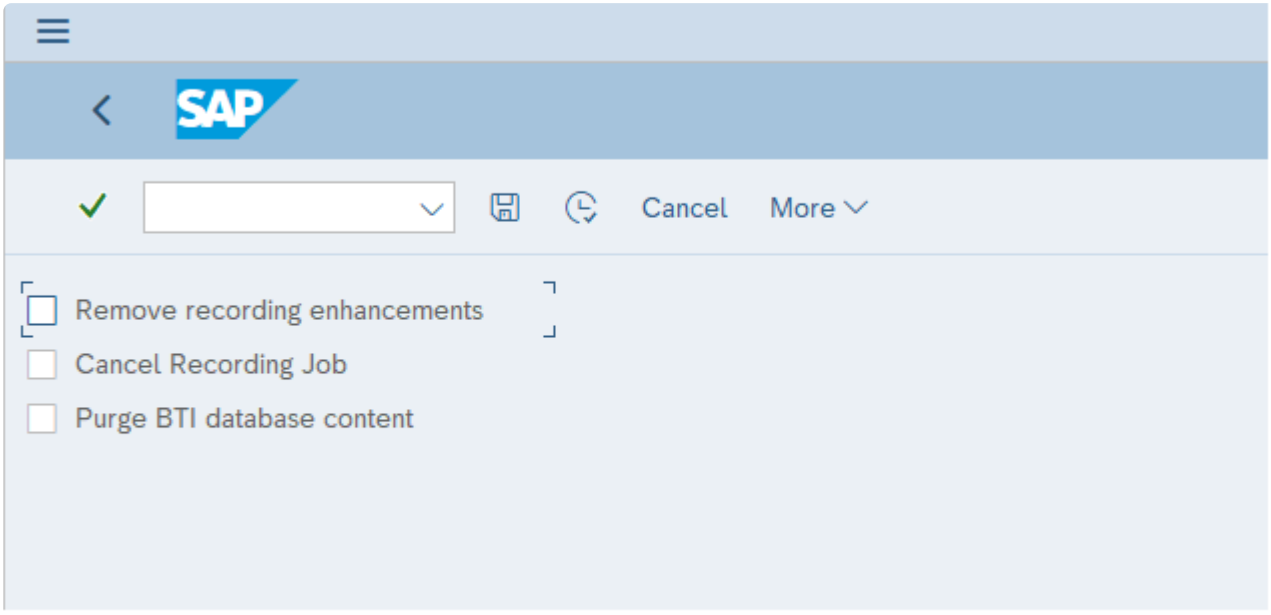
3.7.5.1. Manually Aborting a Recording

A recording can be aborted manually in two ways as below:

1. Manually via Testimony on the central system using the stop button in the Shared Memory Explorer. The popup as below appears with options before you execute the abort of the recording.



2. Manually directly on the source system via the transaction /BTI/AUT_ABORT_REC or program /BTI/AUT_ABORT_REC in SE38. This option is helpful if there is an issue with the RFC connection between the central and source system.



The manual operation of an abort has additional options that can be selected which include:

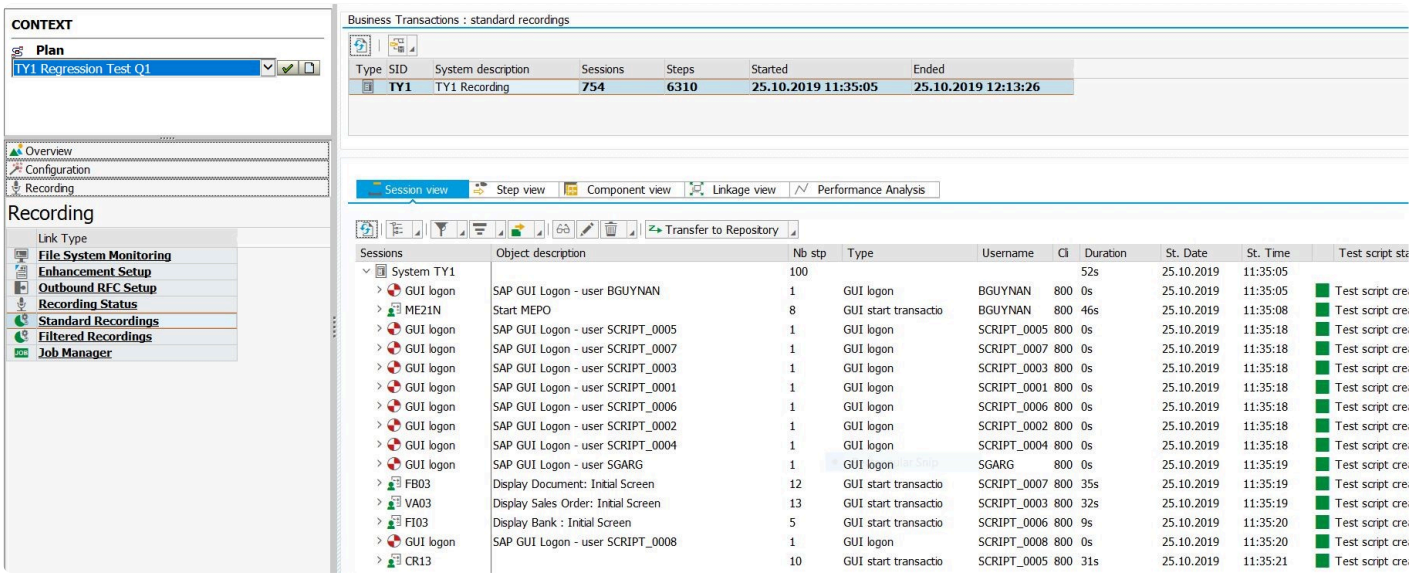
- Making the enhancements inactive (this stops data being recorded into shared memory)
- The recording job saving the data to the DB is cancelled (Job name is /BTI/AUT_SAVE_BTRAN_PERIODIC)
- The /BTI/ DB tables have their data deleted/purged

> ! Note that deleting or purging the data from the /BTI/ tables on the source system means no recorded data can be retrieved into the central system for a later playback

3.8. Review Recordings/Transfer to Repository

Verifying that recording has captured activity

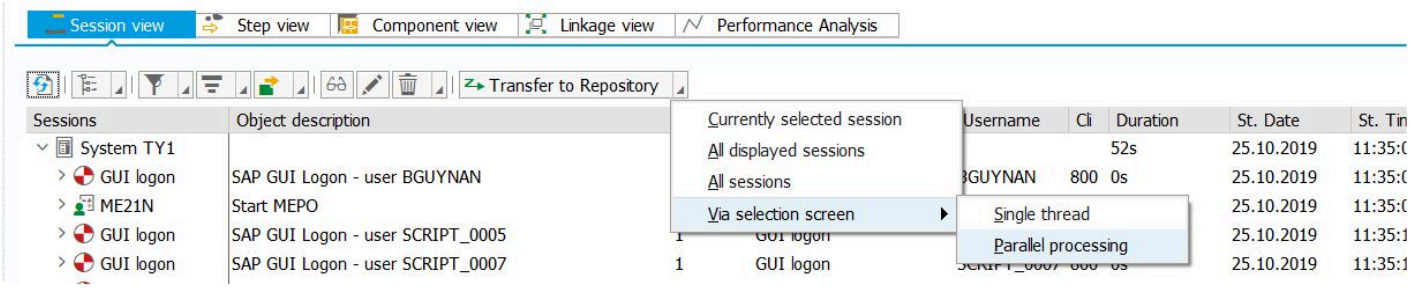
After the recording has been completed, the captured recordings will be moved from the source systems into the central system. You are then able to review what has been captured. To check that your recording has captured activity (and that it was the activity you expected), click on the Standard Recordings or Filtered Recordings item (depending on the type of recording you ran) in the Recordings tray in the left-hand navigation pane.



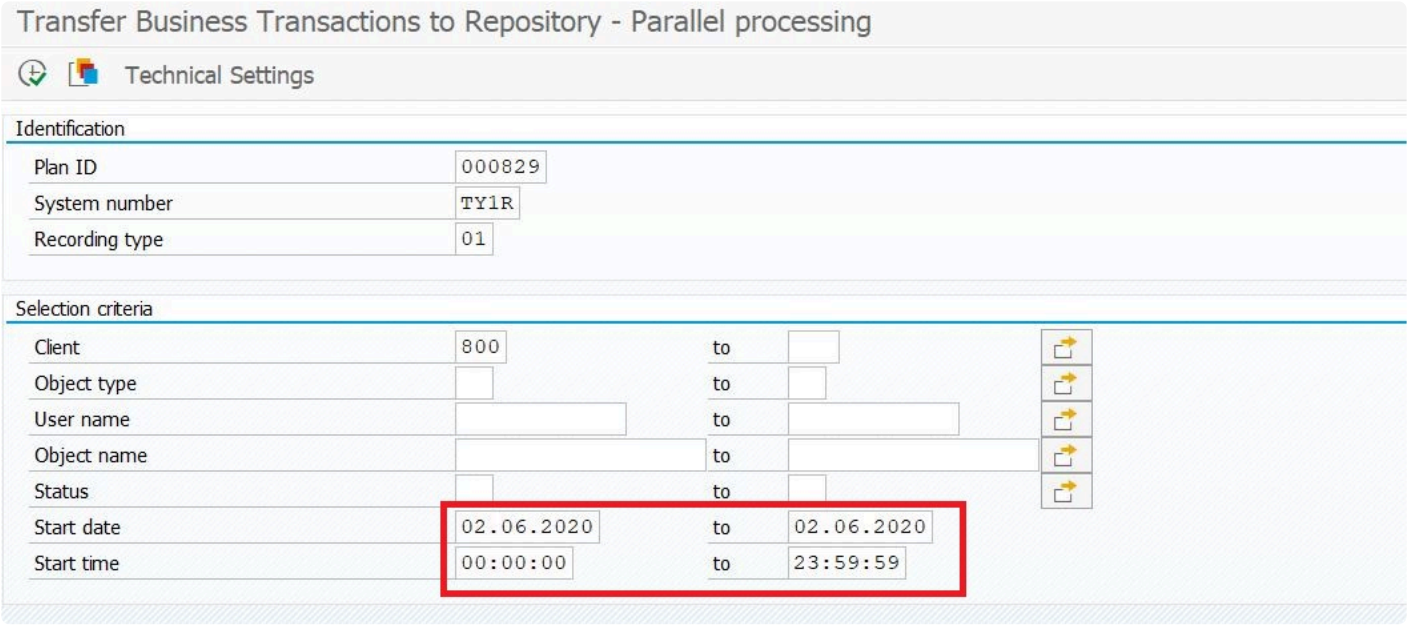
The main thing to validate here is that the Start Date/Time from the recording is what you had planned. It's a best practice to keep your entire recording. You can easily remove individual components later in the process.

Transfer Recording to Repository

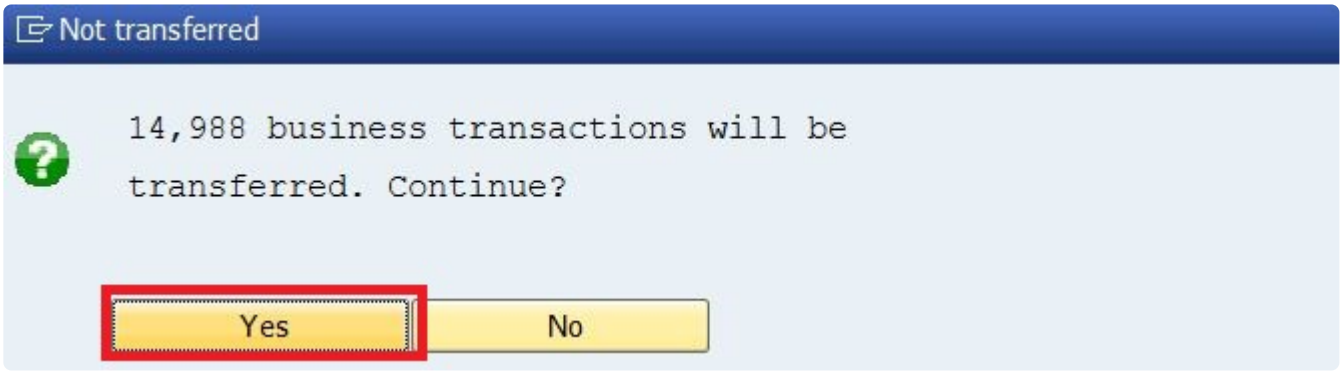
In this step, the "business transactions" that have been generated automatically via the recording process must be first transferred to the test repository. At this stage you ensure that only those recordings are transferred that happened on or after the database snapshot is taken. Dialog/batch processes and user sessions that began prior to the start of the recording will not be considered viable and should not be transferred to the repository. This is performed by choosing the Transfer to Repository drop-down menu and choosing the method for transfer. For large recordings, it is recommended to transfer via selection screen and parallel processing.



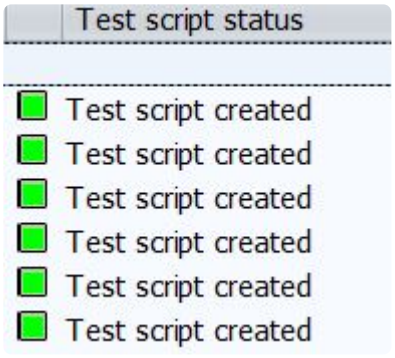
Here is an example where the selection screen is used to select the exact time frame of when the recording and the system backup were in synch to ensure the most accurate recording.



Once you select your transfer option, you'll receive a pop-up box where you can click "Yes" to continue



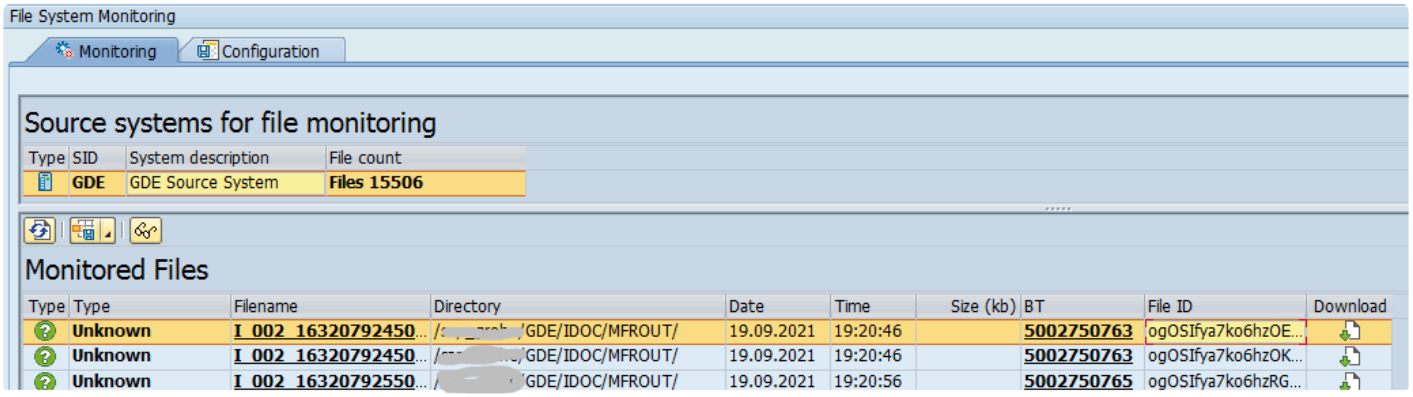
At this point, your recorded scripts are stored in the Testimony Repository and can be used to create an Execution Queue. The Test Script status should show "Test Script Created"


Test script status
Test script created
Test script created
Test script created
Test script created
Test script created
Test script created

3.8.1. Check for application server files

As mentioned in the section on file system monitoring, Testimony will detect for any access to files on the application server during a recording, and copy those files over to the Central System after the recording is complete. You should check to ensure that these files have been successfully brought over to the Central System.

To do this, go to Recording —> File System Monitoring. On the monitoring tab, you can see how many files have been brought across from the source system. By double-clicking on the source system, you can also display details of all of the captured files, including their names and the directory (in the source system) from which they were copied.



3.8.2. The Recording Performance Analysis

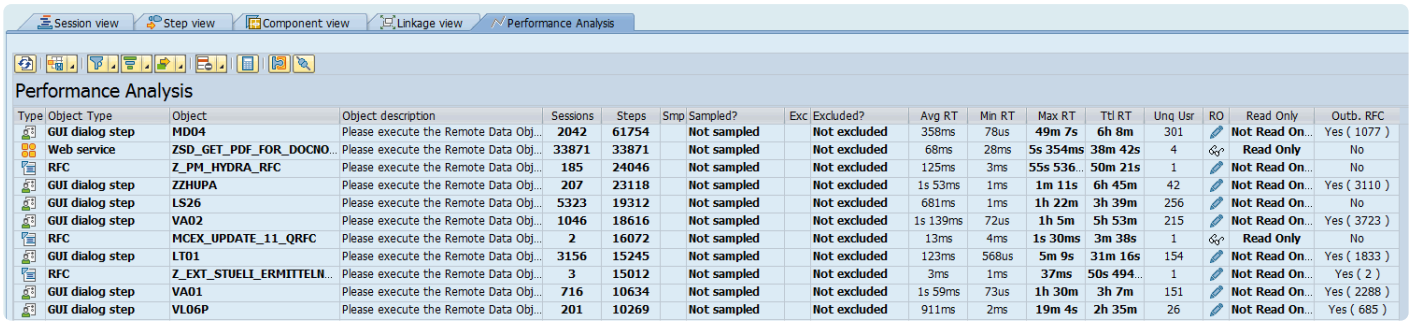
Introduction

One of the keys to the success of any playback is ensuring that you only play back what you really need. This ensures not only that the playback results are relevant, but also that the duration of the playback is not unnecessarily elongated through having to play back unnecessary transactions. The Recording Performance Analysis is a very useful tool to enable you to “trim” the playback of unwanted transactions.

The Performance Analysis Screen

From Recording —> Standard Recordings, go to the Performance Analysis tab and you will be presented with a screen showing relevant information for each object (dialog transaction, batch job, RFC, etc.) that was recorded.

> It is very useful to sort this screen in descending order of steps so that you get information on the most heavily-used objects first.



Type	Object Type	Object	Object description	Sessions	Steps	Smp	Sampled?	Exc	Excluded?	Avg RT	Min RT	Max RT	Ttl RT	Unq Usr	RO	Read Only	Outb. RFC
	GUI dialog step	MD04	Please execute the Remote Data Obj...	2042	61754		Not sampled		Not excluded	358ms	78us	49m 7s	6h 8m	301		Not Read On...	Yes (1077)
	Web service	ZSD_GET_PDF_FOR_DOCNO...	Please execute the Remote Data Obj...	33871	33871		Not sampled		Not excluded	68ms	28ms	5s 354ms	38m 42s	4		Read Only	No
	RFC	Z_PM_HYDRA_RFC	Please execute the Remote Data Obj...	185	24046		Not sampled		Not excluded	125ms	3ms	55s 536...	50m 21s	1		Not Read On...	No
	GUI dialog step	ZZHUPA	Please execute the Remote Data Obj...	207	23118		Not sampled		Not excluded	1s 53ms	1ms	1m 11s	6h 45m	42		Not Read On...	Yes (3110)
	GUI dialog step	LS26	Please execute the Remote Data Obj...	5323	19312		Not sampled		Not excluded	681ms	1ms	1h 22m	3h 39m	256		Not Read On...	No
	GUI dialog step	VA02	Please execute the Remote Data Obj...	1046	18616		Not sampled		Not excluded	1s 139ms	72us	1h 5m	5h 53m	215		Not Read On...	Yes (3723)
	RFC	MCEX_UPDATE_11_QRFC	Please execute the Remote Data Obj...	2	16072		Not sampled		Not excluded	13ms	4ms	1s 30ms	3m 38s	1		Read Only	No
	GUI dialog step	LT01	Please execute the Remote Data Obj...	3156	15245		Not sampled		Not excluded	123ms	568us	5m 9s	31m 16s	154		Not Read On...	Yes (1833)
	RFC	Z_EXT_STUELI_ERMITTELN...	Please execute the Remote Data Obj...	3	15012		Not sampled		Not excluded	3ms	1ms	37ms	50s 494...	1		Not Read On...	Yes (2)
	GUI dialog step	VA01	Please execute the Remote Data Obj...	716	10634		Not sampled		Not excluded	1s 59ms	73us	1h 30m	3h 7m	151		Not Read On...	Yes (2288)
	GUI dialog step	VL06P	Please execute the Remote Data Obj...	201	10269		Not sampled		Not excluded	911ms	2ms	19m 4s	2h 35m	26		Not Read On...	Yes (685)

For each object, the following useful information is shown:

- The number of sessions and steps that called this object
- The current sampling and exclusion status of the object
- Response time (average, min, max, total) information for the object during the recording
- How many unique users called the object during the recording
- Whether or not the object was read only
- Whether or not the object started any outbound RFC calls

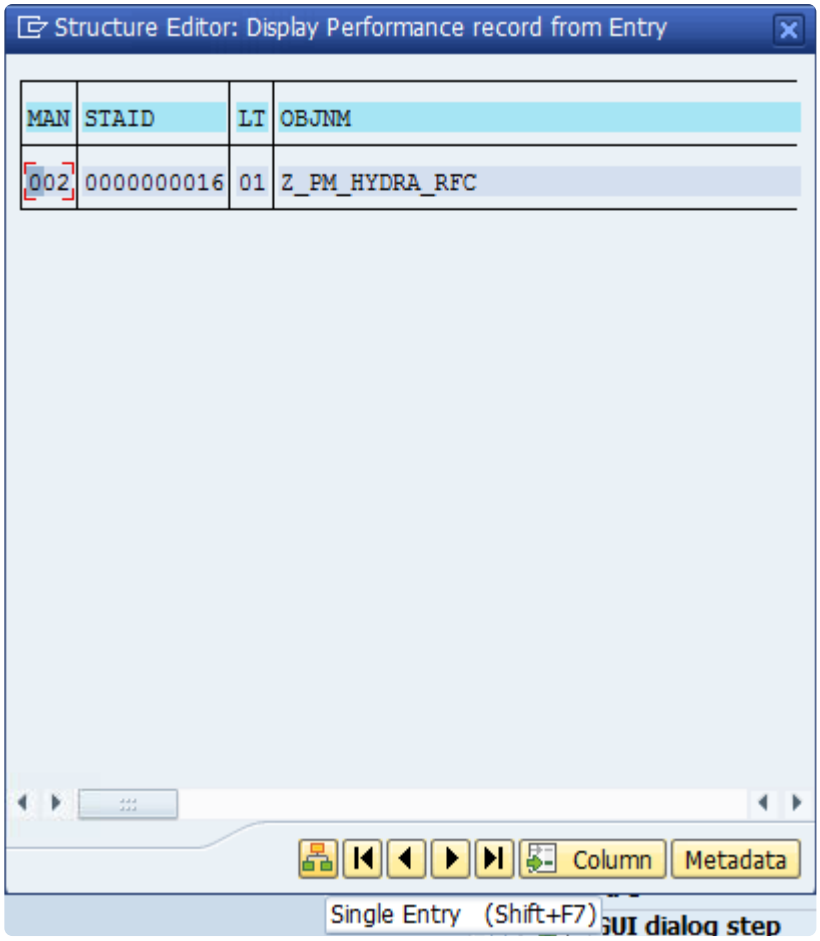
All of this information can be very useful in determining whether or not a particular object should be included in the playback.

There are also two other pieces of detailed information available.

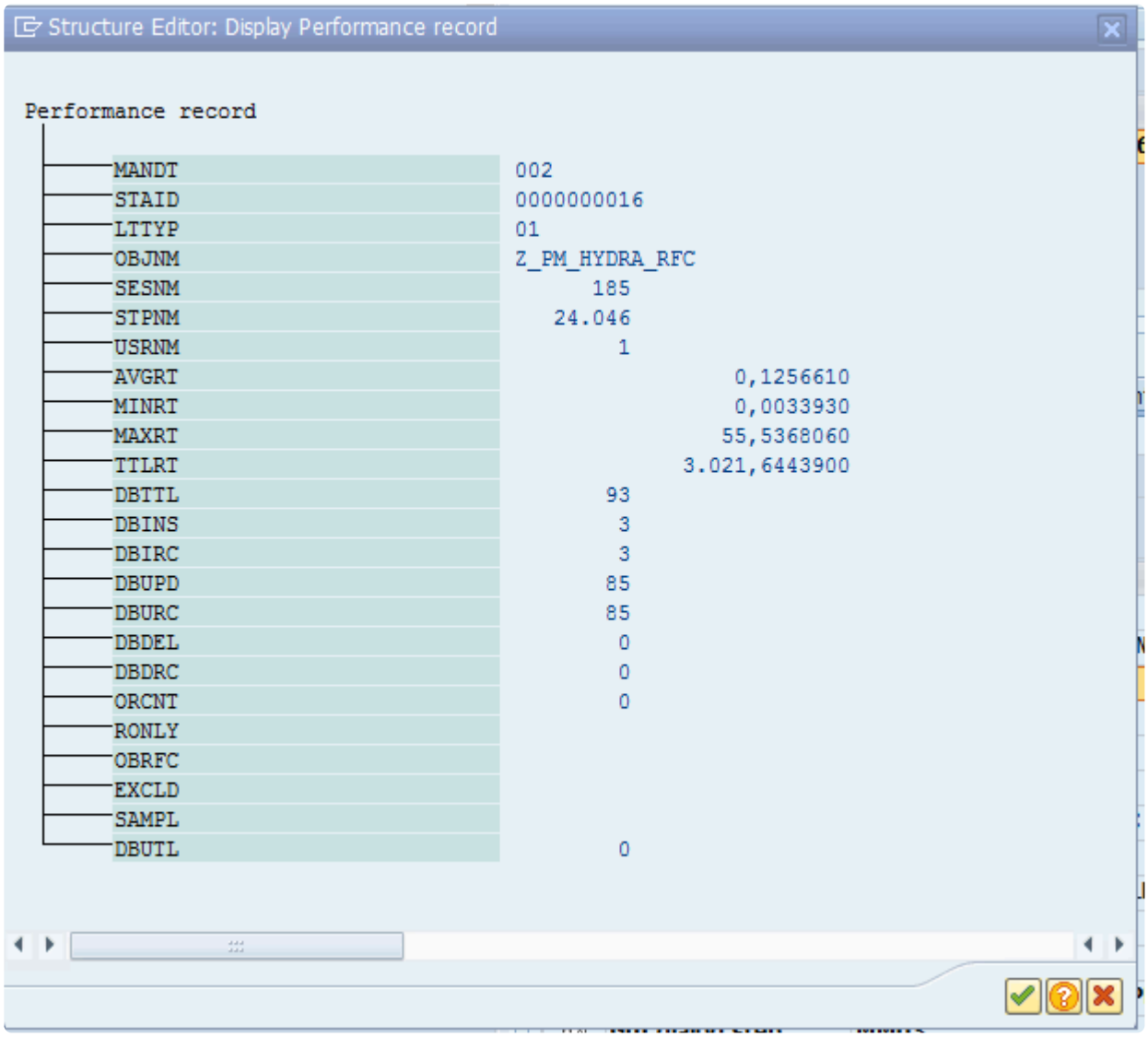
Record details

By selecting a row and clicking on the Record Details button 

You can see the full details of the record.



Clicking on the Single Entry button will show this information in a more easy to read format.



Outbound RFC details

By clicking on the Outbound RFC button:

You can get details of any outbound RFC calls made by the object.

MAN	STAID	LT	OBJNM	RFCNM	RDEST	ORCNT
002	0000000016	04	ZZHUPA	*RFC_CLIENT_COLLECTOR*		8
002	0000000016	04	ZZHUPA	CRM_UIU_ERP_READ_CRM_DOC_FLOW	MP2CLNT002	13
002	0000000016	04	ZZHUPA	ENQUE_READ2	sv000109_GDE_00	4
002	0000000016	04	ZZHUPA	FUNCTION_EXISTS	MP2CLNT002	4
002	0000000016	04	ZZHUPA	HU_GET_RFC_DATA	WAAGEN_LZHEMER_CITRIX	2.148
002	0000000016	04	ZZHUPA	IDOCS_OUTPUT_TO_FILE	a_rfc	885
002	0000000016	04	ZZHUPA	RFC_PING	MP2CLNT002	26
002	0000000016	04	ZZHUPA	Z_RCOM_PRINT	RCOM_SERVER	18
002	0000000016	04	ZZHUPA	Z_ZGROHE_STOCK_GET	sv000109_GDE_00	2
002	0000000016	04	ZZHUPA	Z_ZGROHE_STOCK_UPDATE	sv000109_GDE_00	2

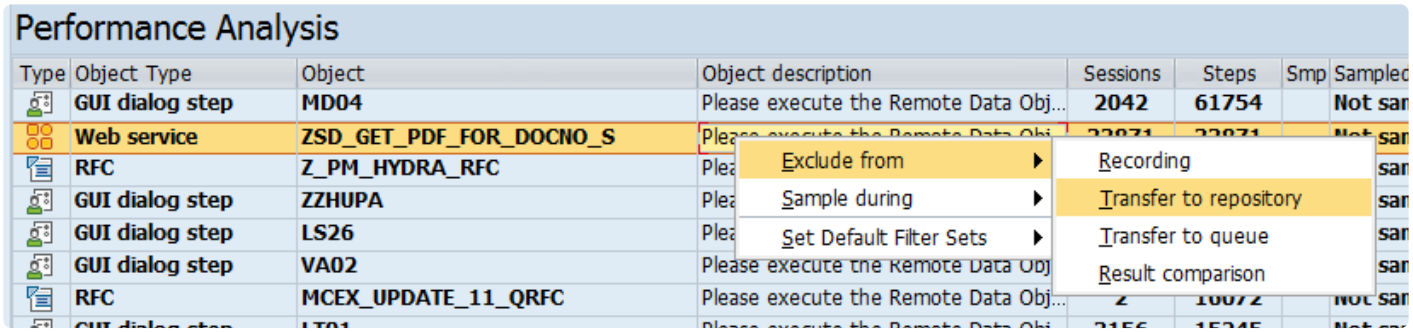
Trimming the playback

There are two ways, from within the performance analysis screen, of trimming objects from the playback.

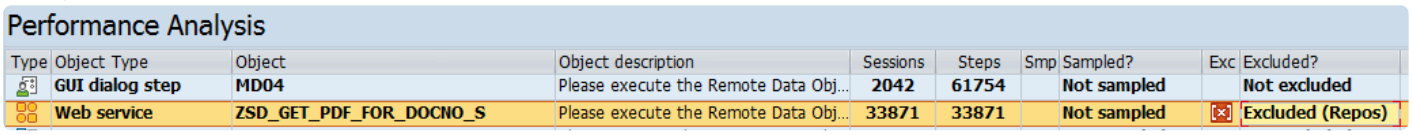
> The functionality in this area makes use of Filter Sets, so please make sure you are familiar with filter sets, exclusions and sampling before proceeding. Documentation for this functionality can be found in this section of the guide.

Exclusion

By using the exclusion functionality, it is possible to completely remove an object from the playback. For example, if we wanted to completely exclude the Web Service ZSD_GET_PDF_FOR_DOCNO_S from the playback, then we would right-click on the row and choose Exclude from —> Transfer to Repository.



This will add an exclusion to your default filter set and the exclusion will be displayed on the Performance Analysis screen.

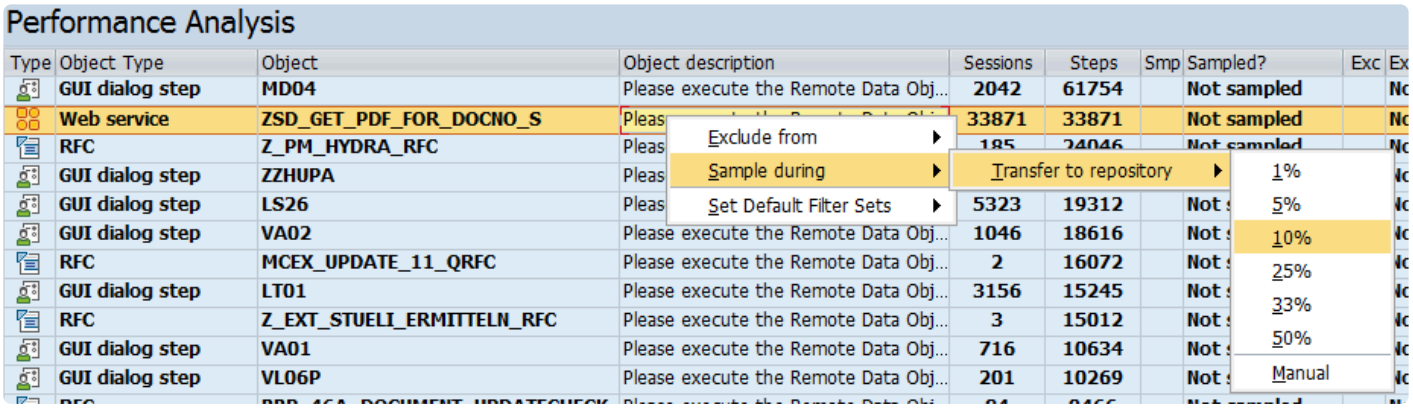


> If you know that you will never want to play back this object, then you can also at this point exclude it from future recordings by right-clicking and choosing Exclude from —> Recording

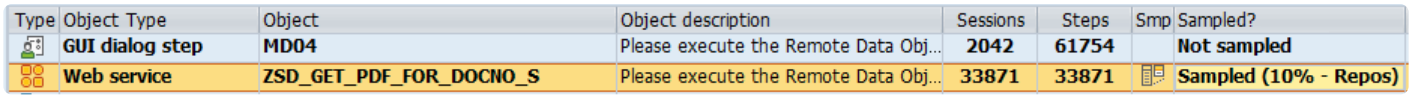
Sampling

If, however, you decide that you still want to test this web service, but that there is no need to execute it over thirty thousand times during the playback, you can instead use sampling. Sampling allows you to only transfer to the repository a percentage of the recorded transactions. So if, for example, we decide that executing this web service 3,000 times will be enough to properly test it, we could sample it so that only 10% of the recorded volume is transferred to the repository and hence included in the playback.

To do this, right-click on the object and choose Sample during —> Transfer to repository —> 10%



Once this has been done, then during the transfer to repository Testimony will randomly select 10% of the recorded transactions to be transferred. The sampling rate is displayed in the Performance Analysis screen.

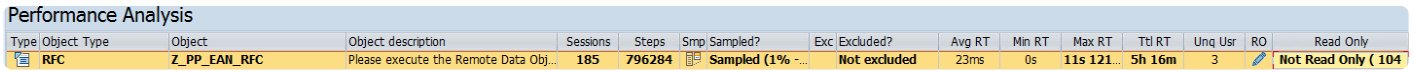


Determining which objects to trim from the playback

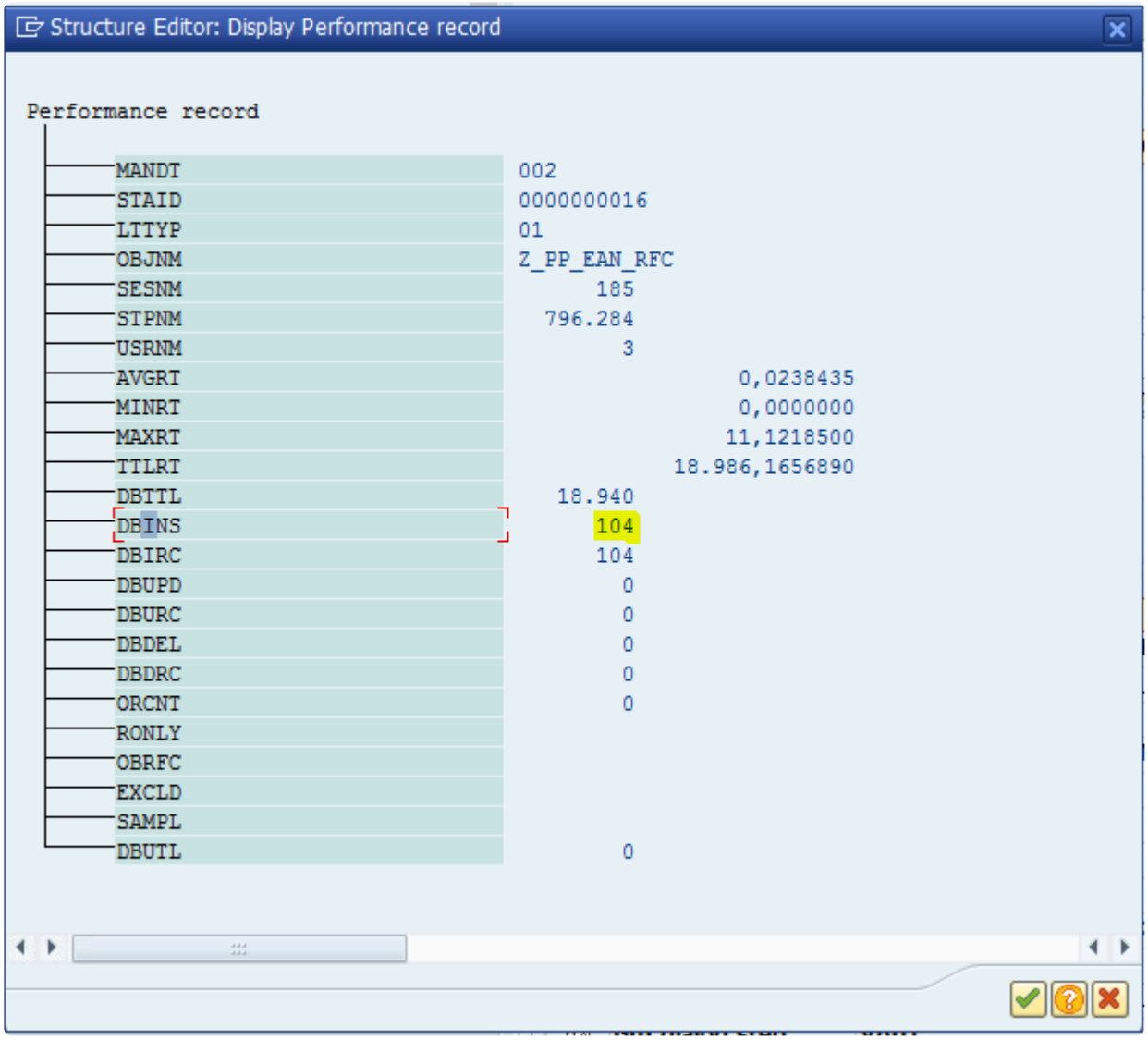
The following are the main things to consider when trying to determine if an object can be trimmed from the playback?

Is it a high-volume read-only object?
Read-only objects that are executed many thousands of time are prime candidates for trimming from the playback. Since they are read-only, we know that they don't create or change data which subsequent objects may rely on. We therefore need to decide, firstly, whether they should be tested at all. If they don't need to be tested, then they can be completely excluded from the playback. Secondly, if they should be tested, we should decide whether we need to test the full recorded volume. If not, then we can use sampling to trim the bulk of them from the playback.

Is it a high-volume "read-mostly" object?
Some objects will be listed as Not Read Only, but the value of the number of writes (shown in brackets in the Read Only column) is very low compared to the number of Steps or Sessions. In the example below, we can see that despite there having been over 750,000 steps recorded, only 104 database updates were executed:



By drilling into the details, we can see that these were inserts into the database.



Based on this, we may be able to decide – perhaps after consulting with the technical or functional experts responsible for this object – that in nearly all cases this object is in fact read-only, and can therefore be trimmed from the playback. As you can see, this has been done for this object in the example, where it is being sampled at a rate of 1%.

Does it have outbound RFC calls to a destination that doesn't exist in the playback system?
At presend, Testimony only supports outbound RFCs on a pilot basis, so in most cases it will be necessary to exclude objects that make outbound RFC calls. By checking the outbound RFC details you can see where the calls are being made to (the RDEST column) and determine if this RFC destination exists in the playback system.

MAN	STAID	LT	OBJNM	RFCNM	RDEST	ORCNT
002	0000000016	04	ZZHUPA	*RFC_CLIENT_COLLECTOR*		8
002	0000000016	04	ZZHUPA	CRM_UIU_ERP_READ_CRM_DOC_FLOW	MP2CLNT002	13
002	0000000016	04	ZZHUPA	ENQUE_READ2	sv000109_GDE_00	4
002	0000000016	04	ZZHUPA	FUNCTION_EXISTS	MP2CLNT002	4
002	0000000016	04	ZZHUPA	HU_GET_RFC_DATA	WAAGEN_LZHEMER_CITRIX	2.148
002	0000000016	04	ZZHUPA	IDOCS_OUTPUT_TO_FILE	a_rfc	885
002	0000000016	04	ZZHUPA	RFC_PING	MP2CLNT002	26
002	0000000016	04	ZZHUPA	Z_RCOM_PRINT	RCOM_SERVER	18
002	0000000016	04	ZZHUPA	Z_ZGROHE_STOCK_GET	sv000109_GDE_00	2
002	0000000016	04	ZZHUPA	Z_ZGROHE_STOCK_UPDATE	sv000109_GDE_00	2

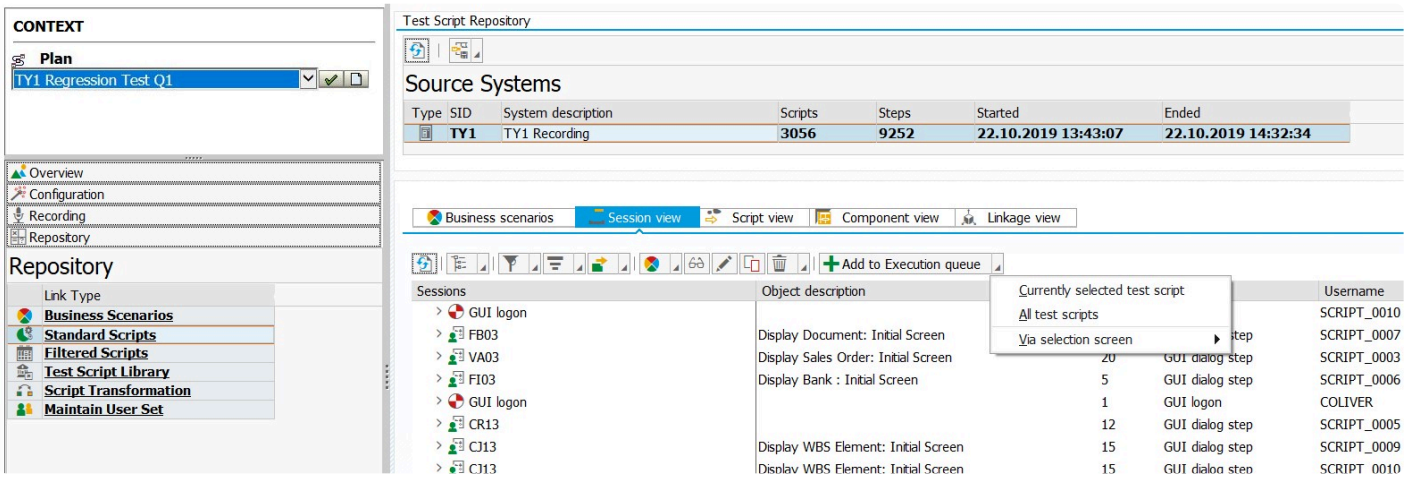
3.9. Building the Execution Queue

Overview

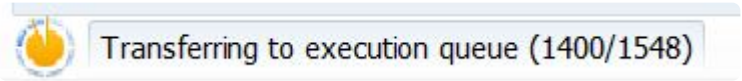
The test scripts that have been created in the test repository must be transferred to the execution queue in preparation for the playback process. For a given Test Plan, only a single active execution queue can exist. Historical execution queues can also be seen here.

Process Steps

1. Select Test Plan to use
2. Navigate to the Repository
3. Select Standard Scripts or Filtered Scripts
4. Double click on the Source System which will display the list of scripts in the Repository for the selected Test Plan
5. Click on “Transfer to Execution Queue”. If you’ve filtered the scripts when you transfered them to the repository, you can select all scripts. Otherwise you may want to narrow your selection via Selection screen

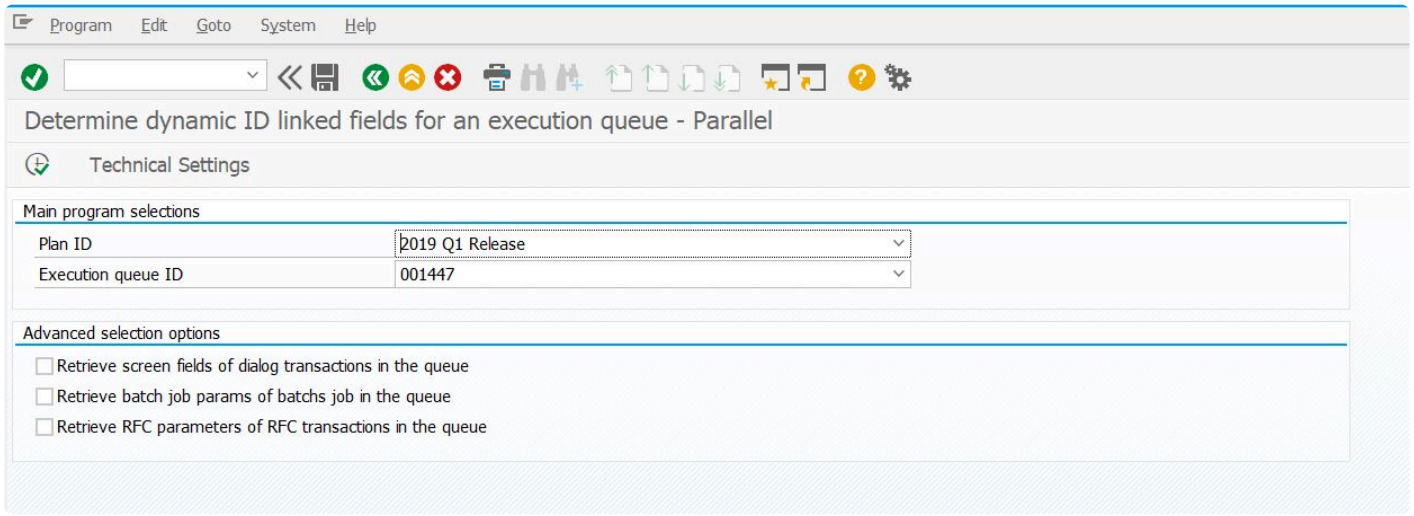


You will see the following status as the transfer is taking place



You will also see the following popup box to ask about the generation of Dynamic IDs.

it is highly suggested that you always run the Dynamic ID. If your recording is large, select “Multi Thread”. You’ll be taken to the follow screen where you select the the Plan ID and Execution ID for which you want to run the Dynamic ID process. If you choose not to run the dynamic ID process at this time, you will need to do so prior to starting the execution queue playback. This is done by selecting the execution queue in the list and clicking the “Dynamic ID Generation” button.



Renaming the execution queue

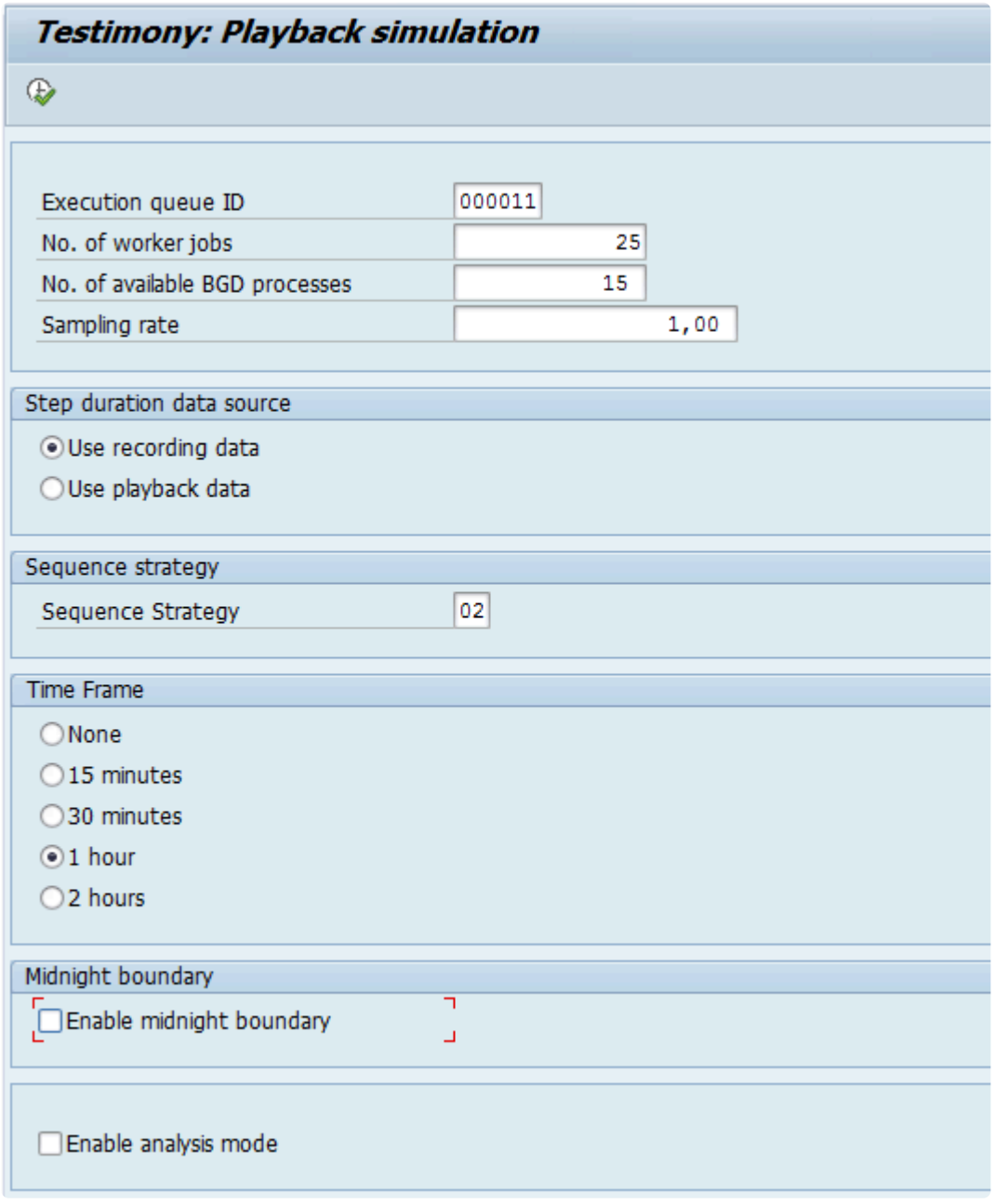
When the execution queue is created, it is given the default name “Test Phase 1”. It is a good idea to rename the execution queue and give it a more meaningful name, especially if you are planning on doing a double playback.

To rename the execution queue, select the queue and click on the “Rename queue” button:

3.9.1. Run the playback simulation

The playback simulation can be useful for checking that you have enough bots defined for the playback, as well as enough background processes defined on the playback system.

To run the simulation, go to Execution —> Execution Queues, select your execution queue and click on the Simulation Playback button:

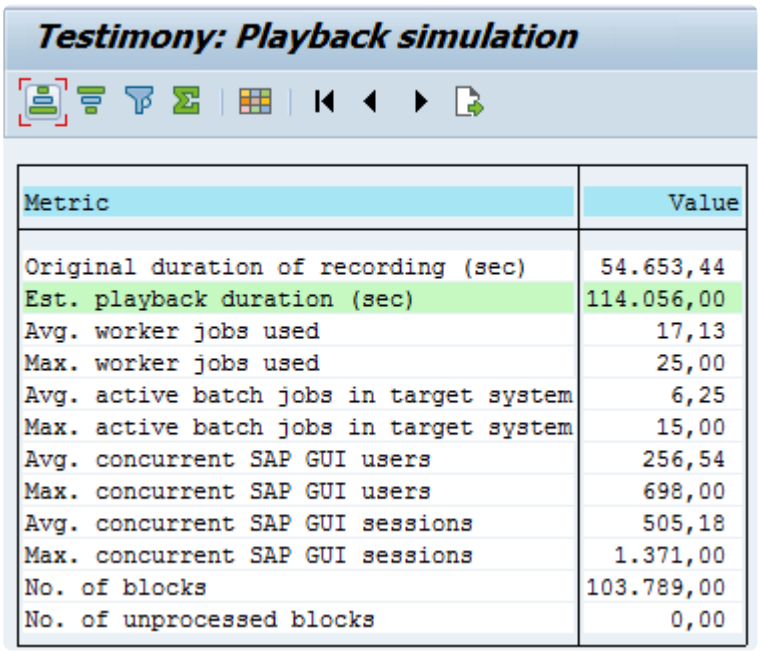


On the selection screen, enter the following information:

- No. of worker jobs: set this to the number of bots you currently have

- No. of available BGD processes: set this to the number of background processes defined on the playback system
- Time frame: if you would like the simulation broken down into time slices, select one of the radio buttons
- Enable midnight boundary: if your recording crossed a midnight boundary, then select this checkbox

When you execute the program, you will be presented with the following output. (Note that for large execution queues it can take some time for this program to run. You may want to run it in background and view the results from the spool.)



Metric	Value
Original duration of recording (sec)	54.653,44
Est. playback duration (sec)	114.056,00
Avg. worker jobs used	17,13
Max. worker jobs used	25,00
Avg. active batch jobs in target system	6,25
Max. active batch jobs in target system	15,00
Avg. concurrent SAP GUI users	256,54
Max. concurrent SAP GUI users	698,00
Avg. concurrent SAP GUI sessions	505,18
Max. concurrent SAP GUI sessions	1.371,00
No. of blocks	103.789,00
No. of unprocessed blocks	0,00

There are two key metrics to consider on this report:

- Max. active batch jobs in target system: if the number shown is equal to the number of available BGD processes you specified on the selection screen, then you could potentially speed up the playback by adding more BGD processes on the playback system.
- Max. concurrent SAPGUI users: if this number is greater than twenty times the number of bots you have, then you may well need to add more bots in order to speed up the playback.

3.9.2. Run the screen size analysis

Introduction

When recording online (dialog) transactions, one thing that Testimony captures is the number of lines that can be displayed on a user's screen. Since it is very likely that users will have a variety of monitors, screen sizes and screen resolutions, different users will be able to display different numbers of lines of output on their screens. Some list-based screen outputs are accessed by selecting a particular row on a screen. For example, a user may be using VA03 to display a sales order. They may use the F4 help to get a list of sales orders and then double-click on, say, the 25th row on the screen to select the sales order they want to view.

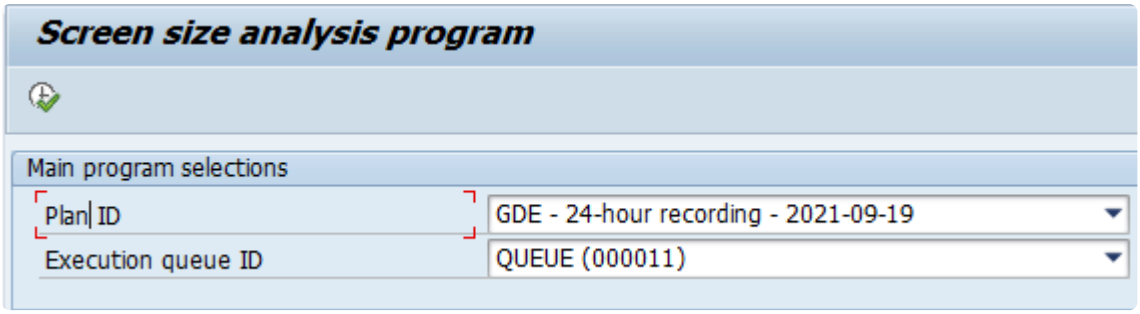
Screen resolution for bot VMs

During a playback, the bots, running on a Windows machine (usually a Virtual Machine) run through dialog transactions in exactly the same way as the users did. In our VA03 example, the bot will also call the F4 help and will also double-click the 25th row on the screen to select the sales order to be viewed. If the screen resolution on the bot VM is not set high enough, then it may be that only 24 rows can be displayed. In this case, the execution of the VA03 script will cause an Error.

> NOTE: Improvements in Testimony v2.40 in the way that Testimony handles selection from lists means that there is much less reliance than previously on the actual row in the displayed list. However, there are still some areas of functionality where the row number is important.

The program /BTI/AUT_BOT_SCRN_SIZE_ANLYSIS can be used to analyse the dialog transactions in an execution queue, and will report on the number of rows displayed on screen.

Execute the program using SE38/SA38 on the Central System and select your Test Plan and Execution Queue.



When you execute the report, output will be shown detailing different "Max Rows Displayed" and how many scripts displayed each number of rows.

MAX NO OF ROWS	NO. OF SCRIPTS
77	5
76	3
75	8
68	4
62	2
60	86
57	9
56	104
55	209
54	4
53	476
52	652
51	141
50	326
49	716
48	270
47	175
46	143
45	136
44	356
43	384
42	762
41	496
40	371
39	4.766
38	1.414
37	2.085
36	1.899
35	983
34	525
33	1.035
32	1.380
31	2.614
30	805
29	1.364
28	224
27	312
26	280
25	1.592
24	2.075
23	1.327
22	1.947

In the example above, we can see that a few users had very large screens, able to display more than 50 rows in a list. However, it looks like most users had screens displaying a maximum of 39 rows or fewer. You should change the screen resolution on the bot VMs to try to accommodate as many list rows as possible. However, in most cases there will be a maximum screen resolution that can be set which may not replicate very large monitors.

Note that if using RDP to connect to the bots, you often can't directly change the screen size in Windows. However, you can set the resolution in the RDP config, which will take effect the next time you connect to the bot VM

3.10. Analyzing Coverage

Overview

A Testimony recording generates, from real-life activity, an entire test script library which covers a high percentage of your actual SAP system usage. How big a percentage this is (i.e., the coverage you have achieved) depends firstly on the length of the recording (a 48-hour recording will naturally cover more work that a 4-hour recording) as well as the point within the business cycle (e.g., month-end, mid-month) in which the recording was active.

In order to better understand the level of coverage that you have achieved with your recording, Testimony includes functionality to perform a Coverage Analysis, comparing what was recorded with your production system's typical workload.

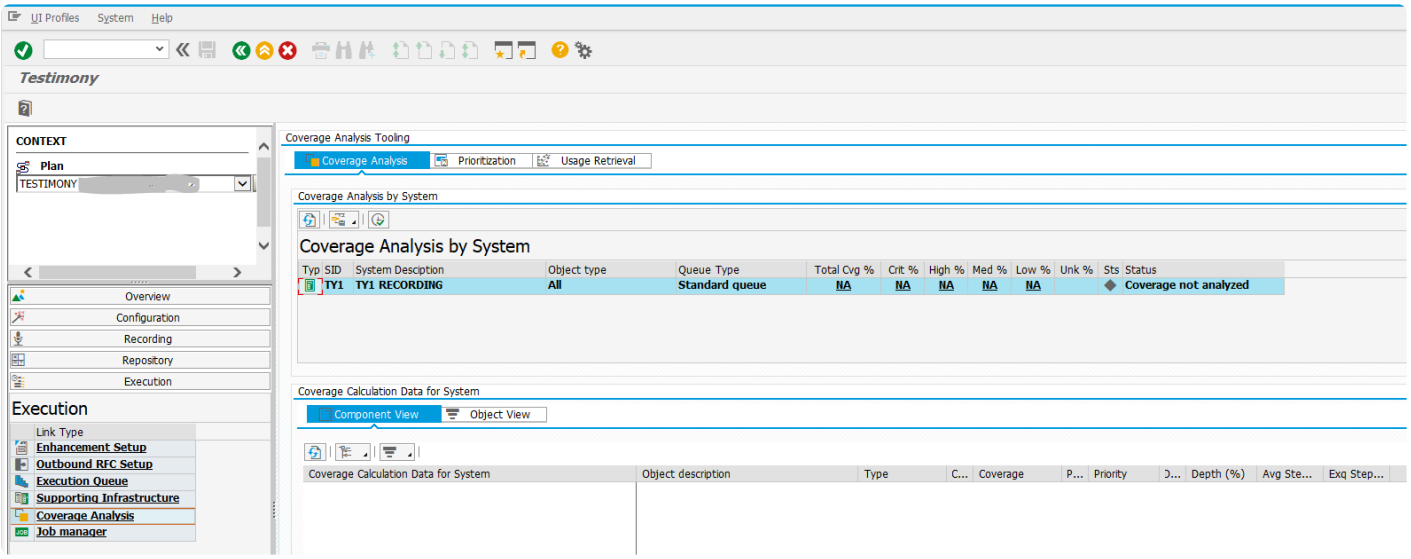
Coverage Analysis consists of three steps:

1. **Usage retrieval**: In this step, performance statistics data is retrieved from the recorded system for a period that you define
2. **Prioritization**: This step takes the performance data and prioritises the individual components (dialog transactions, batch jobs, etc.) according to either how frequently they are typically executed in production, or according to a set of priorities that you determine
3. **Coverage Analysis**: This final steps takes the recorded data from the execution queue and compares it with the usage data. You can then see high-level statistics (e.g., what percentage of critical priority dialog transactions you recorded) as well as detailed information on each dialog transaction, batch job, etc.

Accessing the Coverage Analysis functionality

Since Coverage Analysis compares production usage data with what is in your execution queue, the Coverage Analysis functionality is contained within the Execution section of the main menu in Testimony. However, it is not necessary to have already performed the playback before running the Coverage Analysis process.

The following image shows the screen you will see when you select the Coverage Analysis item from the menu before you have run any of the Coverage Analysis steps.



The following sections give details of the process to go through in order to perform a full Coverage Analysis of your execution queue.

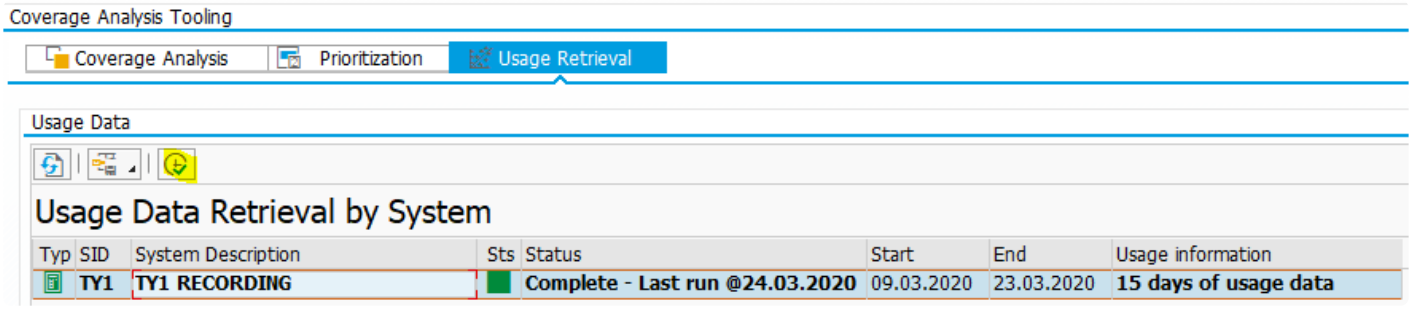
3.10.1. Usage Retrieval

Introduction

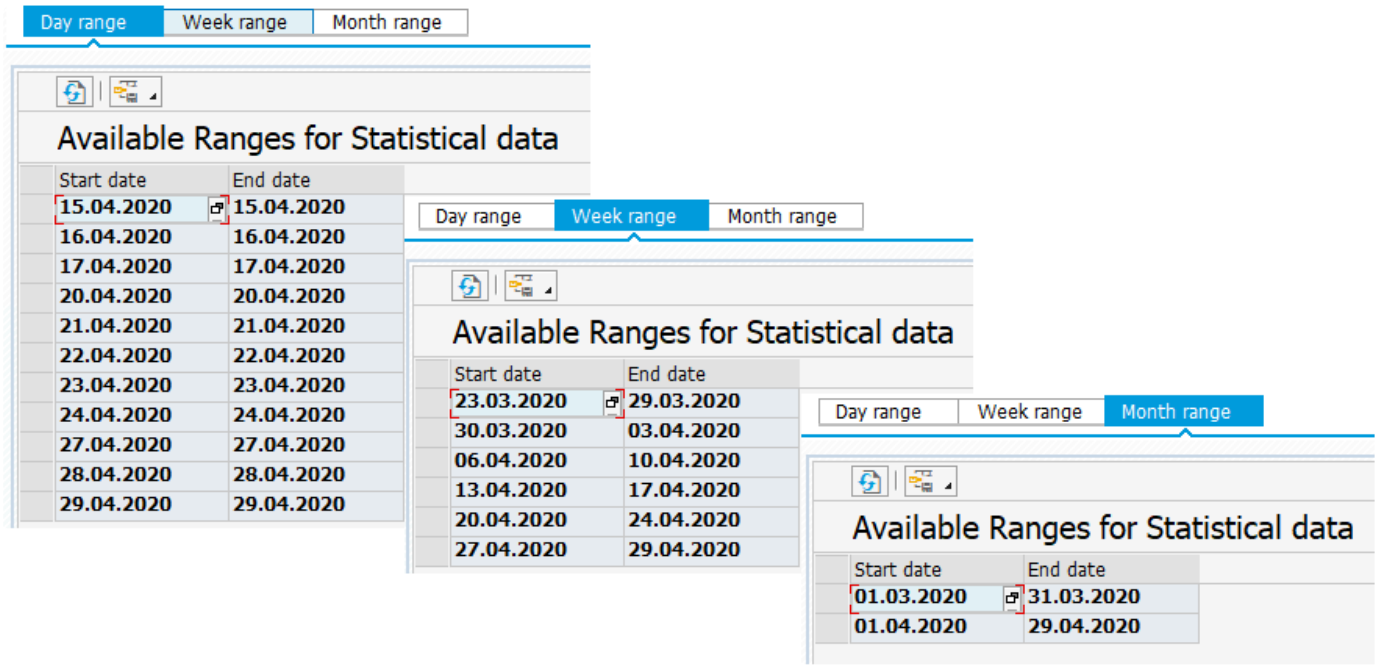
The first step in the coverage analysis process is to retrieve the usage data from the source system. In this step, Testimony will connect to the source system and retrieve workload statistics data (from the ST03N repository) for a period that you choose.

Executing the Usage Retrieval

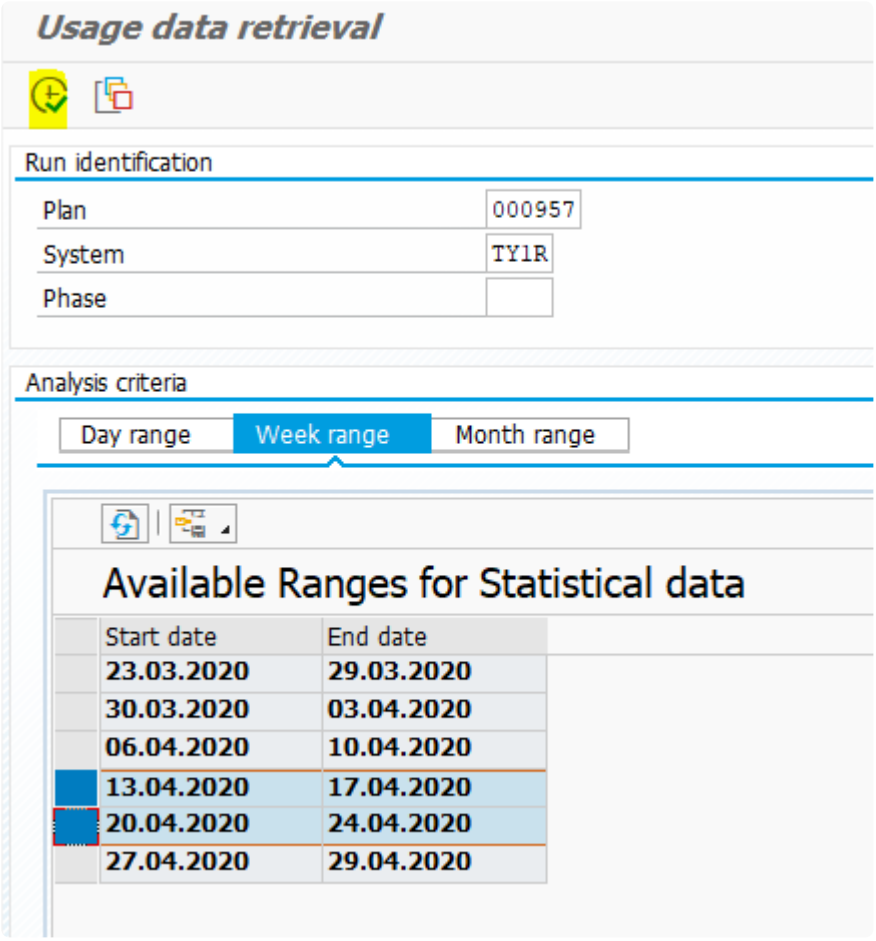
From the Coverage Analysis screen, go to the Usage Retrieval tab, select your source system and click on the Execute button.



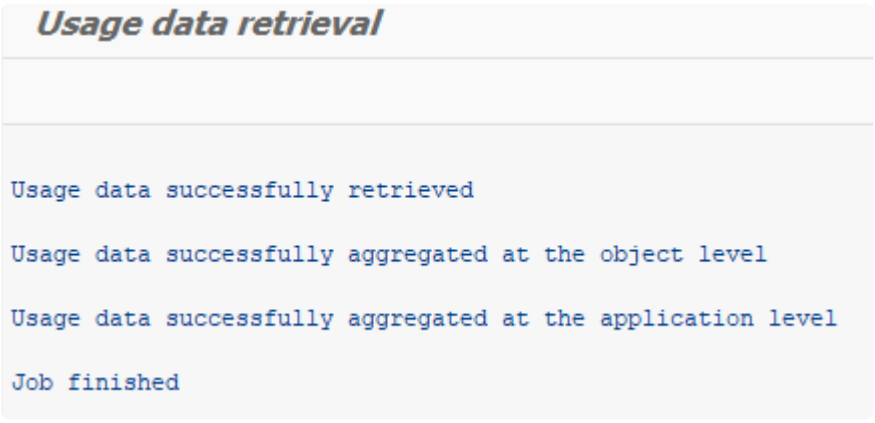
In the selection screen that is displayed you now need to choose the period for which you want to retrieve usage data from the source system. The available dates are shown in three separate tabs, based upon the data available, and the summarisation of data, in the ST03N repository in the source system.



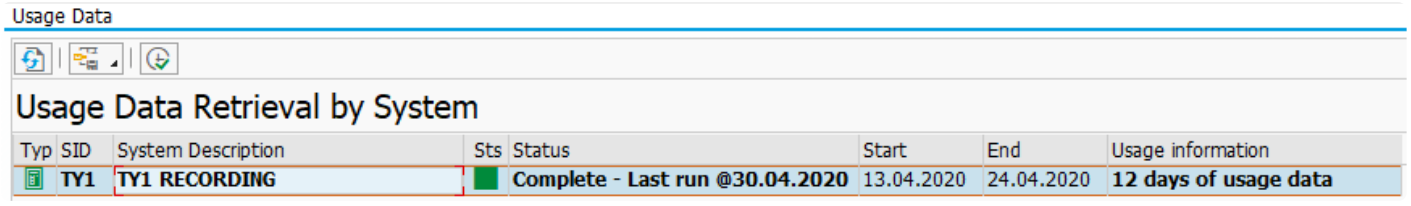
You can choose multiple periods by highlighting more than one row and then retrieve the usage data for the chosen dates by clicking on the execute button. In the example below, we are retrieving data for the two weeks starting 13th April.



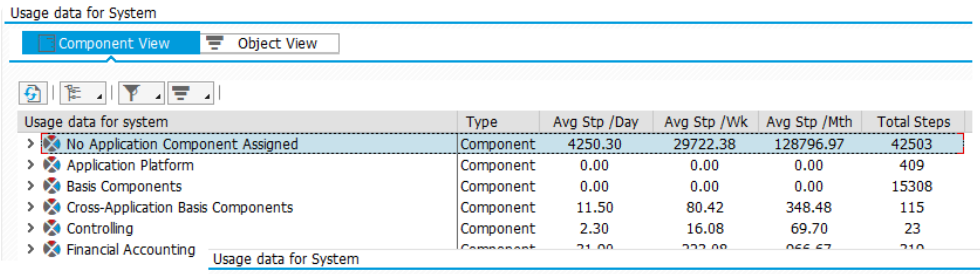
Once the data has been retrieved, the following output is displayed.

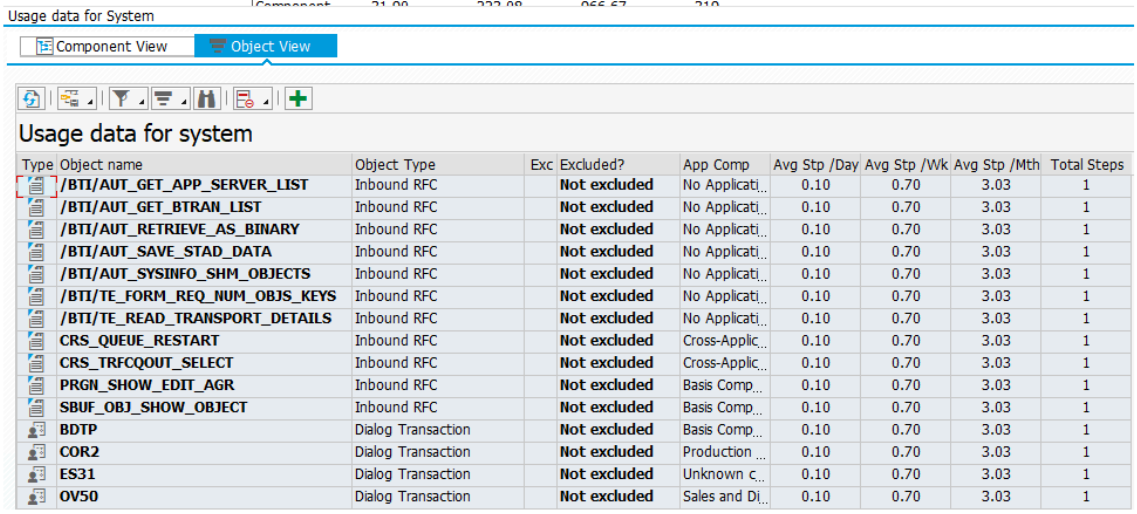


Going back to the Coverage Analysis screen (Usage Retrieval tab) you can see the details of the last usage retrieval run.



Double-clicking on this row will show the details of the data that has been retrieved, either grouped by application component or at the object level:





3.10.2. Prioritization

Introduction

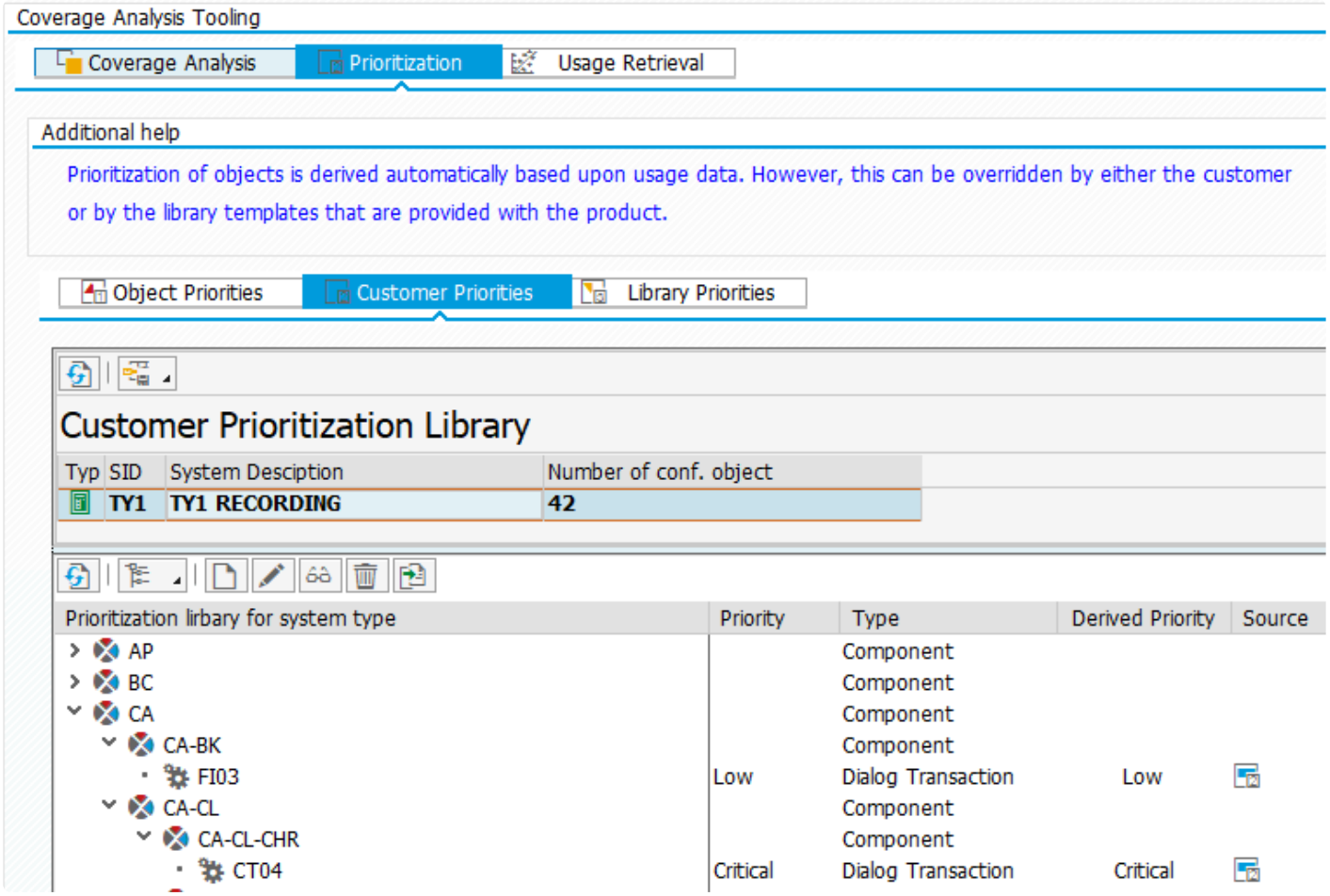
The next step is to prioritise the transactions, batch jobs, etc. that have been retrieved from the source system.

Testimony prioritises the usage data as follows:

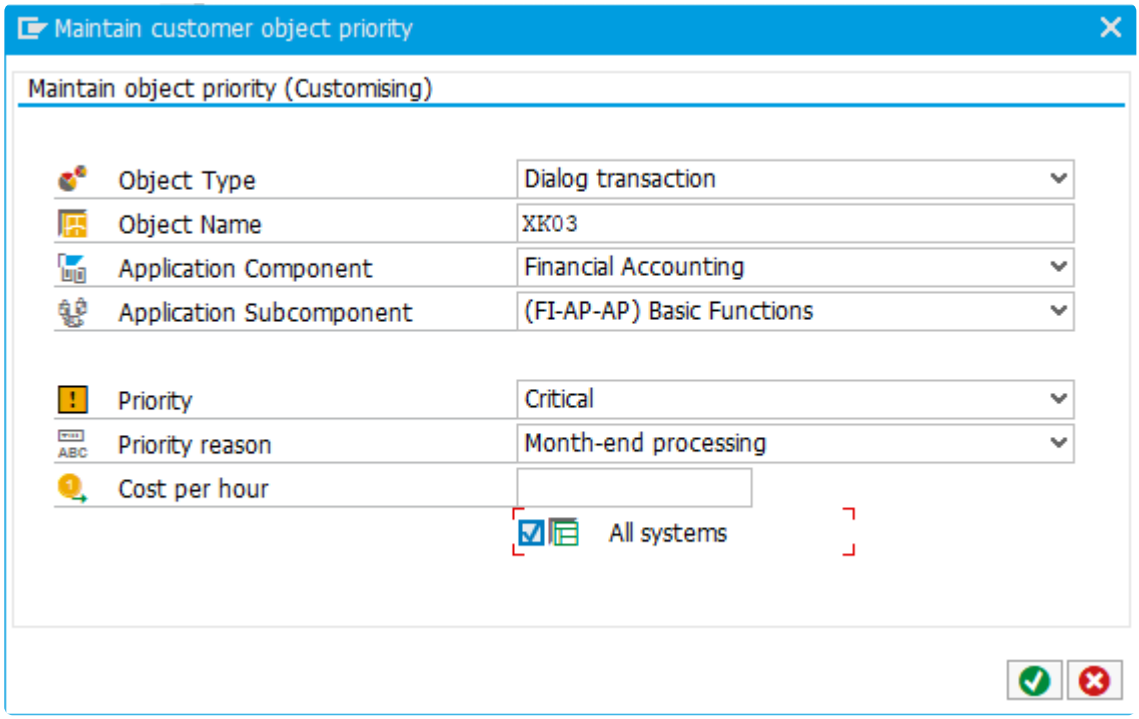
1. First of all, it prioritises by frequency of execution in the source system. The most frequently-executed objects are given the highest priority, and so on down to very infrequently-executed objects which are given the lowest priority. The thresholds used to determine these priorities are determined in the prioritisation run selection screen, discussed below.
2. Next, Testimony can override these usage-based priorities based on Library priorities. These are a set of priorities delivered with the Testimony software where Basis Technologies, in consultation with our customers, has determined that certain common objects should have a priority that is different (either higher or lower) than what might be calculated based on frequency of execution. *Note that at present (Testimony v2.21) we do not currently deliver any library priorities.*
3. Finally, the usage-based priorities can be overridden by priorities set by the customer. This enables the customer to decide that, for example, a dialog transaction that is only infrequently executed (and which would therefore have a low priority according to usage) is important enough to its business that it should have a higher priority.

Specifying customer priorities

From the Prioritization tab of the Coverage Analysis screen, go to the Customer Priorities tab and double-click on the source system. Any customer priorities that have been previously specified will be shown, grouped by application component.



To add a new object to the customer priorities list, click on the Create button and fill in the screen as shown below.

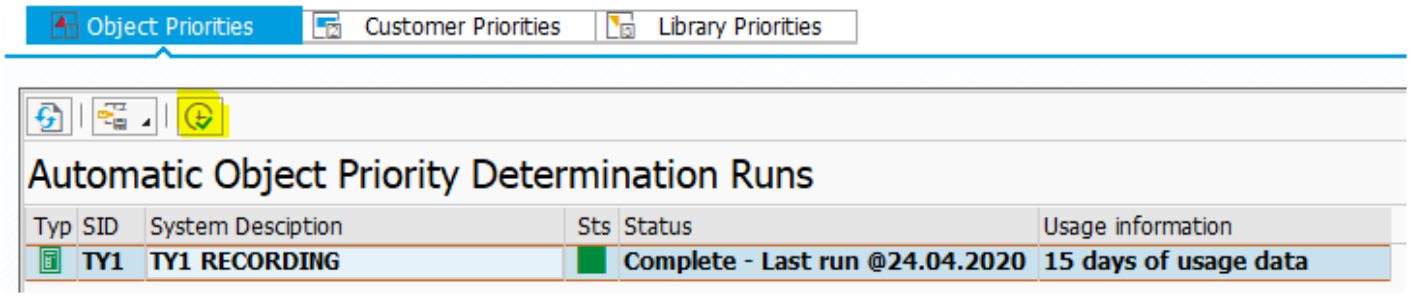


The Object Type, Object Name and Priority fields should be filled in as a minimum. Other fields are optional.

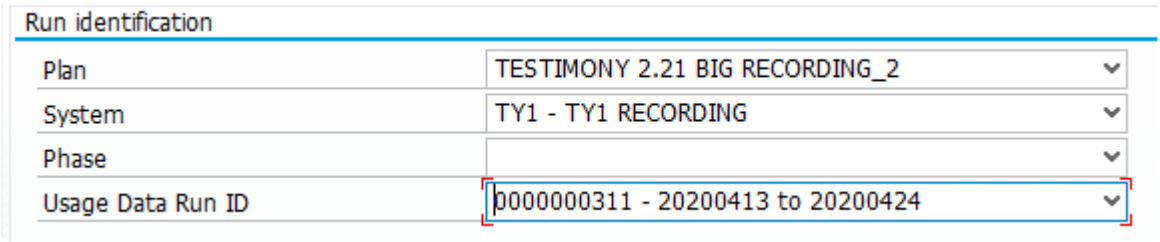
The “All systems” box should be checked if this priority is relevant for all systems that you will be using Testimony to record from.

Performing the Prioritization run

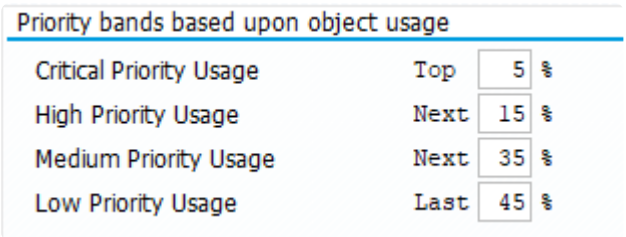
Once your customer priorities have been set, go to the Object Priorities tab, select your source system and click on the execute button.



In the selection screen, enter Usage Data Run ID from the usage retrieval run you recently performed. **NB**, you may need to press Enter to refresh the screen’s fields, and see the correct list of Run IDs



On this screen, you also have the opportunity to influence the automatic usage-based calculation of priorities by entering percentage thresholds for the different priorities.



The thresholds work as follows.

Using the thresholds in the screen shot above, and assuming that the usage retrieval run brought back usage statistics for 100 distinct transactons, then:

- The 5 most-frequently executed transactions would be given a Critical priority
- The next 15 most frequently executed transactions would be given a High priority

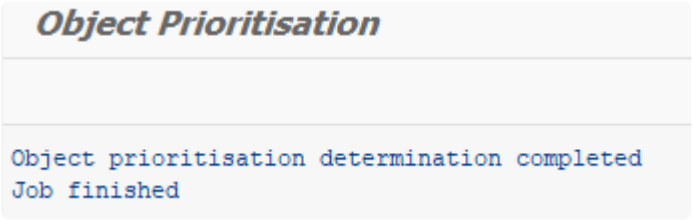
- The next 35 most frequently executed transactions would be given a Medium priority
- The remaining 45 transactions would be given a Low priority

> When changing these thresholds, make sure that their sum adds up to 100%

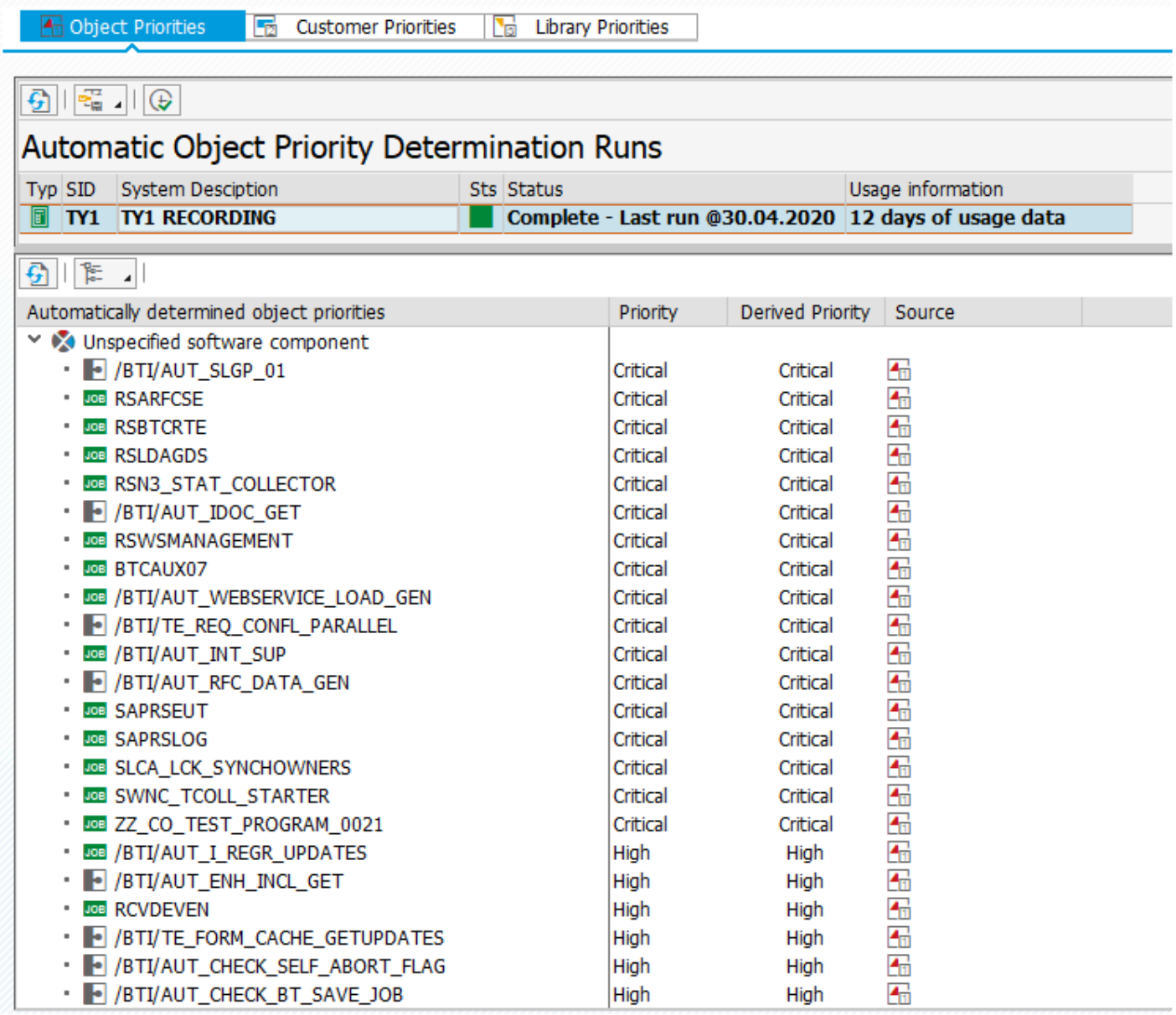
Once you are happy with the thresholds, click on the execute button to perform the prioritisation.

> Remember that the usage-based priorities will be overriden by any customer-defined priorities.

The prioritisation run will return the following output.



You will now be able to see the prioritisation results from the coverage analysis screen.


Object Priorities
Customer Priorities
Library Priorities
Automatic Object Priority Determination Runs
Typ SID System Desciption Sts Status Usage information
TY1 TY1 RECORDING Complete - Last run @30.04.2020 12 days of usage data
Automatically determined object priorities Priority Derived Priority Source
Unspecified software component
/BTI/AUT_SLGP_01 Critical Critical
RSARFCSE Critical Critical
RSBTCRTE Critical Critical
RSLDAGDS Critical Critical
RSN3_STAT_COLLECTOR Critical Critical
/BTI/AUT_IDOC_GET Critical Critical
RSWSMANAGEMENT Critical Critical
BTCAUX07 Critical Critical
/BTI/AUT_WEBSERVICE_LOAD_GEN Critical Critical
/BTI/TE_REQ_CONFL_PARALLEL Critical Critical
/BTI/AUT_INT_SUP Critical Critical
/BTI/AUT_RFC_DATA_GEN Critical Critical
SAPRSEUT Critical Critical
SAPRSLOG Critical Critical
SLCA_LCK_SYNCHOWNERS Critical Critical
SWNC_TCOLL_STARTER Critical Critical
ZZ_CO_TEST_PROGRAM_0021 Critical Critical
/BTI/AUT_I_REGR_UPDATES High High
/BTI/AUT_ENH_INCL_GET High High
RCVDEVEN High High
/BTI/TE_FORM_CACHE_GETUPDATES High High
/BTI/AUT_CHECK_SELF_ABORT_FLAG High High
/BTI/AUT_CHECK_BT_SAVE_JOB High High

3.10.3. Coverage Analysis

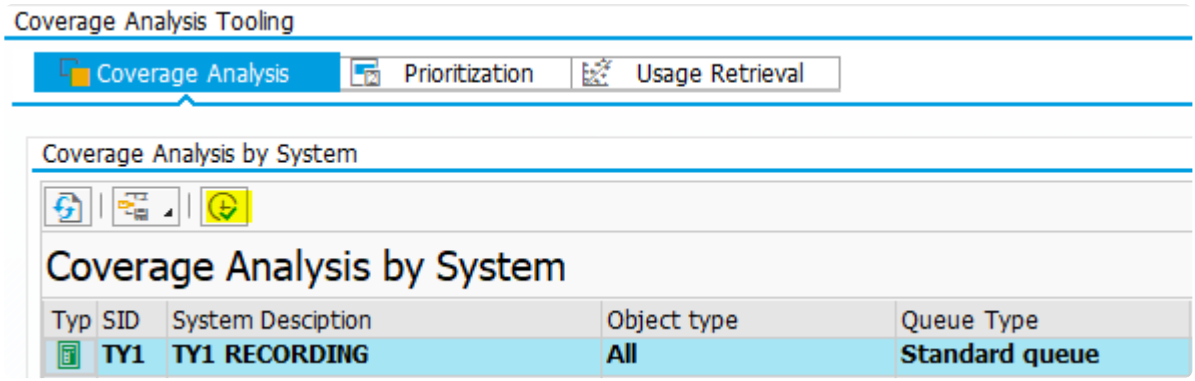
Introduction

Now that the usage information has been retrieved from the source system and the prioritisation run as completed, you can run the coverage analysis proper.

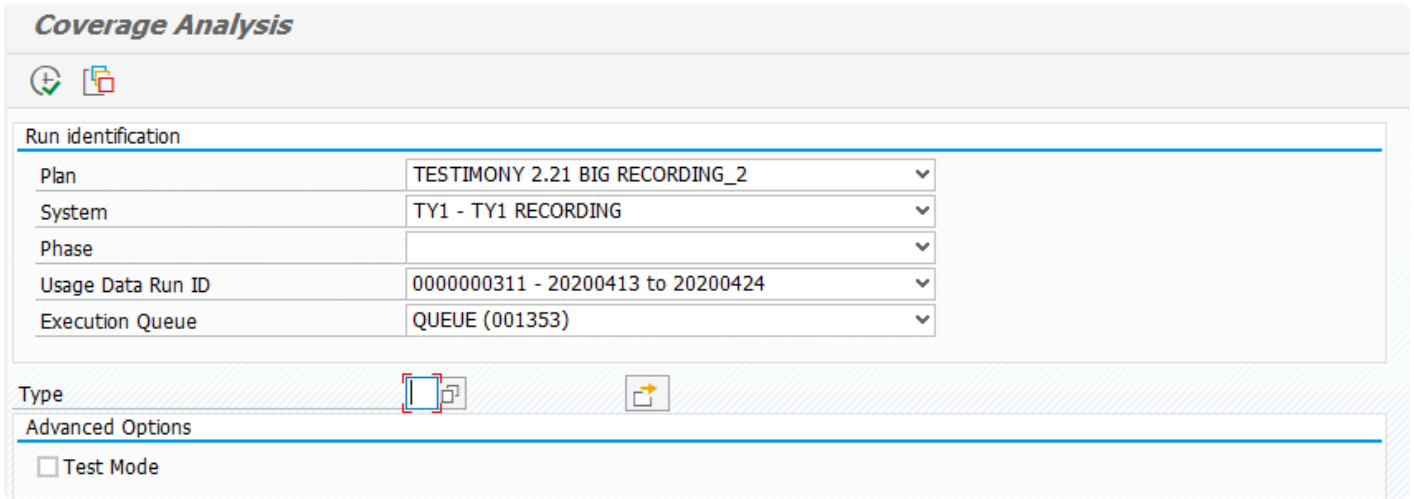
The coverage analysis run will compare the transaction volumes captured in your recording (by reading the execution queue) with the data from the usage retrieval, determining whether or not a particular transaction (or batch job, RFC, etc.) has been adequately covered in the recording.

Executing the Coverage Analysis

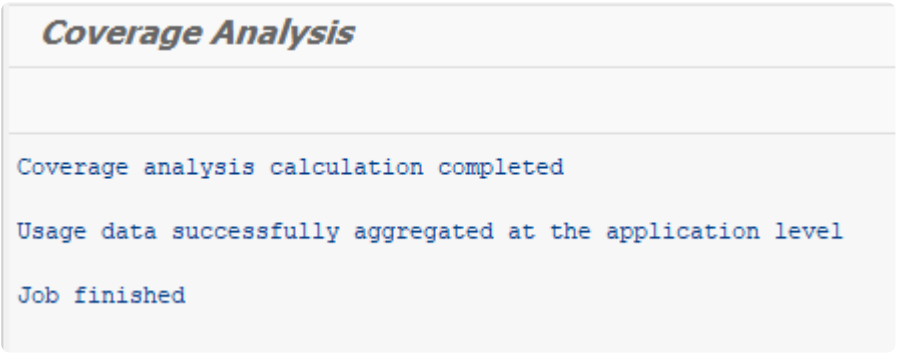
From the Coverage Analysis tab, select your source system and click on the Execute button.



On the selection screen, enter the usage retrieval run and the execution queue you want to compare against. Optionally, you can also restrict the coverage analysis to certain types of workload (e.g., dialog transactions only).
NB, you may need to press Enter to refresh the screen's fields, and see the correct list of Run IDs

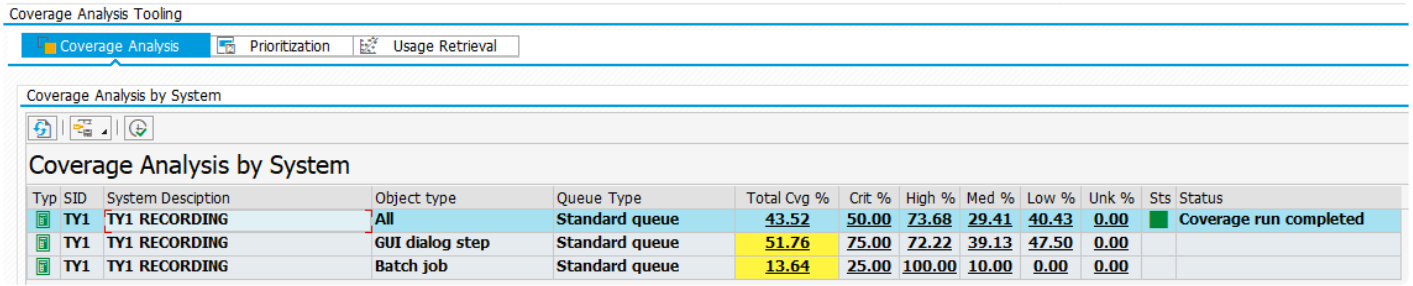


Click on Execute and the coverage analysis will run, displaying this output at the end.



Analysing Coverage Analysis Results

Once the coverage analysis has run, you will see a summary of the results in the Coverage Analysis screen. The summary shows the analysis for the entire execution queue, as well as broken down by different transaction types.

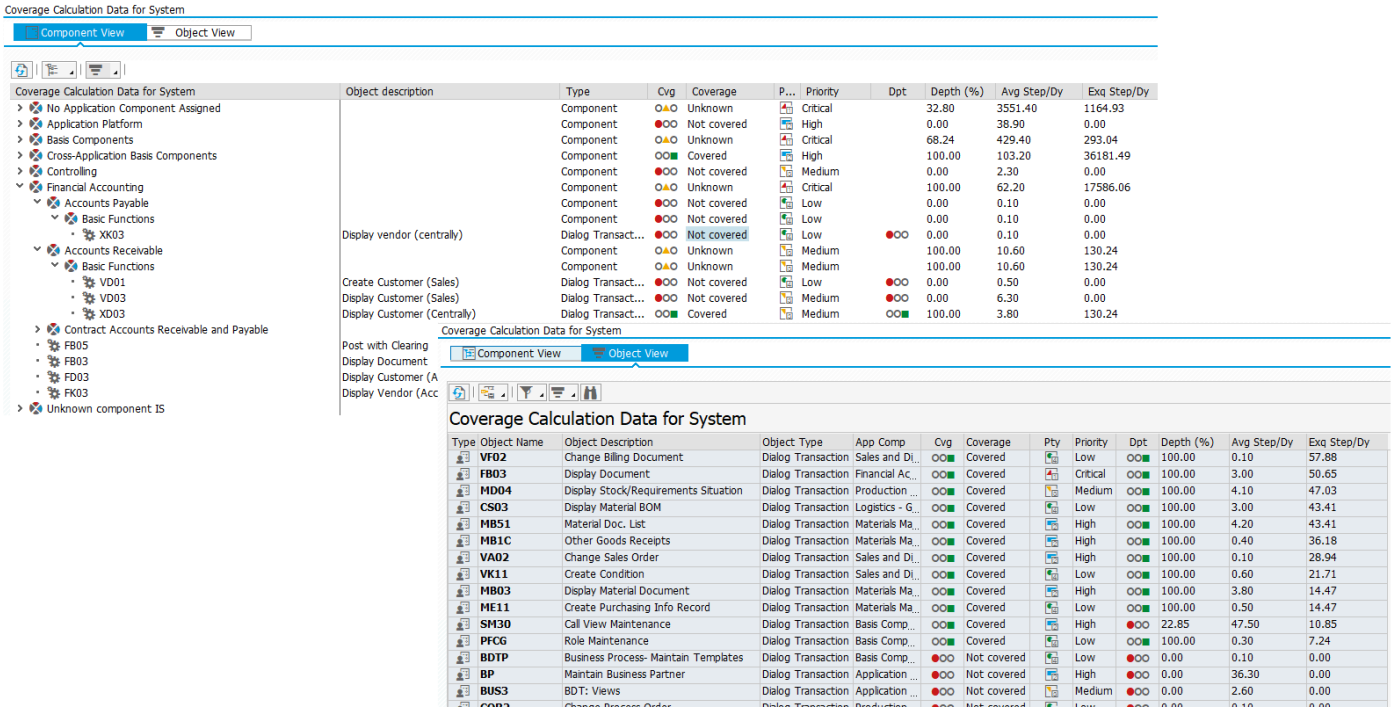


Typ	SID	System Description	Object type	Queue Type	Total Cvg %	Crit %	High %	Med %	Low %	Unk %	Sts	Status
	TY1	TY1 RECORDING	All	Standard queue	43.52	50.00	73.68	29.41	40.43	0.00		Coverage run completed
	TY1	TY1 RECORDING	GUI dialog step	Standard queue	51.76	75.00	72.22	39.13	47.50	0.00		
	TY1	TY1 RECORDING	Batch job	Standard queue	13.64	25.00	100.00	10.00	0.00	0.00		

In the image above, we can see that the overall coverage for all transaction types was 43.52%, with the coverage also broken down by priority and by transaction type.

For a transaction to be covered in the execution queue and count towards the coverage percentage, it must exist at least once in the execution queue.

As well as the results summary, you can double-click on a row in the summary to get details of the coverage broken down by application component as well as at the object level



As you can see from the object view, for each object you are told whether or not it is covered (i.e., whether or not it exists in the execution queue), the priority according to the prioritisation, and the “Depth”.

The Depth of an object’s coverage is a comparison of the number of times it will be executed in the playback (normalised to a 24-hour period) with the average number of executions per day from the usage retrieval.

For example, in the object view in the image above, we can see that transaction MD04 was executed on average 4.1 times per day in the source system. In the execution queue, this transaction has a normalised daily execution rate of 47.03 executions per day. This transaction is therefore covered (it exists in the execution queue) and it has a depth of 100% (its number of executions in the execution queue is greater than or equal to the daily rate in the source system).

On the other hand, transaction SM30 has a source system daily execution rate of 47.5, but an execution queue rate of 10.85. Since the execution queue rate is lower than the source system rate, although it is covered it has a depth of only 22.85%.

3.11. Performing a Playback

Overview

The playback process can be run in approximately the same time as the recording phase. However, this time can be reduced dependent upon the amount of bots you have available as well as the size of the target system(s) you are playing back into. The transaction volume from the recording also has an impact upon the playback time.

Process Steps

- Preparing Playback
- Start Playback
- Monitor Playback
- Stop Playback

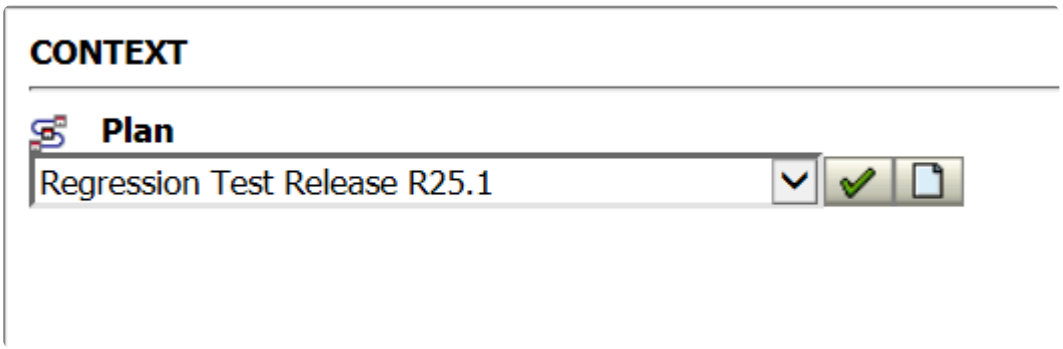
3.11.1. Preparing Playback

Overview

After the Execution Queue is built you are ready to run a a playback in Testimony, you need to first prepare and check the Target and Central systems. The following process steps set out the steps you need for a successful playback.

If you are going to be utilising a double playback then you should also ensure you check out the Double Playback topic.

You should always ensure you are in the correct test plan before commencing activities for double playback by selecting the correct Test Plan from the drop down and click on the green tick.



Process Steps

1. Confirm Target system refresh steps – As per your organisations specific steps. Template steps found in this list here.
2. Run Predicative Difference Analyser
3. Check Central System batch processors
4. Activate playback enhancements
5. Execute Check Steps
6. Reset Date and Time of Target system
7. Start the Bots

3.11.1.1. System Refresh Steps

Below are the steps that should be completed when refreshing the target system from the source system to prepare it for playback. Care should be taken to review these steps against the steps already documented for refreshing systems in the current landscape. Each organisation should create its own list of refresh steps considering the recommended steps for Testimony as below.

Note that some steps are only required if changing the time of the system which is recommended with TimeShiftX.

> ! If upgrading the target system it is important that the recording enhancements are deactivated on the target using program /BTI/AUT_DEACT_ENH via SE38 on the copied system

> ! If the target system is a brand new SAP system ensure that you request the license key from SAP for the period you recorded, otherwise time travelling may not be possible.

> ! Retain the same Client Number (MANDT) between Source and Target

> ✻ Note that parameters should be changed on each app server

#	Description	Required	Notes
1	Activate recording on the source system	Yes	
2	Put the source database into online backup mode	Yes	For details of how to coordinate the backup with the start of the recording, see the documentation here.
3	Perform a full database copy of the source database	Yes	
4	Once source database copy is complete, deactivate online backup mode	Yes	
5	Note the exact timing of the copy so that the transfer to repository step can use this	Yes	

	date and time		
6	Setup SAP target system (system profiles etc)	Yes	
7	Optionally setup the target system in a “ring-fenced” network	Optional	
8	Install TimeShiftX onto application and database server(s) on the target system	TimeShiftX Only	
9	Ensure batch jobs are deactivated when target SAP system is started (profile setting)	Yes	The recommended option is to run btctrns1 (in SE38) to set all batch jobs to Released/Suspended mode, and then go and delete them all from SM37. Only exception should be the event-dependent RDD* jobs, as without these the transports for the release (line 24) won’t work.
10	Start-up SAP target database and app server(s)	Yes	
11	Disable/deactivate recording enhancements on the target using program /BTI/AUT_DEACT_ENH via SE38 on the copied system	Yes	**If you are upgrading your system ensure you run this program before starting the upgrade process**
12	Check parameter sapgui/ user_scripting	Yes	This value needs to be TRUE which allows the bots to perform the playback properly. Only in playback system.
13	Check parameter sapgui/ user_scripting_per_user	Yes	This value needs to be FALSE which allows the bots to perform the playback properly. Only in playback system.
14	Check parameter sapgui/ user_scripting_set_readonly	Yes	This value needs to be FALSE which allows the bots to perform the playback properly. Only in playback system.
15	Check parameter login/ disable_password_logon	Yes	This value needs to be 0 which allows the bots to login to the playback system. Only in playback system.
16	Check parameter login/ disable_multi_gui_login	Yes	This value needs to be 0 this allows multiple logons. Only in playback system.
17	Check parameter rdisp/ tm_max_no	Yes	As Testimony will be logging on with the user load from the source (which could be production) this should be the source setting +50% Only in playback system.
18	Check parameter rdisp/ gui_auto_logout	Yes	This value needs to be 0. This ensures that the bots and their logged on users cannot be automatically logged off by the user

			due to time-limits (for example, if the playback is paused or other issues arise)
19	Check parameter rdisp/ max_wprun_time	Yes	Ensure this value is at least the value set in the production system
20	Check parameter abap/ buffersize	Yes	Ensure this value is at least the value set in the production system
21	Check parameter ztta/ parameter_area	Yes	This value needs to be at least 64000. The value checked can be adjusted by changing PLAYBACK_PARAM_AREA in the General Parameters. Only in playback system.
22	Check parameter snc/ enable	TimeShiftX Only	This value needs to be 0 this is required where TimeShiftX is being operated. Only in playback system.
23	Run post-processing steps such as RFC destination re-pointing, file system re-pointing, printer re-pointing etc.	Yes	
24	Do not delete/remove source SAP users from the target system	Yes	Testimony requires the users from the source system to playback correctly
25	Optionally run BDLS if system ID is to be renamed (e.g. PRD to QA1)	Optional	
26	Optionally use Basis Technologies System Copy GT if BDLS takes quite a long time	Optional	
27	Deploy changes that form the release being tested (e.g., transport requests or perform upgrade) to target system	Yes	
28	Deploy Testimony playback enhancement transport to target system	Optional	If the Testimony playback enhancements transport was not applied to the source system you will need to apply that transport here
29	Ensure Target RFC User exists	Yes	Ensure a **system user** with the role /BTI/AUT_TARGET_RFC (ensure profile role is generated) This user is for the RFC destination defined in the central system.

			While setting up bots for the first time you should also use the role /BTI/AUT_BOT_SIMULATION This is an optional Role that is only required for the BOT Simulation Program. This role provides auths for creating Users, assigning Roles to Users and Deleting Users, which is required during the Bot Simulation and can be removed after the simulation is complete. **Checking the RFC setup with the user at this point is best practice.**
30	Ensure Meta Data RFC User exists	Yes	Ensure a **system user** with the role /BTI/AUT_NCO_METADATA_ACCESS (ensure profile role is generated) This user is for playing back RFCs and collecting the meta data to ensure that they operate correctly the suggested user name is /BTI/AUT_RFM although this can be changed in the General Parameters.
31	Ensure Batch Job User is correct	Yes	Before starting the playback, Testimony will verify the validity and existence of users that are present as creators of batch jobs in the recording data. If one of these users fails this check (i.e. because the user doesn't exist or is not valid), Testimony will use this user to execute the batch job, we would suggest configuring the default batch user in the General Parameters. The user name defaulted into the general parameters is /BTI/AUT_BTC
32	Set the workflow batch user's password	Yes	Use the transaction SWU3 to set the RFC workflow user (WF-BATCH) to the Testimony password as set in the general parameters under the CHECK_DEFAULT_PWD parameter.
33	Disable Personas	Optional	At present, the use of Personas is not supported by Testimony. The main reason for this is that with Personas, a lot of processing happens on the front end, and Testimony, as a tool which records at the back end, cannot capture this information. In order to disable Personas for the playback, in the target system go to transaction /n/personas/settings and add the following entry: DISABLE_WEBGUI = X.
34	Take a backup of the target system	Yes	It is recommended to take a backup as this means the target system can be easily restored back to this state for a second run which means if anything goes wrong the system can be quickly restored and a second playback run, without having to repeat all of the above steps again.
35	Check the SAP system for time inconsistencies	TimeShiftX Only	Use the program RSDBTIME via transaction SE38 to check time inconsistencies between the SAP servers

36	Check current time travel status	TimeShiftX Only	Use the TimeShiftX command “tsx list” to check the current time travel status of your system this will avoid running multiple TimeShiftX commands which might lead to unexpected times on systems and might cause systems to shutdown.
37	Required if the SAP HANA version is HANA 2.0 SP04 or lower and Multi Tenant database is used. Run the SQL commands	TimeShiftX Only	Run the following SQL commands in HANA Studio on the System database (Multi Tenant database) ALTER SYSTEM ALTER CONFIGURATION (‘nameserver.ini’, ‘SYSTEM’) SET (‘delta’, ‘preallocated_nodebuffers’) = ‘1’ WITH RECONFIGURE ALTER SYSTEM ALTER CONFIGURATION (‘indexserver.ini’, ‘SYSTEM’) SET (‘delta’, ‘preallocated_nodebuffers’) = ‘1’ WITH RECONFIGURE
38	If running HANA 2.0 SP4 or lower only. Run the SQL commands	TimeShiftX Only	Run the following SQL commands in HANA Studio on the Tenant database ALTER SYSTEM ALTER CONFIGURATION (‘indexserver.ini’, ‘SYSTEM’) SET (‘delta’, ‘preallocated_nodebuffers’) = ‘1’ WITH RECONFIGURE
39	If running a SAP HANA database server set the environment variable HDB_TIMER=system	TimeShiftX Only	
40	Shut-down SAP target system Application Servers	TimeShiftX Only	
41	Shut-down SAP target system Database Servers	TimeShiftX Only	
42	Setup virtual clock for the ADM users in TimeShiftX on both the target database and app servers	TimeShiftX Only	It is recommended to use the offset option rather than set a specific time to make life easier for multiple app servers.
43	Ensure TimeShiftX virtual clock is set close to start of recording time just before playback commences	TimeShiftX Only	It is recommended to slow time down to ensure that there is time in case there is a delay starting playback or playback is slower than expected, time can be sped up if the midnight boundary is imminent or Testimony will pause and wait.
44	Start SAP target system Database Servers	TimeShiftX Only	
45	Start SAP target system Application Servers	TimeShiftX Only	
46	Check the SAP system time	TimeShiftX Only	Use the program RSDBTIME via transaction SE38 to check the time and check for any inconsistencies between the SAP

			servers

3.11.1.2. Predictive Difference Analyzer

Overview

The function of Predictive Difference Analyzer (PDA) is to allow the target system to be reviewed prior to a playback starting. It works by comparing the objects currently in the execution queue (from the recording of the source system) and comparing those objects with the target and highlighting any potential technical issues. A list of supported objects and check types is found here. When critical differences exist in the target system which cause either mass technical errors or mass failures in the playback, it is better to be proactive in fixing these issues rather than reactive when to a large number of failures/defects.

Any differences between the objects compared are flagged up to the operator and are marked as either:

- **Errors** – Differences in target that are likely to cause issues during the playback
- **Warnings** – Differences to be aware of but not necessarily playback issues

Process Steps

- **Preparation Phase** – Analyses all test scripts in the execution queue and creates a list of technical objects that will be compared between the source and the target
- **Comparison Phase** – For each object identified in the preparation phase retrieve the components from the source and target system. These objects are then compared and results are written for each object (pass, error or warning).
- **Result Phase** – Review all errors that could cause a critical issue on the target system during playback and either fix or remediate before starting playback. Warnings should also be reviewed as well.

> ! Predictive Difference Analyzer can still be utilized for a Baseline playback to ensure that the target system does not contain any critical issues. However it is then run for the Release/ Upgrade Playback once the target has been upgraded or had the release applied this can highlight potential technical issues as early as possible to be fixed before playback.

3.11.1.2.1. Supported Objects and Check Types

Object Types

Inbound RFCs
Dialog
Background/Batch Jobs

Check Types

Inbound RFC:

Existence
Header existence
RFC enabled flag
Signature existence
Syntax check of FUGR of the FM
Parameter checks:
Parameter existence
Paramater classification (exporting importing etc.)
Parameter mandatory status
Parameter length
Parameter data type (deep)

Background/Batch Program:

Existence
Syntax
Technical definition (executable, include, etc.)
Auth object
Screens that have parameters or select-options checked as above
Variant
Existence
Is it obsolete
Variant field existence
Variant field values

Dialog Transaction Code:

Existence
Start program
Syntax of Start program

Auth objects, activities and their values
All screens recorded checked as below
Existence
Type (mainscreen subscreen etc.)
Dimension (coordinates)
Field existence
Field type (I/O, Box, Push, Check etc.)
Field mandatory status
Field length
Field data type
Field visible length

3.11.1.2.2. Preparation Phase

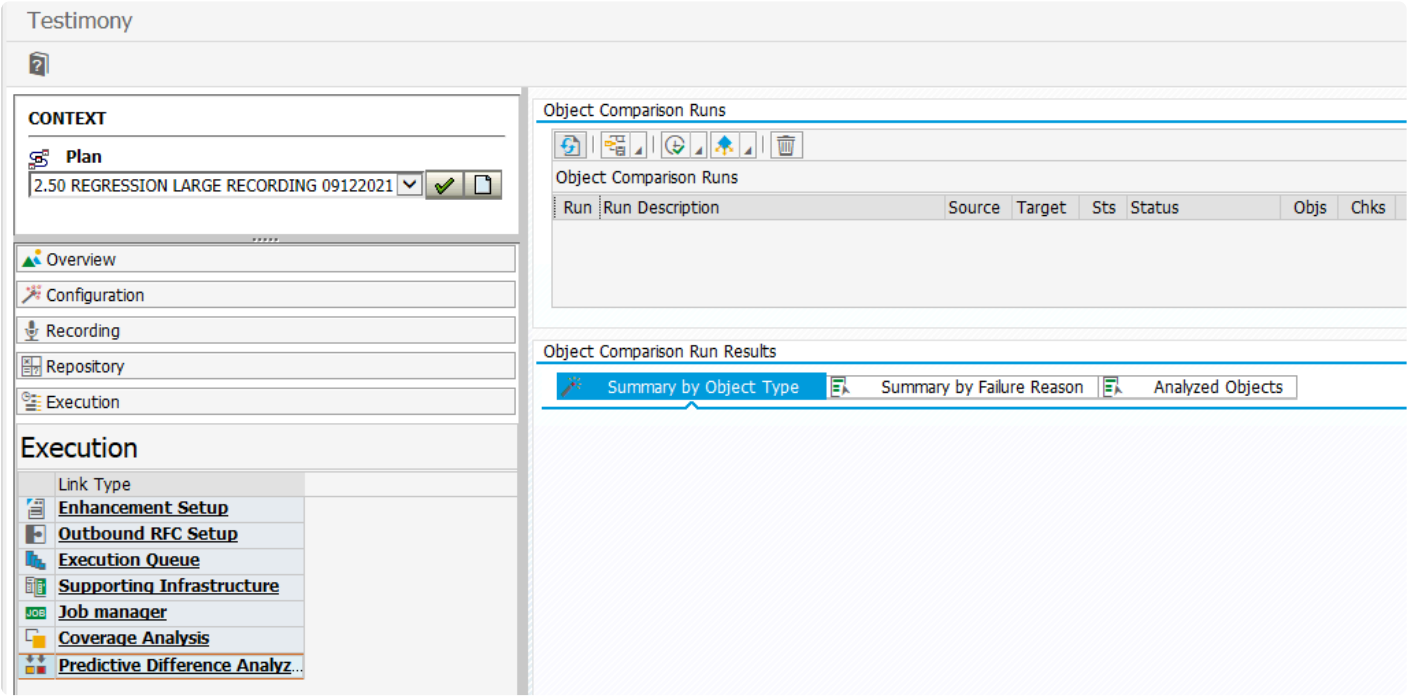
Overview

The preparation program analyses all test scripts in the execution queue and creates a list of technical objects that will be compared between the source and the target.

Execution

Ensure the correct test plan has been selected and navigate to the Predictive Difference Analyzer on the Execution drawer of the Test Plan you wish to analyze.

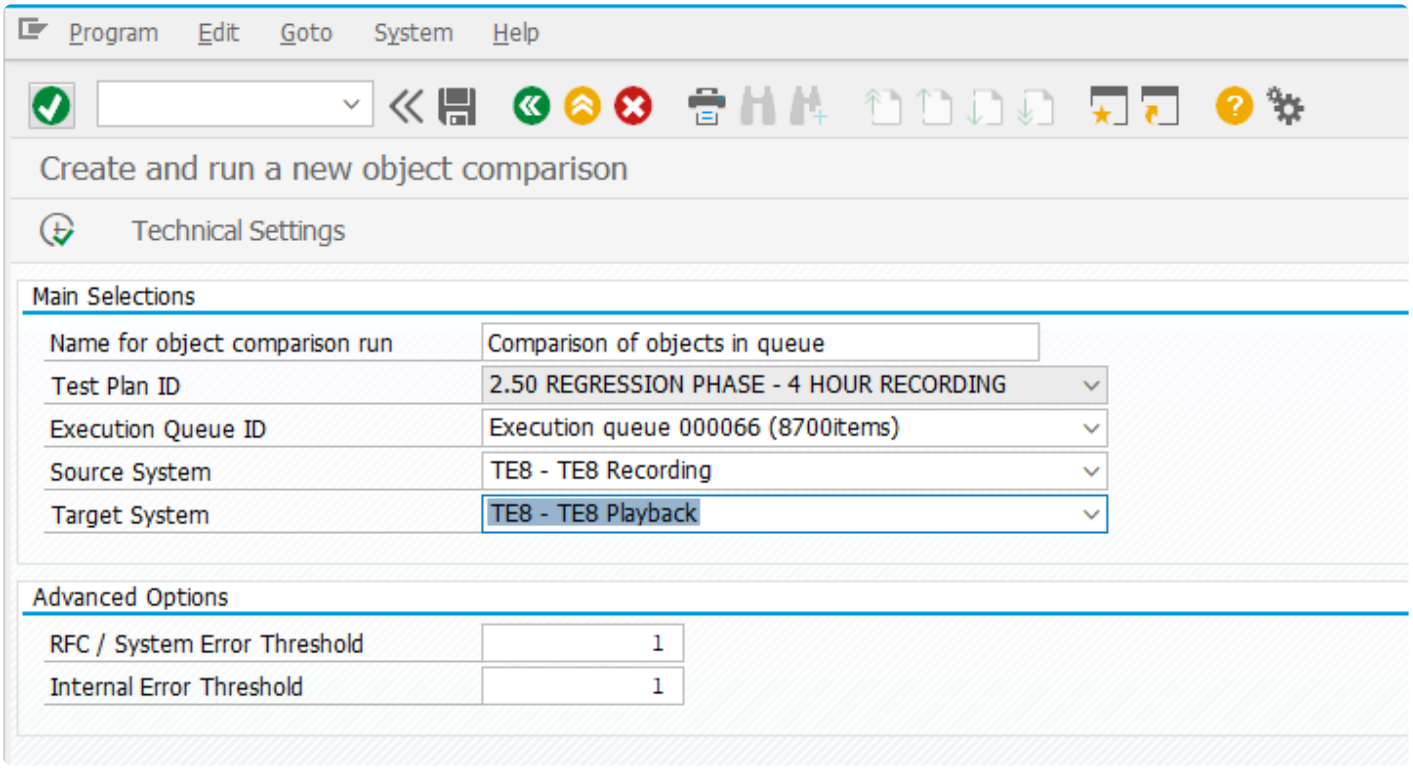
Click on the Prepare button  There are options for single thread or multi-thread via Diffuser.



Use the name for object comparison run field to give the run a name. Then select the execution queue and source and target systems to be used for this comparison.

The multi thread option can be used for large runs utilizing the technical settings button with an interval size of 1000 and selecting the number of batch jobs that are available.

The Advanced Options are for triggering a hard abort if either RFC/System Errors, or Internal Errors are seen. If set to 1 then the program will abort when these types of faults are encountered. If set to 0 then the processing will continue regardless.



Once this program has completed you can start the comparison phase.

3.11.1.2.3. Comparison Phase

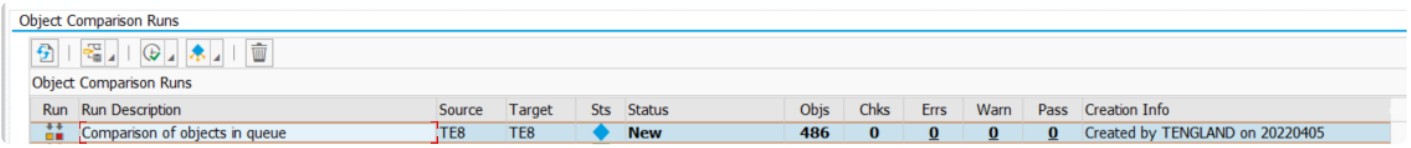
Overview

Running the comparison program takes each object identified in the preparation phase retrieve the components from the source and target system. These objects are then compared and results are written for each object (pass, error or warning).

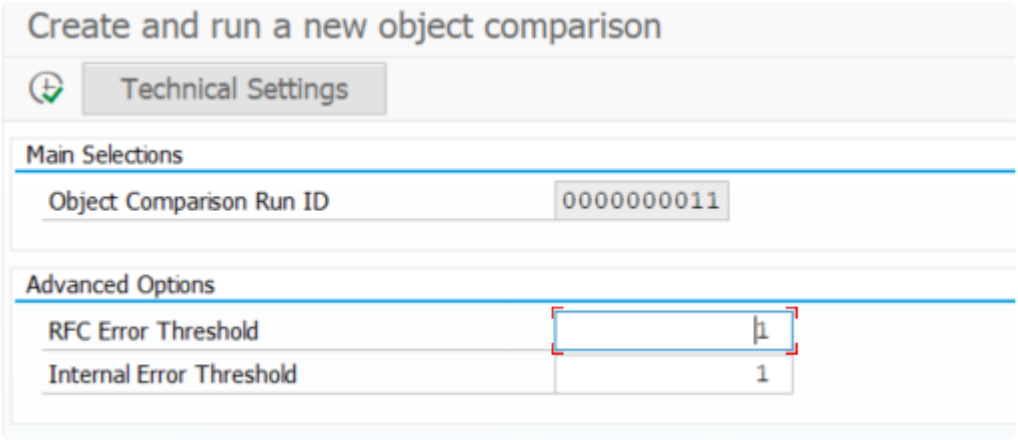
Execution

Select the run that you want to run the comparison for and then click on the Comparison button 

There are options for single thread or multi-thread via Diffuser.



The multi thread option can be used for large runs utilizing the technical settings button with an interval size of 1000 and selecting the number of batch jobs that you have spare. There are no other parameters that require an update.



Once this program has completed you can review the results in the Results phase

3.11.1.2.4. Results Phase

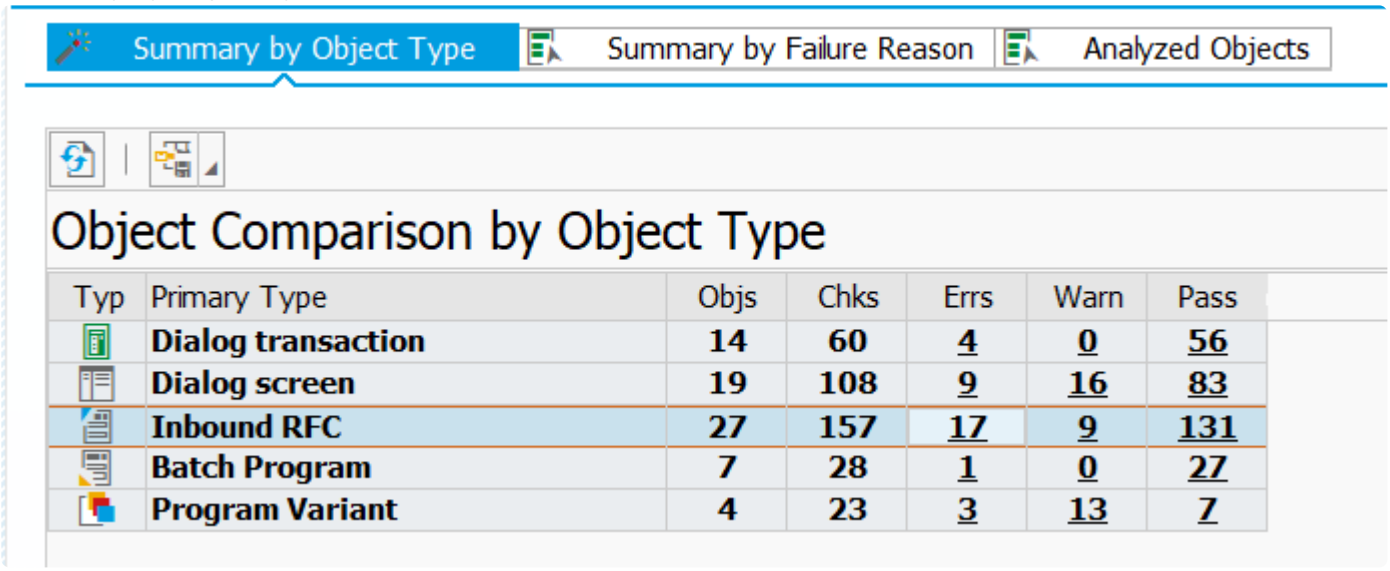
Overview

The object of this phase is to review all errors that could cause a critical issue on the target system during playback and either fix or remediate before starting playback. Warnings should also be reviewed as while these might not technically impact the playback they might be of interest with to the technical or testing team to help them review the quality of the target system.

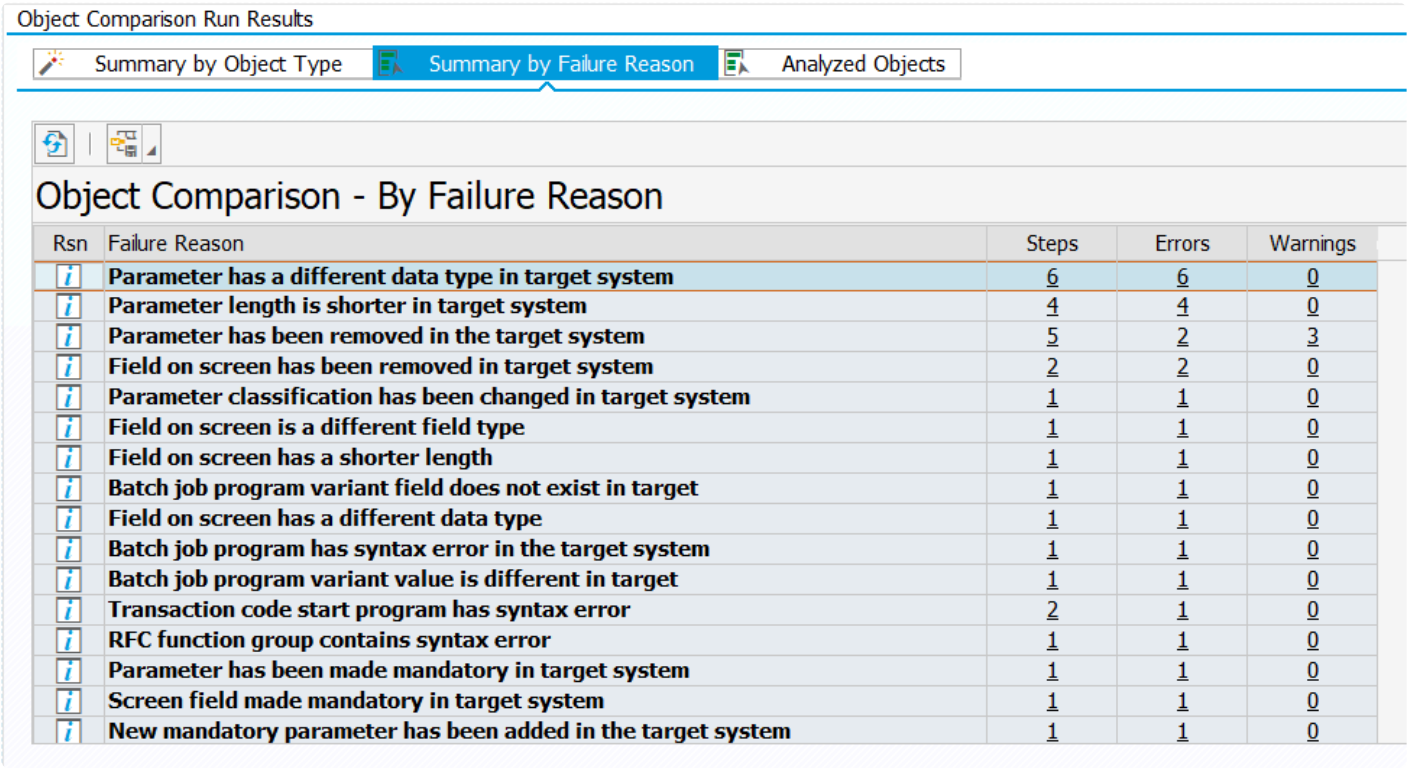
Reviewing Results

Once the comparison run is complete, note that the status should be complete. Double-click the Predicative Difference Analyser run you want to see the results for you can then switch tabs for different views of the results as below. Filters are available in all grids with options to download to a spreadsheet if required. Clicking the hotspots under the underlined numbers allows you to drill into the details for each pass, warning or error. The information buttons  also provide a lot of extra information and help on the issue. There is also an option in some circumstances to repair scripts see the topic Remediation of Test Scripts

Summary by Object Type

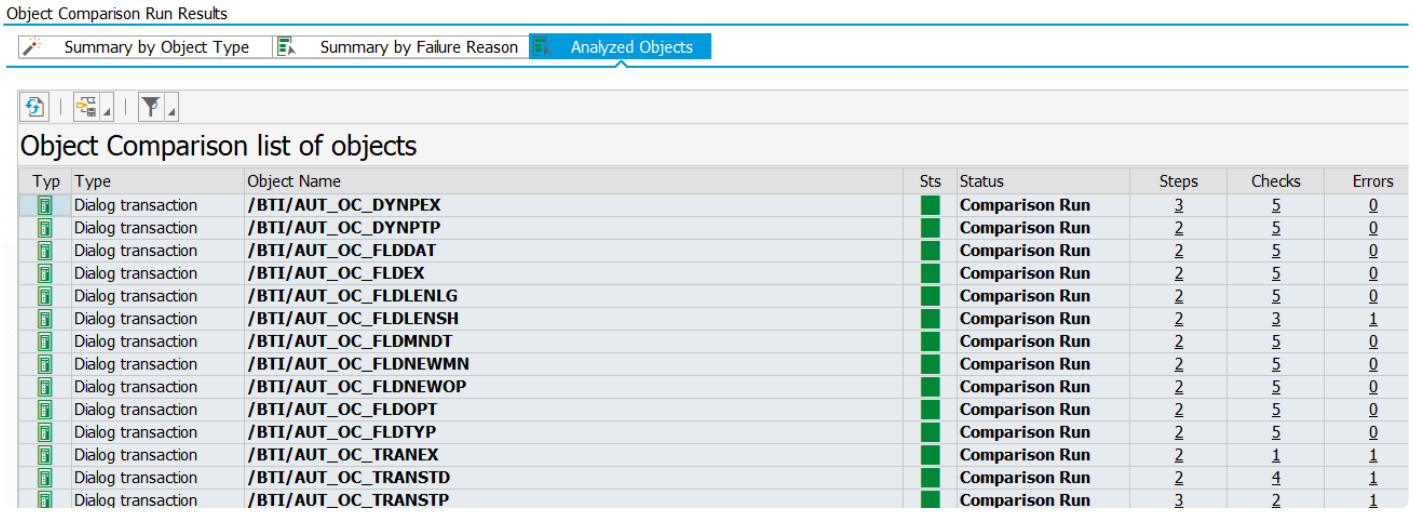


Summary by Failure Reason



Rsn	Failure Reason	Steps	Errors	Warnings
	Parameter has a different data type in target system	6	6	0
	Parameter length is shorter in target system	4	4	0
	Parameter has been removed in the target system	5	2	3
	Field on screen has been removed in target system	2	2	0
	Parameter classification has been changed in target system	1	1	0
	Field on screen is a different field type	1	1	0
	Field on screen has a shorter length	1	1	0
	Batch job program variant field does not exist in target	1	1	0
	Field on screen has a different data type	1	1	0
	Batch job program has syntax error in the target system	1	1	0
	Batch job program variant value is different in target	1	1	0
	Transaction code start program has syntax error	2	1	0
	RFC function group contains syntax error	1	1	0
	Parameter has been made mandatory in target system	1	1	0
	Screen field made mandatory in target system	1	1	0
	New mandatory parameter has been added in the target system	1	1	0

Analyzed Objects



Typ	Type	Object Name	Sts	Status	Steps	Checks	Errors
	Dialog transaction	**/BTI/AUT_OC_DYNPEX**		**Comparison Run**	3	5	0
	Dialog transaction	**/BTI/AUT_OC_DYNPTP**		**Comparison Run**	2	5	0
	Dialog transaction	**/BTI/AUT_OC_FLDDAT**		**Comparison Run**	2	5	0
	Dialog transaction	**/BTI/AUT_OC_FLDEX**		**Comparison Run**	2	5	0
	Dialog transaction	**/BTI/AUT_OC_FLDLENLG**		**Comparison Run**	2	5	0
	Dialog transaction	**/BTI/AUT_OC_FLDLENSH**		**Comparison Run**	2	3	1
	Dialog transaction	**/BTI/AUT_OC_FLDMNDT**		**Comparison Run**	2	5	0
	Dialog transaction	**/BTI/AUT_OC_FLDNEWMN**		**Comparison Run**	2	5	0
	Dialog transaction	**/BTI/AUT_OC_FLDNEWOP**		**Comparison Run**	2	5	0
	Dialog transaction	**/BTI/AUT_OC_FLDOPT**		**Comparison Run**	2	5	0
	Dialog transaction	**/BTI/AUT_OC_FLDTYP**		**Comparison Run**	2	5	0
	Dialog transaction	**/BTI/AUT_OC_TRANEX**		**Comparison Run**	2	1	1
	Dialog transaction	**/BTI/AUT_OC_TRANSTD**		**Comparison Run**	2	4	1
	Dialog transaction	**/BTI/AUT_OC_TRANSTP**		**Comparison Run**	3	2	1

3.11.1.2.4.1. Remediation of Test Scripts

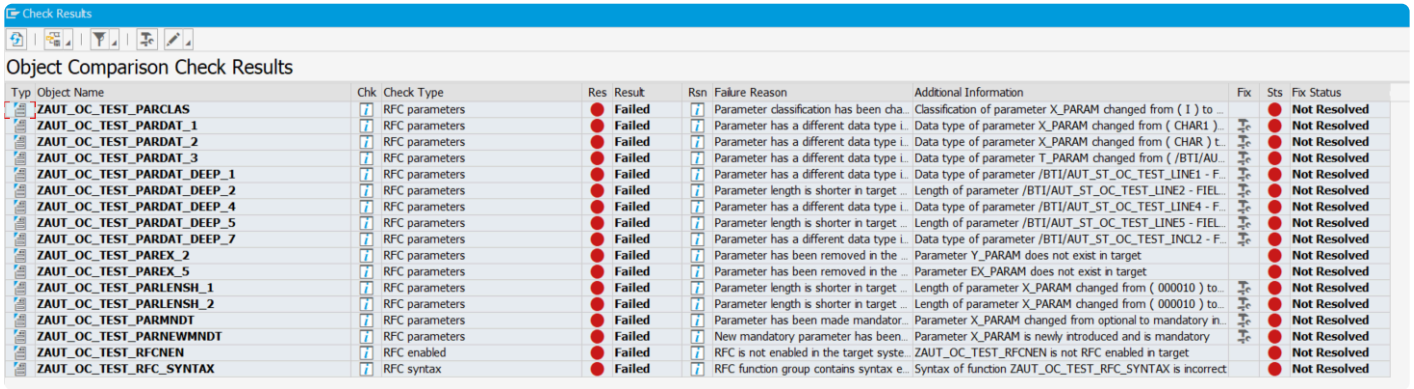
Overview

Testimony currently only supports adding of replacing fields to containers (with default values) for **Dialog Transactions**, and **Inbound RFC**. to help repair scripts that may not operate correctly due to changes in the target system.

Example 1

New mandatory parameter added to an RFC in the Target.
In this example, if the playback were to run then every test script containing this RFC would return a Technical Error. When Predictive Difference Analyzer detects this difference, it allows you to choose to remediate by updating all relevant test script steps and add this parameter to the input contained with a default value that provided. This allows the playback to continue with the ‘known issue’ pre-solved and gives a more complete replay.

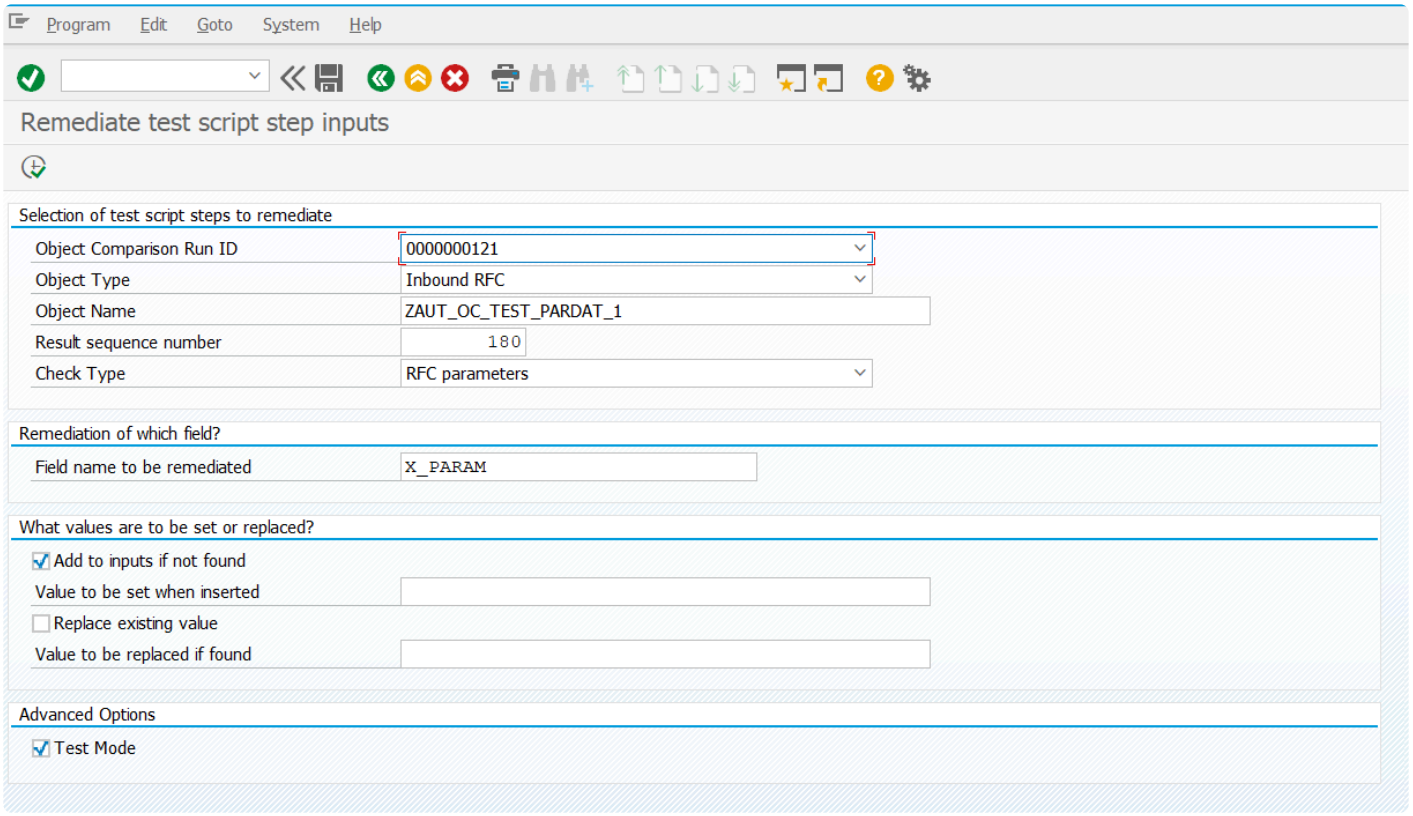
The screenshot below shows the Inbound FRC example again the information buttons also provide a lot of extra information and help on the issue. In this case the fix button takes you to a new screen (second screenshot below) to enable the repair of this script.


Check Results

Object Comparison Check Results

Typ	Object Name	Chk	Check Type	Res	Result	Rsn	Failure Reason	Additional Information	Fix	Sts	Fix Status
	ZAUT_OC_TEST_PARCLAS		RFC parameters		Failed		Parameter classification has been cha...	Classification of parameter X_PARAM changed from (I) to ...			Not Resolved
	ZAUT_OC_TEST_PARDAT_1		RFC parameters		Failed		Parameter has a different data type i...	Data type of parameter X_PARAM changed from (CHAR1)...			Not Resolved
	ZAUT_OC_TEST_PARDAT_2		RFC parameters		Failed		Parameter has a different data type i...	Data type of parameter X_PARAM changed from (CHAR) t...			Not Resolved
	ZAUT_OC_TEST_PARDAT_3		RFC parameters		Failed		Parameter has a different data type i...	Data type of parameter T_PARAM changed from (/BTI/AU...			Not Resolved
	ZAUT_OC_TEST_PARDAT_DEEP_1		RFC parameters		Failed		Parameter has a different data type i...	Data type of parameter /BTI/AUT_ST_OC_TEST_LINE1 - F...			Not Resolved
	ZAUT_OC_TEST_PARDAT_DEEP_2		RFC parameters		Failed		Parameter length is shorter in target ...	Length of parameter /BTI/AUT_ST_OC_TEST_LINE2 - FIEL...			Not Resolved
	ZAUT_OC_TEST_PARDAT_DEEP_4		RFC parameters		Failed		Parameter has a different data type i...	Data type of parameter /BTI/AUT_ST_OC_TEST_LINE4 - F...			Not Resolved
	ZAUT_OC_TEST_PARDAT_DEEP_5		RFC parameters		Failed		Parameter length is shorter in target ...	Length of parameter /BTI/AUT_ST_OC_TEST_LINE5 - FIEL...			Not Resolved
	ZAUT_OC_TEST_PARDAT_DEEP_7		RFC parameters		Failed		Parameter has a different data type i...	Data type of parameter /BTI/AUT_ST_OC_TEST_INCL2 - F...			Not Resolved
	ZAUT_OC_TEST_PAREX_2		RFC parameters		Failed		Parameter has been removed in the ...	Parameter Y_PARAM does not exist in target			Not Resolved
	ZAUT_OC_TEST_PAREX_5		RFC parameters		Failed		Parameter has been removed in the ...	Parameter EX_PARAM does not exist in target			Not Resolved
	ZAUT_OC_TEST_PARLENSH_1		RFC parameters		Failed		Parameter length is shorter in target ...	Length of parameter X_PARAM changed from (000010) to...			Not Resolved
	ZAUT_OC_TEST_PARLENSH_2		RFC parameters		Failed		Parameter length is shorter in target ...	Length of parameter X_PARAM changed from (000010) to...			Not Resolved
	ZAUT_OC_TEST_PARMNDT		RFC parameters		Failed		Parameter has been made mandator...	Parameter X_PARAM changed from optional to mandatory in...			Not Resolved
	ZAUT_OC_TEST_PARNEWMNDT		RFC parameters		Failed		New mandatory parameter has been...	Parameter X_PARAM is newly introduced and is mandatory			Not Resolved
	ZAUT_OC_TEST_RFCNEN		RFC enabled		Failed		RFC is not enabled in the target syste...	ZAUT_OC_TEST_RFCNEN is not RFC enabled in target			Not Resolved
	ZAUT_OC_TEST_RFC_SYNTAX		RFC syntax		Failed		RFC function group contains syntax e...	Syntax of function ZAUT_OC_TEST_RFC_SYNTAX is incorrect			Not Resolved

In this case values can be added to the new mandatory field or in the case of a change of the field old values replaced with a new value.


Program Edit Goto System Help
Remediate test script step inputs
Selection of test script steps to remediate
Object Comparison Run ID
0000000121
Object Type
Inbound RFC
Object Name
ZAUT_OC_TEST_PARDAT_1
Result sequence number
180
Check Type
RFC parameters
Remediation of which field?
Field name to be remediated
X_PARAM
What values are to be set or replaced?
Add to inputs if not found
Value to be set when inserted
Replace existing value
Value to be replaced if found
Advanced Options
Test Mode

3.11.1.3. Batch processes on the central system

Overview

When you start a playback, Testimony will submit a number of background processes on the central system. It is important that enough background work processes are available on the central system to accommodate the Testimony requirements, as well as leaving enough background work processes available for normal operation. (This is especially true if your central system is also a productive Solution Manager system.) The normal requirement is one background job for each bot plus an extra background job to manage the playback. The details on how this operates and is managed by Testimony is as below.

Detailed Information

When a playback is first executed an orchestrator job will start. The orchestrator job manages the whole playback process, and one of its functions is, firstly, to split the playback into blocks. Once the playback blocks have been built, the orchestrator will then start a number of worker jobs. The number of worker jobs that will be started is governed by the Testimony general parameter PLAYBACK_MAX_BLOCKS. The number of worker jobs (and hence the value of the parameter PLAYBACK_MAX_BLOCKS) should be set to the number of bots that you are using.

General parameters

Type	Parameter type	Technical name	Parameter description	Help	Current Value	Default Value
▶	**Playback**	**PLAYBACK_MAX_BLOCKS**	**Max. number of parallel server threads**	ℹ	25	3

Once these jobs have been started, the orchestrator will allocate blocks for processing to the worker jobs. Each worker job will then work through the scripts in its block, allocating them to bots for playback. Both the orchestrator job and the worker jobs will remain active for the whole of the playback.

This means, therefore, that the total number of jobs required on the central system by Testimony for a playback is 1 + PLAYBACK_MAX_BLOCKS.

The jobs can be seen in SM37 as below.

JobName	Spool	Job doc	Job CreatedB	Status	Start date	Start Time	Duration(sec.)	Delay (sec.)
/BTI/AUT_EXEC_QUEUE_PLAY			G181272	Finished	25.11.2021	17:27:27	2.997	0
ogOIb0V07joJmPC3rr60MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPC5Qmj0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPC6Ma}0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPC7}6X0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPC78Uq0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPC8nNo0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPC9a0B0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCAN9J0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCBBdb0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCC3{T0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCD0LC0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCDxDF0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCEoU20MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCFhV10MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCGYwX0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCHPcr0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCIGaa0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCJ8i30MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCK5rp0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCKv2c0MG			G181272	Finished	25.11.2021	17:27:34	2.966	0
ogOIb0V07joJmPCLsa10MG			G181272	Finished	25.11.2021	17:27:35	2.965	0
ogOIb0V07joJmPCMilN0MG			G181272	Finished	25.11.2021	17:27:35	2.965	0
ogOIb0V07joJmPCNdcK0MG			G181272	Finished	25.11.2021	17:27:35	2.965	0
ogOIb0V07joJmPCOVsA0MG			G181272	Finished	25.11.2021	17:27:35	2.965	0
ogOIb0V07joJmPCPQr50MG			G181272	Finished	25.11.2021	17:27:35	2.965	0
*Summary							77.142	0

The orchestrator job is called /BTI/AUT_EXEC_QUEUE_PLAY, and the worker jobs are given a dynamically-generated name.

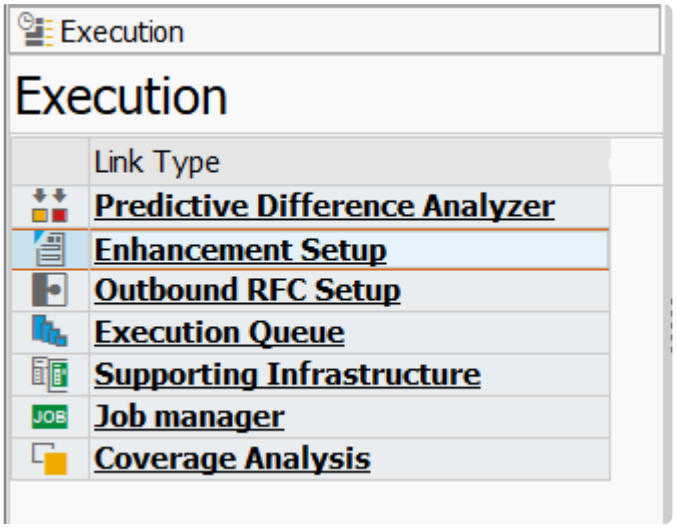
3.11.1.4. Activate Playback Enhancements

Overview

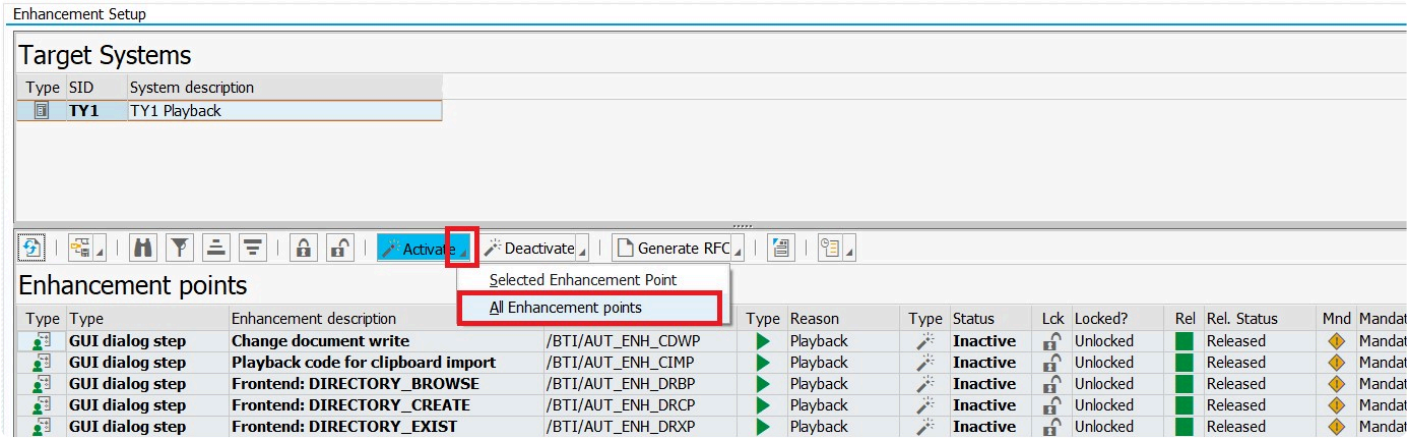
Before you start a playback the Testimony playback enhancements must be activated to ensure that Testimony can execute the playback correctly.

Process Steps

Navigate to the Execution queue and then select the Enhancement Setup option.



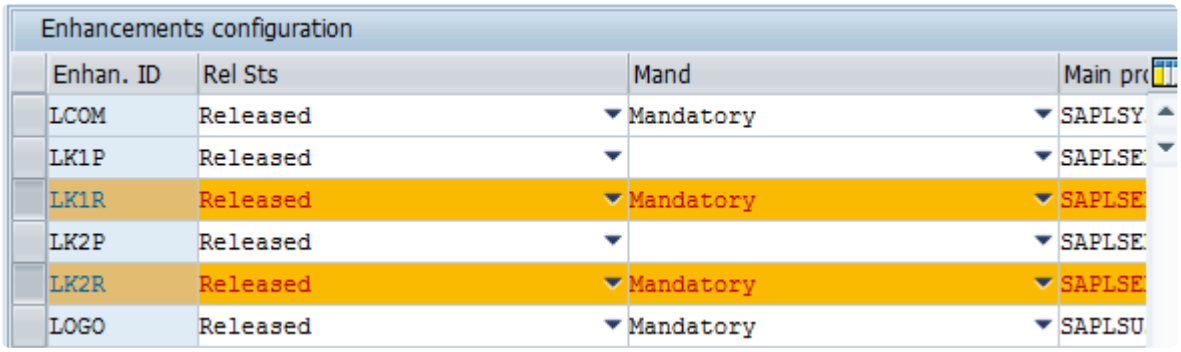
Next click on the “Activate” button and select “All Enhancement Points”. When Testimony is first installed, it comes with a set of playback enhancements which must be activated before starting a playback. While many of these are delivered as mandatory enhancements, some are delivered as optional enhancements because they won’t be relevant for all customers.



If there are areas of functionality that are important to you which are covered by optional enhancements, then rather than have to individually activate them you can change the configuration so that these are marked as mandatory.

Updating enhancement configuration

Enhancement configuration is updated via table /BTI/AUT_C_ENHT, which can be edited using transaction SM30. In the table, optional enhancements have a blank entry in the “Mand” Column. Change this entry to Mandatory for the enhancements you want to change.



> ! Make sure that you are changing the setting for the correct enhancement. Pay particular attention to the type of enhancement (Recording or Playback) and the name of the enhancement.

3.11.1.5. Execute Check Steps

3.11.1.6. Reset Target Date and Time

Overview

As many SAP transactions are date-bound the date entered on the screen is validated against the system date. For example, if Testimony records a financial posting on 1st May, but the playback is run on 5th May, then the posting transaction will fail in the playback with a “Cannot post in the past error” as the playback will try to post with the recorded date of 1st May.

Therefore, to enable the playback to run smoothly then the date and time, primarily the date of the system needs to be reset to the same as the recording. To achieve this Basis Technologies recommends that you utilise TimeShiftX as per the steps below.

Process Steps

Firstly, you should ensure that you are familiar with the manual for the correct version of the TimeShiftX product for your Operating System.

Identify all the OS user accounts that run the SAP system, the DB, and any OS-level users that facilitate the operation of the system, such as SIDADM, SAPserviceSID, SMDAgent, etc . Time travel will need to be activated for these accounts later in the process.

Shutdown both the SAP system app servers and the Database.

Note: replace sidadm with the OS User accounts identified as running the SAP system.

```
# su - <sid>adm
$ tsx set -u <sid>adm -d  -4
$ date
```

The time and date should be four days in the past. The number after the -d indicates how many days you want to shift time by -4 means minus 4 days.

When the SAP target system is restarted you should check the target system is time travelling correctly as per the topic here.

Again should you require more complex time travelling on your system please consult the TimeShiftX manual for your version of that product.

Troubleshooting

To reset the time travelling of a system the syntax is as below:

```
$ tsx reset
```

```
$ date
```

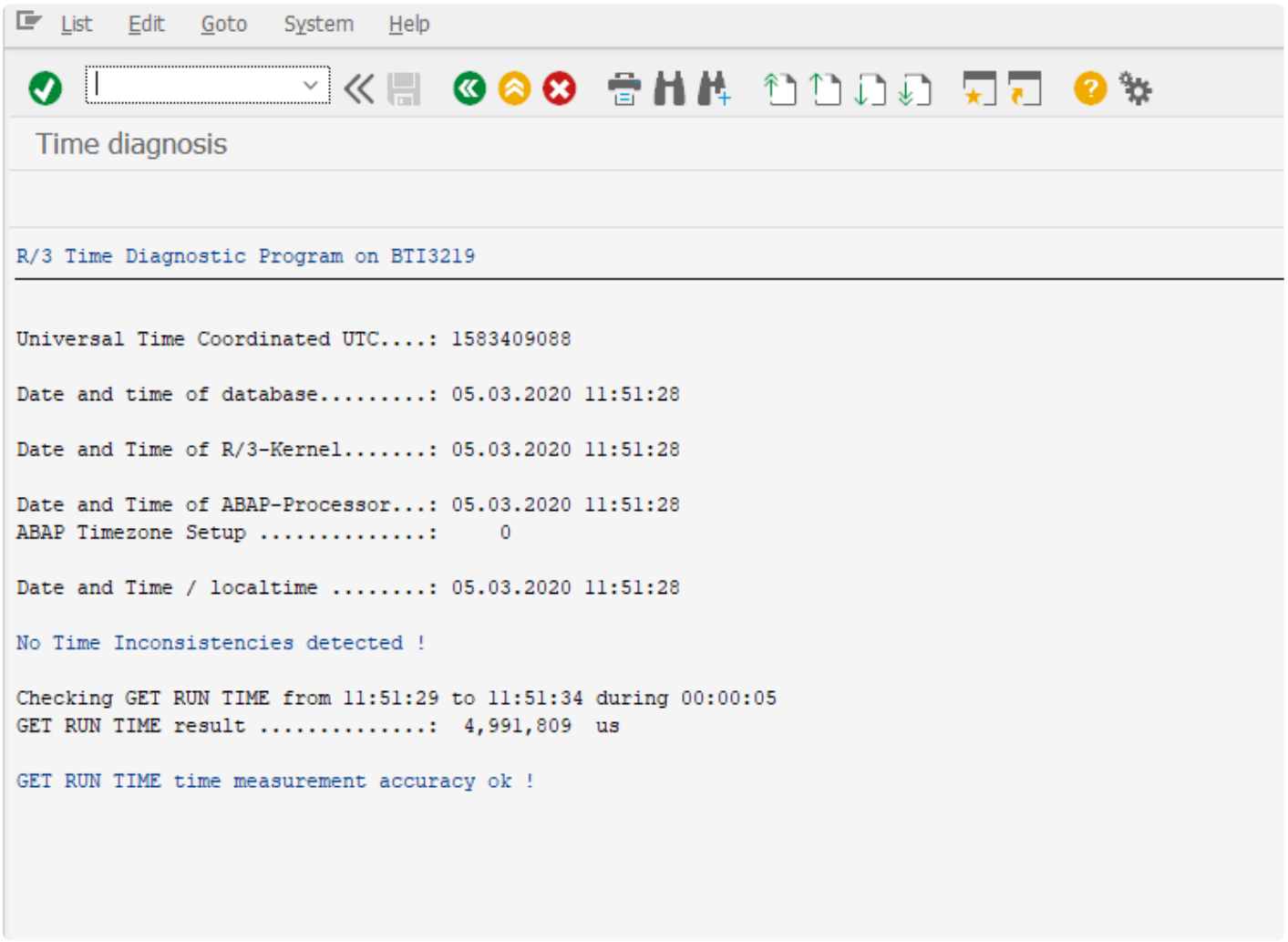
The present time and date should be displayed.

If you have serious issues then please email support@vornexinc.com

3.11.1.6.1. Checking time travel on SAP

The standard SAP program RSDBTIME can be run via SE38 to verify the system time and date on the different components that make up the SAP system.

This program should be run before and after time travelling to firstly verify there are no time mismatches on the system before time traveling and secondly after time traveling is started to ensure that the time travelling is working on all parts of the system.



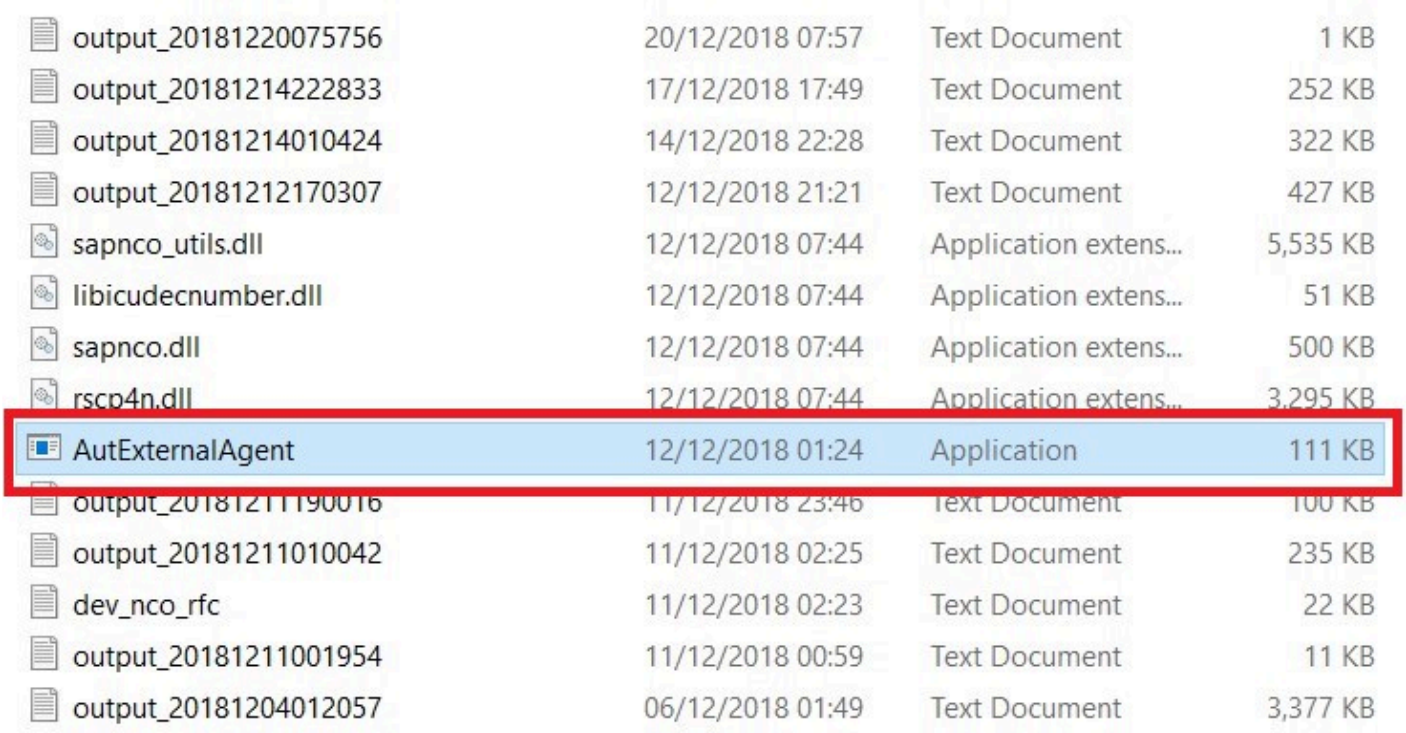
3.11.1.7. Start the Bots

Overview

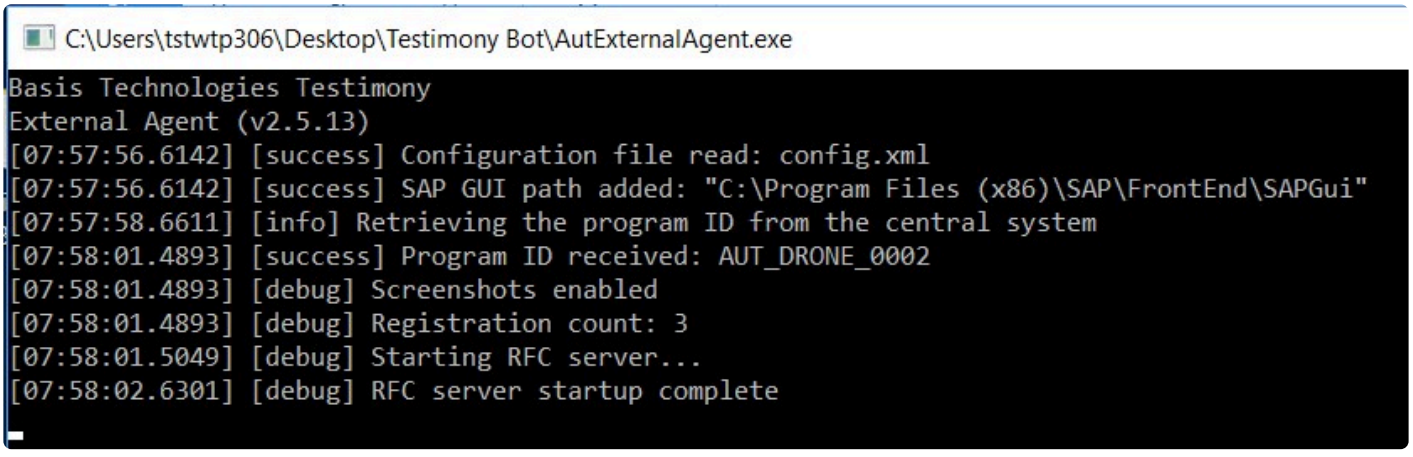
The playback is executed via the Testimony Bots that run on Windows machines. The size of the recording will drive the number of Bots required.

Process Steps

The first step is to navigate to the relevant folder on your Bot machines (local or VM) and double-click on the AutExternalAgent executable file.



The Bot will start in a command prompt window.

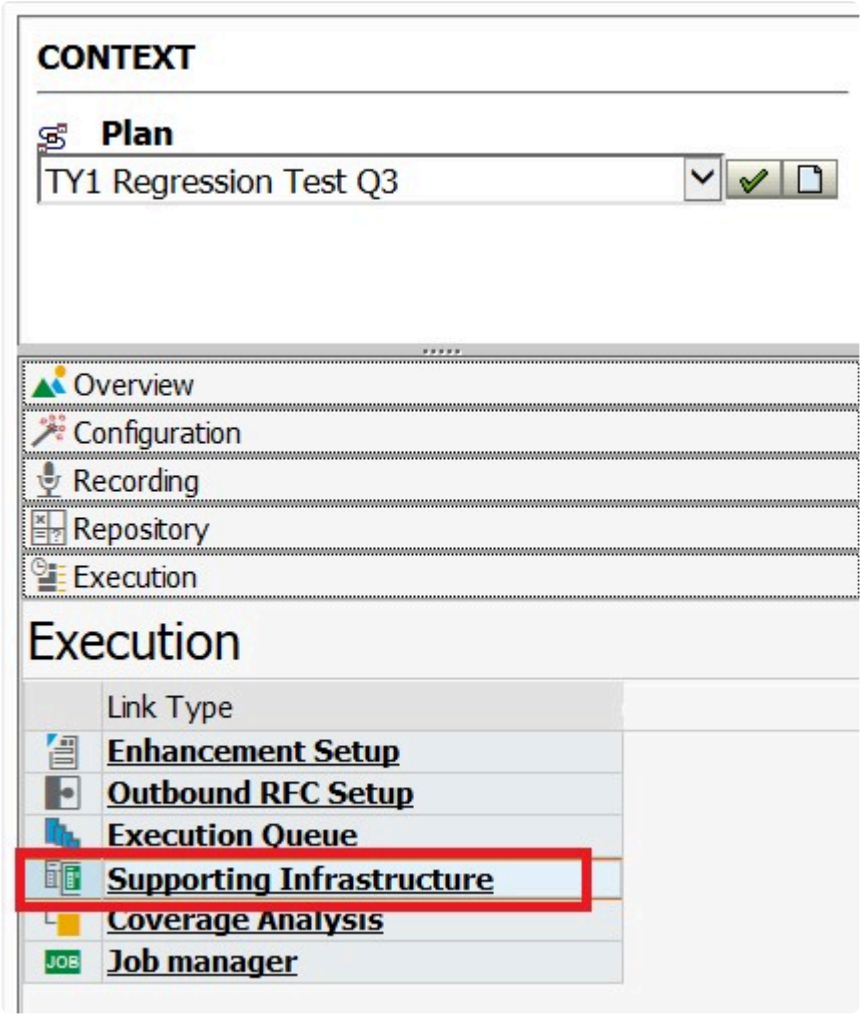


Startup is complete once you see the “RFC server startup complete” message.

Repeat for all Bots that you want to start for this playback.

Validate the BOTs are running with their screens unlocked. This step is essential for recording screenshots. The screenshots require an active graphics context, which is disabled when the Windows desktop screen is locked.

You can verify that the Bots have successfully registered with Testimony by navigating to Execution -> Supporting Infrastructure in the central system.



In the right-hand pane, you should see that the Bots you started have a status of Ready.

Additional Supporting Systems

Typ	Sys Type	Sts	Agent Status	System ID	Terminal ID	Host User	ID	IP Address	Registered Name	OS	Op. System
	Playback Bot		**Inactive**	**KMC laptop**	DESKTOP-S599L7R		1				Windows 8.1
	Playback Bot		**Inactive**	**DC Laptop**	XPS13-DCAPO		5				Windows 10
	Playback Bot		**Inactive**	**BP Connect**	BTI1102		6				Windows Server 2008
	Playback Bot		**Ready**	**VM1**	SKLRWX7BVDPF2-L		2		**AUT_DRONE_0002**		Windows 10
	Playback Bot		**Inactive**	**VM2**	USUNWX745FPF2-L		3				Windows 10
	Playback Bot		**Inactive**	**VM3**	SKLRW787ZVMX1-L		4				Windows 7
	Playback Bot		**Inactive**	**VM4**	USUNWX1F6DPF2-L						Windows 10

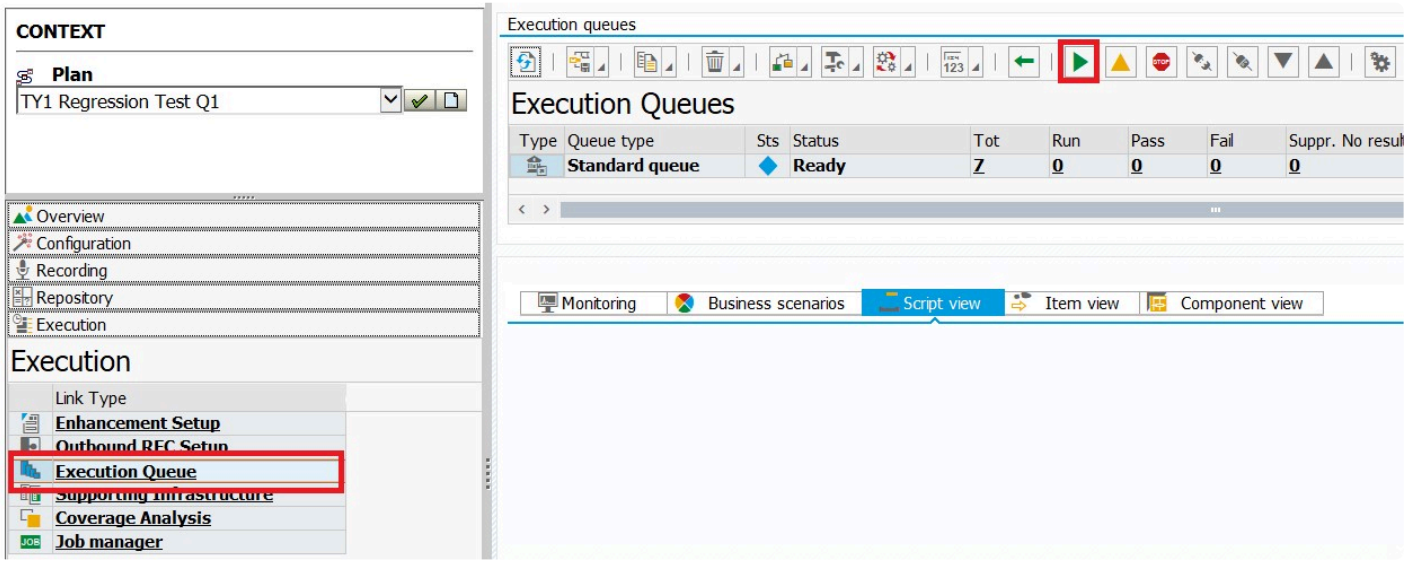
3.11.2. Start Playback

Overview

Once the execution queue is built from the repository, the target system is prepared and the Bots are started, the execution queue can be started

Process Steps

1. Select Test Plan to use
2. Navigate to the Execution Queue
3. Double check the Check Steps
4. Click on “Start Execution Queue” button

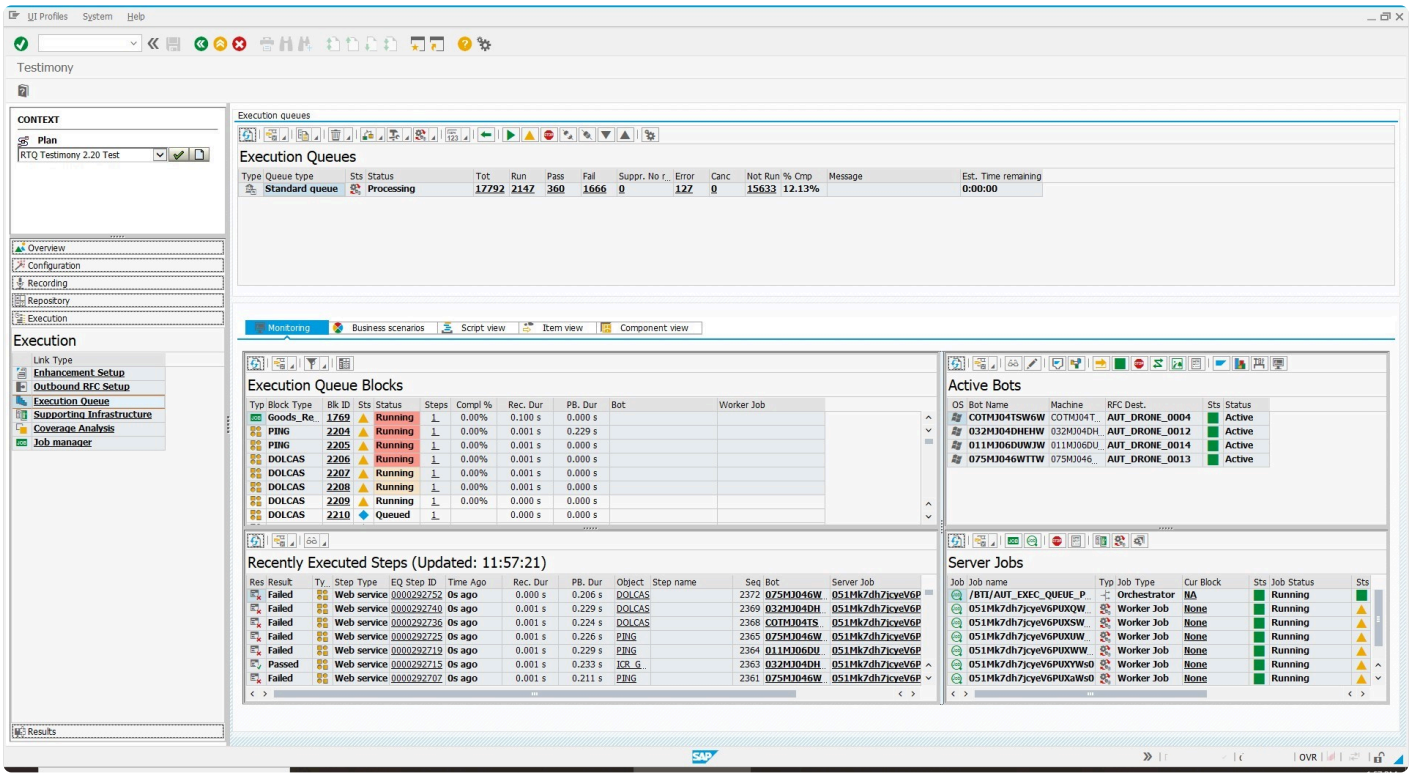


When the “Start Execution Queue” button is selected, the check steps will be “checked” to ensure that all have passed ok. If not, you will be prompted as to whether you want to proceed. The first step of the job is to run the “preparation steps”. You do not need to run these manually as the execution queue will automatically do this. If any issues arise as a result of this, you will need to investigate the preparation step logs in order to determine if you would still like to proceed or not.

Once the status of the execution changes to “In Process”, this means that the queue is running. At this point, you should begin to monitor the playback.

3.11.3. Monitor Playback

From within the Execution Queues window, click on the Monitoring tab to see status information for the playback.



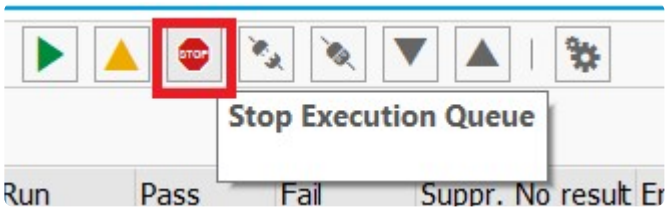
This shows information on (reading clockwise from top-left) the currently-running execution queue blocks; the active Bot(s); the jobs that are running in the central system to manage the playback; and the most recently executed steps.

You can refresh the top-level to update all the individual results windows, or you can refresh each window independently. You can also log into the Target system to view currently running activities.

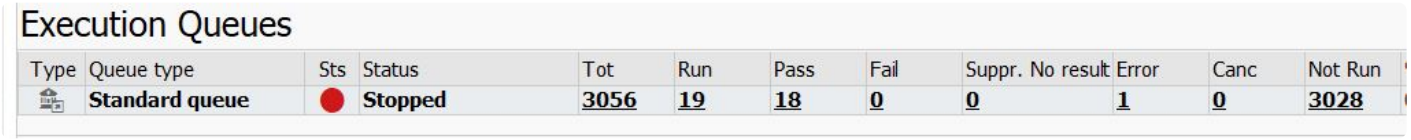
If you have visibility of the bot machines, you should start to see the bots performing the playback steps. This might be via the SAP GUI, but equally the console will show you if it is running other steps such as RFC's and batch jobs.

3.11.4. Stop Playback

In normal circumstances, you would allow a playback to complete running all of its scripts, at which point it will stop automatically. However, if you need to stop a playback early for any reason, then you can do so by pressing the Stop Execution Queue button.



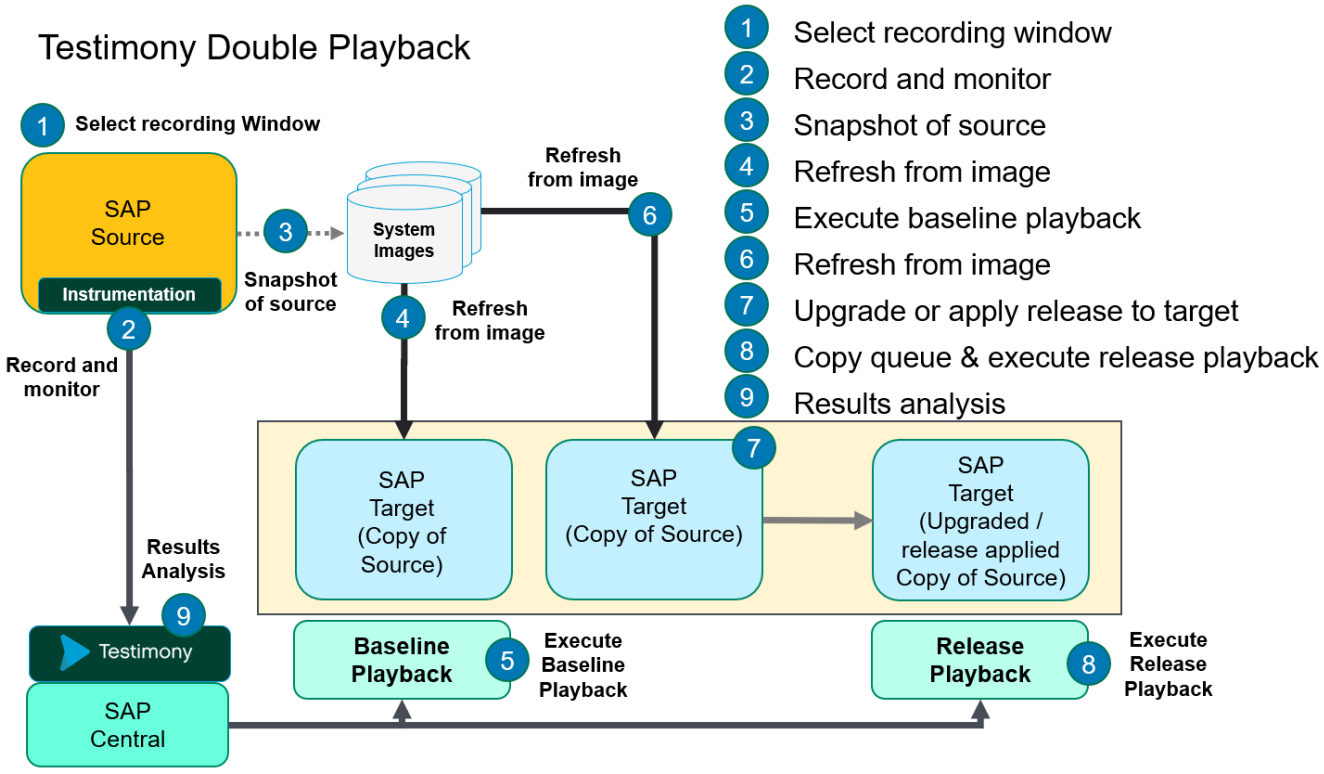
The playback will enter status Stopping, followed by Post-Processing and then Stopped.



3.12. Double Playback

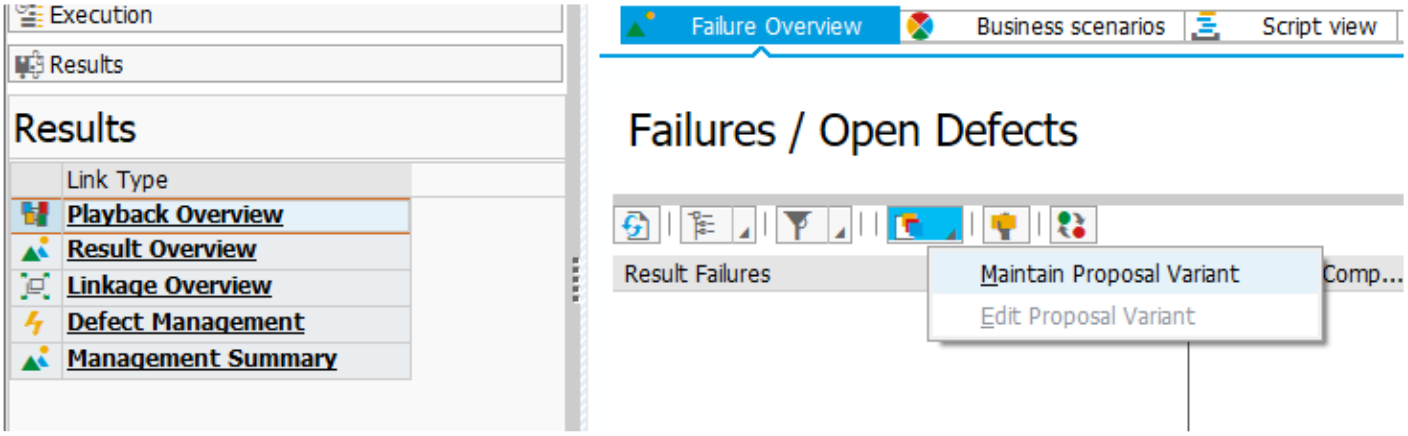
One of the challenges with Testimony producing a clear set of defects that are purely connected to the release being tested is the “noise” that can exist in target systems from slight differences in environmental setup of these systems. While suppressions and exclusions can help, this can take time to setup to have the desired impact and runs the risk of masking genuine regression defects.

The Double Playback option helps to filter out this “noise”. This operates by running a “baseline” playback into an identical copy of the source system, with no changes or release candidate applied. The first execution queue is then copied and a second playback is then required. This time the identical copy of the source system is upgraded or has the changes or release candidates applied. The failures in the second (release) queue are compared to the baseline. If the failures occurred in the second playback for the same reason, these are excluded from defect proposal generation for the second queue therefore reducing the noise. The overview of the steps to perform a Double Playback are show in the diagram below:

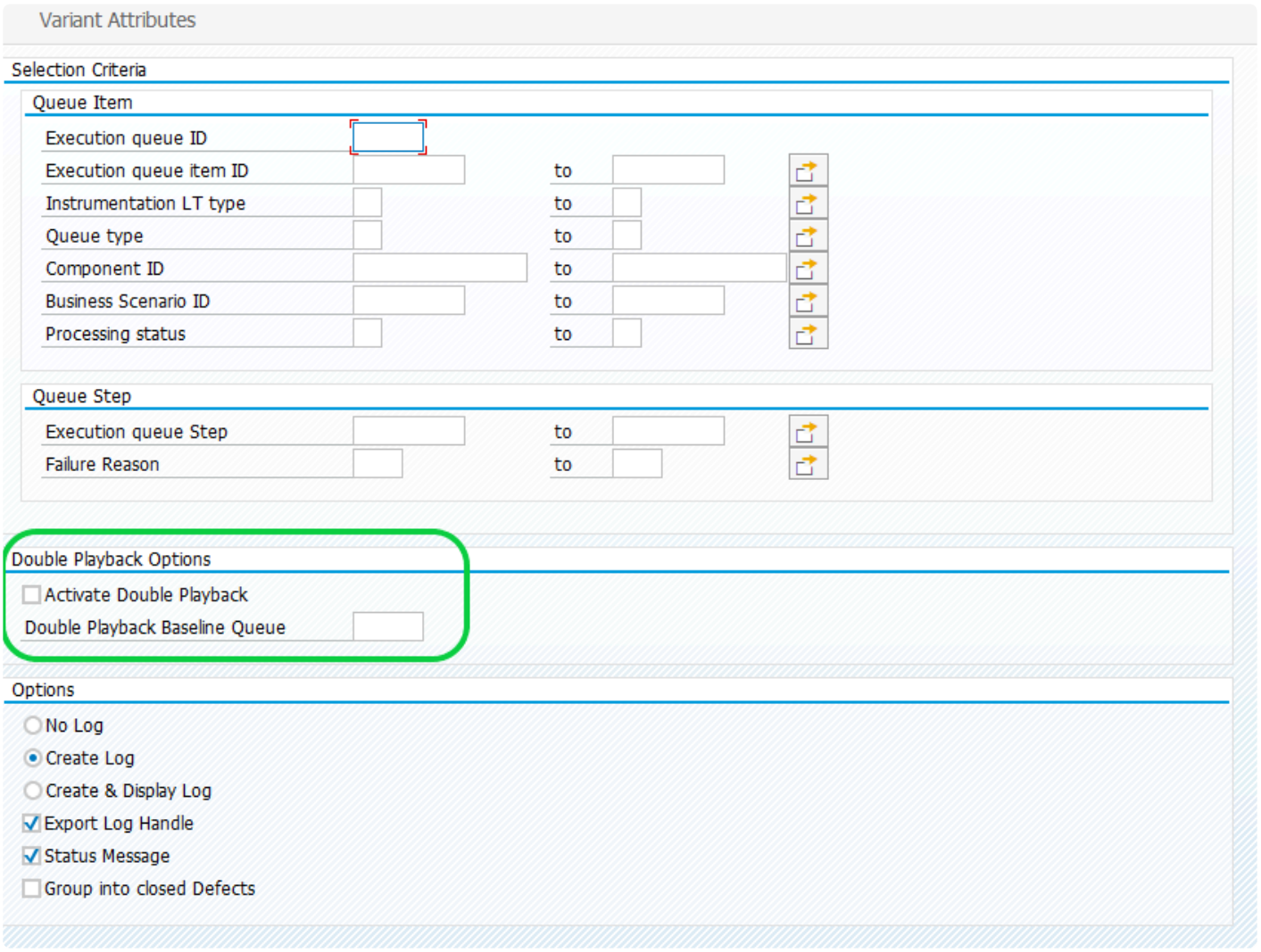


The comparison of the two playback runs occurs with the Defect Proposal

Select a different variant prior to the Defect Proposal.



Ensure that the Double Playback check box is selected and enter the Execution Queue ID of the “Baseline” against the Double Playback Baseline Queue



You can also run the Defect Proposal as a standalone program /BTI/AUT_PROPOSE_DEFECTS

3.12.1. Double playback Preparation

Parameter settings

Before starting the second playback, the following parameters should be set in Configuration —> General Parameters.

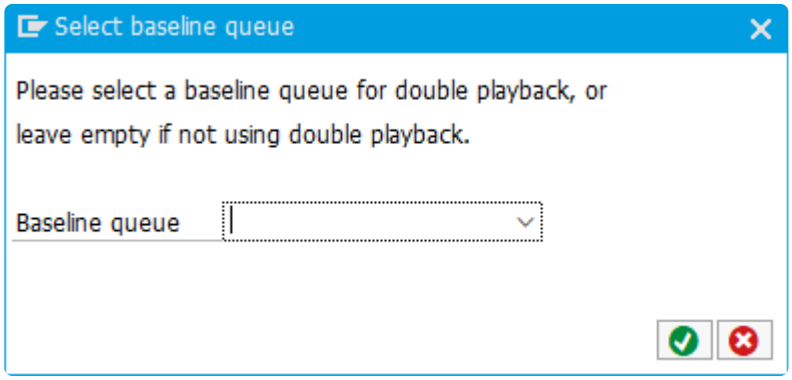
▶	Playback	FORCE_BASELINE_SEQ	Force sequence from baseline queue	i	X
▶	Playback	FORCE_BASELIN_RESULT	Force results from baseline queue	i	X

FORCE_BASELINE_SEQ
This parameter forces the second playback to use the exactly the same script sequence as was used during the baseline playback. This ensures that there are no unintended consequences of running scripts in a slightly different order.

FORCE_BASELIN_RESULT
This parameter forces the second playback to mimic the same execution results (for failed/errored/cancelled scripts) as they were experienced during the baseline playback. So, for example, if during the baseline playback a script was cancelled, then it will also be marked as cancelled in the second playback. This avoids any false positives arising from scripts executing that were not successfully executed during the baseline.

Starting the second playback

When starting the second playback, you will receive a pop-up asking you to select the baseline queue against which the results of this playback should be compared.



3.12.2. Causes of noise during playbacks

Introduction

In any Testimony playback, even into an identical system with no changes, some script failures are inevitable. These might be caused by environmental differences between the recording and playback systems; by limitations in the wider playback environment (e.g., some organisations choose not to install MS Office on the bot VMs); by slight differences in execution timings; or by certain aspects of the ways that SAP and Testimony work.

These failures during a playback constitute “noise” which can mask the genuine, regression defects that Testimony is looking for.

With a double playback, we seek to strip out this noise, therefore making the job of identifying genuine regression defects much easier.

Reasons for defects during first or second playbacks

Even where a recording system and a playback system are identical (i.e., no release / upgrade / patching changes have been deployed to the target system before starting the playback), you can always expect to see some defects (differences between the output from the recording and the output from the playback) because of fundamental ways in which SAP and Testimony work. This section highlights some common reasons for seeing playback defects.

In addition to these, the Testimony Testers’ Guide explains different types of defects, some of which are caused by factors that do not point to genuine regression failures.

User logons not captured

Scenario:

- User1 logs on at 09:00
- Recording is switched on at 09:30
- At 10:00 User1 creates sales order 1234
- At 10:30 User2 logs on
- At 11:00 User2 changes sales order 1234
- At 11:30 User3 logs on
- At 12:00 User3 displays sales order 1234

In this scenario, because User1 logged on before the recording was switched on, Testimony didn’t capture their logon. By default, Testimony will discard any activity for a user without a logon, so the creation of sales order ABC1 is not played back. Because of this, the activities of User2 & User3 will fail, because the sales order they are trying to access doesn’t exist.

There is an option in Testimony to create logon scripts for users where we didn't capture the actual logon. However, when this is switched on we still discard any activity in the user session (i.e., the particular SAPGUI window) for which we didn't have a logon . (This is so that, for example, we don't try to play back a transaction from the middle of the screen logic.) So in our scenario above, sales order ABC1 wouldn't have been created (and hence the subsequent transactions would have failed) because User1 was working in one session. However if, after the start of the recording, User1 had opened another SAPGUI session and then created ABC1, the other two transactions would have worked.

This scenario would result in failures during the first playback, but these would also arise in the second playback and so would be filtered out by the defect proposal run after the second playback, since the failures would be for exactly the same reasons. We therefore know that these failures do not indicate genuine regression defects.

Locking

Scenario 1:

- User1 goes to change sales order 1234
- A few seconds later, User2 also tries to change sales order 1234, and receives a "Sales Order is locked" message
- A few seconds later, User1 finishes their change and saves the sales order
- A few seconds later, User2 tries to change the sales order again and this time is able to

Because locking is so transient (with each lock often lasting a few seconds or less), it's possible that during the playback User2's change is executed after the lock on the sales order has already been released. In this case, User2 won't receive the "Sales Order is locked" message. This will lead to a failure (Different Message) in the script.

The converse is also true.

Scenario 2:
User3 goes to change sales order 1234
A minute later, User4 changes sales order 1234. Because User3 has finished the change, there is no lock, so User4's change proceeds as normal

During the playback, it's possible that User4's change is executed while User3's change is still running. This will lead User4 to receive a "Sales Order is locked" message, which will cause their script to fail.
Because locks are so transient and are dependent on the timing and sequencing of calls, it is possible that lock errors might occur in the first playback, but not in the second; or in the second playback but not in the first. (Where a locking error occurs in both playbacks, this will be filtered out by the Double Playback functionality.)

Timing / Sequencing

Scenario1 :

- Batch Job A updates stock levels in a warehouse. It starts at 15:00 and finishes at 16:00
- At 16:30 User1 creates an order which checks the stock level. There is enough stock of the material, so the order is created

During the playback it's possible that User1's order creation runs before Batch Job A starts (or while the job is still running). In this case, it's therefore possible that when they do the stock check there isn't enough stock for the order so an error message appears, causing the script to fail.

It is most likely that an error of this type seen in the first playback will also be seen in the second and will therefore be filtered out by Double Playback. However, it is still possible that timing issues may occur in only one of the playbacks.

Scenario 2:

- A series of 25 user transactions executes between 15:00 & 16:00, each updating the statuses of various orders to "ready for delivery"
- A batch job starts at 16:15 to process orders marked as ready for delivery. For each order found, a message is output to the job log.

This scenario can be considered as the obverse of the previous scenario. In this instance, a batch job is processing updates made by several dialog transactions. During the first playback, the batch starts before any order status updates have been made. This results in a "no orders to process" message in the job log which, because this was not seen in the recording, results in the batch job being flagged as a failure. If, during the second playback, the batch job again starts before any orders had been updated, we would again see this failure and this would be filtered out by the Double Playback functionality.

Regression failures causing non-regression failures

During a second playback in the Double Playback scenario, it is possible that a genuine regression failure in one script can lead to non-regression failures in other scripts.

Scenario:

- Batch Job A updates stock levels in a warehouse. It starts at 15:00 and finishes at 16:00
- At 16:30 User1 creates an order which checks the stock level. There is enough stock of the material, so the order is created

This is the same scenario as above, but in this case during the first playback both the batch job and User1's transaction were executed in the correct order and completed successfully. However, during the second playback a regression error has been introduced in Batch Job A causing it to fail without updating the stock levels. This results in a failure of User1's transaction, even though the transaction itself has no regression error.

3.13. Review Playback Results

Overview

When Testimony records activity in a system, it captures **inputs**, **outputs** and certain **linkages** that occur in-between. When the playback is executed in the test system, Testimony will automatically check that the output in the playback matches the output that was received in the recording. If linkages (for example, change documents) were recorded, it will compare these as well. Once a playback has completed, there are various ways of reviewing the results of the playback. These are described in the following sections.

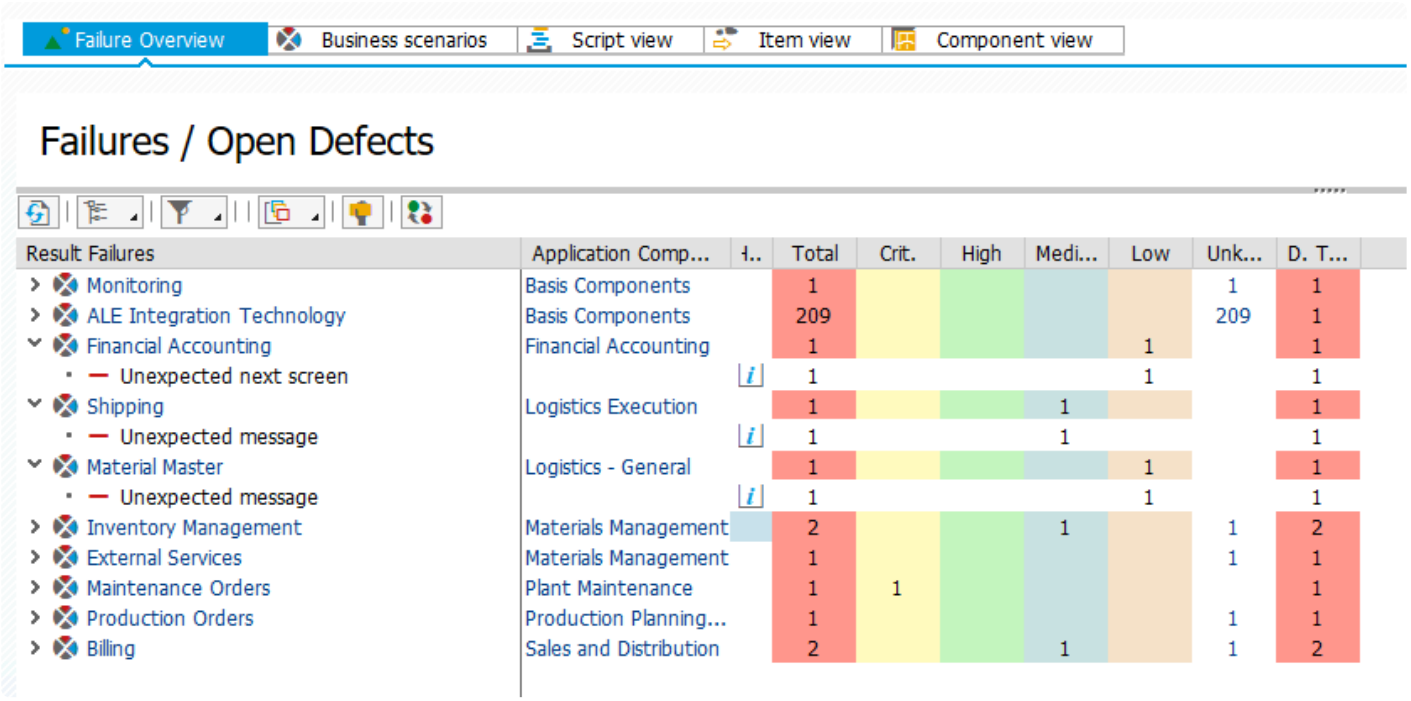
3.13.1. Playback Overview

Introduction

The Playback Overview, accessed from within the Results tray, shows the results of the playback in various different ways, as explained below.

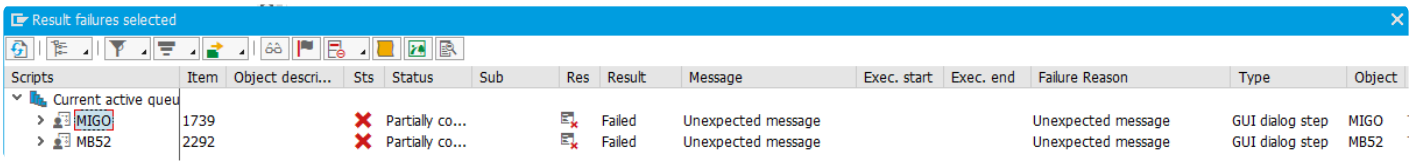
Failures / Open Defects

In this tab you can see the failures from the playback grouped according to their SAP application component and priorities from the Coverage Analysis. Expanding each.application component tree, you can see the types and numbers of errors (e.g., Unexpected next screen or Unexpected next message).



Result Failures	Application Comp...	!	Total	Crit.	High	Medi...	Low	Unk...	D. T...
Monitoring	Basis Components		1					1	1
ALE Integration Technology	Basis Components		209					209	1
Financial Accounting	Financial Accounting		1				1		1
Unexpected next screen		i	1				1		1
Shipping	Logistics Execution		1			1			1
Unexpected message		i	1			1			1
Material Master	Logistics - General		1				1		1
Unexpected message		i	1				1		1
Inventory Management	Materials Management		2			1		1	2
External Services	Materials Management		1					1	1
Maintenance Orders	Plant Maintenance		1	1					1
Production Orders	Production Planning...		1					1	1
Billing	Sales and Distribution		2			1		1	2

By clicking on the number hot-links in the display, you will show the individual errors for that area. For example, clicking on the number 2 in the Total column for Inventory Management will show you the two individual errors received, in this case that we received unexpected error failures in transactions MIGO and MB52.



From here you can investigate the errors using the Investigate Screen functionality.

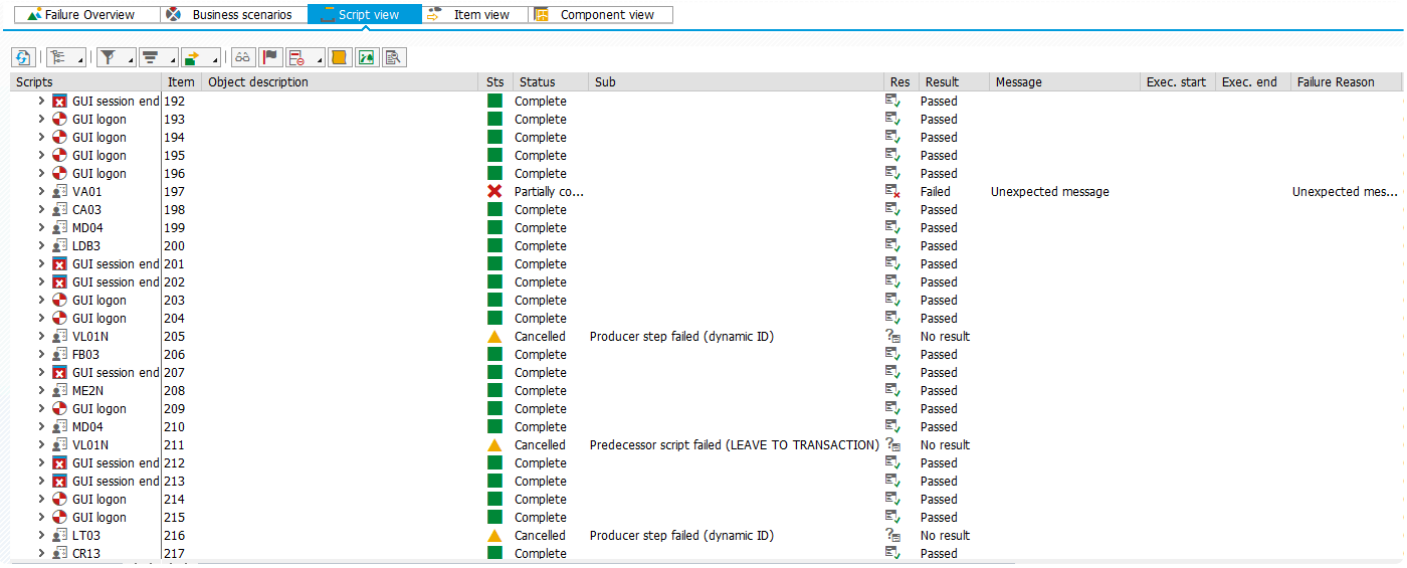
This is also the screen from which you generate defects to be investigated as part of the regression test

process. By clicking on the “Propose Defects” button, Testimony will analyse all of the reported failures and group them into defects. More information on defects can be found in the Testers’ Guide here.

3.13.2. Script statuses and execution results

When reviewing the results of a playback (either when it has finished or during its execution) it is important to understand the various possible statuses of the individual scripts (and script steps) being played back.

These statuses can be most easily seen once a playback has finished under Results —> Playback Overview —> Script View tab. The columns “Status” and “Result” show the success or otherwise of the scripts in the playback.



Scripts	Item	Object description	Sts	Status	Sub	Res	Result	Message	Exec. start	Exec. end	Failure Reason
GUI session end	192			Complete			Passed				
GUI logon	193			Complete			Passed				
GUI logon	194			Complete			Passed				
GUI logon	195			Complete			Passed				
GUI logon	196			Complete			Passed				
VA01	197			Partially co...			Failed	Unexpected message			Unexpected mes...
CA03	198			Complete			Passed				
MD04	199			Complete			Passed				
LDB3	200			Complete			Passed				
GUI session end	201			Complete			Passed				
GUI session end	202			Complete			Passed				
GUI logon	203			Complete			Passed				
GUI logon	204			Complete			Passed				
VL01N	205			Cancelled	Producer step failed (dynamic ID)		No result				
FB03	206			Complete			Passed				
GUI session end	207			Complete			Passed				
ME2N	208			Complete			Passed				
GUI logon	209			Complete			Passed				
MD04	210			Complete			Passed				
VL01N	211			Cancelled	Predecessor script failed (LEAVE TO TRANSACTION)		No result				
GUI session end	212			Complete			Passed				
GUI session end	213			Complete			Passed				
GUI logon	214			Complete			Passed				
GUI logon	215			Complete			Passed				
LT03	216			Cancelled	Producer step failed (dynamic ID)		No result				
CR13	217			Complete			Passed				

Before diving into the detail of the different statuses and execution results in the following sections, it is worth explaining the difference between a **Status** and a **Result**, as this often causes some confusion.

Playback statuses

The **status** of a script is the final technical status after playback. It tells you information on whether, and how far, the script was able to be executed (or was chosen to be executed) during the playback. The following statuses are possible.

- Complete: All steps of the script were executed
- Partially complete: One or more (but not all) steps of the script were executed
- Error: A technical error occurred which prevented one of the script steps from being executed
- Cancelled: Testimony decided not to run this script
- Suppressed: One of the steps in the script failed, but the error was suppressed through configuration
- Not processed: The script was not executed, nor was it attempted to be executed

Execution results

There are only three possible **results** for a script:

- Passed: All steps of the script were executed, and the outputs from the playback matched the outputs from the recording
- Failed: One of the steps in the script had a different output in the playback than it did in the recording
- No result: The status of the script meant that no result could be determined

Mapping statuses to results

Some **statuses** can have only one possible **result**, whilst others can have more than one. The following table maps the possible status & result combinations.

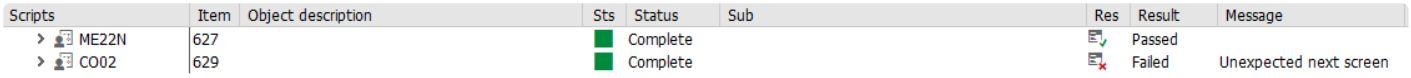
	Passed	Failed	No result
Complete	X	X	
Partially complete		X	
Error			X
Cancelled			X
Suppressed	X		
Not processed			X

The different statuses and execution results are discussed in the following sections.

3.13.2.1. Status: Complete

A status of Complete means that all of the steps in a script were able to be executed during the playback. However, this does not necessarily mean that the script was *successfully* executed.

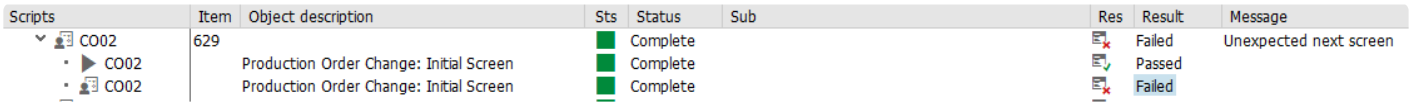
Compare the two scripts shown below:



Scripts	Item	Object description	Sts	Status	Sub	Res	Result	Message
ME22N	627			Complete			Passed	
CO02	629			Complete			Failed	Unexpected next screen

As you can see, the first (ME22N) has a **status** of Complete and a **result** of Passed. However the second (CO02) has a **status** of Complete but a **result** of Failed.

So in the case of scripts with a status of Complete, you need to check the script result in order to determine whether or not the script has actually been successfully played back. In fact, the combination of *Status=Complete* and *Result=Failed* always means that a failure occurred on the last step of the script. This is shown if we expand the failed script for CO02, which shows that it was the second (and final) step of the script that actually failed:



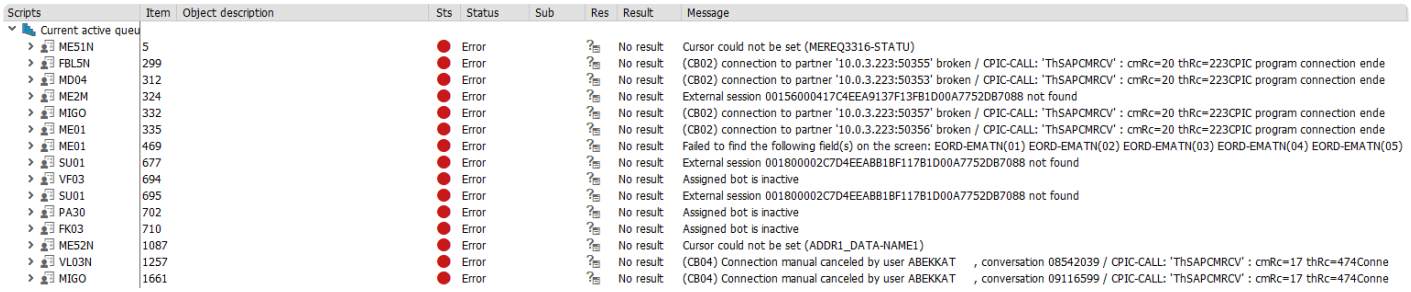
Scripts	Item	Object description	Sts	Status	Sub	Res	Result	Message
CO02	629			Complete			Failed	Unexpected next screen
CO02		Production Order Change: Initial Screen		Complete			Passed	
CO02		Production Order Change: Initial Screen		Complete			Failed	

To summarise:

- A script with Status=Complete can have two possible Results: Passed or Failed
- All scripts with Result=Passed will have Status=Complete
- A script with Status=Complete and Result=Failed shows that a failure occurred on the *last step* of the script
- Scripts with Result=Failed will generate defects once the Defect Proposal is run.

3.13.2.2. Status: Error

Technical problems sometimes mean that the bot, which is responsible for script execution, cannot actually execute a script step. In this case, the script will be shown to have a status of **Error**.



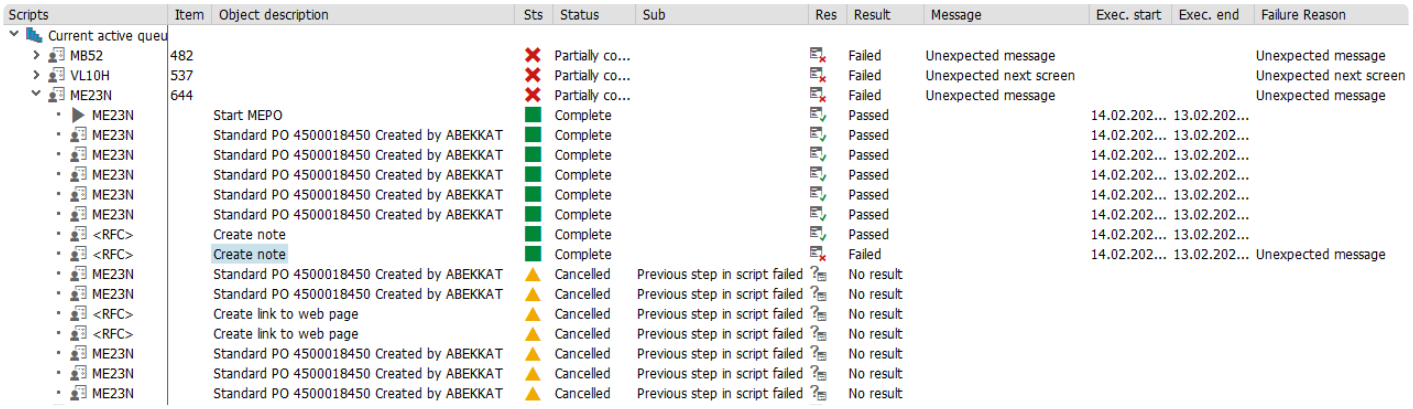
Scripts	Item	Object description	Sts	Status	Sub	Res	Result	Message
Current active queu								
ME51N	5			Error			No result	Cursor could not be set (MEREQ3316-STATU)
FBL5N	299			Error			No result	(CB02) connection to partner '10.0.3.223:50355' broken / CPIC-CALL: 'ThSAPCMRCV' : cmRc=20 thRc=223CPIC program connection ende
MD04	312			Error			No result	(CB02) connection to partner '10.0.3.223:50353' broken / CPIC-CALL: 'ThSAPCMRCV' : cmRc=20 thRc=223CPIC program connection ende
ME2M	324			Error			No result	External session 00156000417C4EEA9137F13FB1D00A7752DB7088 not found
MIGO	332			Error			No result	(CB02) connection to partner '10.0.3.223:50357' broken / CPIC-CALL: 'ThSAPCMRCV' : cmRc=20 thRc=223CPIC program connection ende
ME01	335			Error			No result	(CB02) connection to partner '10.0.3.223:50356' broken / CPIC-CALL: 'ThSAPCMRCV' : cmRc=20 thRc=223CPIC program connection ende
ME01	469			Error			No result	Failed to find the following field(s) on the screen: EORD-EMATN(01) EORD-EMATN(02) EORD-EMATN(03) EORD-EMATN(04) EORD-EMATN(05)
SU01	677			Error			No result	External session 001800002C7D4EEABB1BF117B1D00A7752DB7088 not found
VF03	694			Error			No result	Assigned bot is inactive
SU01	695			Error			No result	External session 001800002C7D4EEABB1BF117B1D00A7752DB7088 not found
PA30	702			Error			No result	Assigned bot is inactive
FK03	710			Error			No result	Assigned bot is inactive
ME52N	1087			Error			No result	Cursor could not be set (ADDR1_DATA-NAME1)
VL03N	1257			Error			No result	(CB04) Connection manual canceled by user ABEKKAT , conversation 08542039 / CPIC-CALL: 'ThSAPCMRCV' : cmRc=17 thRc=474Conne
MIGO	1661			Error			No result	(CB04) Connection manual canceled by user ABEKKAT , conversation 09116599 / CPIC-CALL: 'ThSAPCMRCV' : cmRc=17 thRc=474Conne

In the Message column, you can see the reason for each of these errors and, as mentioned above, these are all technical in nature and resulted in the Testimony bot being unable to execute a script step. For example, we can see that there were some connectivity issues (“connection to partner xxxxx broken”) and it also looks like one or more of the bots were shut down during the playback (“Assigned bot is inactive”).

Note that scripts with Status=Error do not generate defects.

3.13.2.3. Status: Partially complete

Where a script has been executed, but one of the steps (other than the last step) has failed, then this script will have a status of “Partially complete”.



Scripts	Item	Object description	Sts	Status	Sub	Res	Result	Message	Exec. start	Exec. end	Failure Reason
Current active queu											
MB52	482			Partially co...			Failed	Unexpected message			Unexpected message
VL10H	537			Partially co...			Failed	Unexpected next screen			Unexpected next screen
ME23N	644			Partially co...			Failed	Unexpected message			Unexpected message
ME23N		Start MEPO		Complete			Passed		14.02.202...	13.02.202...	
ME23N		Standard PO 4500018450 Created by ABEKKAT		Complete			Passed		14.02.202...	13.02.202...	
ME23N		Standard PO 4500018450 Created by ABEKKAT		Complete			Passed		14.02.202...	13.02.202...	
ME23N		Standard PO 4500018450 Created by ABEKKAT		Complete			Passed		14.02.202...	13.02.202...	
ME23N		Standard PO 4500018450 Created by ABEKKAT		Complete			Passed		14.02.202...	13.02.202...	
ME23N		Standard PO 4500018450 Created by ABEKKAT		Complete			Passed		14.02.202...	13.02.202...	
<RFC>		Create note		Complete			Passed		14.02.202...	13.02.202...	
<RFC>		Create note		Complete			Failed		14.02.202...	13.02.202...	Unexpected message
ME23N		Standard PO 4500018450 Created by ABEKKAT		Cancelled	Previous step in script failed		No result				
ME23N		Standard PO 4500018450 Created by ABEKKAT		Cancelled	Previous step in script failed		No result				
<RFC>		Create link to web page		Cancelled	Previous step in script failed		No result				
<RFC>		Create link to web page		Cancelled	Previous step in script failed		No result				
ME23N		Standard PO 4500018450 Created by ABEKKAT		Cancelled	Previous step in script failed		No result				
ME23N		Standard PO 4500018450 Created by ABEKKAT		Cancelled	Previous step in script failed		No result				
ME23N		Standard PO 4500018450 Created by ABEKKAT		Cancelled	Previous step in script failed		No result				

As you can see from the above, the script for transaction ME23N has a **status** of Partially complete and a **Result** of Failed. The eighth step of the script is the step that actually failed. It has a **Status** of Complete (because Testimony could execute the step) and a **Result** of Failed because Testimony detected an unexpected message during the playback. The remaining steps of the script have a **Status** of Cancelled, since Testimony automatically cancels the remaining steps of a script if a step fails.

To summarise:

- If a step, other than the last step, of a script fails, then the script will have a **Status** of Partially complete and a **Result** of Failed
- The step which failed will have a **Status** of Complete and a **Result** of Failed
- All scripts with a **Status** of Partially complete will result in defects being generated once the defect proposal has been run

3.13.2.4. Status: Suppressed

In Testimony it is possible to suppress a script failure so that it doesn't generate a defect. You might do this, for example, if the failure is as a result of the environment in which the playback or the bots are running. (For example, if you have not installed MS Office on the bots, some scripts which export data directly to Excel will fail.) More information on suppression can be found in the [Tester's Guide here]

If a script failure has been suppressed, then the script will have a **Status** of Suppressed and a **Result** of Passed.

Scripts	Item	Object description	Sts	Status	Sub	Res	Result	Message	Exec. start	Exec. end	Failure Reason
MM01	526			Suppressed			Passed	Unexpected message			Unexpected message

Scripts	Item	Object description	Sts	Status	Sub	Res	Result	Message	Exec. start	Exec. end	Failure Reason
MM01		Create Material 9277 (Configurable material)		Complete			Passed		14.02.202...	14.02.202...	
MM01		Create Material 9277 (Configurable material)		Complete			Passed		14.02.202...	14.02.202...	
MM01		Create Material 9277 (Configurable material)		Complete			Passed		14.02.202...	14.02.202...	
MM01		Create Material 9277 (Configurable material)		Complete			Passed		14.02.202...	14.02.202...	
MM01		Create Material 9277 (Configurable material)		Complete			Passed		14.02.202...	14.02.202...	
MM01		Transportation Group 18 Entries		Complete			Passed		14.02.202...	14.02.202...	
MM01		Create Material 9277 (Configurable material)		Complete			Passed		14.02.202...	14.02.202...	
MM01		Loading Group 7 Entries		Complete			Passed		14.02.202...	14.02.202...	
MM01		Create Material 9277 (Configurable material)		Complete			Passed		14.02.202...	14.02.202...	
MM01		Create Material 9277 (Configurable material)		Complete			Passed		14.02.202...	14.02.202...	
MM01		Create Material 9277 (Configurable material)		Complete			Failed		14.02.202...	14.02.202...	Unexpected message
MM01		Create Material 9277 (Configurable material)		Cancelled	Previous step in script failed		No result				
MM01		Create Material 9277 (Configurable material)		Cancelled	Previous step in script failed		No result				
MM01		Create Material 9277 (Configurable material)		Cancelled	Previous step in script failed		No result				
MM01		Create Material 9277 (Configurable material)		Cancelled	Previous step in script failed		No result				
MM01		Create Material 9277 (Configurable material)		Cancelled	Previous step in script failed		No result				
MM01		Create Material 9277 (Configurable material)		Cancelled	Previous step in script failed		No result				
MM01		Create Material 9277 (Configurable material)		Cancelled	Previous step in script failed		No result				

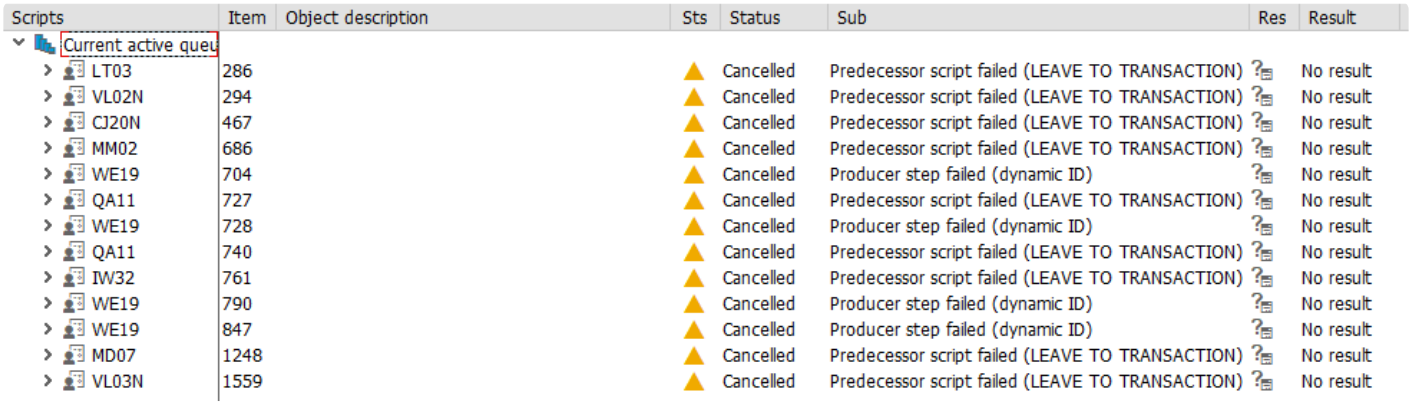
The step which failed will have a **Status** of Complete and a **Result** of failed. The remaining steps in the script will be cancelled.

To summarise:

- When a failure has been suppressed, the script will have a **Status** of Suppressed and a **Result** of Passed
- The failed step will have a **Status** of Complete and a **Result** of Failed
- Suppressed scripts do not result in the defects.

3.13.2.5. Status: Cancelled

Sometimes during a playback, Testimony will decide that a script cannot or should not be executed. These are shown with a **Status** of Cancelled and a **Result** of No result.

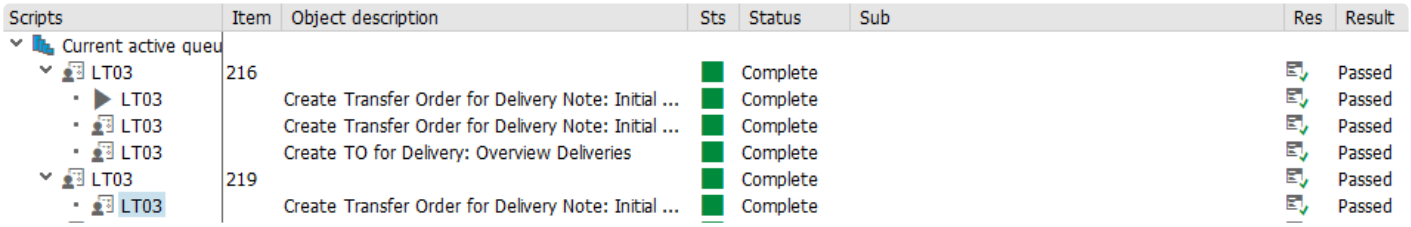


Scripts	Item	Object description	Sts	Status	Sub	Res	Result
Current active queu							
LT03	286			Cancelled	Predecessor script failed (LEAVE TO TRANSACTION)		No result
VL02N	294			Cancelled	Predecessor script failed (LEAVE TO TRANSACTION)		No result
CJ20N	467			Cancelled	Predecessor script failed (LEAVE TO TRANSACTION)		No result
MM02	686			Cancelled	Predecessor script failed (LEAVE TO TRANSACTION)		No result
WE19	704			Cancelled	Producer step failed (dynamic ID)		No result
QA11	727			Cancelled	Predecessor script failed (LEAVE TO TRANSACTION)		No result
WE19	728			Cancelled	Producer step failed (dynamic ID)		No result
QA11	740			Cancelled	Predecessor script failed (LEAVE TO TRANSACTION)		No result
IW32	761			Cancelled	Predecessor script failed (LEAVE TO TRANSACTION)		No result
WE19	790			Cancelled	Producer step failed (dynamic ID)		No result
WE19	847			Cancelled	Producer step failed (dynamic ID)		No result
MD07	1248			Cancelled	Predecessor script failed (LEAVE TO TRANSACTION)		No result
VL03N	1559			Cancelled	Predecessor script failed (LEAVE TO TRANSACTION)		No result

There are various reasons why Testimony might decide to cancel a script. These are discussed below.

Predecessor script failed (LEAVE TO TRANSACTION)

This cancellation reason arises in certain cases where “chained transactions” are executed by a user. In many cases in SAP, the end of processing of a transaction (e.g., hitting the back button or pressing save) will take the user back to the initial screen for the transaction. This is implemented in ABAP via a LEAVE TO TRANSACTION command. When Testimony builds the execution queue for these chained transactions, only the first call of the transaction has a “Start Transaction” step. The remaining transactions in the chain, although treated as separate scripts by Testimony, do not have a separate Start Transaction step, as the context from the previous transaction gives Testimony all of the information it needs to execute the transaction again.



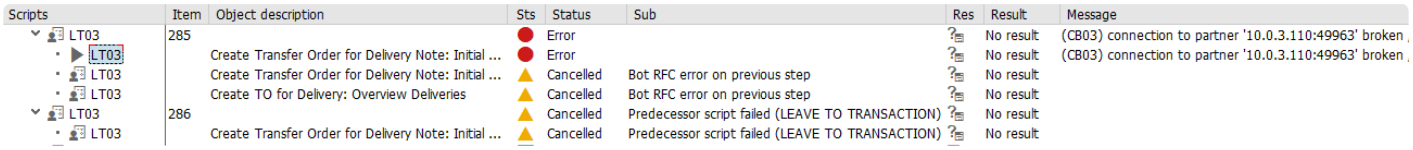
Scripts	Item	Object description	Sts	Status	Sub	Res	Result
Current active queu							
LT03	216			Complete			Passed
LT03		Create Transfer Order for Delivery Note: Initial ...		Complete			Passed
LT03		Create Transfer Order for Delivery Note: Initial ...		Complete			Passed
LT03		Create TO for Delivery: Overview Deliveries		Complete			Passed
LT03	219			Complete			Passed
LT03		Create Transfer Order for Delivery Note: Initial ...		Complete			Passed

In the example above, we can see that a user has executed transaction LT03, but only the first script has a Start Transaction step (shown by the “Play” icon on the first step). In this case, the user started the transaction (first step); entered some data (second step); and then clicked on save (third step). Because of the LEAVE TO TRANSACTION command in the SAP code for this transaction, the user was then taken back to the initial screen of LT03. This spawned a new Testimony script, but without a Start Transaction step.

One effect of this method of handling chained transactions in Testimony is that if one of the scripts in the transaction fails or errors, then the remaining scripts in the chain must be cancelled, because these

remaining scripts do not have a Start Transaction step.

This can be seen in the example below.



Scripts	Item	Object description	Sts	Status	Sub	Res	Result	Message
LT03	285			Error			No result	(CB03) connection to partner '10.0.3.110:49963' broken
LT03		Create Transfer Order for Delivery Note: Initial ...		Error			No result	(CB03) connection to partner '10.0.3.110:49963' broken
LT03		Create Transfer Order for Delivery Note: Initial ...		Cancelled	Bot RFC error on previous step		No result	
LT03		Create TO for Delivery: Overview Deliveries		Cancelled	Bot RFC error on previous step		No result	
LT03	286			Cancelled	Predecessor script failed (LEAVE TO TRANSACTION)		No result	
LT03		Create Transfer Order for Delivery Note: Initial ...		Cancelled	Predecessor script failed (LEAVE TO TRANSACTION)		No result	

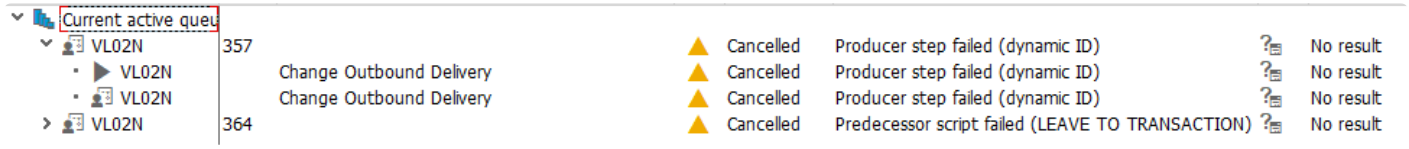
Here we can see that the first step of the first LT03 script had an error as a result of a connectivity. As usual, the remaining steps of this script were cancelled, but in addition the second script in the chain was also cancelled for the reasons given above.

Producer step failed (dynamic ID)

Testimony's dynamic ID process ensures that where the execution of one transaction is dependent on the results of another transaction, that the two transactions are linked together. This might be because, for example, one script creates a document, and a subsequent script changes that document. The dynamic ID process ensures that:

1. The linked scripts are executed in the correct sequence, so that the document creation is always executed before the document change
2. If the document number that is generated when the document is created in the playback is different to the number that was generated in the recording, the subsequent change document transaction will use the new number rather than the old number (so that the correct document is being changed)
3. If the document creation script fails in the playback, then the subsequent change document script is cancelled, to avoid unnecessarily trying to change a document which Testimony knows doesn't exist

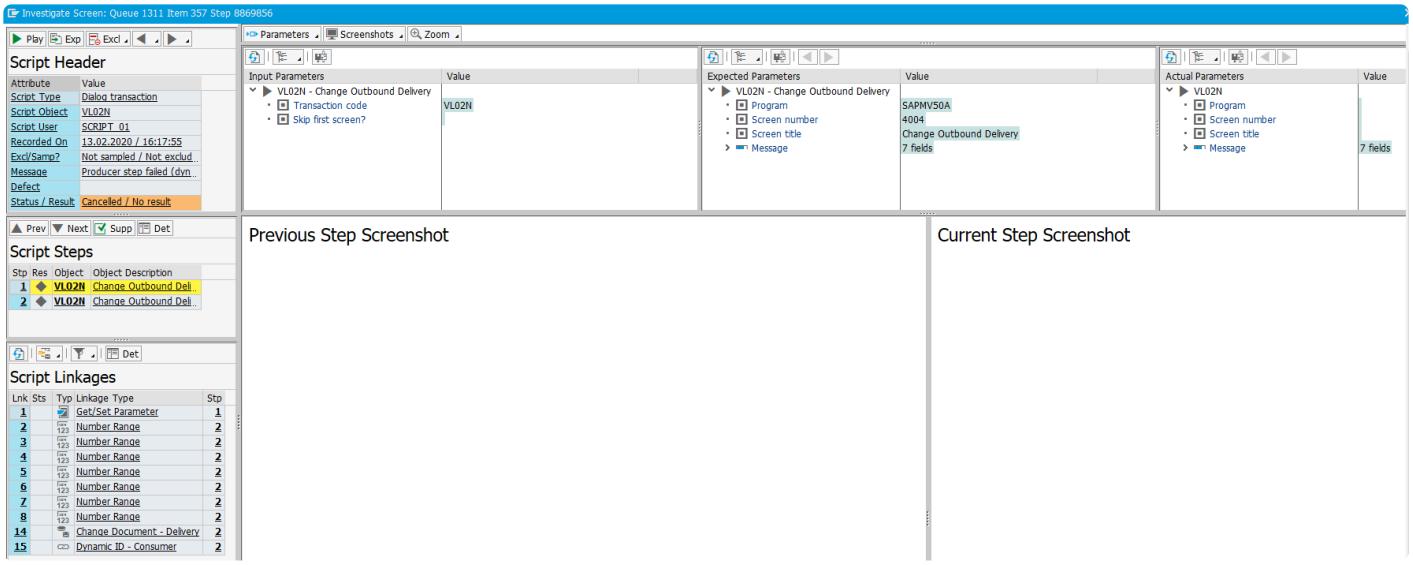
In the 3rd case above, the change document transaction would be given a **Status** of Cancelled with a **Sub-status** of "Producer step failed (dynamic ID)" and a **Result** of No result.



Current active queu					
VL02N	357		Cancelled	Producer step failed (dynamic ID)	No result
VL02N		Change Outbound Delivery	Cancelled	Producer step failed (dynamic ID)	No result
VL02N		Change Outbound Delivery	Cancelled	Producer step failed (dynamic ID)	No result
VL02N	364		Cancelled	Predecessor script failed (LEAVE TO TRANSACTION)	No result

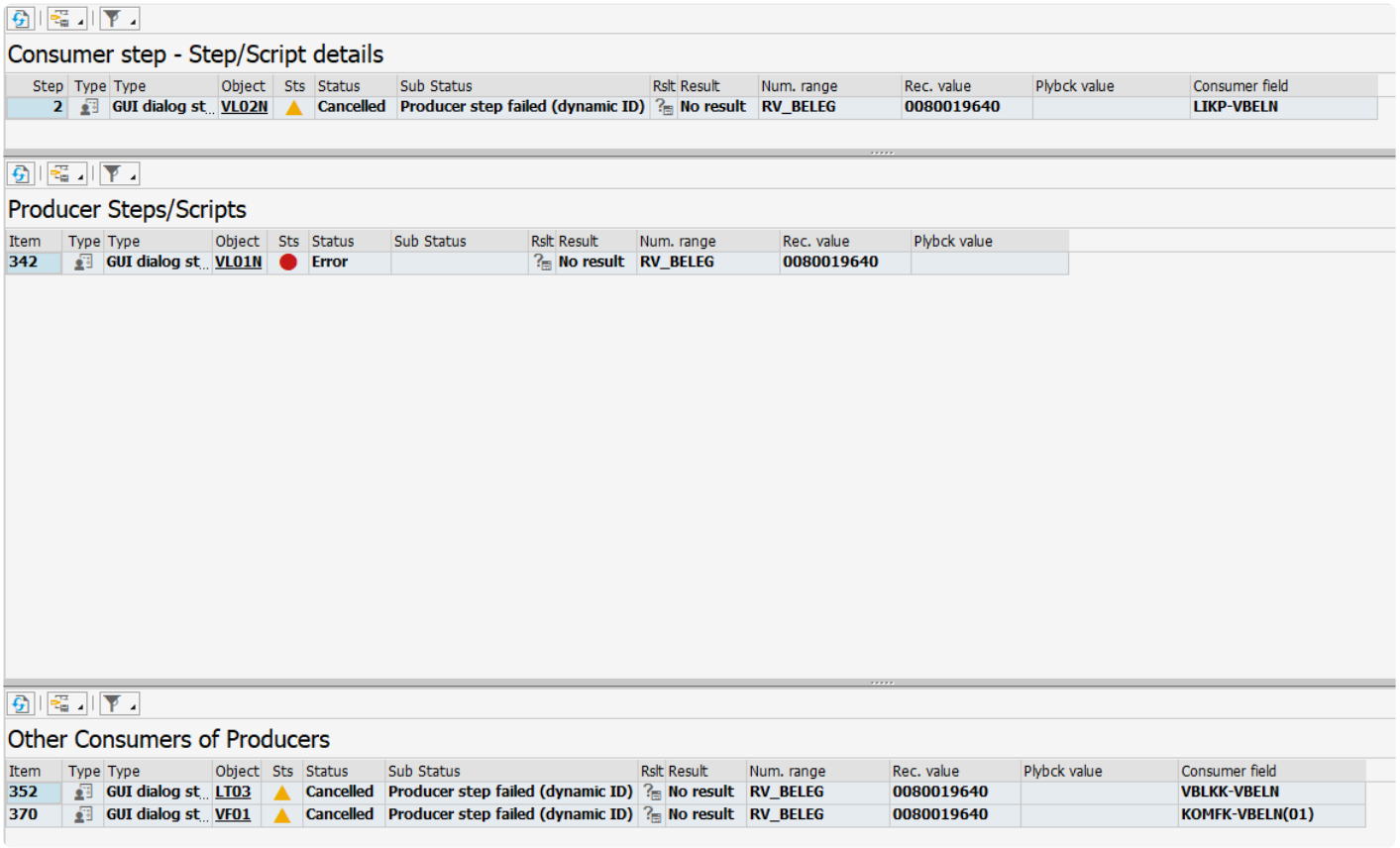
In the example above, a VL02N transaction has been cancelled with the "Producer step failed (dynamic ID" sub-status.

We can check to see which producer step caused this by going into the investigate screen for the cancelled script.



The Script Linkages section at the bottom left of the screen shows any objects or scripts that have been linked to this script. Here we can see that there is a “Dynamic ID – Consumer” linkage, which shows that this script is the consumer of another script. If we click on this item in the linkages list, we can see details of (reading from top to bottom):

- The current script / step (“Consumer step – Step/Script details”)
- The producer, i.e., the script / step on which this step is dependent (“Producer Steps/Scripts”)
- Any other scripts that are dependent on this producer (“Other Consumers of Producers”)



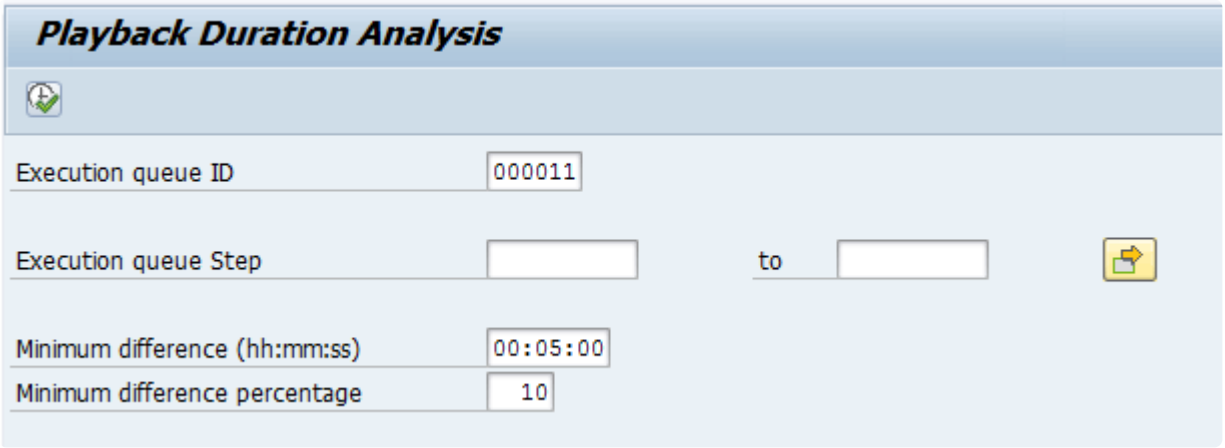
So, in the example above, we can see that our VL02N script was cancelled because it was dependent on a previous VL01N transaction which had a **Status** of Error. Because this producer script failed, the VL02N transaction, as well as an LT03 and a VF01 transaction, were all cancelled.

3.13.3. Playback duration analysis

During a recording, Testimony captures the response times (through the STAD data) of all recorded objects. It is often useful to compare these runtimes with the times taken during a playback, to see if there are any processes which take significantly longer during the playback. This might be, for example, because a batch job makes calls to an external system which isn't available in the playback environment. In this case, during a playback the job may have to wait for each RFC call to time out, causing the runtime of the job to be significantly higher in the playback.

Program /BTI/AUT_DURATION_ANALYSIS can be used to analyse the durations of scripts during the playback in comparison with their durations in the recording.

Call the program using SE38/SA38 in the Central System.



Enter the execution queue ID of the playback you want to analyse (you can get this from table /BTI/AUT_EXQH) and then values for the Minimum difference time and Minimum difference percentage. (In the example above, we are looking for steps where the playback duration was at least 5 minutes longer than the recording duration, AND the playback duration was at least 10% longer than the recording duration.)

Once you have the output, you can filter on the Type column if you're particularly interested in one type of object (for batch jobs, for example, you would filter on Type=08).

Queue step	Item ID	T	Object	Recording duration	Playback duration	Difference	% Difference
0001833744	0000686841	08	3RD_PARTY_IDB	157,0000000	472,0000000	315,0000000	200,64
0001833679	0000686686	08	3RD_PARTY_IDB_HOURLY	157,0000000	472,0000000	315,0000000	200,64
0001832236	0000691516	08	C0000KAL+RKAZCO43_ZUSCHL	463,0000000	890,0000000	427,0000000	92,22
0001676461	0000640001	08	D0000_BE_AB_LISTE_CSV	402,0000000	1.098,0000000	696,0000000	173,13
0001678800	0000640230	08	D0000_DK_AB_LISTE_CSV	456,0000000	1.072,0000000	616,0000000	135,09
0001613569	0000622065	08	D0000_GROHE_STOCK	0,1000000	908,0000000	907,9000000	907.900,00
0001628385	0000627044	08	FI_HOAG_NACHBUCHUNG	1.074,0000000	1.514,0000000	440,0000000	40,97
0001682654	0000639882	08		1.164,0000000	1.585,0000000	421,0000000	36,17
0001621325	0000625937	08	LR_ZQMFK_UPDATE_RECEIVINGS	5,0000000	2.013,0000000	2.008,0000000	40.160,00
0001694106	0000651532	08	M0000_QUERY_AS_JOB_KDAUFT1	1.286,0000000	2.892,0000000	1.606,0000000	124,88
0001746866	0000667941	08	M0000ZPP_PLAF_LSP_QUOTA	536,0000000	2.300,0000000	1.764,0000000	329,10
0001712545	0000654867	08	MM0299EINK+P4T+VMI_0299	201,0000000	2.326,0000000	2.125,0000000	1.057,21
0001712815	0000654925	08	MM7301EINK+P4T+VMI_7301	2.043,0000000	2.486,0000000	443,0000000	21,68
0001756170	0000668737	08	MM7301EINK+P4T+VMI_7301_VDR30	1.568,0000000	2.005,0000000	437,0000000	27,87
0001679279	0000639951	08	PP_QUERY_PUHL_ERS	775,0000000	1.345,0000000	570,0000000	73,55
0001910199	0000706473	08	PP0202_PROD_ORDER_AV_CHECK	481,0000000	2.341,0000000	1.860,0000000	386,69
0001739185	0000664892	08	PP0301_PROD_ORDER_AV_CHECK	856,0000000	2.309,0000000	1.453,0000000	169,74
0001900304	0000704797	08	PP1702_MONTAGE_ABLIEF_LISTE	670,0000000	1.515,0000000	845,0000000	126,12
0001912053	0000707041	08	PP7301_PROD_ORDER_AV_CHECK	463,0000000	2.311,0000000	1.848,0000000	399,14
0001613832	0000623089	08	SAP_IDOC_NACHBUCHEN_DESADV_51	2.551,0000000	3.288,0000000	737,0000000	28,89
0001689964	0000649992	08		1.628,0000000	2.495,0000000	867,0000000	53,26
0001712529	0000654740	08	SD5201_TH_TERMINIERUNG	1.680,0000000	2.304,0000000	624,0000000	37,14
0001844313	0000692878	08	ZBC_QUERY_CALL_AS_JOB 03:00	691,0000000	2.015,0000000	1.324,0000000	191,61
0001919987	0000707070	08		768,0000000	1.103,0000000	335,0000000	43,62

3.13.4. Root Cause Analysis

Overview

The function of Root Cause Analysis (RCA) is to allow the users managing defects to see if there is an underlying change (SAP transport) related to the defect. Testimony leverages ActiveControl (A change and release product also built by Basis Technologies) to enable Testimony to look inside the transport to determine the underlying objects changed as a result of the release. These changed objects are checked to see if any defects have been raised for the transaction code, batch job, RFC or web service etc. an underlying change found it linked to the defect and the Root Cause Analysis flag is raised against that defect. Also note that RCA is designed to be run once the execution of the playback is completed, if utilising double playback this will be after the second run where you are testing your release.

Process Steps

To ensure the correct setup and running of Root Cause Analysis (RCA) please check the following steps.

- Setup of RCA
- Link Creation
- Build Lists
- RCA Results

A prerequisite of RCA is for ActiveControl 8.31 (as a minimum) to be deployed to the target system. If your organisation is already running ActiveControl please speak with your Testimony consultant who can check the version deployed and ensure there is no disruption to your ActiveControl installation.

3.13.4.1. Setup of RCA

Overview

Before Root Cause Analysis (RCA) can be run successfully there are a set of steps that need to be followed. If Root Cause Analysis is already setup on your systems then you can proceed to the linkage creation step.

Setup Steps

1. ActiveControl 8.31 (as a minimum) needs to be deployed to the target system. If your organisation is already running ActiveControl please speak with your Testimony consultant who can check the version deployed and ensure there is no disruption to your ActiveControl installation.
2. Ensure Testimony's General Parameter "ROOT_CAUSE" is turned on – set to 'X'.
3. The RCA function is, by default, visible only to the Test Engineer profile. Or an other UI profile might need to be changed so that Root Cause Analysis is visible. See the UI profile topic for more details.
4. Execution queue required and target system with transports applied

3.13.4.2. Link Creation

3.13.4.3. Build Lists

Purpose

Build Lists are used to understand the root cause of failures in a playback by determining which changes resulted in the failure. They represent the set of changes (transport requests) that are being tested by Testimony.

Audience / Users

Test Engineers

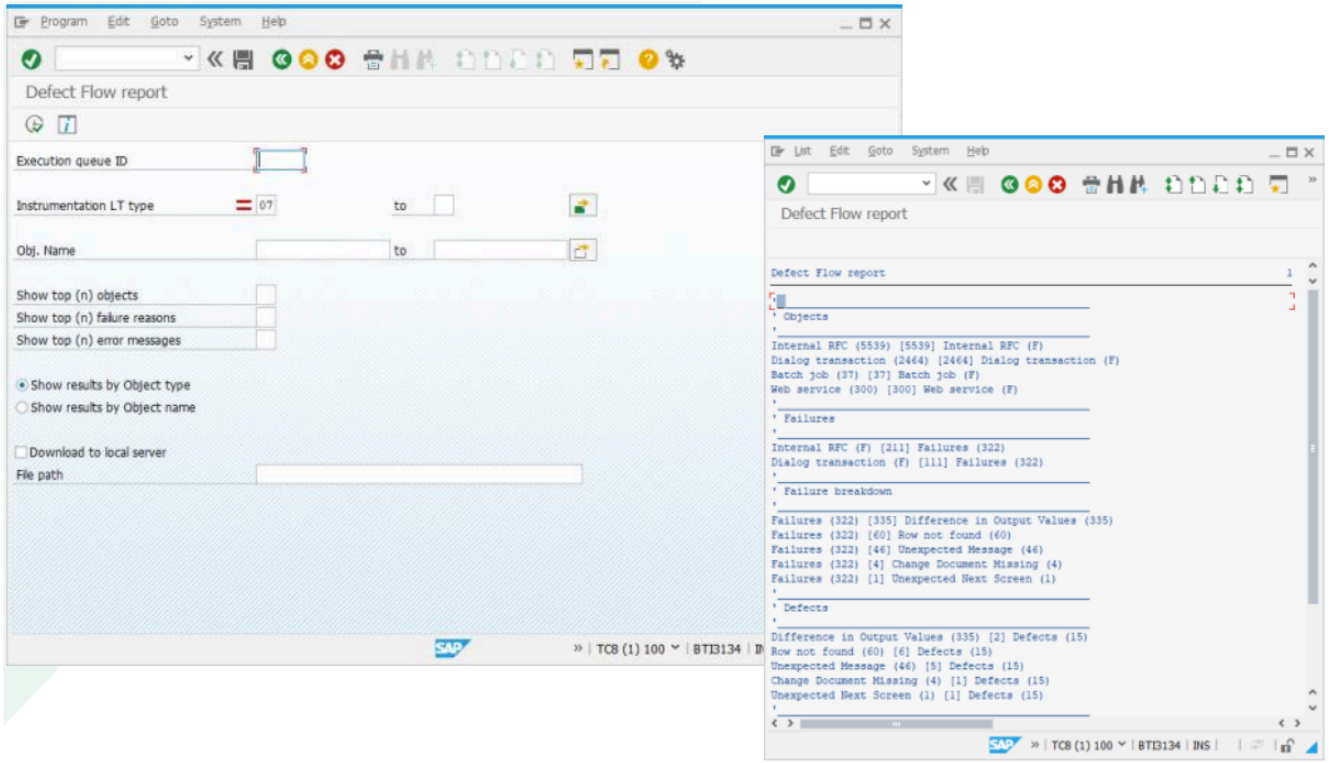
3.13.4.4. RCA Results

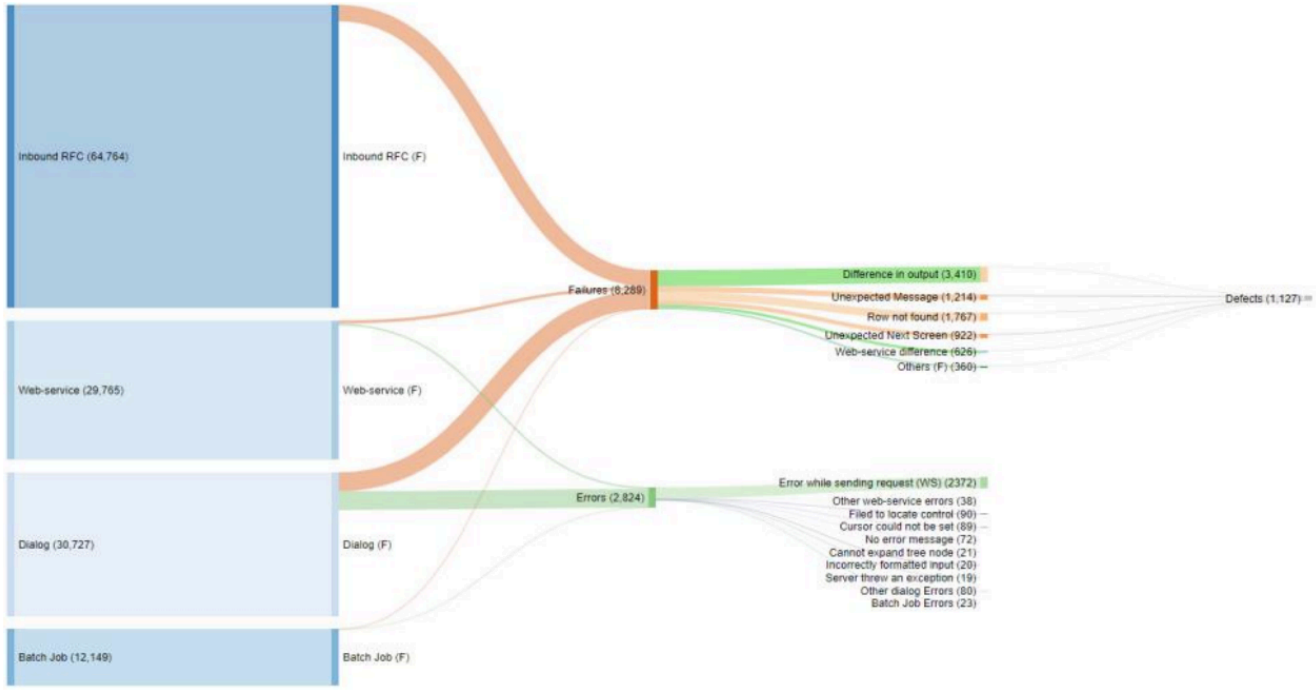
3.14. Useful Programs and Reports

There are utilities which are not included in Testimony's menu-system as they are intended for Administrator use only.

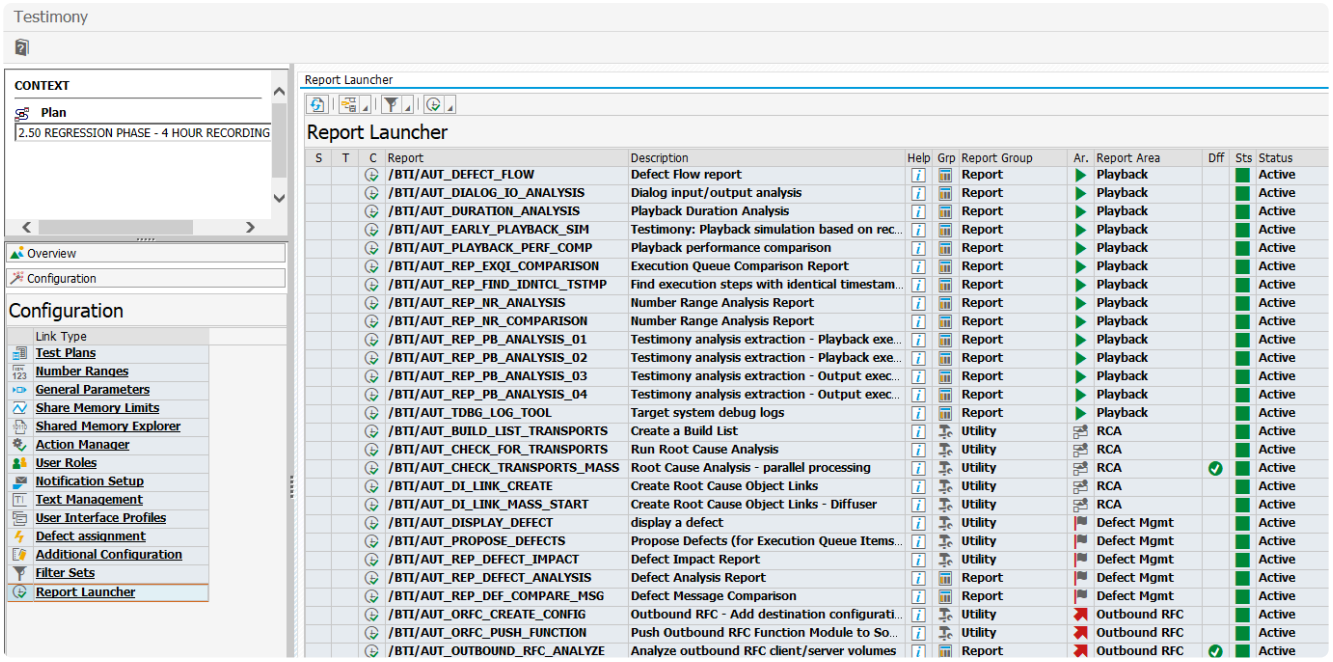
- Recording Extractor
  Output the contents of a recording for non-regression testing purposes.
  The Transaction /BTI/AUT_REXT, linked to program /BTI/AUT_TESTIMONY_EXTRACTOR will download an unformatted XML document which can then be used with third-party tools.

- Defect Flow Report
  It is very useful to visually display the defects, and their source-fault types, in a Sankey diagram. The data for such an output can be downloaded via the transactions /BTI/AUT_DEFECT_FLOW.
  The data can then be uploaded to a website such as SankeyMatic; Builder





- Report Launcher
  A single place where all the reports and transactions available in Testimony can be accessed. Identified by Central, Source or Target, they can be operated via an RFC connection into the relevant Source and Target systems.



4. Glossary

Bot
The bot is an executable program that resides on a windows machine (normally a virtual machine). During playback the bot machines log on as the recorded users and execute the recorded transactions. The requirements for bot setup can be found here.

Central System
This is the primary SAP system in which Testimony is installed and is operated. Testimony users log on to the Central system to configure Testimony, create test plans, start recordings and playbacks and analyse the results.

Check Steps
These are run manually before a recording or playback to validate the environment is ready to perform those functions. The Testimony Administrator should run the check steps and review the results prior to performing a recording or playback.

Coverage Analysis:
Allows the comparison of recorded data from the execution queue with the usage data. It provides high-level statistics (e.g., what percentage of critical priority dialog transactions you recorded) as well as detailed information on each dialog transaction, batch job, etc.

Double Playback
A double playback is where two playbacks are used to reduce the potential for false positives. The first playback occurs with no changes deployed to the target system and is termed the baseline playback. The second playback called the release or upgrade playback is competed on the target system with the release or the upgrade applied .Defects are then only raised on differences detected in the second playback, this helps to screen out environmental issues from the baseline.

Dynamic IDs
These are used to link scripts which use the same data, for example a purchase order number. If the creation of a purchase order fails during the playback then Testimony recognises that there is no point running a subsequent script that approves this purchase order. Testimony will therefore cancel the execution of the order approval script. Testimony will also recognize if a different order number is generated during playback and will adjust subsequent scripts to use this new number rather than the recorded number.

Enhancements
To record and playback Testimony has enhancements on the source or target system to enable the recording or playback to operate correctly. These are switched on before recording or playback and are automatically deactivated at the end of the recording or playback in the “Post-Processing Steps”. Should a recording or playback be stopped unexpectedly or due to a technical error, the Testimony Administrator should manually deactivate the enhancements.

Execution Queue

The execution queue is built when scripts are added from the repository and contains the scripts to be played back. Logic is built into the “Add to Execution Queue” process that idnetifies and establishes linkages between related scripts as the execution queue is being built.

Filtered Recording
A filtered recording is used when you want to record a subset of users, transactions, objects, or transaction types rather than all activity on the source system. It is typically used for testing purposes to ensure that the setup from central to source system has been completed correctly.

Filter Sets
Filter sets have two main uses: to exclude certain objects (transactions, batch jobs, etc.) from a recording; and to provide special handling of error cases during a playback. For example, if you want Testimony to ignore all occurrences of transaction SM21 from the recording, then adding this transaction to the recording filter set will achieve this. If you want to ignore occurrences of message E123 from a particular screen, you can set this message as an exclusion in the comparison filter set. Filter sets can also be defined for the transfer to repository (most commonly for setting up transaction sampling) and for the transfer to the execution queue, although this is less frequently used. This topic should be further studied via the Filter Sets section here.

Linkages
Testimony records activity deeper than just the UI so that objects such as change documents and number ranges are also observed and recorded. These objects are used to create relationships, or linkages, between scripts so that dependencies can be enforced and validated. These can then be checked at playback and during results analysis to ensure that these match, providing a deeper level of testing.

Notifications
Testimony can be configured to send out notifications when certain actions are executed or to provide regular updates on ongoing actions. Notifications are managed through the notification setup in the configuration tray.

Playback
The playback is the execution of the scripts in the execution queue, via the bots, on the target system. The playback executes the scripted activity and generates the test results for comparison and analysis.

Post-Processing Steps
These are run automatically after a recording or playback is completed. Any errors in post-processing will cause a hard stop preventing the status from moving to complete. If errors are found the operator should investigate these errors to determine if they need to be manually resolved.

Preparation Steps
These are run automatically before a recording or playback starts. Any errors will cause a hard stop preventing the recording or playback from starting. Errors should be resolved before attempting to restart the recording or playback.

Recording

A recording (either Filtered or Standard) is the process by which actions on the source system are captured by Testimony.

Repository
The repository is a staging post for recorded transactions. Once all recorded transactions have been stored in the Central System, they are transferred to the repository (potentially with some filtering) before being transferred to the execution queue for playback. The repository is where any manipulation or deletion of sessions should take place, since if a mistake is made the sessions can be restored by transferring them again from the recorded data.

Sampling
Sampling is part of Filter Set functionality and is only set at the “Transfer to Repository” stage. Sampling allows the operator to decrease the volume of a set of activities without negatively impacting the validity of the test results. Since scripts will often modify data that will be used by later scripts, sampling is designed to be used for display or read-only activities that do not manipulate data. The idea behind it is as follows:
If you recorded a read-only process that runs every 5 minutes for the duration of the recording, you can play back a small percentage of those processes to help reduce playback times while still testing that process. Sampling can be used for Dialog, Batch and RFC processes.

Shared Memory
Testimony utilises the shared memory of the source system to save the recorded data temporarily before writing it to the database. This is so that the source system does not see a significant increase in I/O activity during a recording. To prevent any negative impact on source system performance Testimony will stop the recording if it runs too low on available shared memory. The recommended settings for the shared memory parameters are here.

Source System
This is the system that is recorded and therefore acts as the source for the recording. In BAU operation of Testimony, this is usually the production system.

Standard Recording
A standard recording records all activities, excluding any defined exceptions in the “Filter Sets”.

Target System
This is the regression test system into which recorded scripts are played back via the Bots. It is recommended that the Target system is dedicated for use with Testimony and is refreshed with a point-in-time backup of the Source system taken as of the start of the recording.

Test Plan
A test plan is the logical container for the recording, playback, and results of a test scenario. When setting up a test plan the operator will define a Source system, Target system, system mapping and authorizations for users. To simplify the test plan creation process, test plans can be copied for scenarios using the same Source and Target systems.

5. Software Support

After appropriate testing on your SAP test systems, you are now ready to use Testimony within your productive landscape. Remember that Testimony is dormant in your SAP system until activated for recording. You must ensure that Testimony recordings are deactivated after the required recording period in order to ensure that Testimony returns to its dormant state.

5.1. Support from Basis Technologies

Raising Support Tickets

To request support from Basis Technologies on any issue relating to our product sets (ActiveControl, Transport Expresso, DevOps, Testimony, Diffuser, BDEx Utilities or Transformation), support can be requested from Basis Technologies by submitting a request via our support portal link here.

Submitting your request will automatically create a ticket in Zendesk, the ticketing tool used by Basis Technologies.

Require additional Information or Services?

If additional information or services relating to any of Basis Technologies product sets is required, you can contact us via the support portal link here , or alternatively by contacting your assigned Basis Technologies Account Director.