Testimony - Release Notes

2.40 — Last update: 6 September 2022

Basis Technologies

Table of Contents

Introduction

Testimony v2.40 was released in June 2021.

Previous releases of Testimony are detailed in separate Release Notes:

- **Testimony v2.30** (Released November 2020)
- **Testimony v2.21** (Released March 2020)
- **Testimony v2.20** (Released November 2019)
- **Testimony v2.13** (Released June 2019)
- **Testimony v2.12** (Released May 2019)
- **Testimony v2.11** (Released April 2019)
- **Testimony v2.10** (Released February 2019)
- **Testimony v2.09** (Released January 2019)
- **Testimony v2.08** (Released December 2018)
- **Testimony v2.07** (Released November 2018)
- **Testimony v2.06** (Released October 2018) – **Interim Release Only**
- **Testimony v2.05** (Released October 2018)
- **Testimony v2.04** (Released October 2018)
- **Testimony v2.03** (Released September 2018) – **Interim Release Only**
- **Testimony v2.02** (Released September 2018) – **Interim Release Only**
- **Testimony v2.01** (Released September 2018) – **Interim Release Only**
- **Testimony v2.00** (Released August 2018)

SAP Certification

Testimony is a SAP certified solution:

- Certified for deployment on SAP NetWeaver 7.50 via the SAP integration scenario ABAP Add-On Deployment for SAP NetWeaver (**SAP report 12649**)
- Certified for deployment on SAP S/4HANA 1709 via the SAP integration scenario ABAP Add-On Deployment for SAP S/4HANA (**SAP report 12657**)
- SAP Solution Manager Ready functionality

All Testimony SAP components exist within Basis Technologies' own namespace **/BTI/**.

General Enhancements

This section describes the general enhancements that have been delivered in Testimony v2.40.

The key areas described in this section are:

- **Configuration parameters** – New parameters added in this release
- **Check steps** – This includes new check steps that have been introduced in this release including enhancements to existing check steps
- **Playback system parameters** – Any playback system parameters that have been identified for importance for playbacks
- **User preference configuration** – New default user preference configuration
- **Exclusion configuration** – Describes the new entries added to the default filter set containing defaulted exclusion entries
- **Enhancements** – Changes to the enhancements in both recording and playback

Configuration Parameters

A number of new parameters have been added in this release of Testimony:

Parameter Name	Area	Description
TBD	Recording	TBD
TBD	Recording	TBD

Check steps

The following check steps have been created or changed in this release:

Check Step	Phase	Type	Description
TBD	Recording	Preparation	TBD.
TBD	Recording	Post Processing	TBD.

Playback system parameters

No profile parameters have been added or changed during the playback check step in 2.40.

User Preference Configuration

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

! 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**.

Exclusion Configuration

The default values delivered in v2.40 for the exclusion configuration in Filter Sets have been enhanced.

There are now approximately **250 objects** in the default configuration for exclusions with batch jobs, dialog transactions, inbound RFC's and inbound web-services. Please review these entries prior to starting the process and ensure that you agree on the configured entries.

> Don't forget you can always over-ride the standard delivered filter set and configure your own. This can also be done after reviewing the usage data from transaction ST03 or STAD. However, you can also do this configuration after a recording in the "Performance Analysis" area of Testimony.

Enhancements

Recording Enhancements

The following changes have been done with regards to the recording enhancements available in Testimony 2.40.

1. **Enqueue / Dequeue #1** – New enhancement point for Lock / Unlock request capture. This was available in 2.30, but not “released”. It has now been released in 2.40.
2. **Enqueue / Dequeue #2** – New enhancement point for Lock / Unlock request capture. This was available in 2.30, but not “released”. It has now been released in 2.40.

1. **App Server File Discovery** – New enhancement point to discover application server files so that the configuration for them can be done

> The new application server file discovery enhancement does not require the recording to be turned on. You simply need to activate the enhancement and from that point on, all application server files that are accessed (opened for reading, writing or appending) will be logged. Once you deactivate this enhancement point, the logging will stop. A table on the source system will be populated with all files that have been accessed during the period of activation. A program is also available to take the entries in this table from the source system and bring them into the central system.

The lock/unlock linkage capture is available in 2.40 in a released state but optional rather than mandatory. However, due to the high volumes expected with these two new linkages, we are advising that you manually activate these and trial them in production recordings with a gradual ramp up approach.

Key Enhancements

The key areas that have been enhanced in v2.40 are listed below:

- **Refcatored recordings of linkages** – Optmize recording performance issues with linkage capture
- **Double Playback** – Significantly reduces the noise of false positives in the playback
- **Defect Management enhancements** – Extensions to defect management
- **Advanced Row Selection enhancements** – Multiple enhancements to the advanced row selection capabilities
- **Change Document dependencies** – Create dependencies between scripts based upon change documents
- **Synthetic Load enhancements** – Extensions to SLG to support batch jobs and inbound RFC
- **Usability enhancements** – Various usability enhancements such as multiple field sorting on all grids
- **Fiori / UI5 App Support (BETA)** – **BETA** Extensions to support further UI5 control coverage
- **Outbound RFC patch (BETA)** – Resolved performance issues holding back pilot of fuctionality
- **Discovery of recording window** – Analyze and determine the most appropriate recording window

Refactored recording of linkages

One of the major enhancements delivered in 2.40 has been the internal process of capturing of “linkages” during the recording. This process has been overhauled for extremely high volume linkage capture.

The previous algorithm for capturing of linkages during the recording saw internal tables of linkages getting larger and larger as each linkage was captured. This was fine for smaller volumes, but as the volumes increased (in particular with long running batch jobs), this added an overhead that was not acceptable in these exceptional circumstances.

The logic has been reworked so that as a business transaction progresses, the linkages that are captured are also saved down on a regular basis during the recording (rather than accumulating and getting larger and larger).

Some further optimization in this area regarding batch jobs is envisaged for the next release.

Double Playback

A new capability available in 2.40 is the ability to significantly reduce the “noise” of false positives during playbacks. This is achieved by running a “base-line” playback into a precise database copy that allows the customer to understand the “noise” in the system. A second playback is then required (separate execution queue) that is done into a system that contains the changes to be tested. The failures in the second (latter) queue are compared to the base-line. If they happened there for the same reason, these are excluded from defect proposal generation for the second queue – thus reducing the noise.

The following process is required to achieve this:

1. Run a playback into an exact copied system (same database, no changes)
2. Copy the execution queue within the same test plan
3. Restore the target system to the snapshot
4. Deploy changes that are to be tested
5. Re-run the playback of the second queue
6. Run defect proposal with new parameters to activate double playback

The defect proposal program (**/BTI/AUT_PROPOSE_DEFECTS**) now has additional parameters to control double playback. The flag “Activate Double Playback” must be selected. You must then also specify the execution queue ID of the “base-line” queue.

Defect Management Enhancements

Defects and management of those defects is a fundamental aspect to Testimony. Defects aggregate the failures (differences) detected by Testimony into a single grouping so that they are more easily managed and worked. The management of defects has been focused on in release 2.30 in a number of areas extending its functionality and making things more usable.

Key Improvements / Enhancements

1. Mass-Assign multiple defects to the same user or group
2. Allow configuration of defect statuses
3. Ability to set defect type and sub-type
4. Quick filters for defects
5. Closure reasons for defects
6. Auto populate of defect priority from object priority
7. Suppression of defects
8. Ability to show difference between two defects
9. Defect History fixes
10. Various minor bug fixes to defect management functionality

A few of the items above will be described in more detail in the following sub-sections.

Configuration of Defect Statuses

It is now possible to configure your own defect statuses within the product. As default, Testimony will ship some pre-defined statuses but these can then be changed by the customer according to requirements. The defect status controls the current status of a given defect. It was found that numerous customer's would like to configure this field and the values for they wanted varied from one customer to the next. Thus, this functionality was prioritised for release 2.30.

! It is important to note that there a few critical statuses that cannot be removed and the dialog maintenance windows for managing this defect status field will restrict the user from removing or changing these critical status values.

* The table to be configured via SM30 is **/BTI/AUT_C_BTSG**. Please use transaction SM30 to maintain this table in the central system only.

! The following statuses cannot be changed – "Proposed", "New", "Assigned" and "Suppressed". Additional statuses are shipped, but these additional ones can be changed or removed if desired.

Configuration of Defect Type

In order to facilitate enhanced management of defects, it is now possible to configure both defect types and defect sub-types.

The configuration tables are **/BTI/AUT_C_BTTY** and **/BTI/AUT_C_BTSY** for the defect type and sub-type respectively. Please use transaction SM30 upon the central system to manage these tables.

Both of these new fields have been added to the main defect management grid. Additionally, the user can also use filtering upon these two fields.

It is not possible to set the default values for these fields when the defects are automatically proposed. However, you can quickly edit these fields when the defects are being triaged in order to classify them according to your requirements. Note that these two new fields do not need to be used at all.

Configuration of Closure Reasons

Prior to release 2.30, a defect could be put into an appropriate status as required by the user. However, this capability did not make it clear why the defect was put into that status in the first place. In particular, when defects were closed, it was often wanting to be reported whether it was closed because of such reasons as that the problem was resolved, the problem was expected, the problem could not be understood or a false negative was identified.

For this reason, it is now possible to configure and specify a “closure reason” against the defect which can be set when the defect is worked by the testing team.

Closure reasons are configured in table **/BTI/AUT_C_** which can be maintained via transaction SM30 in the central system.

Linkage Validation After Playback

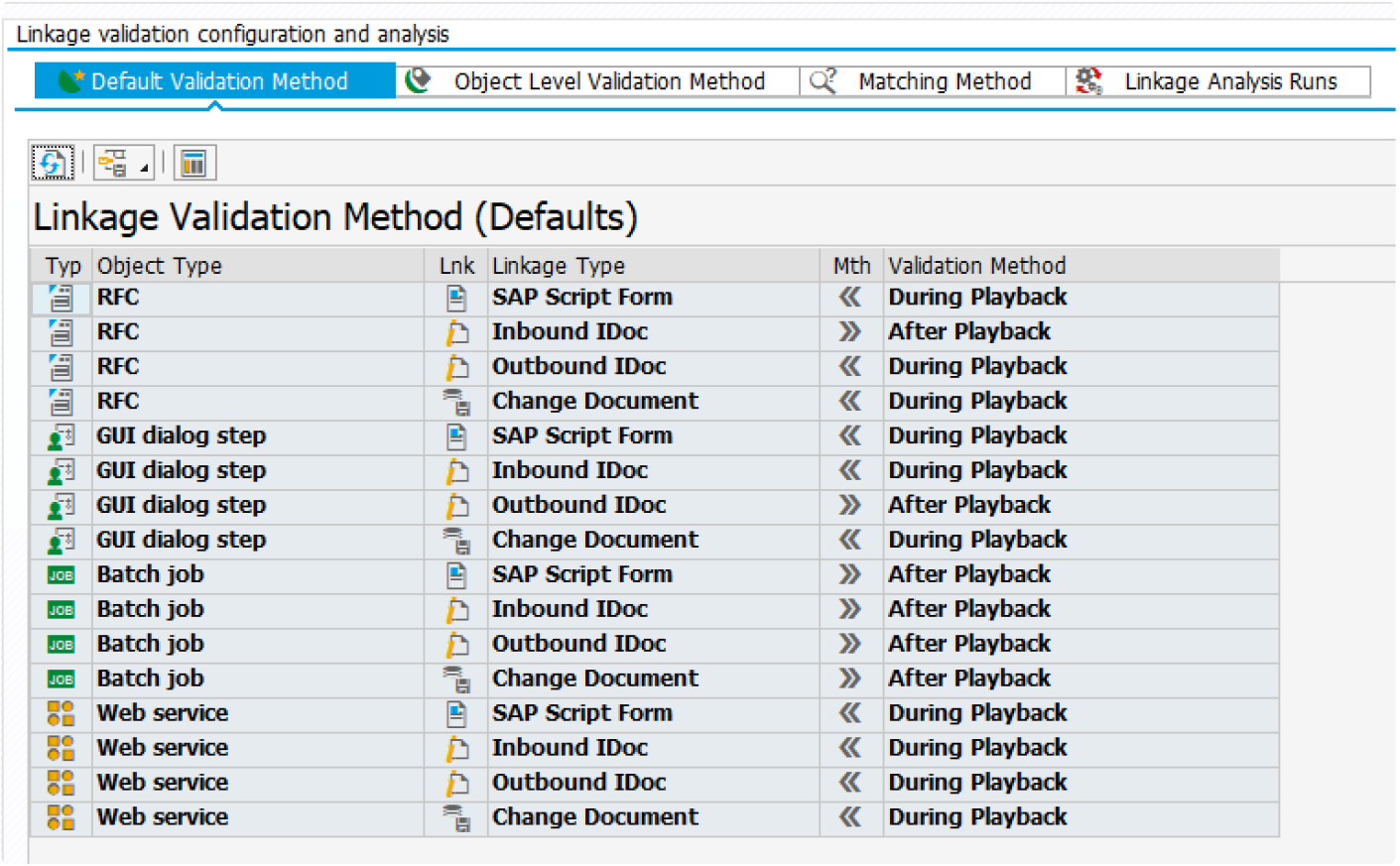
A major issue with validation of linkages for batch jobs in particular was that there would often be slight discrepancies in timing and data during the playback. This meant that during the recording, a batch job (for example) may produce 1000 outbound IDoc's, but during the playback it would produce 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, was minimal. Thus, Testimony would almost always flag these batch jobs up with a failure thus creating a defect for investigation.

In the latest release of Testimony, you are able to mark a particular object type (e.g. batch job) along with the corresponding linkage type (e.g. outbound IDoc's) such 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.

Default Validation Methods

Default configuration is delivered in 2.30 for the way in which linkages are validated. This is by object type and by linkage type. The default configuration is the following:


Linkage validation configuration and analysis

Default Validation Method | Object Level Validation Method | Matching Method | Linkage Analysis Runs

Linkage Validation Method (Defaults)

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
	Batch job		SAP Script Form	»	After Playback
	Batch job		Inbound IDoc	»	After Playback
	Batch job		Outbound IDoc	»	After Playback
	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

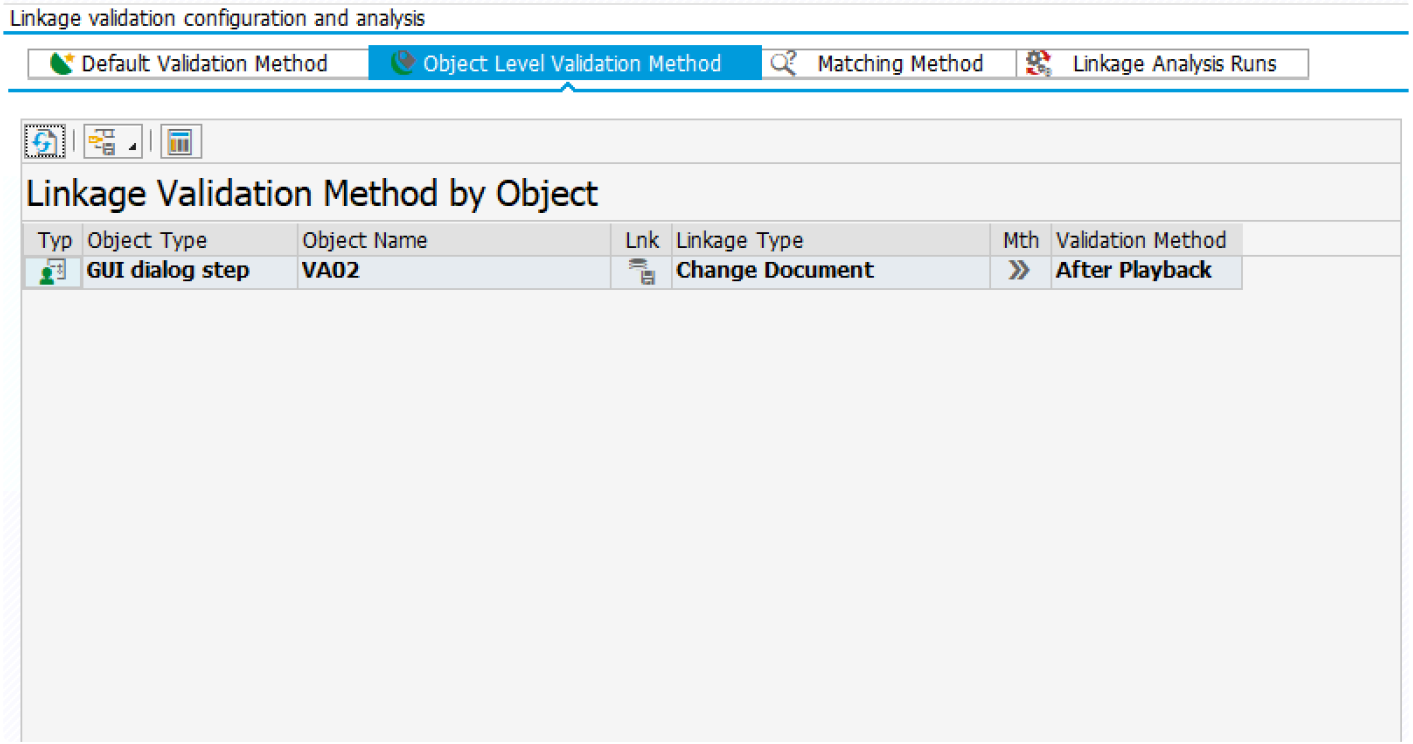
As can be seen, the default validation methods are delivered in 2.30 as default for batch jobs only. You can over-ride these defaults with your own settings as per customer requirements. There are really only two settings here – either the linkages of that particular type are validated **during the playback** or they are done **after the playback**.

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

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.

Intelligent playback row selection

Key Information

A common problem during playback's was that there was no tolerance in the way that bot performed row selections in ALV grids or line selections on ABAP lists. If the data within the SAP system was slightly different, then often the wrong ALV grid row would be selected or the wrong ABAP list line. This lead to many false negatives leading to a greater number of defects.

In this new release of Testimony, the recording will now capture the data within the row(s) of the ALV grid row that was interacted with by the user. Similarly, this is now done for the lines selected in ABAP lists. During the playback, this data is used to more intelligently select the correct row or line.

All of this will lead to more accurate playbacks with less false negatives.

✻ In additional to capturing the data of the row of ALV grid in the recording, additional information within the ALV grid is also used. The columns that are marked as "keys" are used first to more accurately identify the row in the recording compared to the same row in the playback. If this fails (no rows can be identified), then the raw data is then used.

✻ If no row/line can be found based upon the data, the final fall back position is to use the same row/line number from the recording.

! We do not currently support **tree controls** in Testimony and more accurate selection of tree nodes. This is planned for the next release of Testimony or potentially as a patch to this release once ready.

Outbound Web-Service

Key Information

It is now possible to do service virtualization on outbound web-service calls.

In order to do this, you must first activate a new enhancement point in the recording which will capture the outbound web-service calls as linkages against the objects that instigate them. The data captured is the request header and body of the outbound web-service calls.

> Note that there may be more than one outbound web-service call for a given object.

Then, during the playback, a separate enhancement point for outbound web-services must be activated. When the object is triggered by the bot in the playback, any outbound web-service calls will not actually go out to the external system, but will instead be immediately passed back the same data as that in the recording.

> The vast majority of customers running SAP systems will using the RFC protocol for outbound calls to external systems. A smaller percentage of outbound calls will be done via web-services.

New Linkages for Sequencing

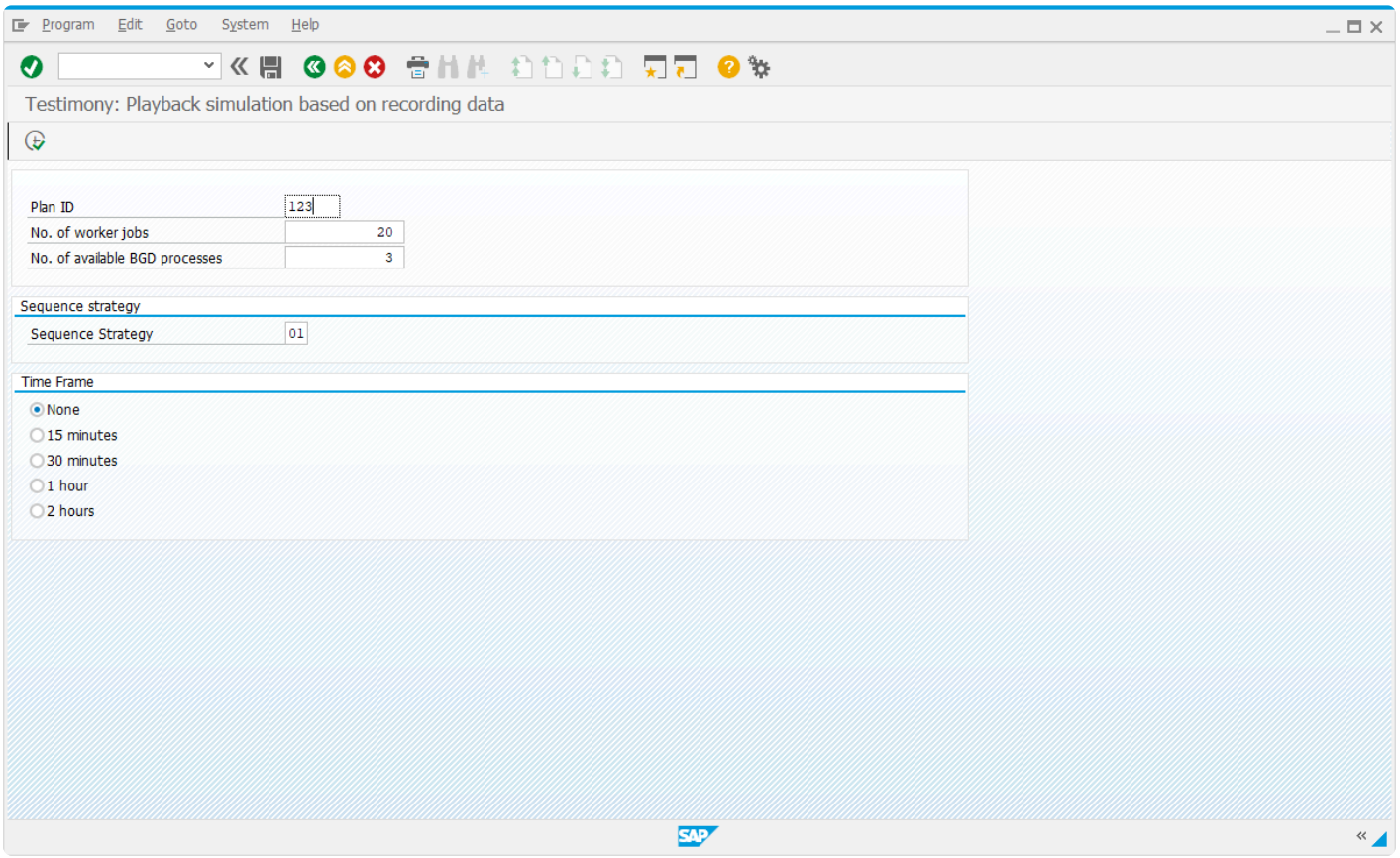
With Tesitmony v2.21 a number of enhancements have been made to the playback simulation program to allow you to understand the requirement on the bots prior to building the execution queue. In this version of the pre-playback simulation, you must still have created a recording, and it is on the data in this recording that the pre-playback simulation will process.

The key output of the pre-playback simulation is to give you a heads up on the expected run-time of the playback and how many bots you will require.

The program name is **/BTI/AUT_EARLY_PLAYBACK_SIM**. It should be run upon the central system only. You specify the test plan ID and the number of worker jobs you have available for use in the central system to facilitate the playback. You should also specify how many background jobs you expect to have available on the target system when you run the playback.

The sequencing strategy should be set to 02 (full throttle) unless for smaller playbacks you intend to use sequencing strategy 01. You can also specify the time frame to obtain an estimate on what the playback will do every N minutes.

A sample of the select-options for the program is shown below:



Pre-playback Simulation Program

Authorization / Dump Failures

Two new failure reasons have been introduced in the 2.30 release that are common differences detected by Testimony. This depends upon the scenario under test, but nonetheless they offer enormous value for detected potential regression scenarios.

The two new failures reasons are:

1. Authorization Failure
2. Short-Dump Failure

Each of these two new failure reasons are described in the following sub-sections.

Authorization Failure Reason

The new authorization failure reason has been introduced to automatically detect scenario's where in the recording a transaction was able to be run by the given user, but in the playback the same object cannot be run as the user is no longer authorized.

Example Scenario

A common scenario for use of this scenario in regression test cycles is when authorization changes are being introduced either knowingly or unknowingly. If there are hundreds or thousands of users to be managed and a rationalised list of common roles, then a small change to alter permissions can have a drastic effect on users once the change is promoted to production.

> This applies to all object types not just SAP GUI dialog transactions. It covers batch jobs, inbound RFC calls and inbound web-services. In the next release, this will be extended to Fiori apps where users are no longer able to start their apps or execute functions within the app.

Process

All authorization failures will be detected automatically by Testimony **during** the playback for all object types. When a failure is detected by Testimony in the playback, the last authorization failure for the given user is checked by the playback engine and its details stored.

> The failure reason will be accessible from the playback overview and from within the test script view for the execution queue. It is also visible within the new investigate screen.

Short Dump Failure Reason

A further common use case for testing is ensuring that new changes do not abnormally terminate during execution. This is referred to as a short-dump where the ABAP program abends unexpectedly. This could happen for numerous reasons but any introduction of new changes into a production system can result in this behaviour unless it has been fully tested.

At numerous customers we have seen how it would be very useful to know when a test script abnormally terminates during playback. This would allow the customer to focus on those failures since often this is a fundamental flaw in the new change.

The new failure reason is auomatically set during the playback as each test script executes. This applies to all object types – dialog transactions, inbound RFC, batch jobs and inbound web-services. This will be extended to Fiori in the next release.

If a test script fails due to a short-dump, it will have its failure reason updated appropriately. The playback overview will break down the failures by application component and failure reason.

Example Scenario

A customer has implemented an user exit in work management that calls a SAP standard external subroutine. After upgrading the system with a support pack, it is found that this new subroutine has an additional mandatory parameter. During playback, this mismatch causes a short-dump in all test scripts that trigger that user exit. The short-dump reason is clearly visible in the playback overview and can be focused on by the test team to remediate the issue in time for retest and production deployment.

Outbound RFC enhancements (Beta)

Recording window discovery

Other Product Changes

The following additional changes have been made in the Testimony v2.30 release.

Testimony Functionality	Enhancement Description
Exclusion of RFC parameters	Ability to exclude particular parameters from the comparison for a given RFC. This must be the entire parameter name, not a sub-component
External session not found issue	Z includes rather than /BTI/AUT includes during enhancement activation
Inbound / Outbound RFC data size limits	New parameter to restrict the amount of data that can go into SHM
SAP GUI for HTML support	Full support for SAP GUI for HTML. Internally tested and at SAP
Utility to delete test plan data	Used for internal purposes first. Can be used carefully in exceptions
Investigate container changes	Various usability changes for viewing expected / actual outputs
Bot RFC playback anomalies	Inbound RFC scale out testing. Data types not matching up
Coverage Analysis and filter sets	Coverage analysis now takes filter sets into account
IDoc’s added in error	If partner profiles deactivated on target, IDoc’s added in error now ok
Logon scripts when SHM filled	When shared memory filled up, logon’s weren’t captured
Protect beta functionality via password	Ensure beta functionality cannot be inadvertently activated by customers
Missing roles from build	Two key roles were missing from the standard build program
Dynamic ID broken in block build	Fix delivered to resolve issue where scripts were left unprocessed
S/4 HANA transaction testing	Issues found during internal testing on S/4 HANA 1909 system HD4
Block building memory	Memory consumption during block building was very high

usage	
Inbound RFC exception handling	Additional handling of edge case exceptions during inbound RFC
Residual inbound RFC in SHM	Failed RFC calls were remaining resident in SHM during recording
Flag to indicate if written to DB	Business transactions have new flag indicating if it was written to DB
Unprocessed scripts for inb. RFC	Large inbound RFC blocks containing circular dependencies resolved
Exclusions on the fly	This had issues in 2.21 (and prior) so this was fixed
Open new session using /o	This never worked and caused anomalies in the playback
Transaction variants supported	This caused issues during the playback but is now resolved
Central / Source client issue	When these were the same system, client was causing issues
SAP GUI session number on S/4	Critical issue identified in S/4 systems with kernel assigning session number
User preferences issues in playback	User layout tables deleted during playback in certain t-codes
Batch job number in descriptor	More accurate change docs assigned by using job number in selection

Notes on S/4 HANA Systems

Testimony is built to function on a central system which records one or more source systems and plays back that recording into one or more target systems. These 3 system types would typically be separate systems. All of the systems must be ABAP stack and on Netweaver release 7.01 and above.

In the case of S/4 HANA systems (1610, 1709, 1809 and 1909), it is possible to install Testimony for recording and playback purposes. The central system would traditionally be a separate Solution Manager system.

> ✻ It is important to understand that this process is not just for S/4 HANA systems. It is for any SAP system that has a Netweaver kernel of 7.53 and above. This could be relevant to ECC systems.

Before you record your S/4 HANA (or NW 7.53+ kernel) system, please liaise with Basis Technologies who will provide you with an additional transport request that should be applied to both your central system and your source S/4 HANA system. The same transport should then be applied to your S/4 HANA playback system prior to starting a playback.

This transport should be applied on top of the main recording transport request into that system.

> ! **It is critical**, that you only apply this transport out of normal system hours. Please work directly with your main point of contact at Basis Technologies in order to seamlessly deploy this aspect of Testimony.

Bug Fixes

The following bugs have been addressed in the Testimony v2.30 release. Please note that this is not an exhaustive list of bug fixes, it is purely intended as a summary of the main issues reported by existing customers in the most commonly used areas of the tool.

Testimony Functionality	Bug that has been fixed
Playback Summary Generation	At the end of a playback, the playback summary was not being automatically generated. This is now resolved.
Editing Defect comments	A fix was implemented for editing defect comments that sometimes threw an exception.
Progress in bot monitoring UI	A minor fix to the progress indicator when refreshing the bot monitoring grid.
Prevent multiple bot instances	The bot executable could previously be started up on the same machine under the same Windows user leading to anomalies in the playback. This has now been prevented.
Resuming after midnight boundary	When not enough time was left in the playback prior to midnight, the playback could not be resumed since it would immediately pause again. This is now resolved.
Scripts incorrectly run on next day	In some cases, scripts were incorrectly run on the following day after the midnight boundary was reached. This no longer happens.
Step level suppressions created defectsl	Step level suppressions are now correctly set to a state of passed and will no longer create defects.
Missing authorization prep step failure	A missing authorization object in the default provided role /BTI/AUT_SOURCE_RFC was causing a preparation step to fail. This is now resolved.