

PUBLIC

SAP Datasphere, SAP BW bridge
Remote Conversion Guide

Version 4.0
November 2025

Contents

1. Introduction 4

1.1 Disclaimer 4

1.2 Overview and Prerequisites 5

1.3 Sequence and Documentation 11

2 Preparations for the Remote Conversion - Scope Collection Check 13

2.1 Collect Scope for Transfer 14

2.2 Define Object Mapping and Store Object List 16

3 Realizing the Remote Conversion 21

3.1 Package Settings (Active Phase) 24

3.1.1 Load Initial Package Settings 24
3.1.2 Define RFC Destinations 24
3.1.3 Verification of Active Packages 25

3.2 Collect Scope and MetaData Transfer (Active Phase) 26

3.2.1 Collect Scope for Transfer (Data flow) 27
3.2.2 Define Object Mapping and Store Object List 31
3.2.3 Prepare Propagate Requests 33
3.2.4 Map InfoObjects for InfoProvider 35
3.2.5 Select Main Partition Criteria for Semantically Partitioned 35
3.2.6 Determine Usage of Involved Objects 36
3.2.7 Checklist for Usage of Involved Objects 36
3.2.8 Confirm Metadata in Target System with Data Transfer Prep 37

3.3 Setup Phase 39

3.3.1 Make Settings for Selected BW Object Tables 39
3.3.1.1 Original System: Read Object Mappings 40
3.3.1.2 Original System: Identify BW Table Pools for Data Migration 40
3.3.1.3 Original System: Collect and Transfer Package Settings Information 41
3.3.1.4 Target System: Identify Semantic Differences between InfoCubes and ADSOs 41
3.3.1.5 Target System: Identify InfoCubes for Migration 42
3.3.1.6 Target System: Generate Dimension and Fact Tables for InfoCubes 43
3.3.1.7 Target System: Build Table Mappings between Mapped Objects 44
3.3.1.8 Target System: Display DDIC Differences Between Mapped Objects 46
3.3.1.9 Target System: Configure MIGSID Conversion Rule for Master Data ID Fields 47
3.3.1.10 Target System: Make Settings to Facilitate Program Generation 48
3.3.2 Conversion Setup (Logical System Rename) 49
3.3.2.1 Select Conversion-Relevant InfoObjects 49
3.3.2.2 Define Mapping Values 50
3.3.2.3 Make Settings: Identify Conversion-Relevant Tables and Fields 51
3.3.2.4 Make Settings: Modify Conv-Relevant Customizing for Table-Field Combination 52
3.3.2.5 Make Settings: Validate Mapping Consistency 53
3.3.3 View BW Objects in the Tablepool Browser (Optional) 54

3.4 Generation Phase 55

3.4.1 Make Technical Settings and Generate Programs: Configure Technical Settings 56
3.4.2 Make Technical Settings and Generate Programs: Count Records in Parallel 57

3.4.3 Make Technical Settings and Generate Programs: Make Settings for BW Administration Tables.....59
3.4.4 Make Technical Settings and Generate Programs: Parallel Read Setup.....61
3.4.4.1 Analyze and Prepare Tables for Parallel Read (Optional).....61
3.4.4.2 Monitor Parallel Read Status (Optional).....64
3.4.5 Verify Consistency of Table Portions.....64
3.4.6 Create Sequences to Generate and Execute Reports.....65
3.4.7 Deactivate InfoProvider Tables Containing Data.....66
3.4.8 Generate Programs for Data Migration, Conversion or Deletion.....67
3.4.9 Execute RSRV Checks in the Original System.....70

3.5 Preprocessing Programs During System Lock.....74
3.5.1 Copy Delta Queues from SAPI to ODP Technology.....74
3.5.2 Confirm that delta was loaded for the cloned DataSources.....76
3.5.3 Confirm the INIT SIMU Procedure.....76
3.5.4 Make Settings to Prepare to Lock all Data Changes.....78
3.5.4.1 Validate Migration Configuration.....78
3.5.4.2 Synchronize Delta Queues between SAPI and ODP.....79
3.5.4.3 Extract from PSAs of DataSources and Error Stacks.....80
3.5.4.4 Master Data Activation.....81
3.5.4.5 Propagate requests.....82
3.5.4.6 Lock All Data Changes.....83
3.5.4.7 Prepare Request Mapping for Transfer.....85
3.5.4.8 Activate Task Sequences.....86
3.5.4.9 Configure Execution Dependencies for Migration.....88
3.5.4.10 Delete Tasks Assigned to Empty Tables.....88
3.5.5 Lock Data Changes and Prepare SID Mappings.....90

3.6 Migration Phase.....90
3.6.1 Make Runtime Settings for Migration.....91
3.6.2 Reset Cluster Data Deletion Date.....95
3.6.3 Start Migration.....95
3.6.4 Monitor: Transformation Status (Optional).....96
3.6.5 Confirm Data Selection and Unlock Data Loading.....98

3.7 Post Processing Activities During System Lock.....100
3.7.1 Rebuild Hierarchy and Reset Number Ranges.....100
3.7.1.1 Rebuild Hierarchy tables (/BI+/H*, /BI+/I*).....100
3.7.1.2 Reset Number Ranges.....101
3.7.1.3 Set All Queries for Generation.....102
3.7.2 Execute RSRV Checks in the Target System.....102
3.7.3 View Time-Related Migration Statistics (Optional).....104

3.8 System Settings After Migration.....106
3.8.1 Truncate Generated InfoCube Tables (Optional).....107
3.8.2 Delete Data Clusters in the Active Package (Optional).....108
3.8.3 Set Package Status to Completed.....110

4 Datasphere Integration.....112

4.1 Space Management Overview.....113

4.2 Data Integration Monitor: BW bridge Cockpit.....114

4.3 Data Builder: Import Entities and Space Sharing.....115

5 Data Transfer to a 2-tier Tenant Landscape.....121

1. Introduction

1.1 Disclaimer

SAP BW bridge is a feature of Datasphere and offers the possibility to use BW objects in SAP Datasphere. The ABAP Tenant (where the BW modeling happens) is integrated by its own space into SAP Datasphere.

The tables of the ABAP Tenant (e.g. ADSOs, InfoObject data tables) can be exposed as remote tables in the BW bridge Space of Datasphere and shared to be consumed by other spaces.

The BW bridge space has limitations and its main functionality is to expose the shared remote tables to other spaces. Data Flows, views etc. using the remote tables can only be created in other spaces based on the shared remote tables.

For naming convention, this document will refer to the following terms as:

- *Sender* system: the source/original BW system from where the Metadata and Business Data should be transferred.
- *Receiver* system: the target BW system (BW/4HANA or BW bridge) that will receive the Metadata and Business Data from the sender BW system.

The Remote Conversion can be used for transfers to BW/4HANA (on-premise) and BW bridge (cloud). There can be some differences when using for one or another and this will be mentioned along the document when applied/necessary.

The sample execution described along this document is based on a single tenant (1-tier) BW bridge landscape.

The chapter 5 provides an overview about the data transfer process to a 2-tier tenant landscape (development and production) and also refers a blog with further information.

This document aims to cover the end-to-end process of the Remote Conversion by using practical examples. It also includes tips and recommendations based on experience with real customer projects.

Some activities require access to the BW bridge backend (SAP GUI), however this is not available for customers and partners. For such activities, or if you encounter errors when running a Remote Conversion, please contact SAP by opening an incident under the components BW-B4H-CNV-RMT or CA-LT-BW4.

1.2 Overview and Prerequisites

You can create your BW objects and data models in BW bridge via a greenfield approach (new implementation) and load data to them from the connected source systems. However, when you already have an operational BW or BW/4HANA system and wishes to move to SAP Datasphere, BW bridge. It can also be helpful and convenient to use an automated way to transfer such objects without the need to recreate them from scratch.

When migrating to SAP BW bridge, the SAP tooling offers 2 (two) automated paths:

- **Shell Conversion**: for metadata transfer (see the Conversion Guide for further details)
- **Remote Conversion**: for metadata and business data transfer.

When the receiver is a BW/4HANA system, there is an additional path called In-Place Conversion (not applied to BW bridge). For information regarding the migration to BW/4HANA, see the BW/4HANA Conversion Guide, the BW/4HANA Simplification List and the SAP Note 2383530.

We can see the Shell Conversion as a subset of the Remote Conversion. By using the Remote Conversion, it is possible to automatically transfer the already existing data of the sender system. This transfer (metadata and business data) is done via RFC.

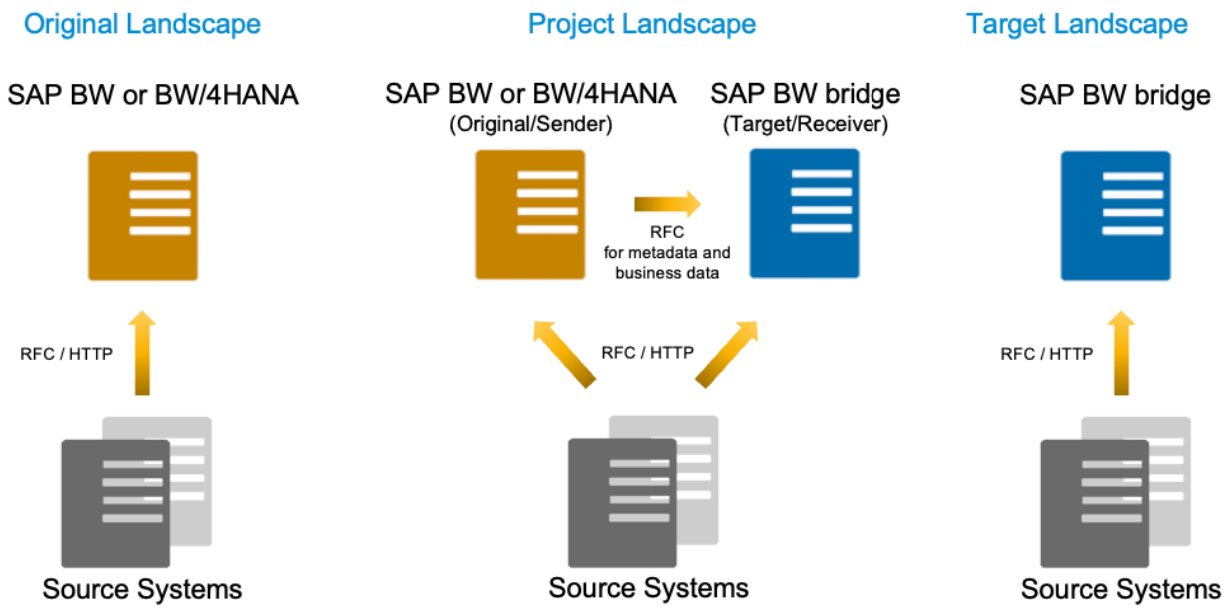


Figure 1: Remote Conversion overview

There are some differences between Shell and Remote Conversion in terms of the scope collection logic for the metadata transfer, especially because of the relationship and dependencies of the request management. When using the Remote Conversion, data and

request Information need to be copied and converted from the sender into the receiver system (Request ID to TSN), therefore all objects on which a data request is touching become relevant for the scope collection in order to ensure consistency.

The metadata transfer converts object types (TLOGO). For example, Classical DSOs and InfoCubes to Advanced DSOs; PSAs to Advanced DSOs (depending on the data flow, it can be optional or mandatory); InfoPackages to DTPs. For PSA-ADSO replacement, check SAP Note 2949412

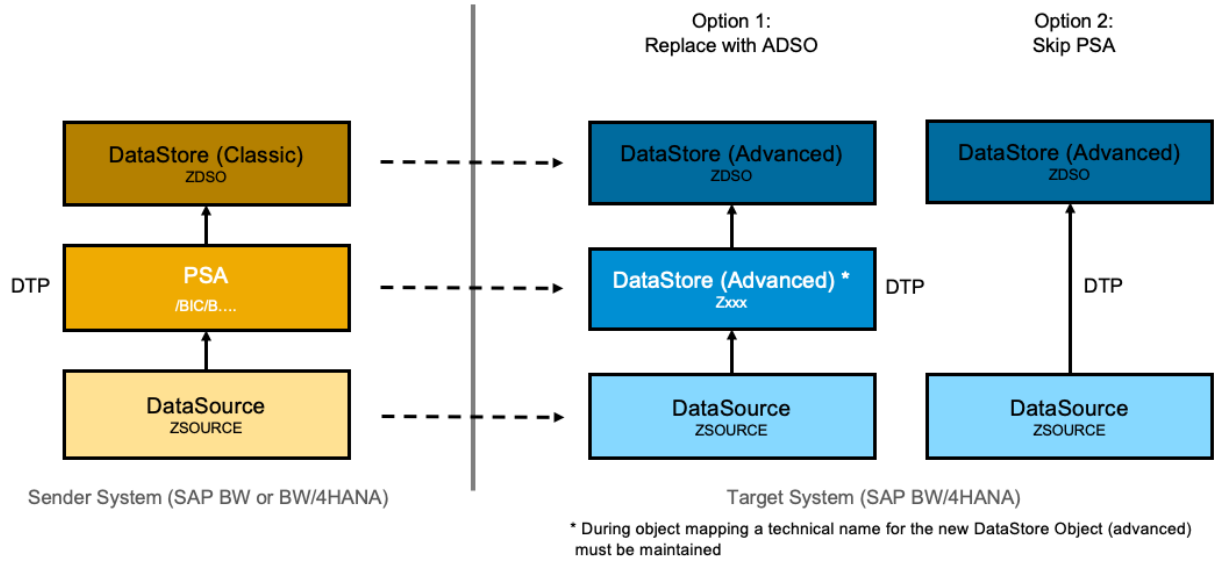


Figure 2: Metadata transfer

For the business data transfer, programs are automatically generated for each table in scope to select, transfer and insert the respective data into the receiver system. Cluster tables are used to control and manage the data transfer process.

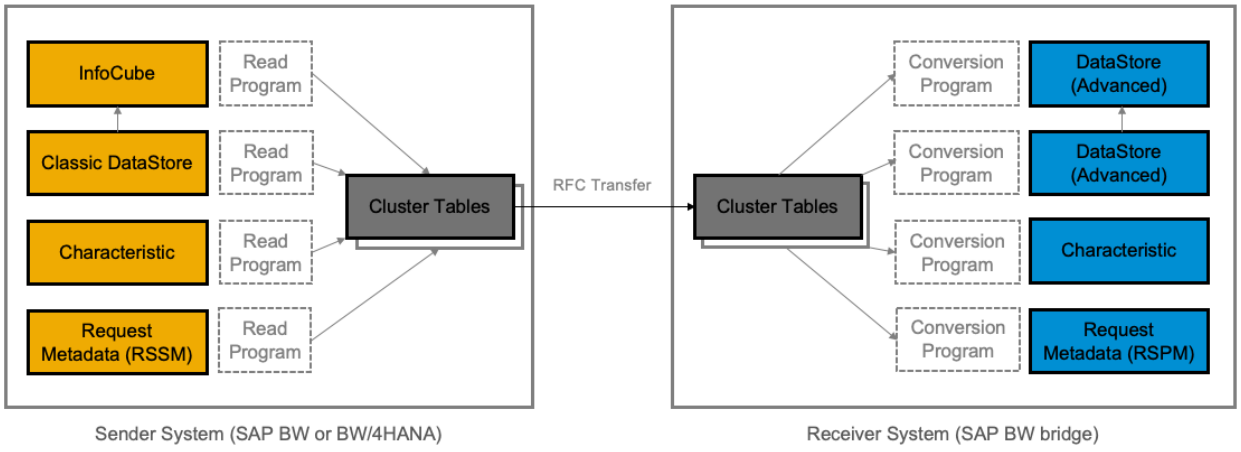


Figure 3: Remote Conversion logic

Besides the business data transfer automation, another important capability of the Remote Conversion is the delta queue cloning. Delta queues are cloned and synchronized before the data transfer, which is beneficial because it ensures consistency of the delta loads after the data transfer is finished. For example, a delta DataSource that already loads data into the sender system will be cloned and start to extract deltas also to the receiver system.

No downtime is required (data load freeze is recommended to speed up the cloning and synchronization). This capability allows a smoother transition to BW bridge or BW/4HANA through the co-existence of both sender and receiver systems to get data loads from that DataSource. This can be beneficial if you do not want to immediately shutdown/decommission your actual productive sender system (e.g. if you wish to extend the validation period with business users and analysts or implement a hybrid scenario). The Shell Conversion does not offer this capability and in that case all delta DataSources in scope should be manually initialized/loaded to the receiver system after the metadata transfer.

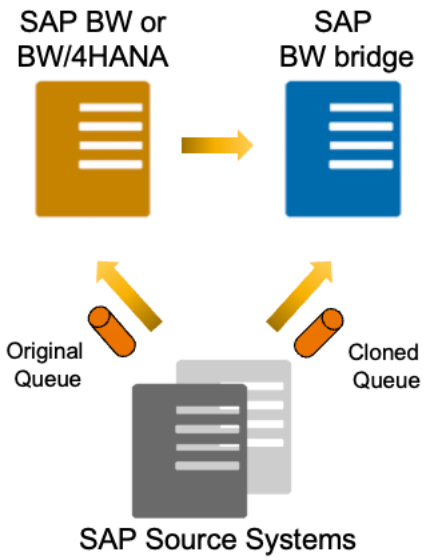


Figure 4: Delta queue cloning

The Remote Conversion to BW bridge is supported by the tooling from BW 7.3 (SP10 and higher). The DMIS add-on installation is required (not needed for the Shell Conversion).

If you are using the embedded BW in ECC or S/4HANA, it is possible to use the migration tools for a transfer to either **BW/4HANA** or **Datasphere BW bridge**.

This is valid for both Shell and Remote Conversions; the pre-requisite is that the sender system should be on NetWeaver 7.51 or 7.58 releases

For detailed information about pre-requisites and preparations, see:

⇒ SAP Note 3141688 - Conversion from SAP BW or SAP BW/4HANA to SAP Datasphere, SAP BW Bridge.

This note contains a .zip file (SAP_Bridge_Transfer_Note_Analyzer_YYYY-MM-DD.zip) with .xml files for each scenario. There are 2 (two) for the Remote Conversion: 1 (one) for when the sender system is on BW 7.x and 1 (one) for when the sender system is on BW/4HANA.

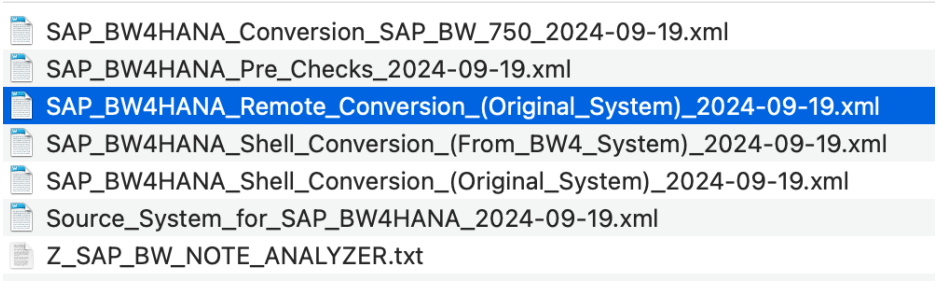


Figure 5: Note Analyzer .zip file

You should import this file to the sender system by using the Note Analyzer report (is also included into the above .zip file).

The file Source_System_for_SAP_BW4HANA_YYYY-MM-DD.xml is also required to be imported to the source systems which are connected to the sender system (e.g. ECC, SCM, etc.). This is required especially for ODP enablement of the DataSources (see SAP Notes 2481315 and 2232584)

The imported .xml file contain a list of notes which are required to be implemented to enable the Remote Conversion. It brings enhancements and corrections in a frequent manner therefore it is important and recommended to always have the latest version implemented.

Differently when the receiver system is on BW/4HANA on-premise, there is no .xml file to be imported to the receiver system as BW bridge is on cloud.

Differently when the receiver system is on BW/4HANA, backend access to BW bridge via SAP GUI is not possible for partners and customers. BW bridge objects and data flows are only accessible via Eclipse/HANA Studio (BWMT).

Connect the Sender System to SAP BW bridge

The sender system must be connected to the SAP BW bridge tenant using the following steps:

- Add the SAP Datasphere Subaccount in the Cloud Connector
  - Before connecting the sender system to your SAP BW bridge tenant, you must add the SAP Datasphere subaccount to the Cloud Connector. See SAP Help for detailed instructions.

- Add a Service Channel to the SAP BW Bridge Tenant in the Cloud Connector
  - The sender system must be able to call the SAP BW bridge tenant via RFC. Therefore, a service channel must be added in the Cloud Connector. See SAP Help for detailed instructions.

- Create a Communication System in the SAP BW bridge Tenant
    - A communication system is a specification of a system that represents a communication partner and the technical information required for the communication, such as the user for inbound communication. The sender system must be created as communication system in the SAP BW bridge tenant:
        i. Log on to the SAP BW bridge Cockpit. See SAP Help for detailed instructions.
        ii. In the 'Communication Management 'section, select the app 'Communication Systems'.
        iii. Click 'New' to add a new Communication System.
        iv. In the 'New Communication System' dialog, enter a 'System ID' and a 'System Name' and choose 'Create'. We recommend choosing *MIGRATE_<SID>* as System Name, e.g., *MIGRATE_KST* if the SID of the sender system is KST.
        v. Under 'Technical Data', maintain the following values:
            a. Enter an arbitrary name as 'Host Name'.
            b. Set 443 as 'Port.'
        vi. Under 'Users for Inbound Communication', click (Add) to maintain the user that is used for the inbound communication in the SAP BW bridge tenant. Select an existing user or click New User to create a new user. The required authorizations are automatically granted to this user when creating the communication arrangement (see below). The authorizations are pre-defined by SAP in the communication scenario used by the communication arrangement.
        vii. Save the new communication system.

- Create a Communication Arrangement in the SAP BW Bridge Tenant
    - A communication arrangement is a runtime description of a specific communication scenario. It describes which communication partners communicate with each other in the scenario and how they communicate. A communication arrangement for the migration scenario must be created in the SAP BW bridge tenant:
        i. Log on to the SAP BW Bridge Cockpit.
        ii. In the 'Communication Management' section, select the app 'Communication Arrangements'.
        iii. Click 'New' to add a new Communication Arrangement.
        iv. In the 'New Communication Arrangement' dialog, enter the following values:
            a. Open the value help for 'Scenario' and choose both predefined communication scenarios *SAP_COM_0691 (SAP DWC BW Bridge - Migration Integration)* and *SAP_COM_0818 (SAP BW Bridge - Data Migration Integration)*. Here is the main difference with the Shell Conversion, because the Remote Conversion also needs *SAP_COM_0818.*
            b. Discard the proposal for the 'Arrangement Name'. Instead, use the technical name of the source system as 'Arrangement Name' and choose 'Create'.
        v. In the 'Common Data' section, choose the communication system you created earlier. The system automatically fills in:

a. the ‘User Name’ in the ‘Inbound Communication’ section with the user you maintained as ‘User for Inbound Communication’ in the communication system.
b. the required ‘Inbound Services’.

vi. The User for Inbound Communication automatically receives the authorization profile pre-defined by SAP in the communication scenario, i.e., the authorizations required to invoke the RFC call from the sender system.
vii. Save the new communication arrangement.

- Create an RFC Destination in the Sender System
  - During shell transfer, when maintaining the scope of the transfer in the task list, you must choose an RFC destination pointing to the receiving SAP BW bridge tenant. This RFC destination must be created in the sender system:
    - 1. Log in to the sender system.
    - 2. Call transaction SM59.
    - 3. Choose ‘Create’ to create a new RFC connection.
    - 4. In the ‘Create Destination’ dialog, enter a name for the RFC Destination and choose ‘RFC connection to ABAP system’ as Connection Type.
    - 5. Under ‘Description’, enter a meaningful description.
    - 6. On the ‘Technical Settings’ tab, enter the hostname of the Cloud Connector (without https:// and without port number) as ‘Target Host’ and the local instance number you defined in the Cloud Connector for the service channel as ‘Instance Number’.
    - 7. On the ‘Logon & Security’ tab, enter the ‘Language’, 100 as ‘Client’ and ‘User’ and ‘Password’ of the user you defined as ‘User for Inbound Communication’.
    - 8. Save the new RFC destination.
    - 9. Perform a ‘Connection Test’ to check if the RFC destination works as expected. In case of errors, check the error messages in the ‘Problems’ view and right-click the error messages to see if the ‘Full Description’ contains additional information on the root cause. Check the settings of the involved firewalls with your infrastructure team if the connection test is failing.

1.3 Sequence and Documentation

The Remote Conversion to BW bridge has the following sequence:

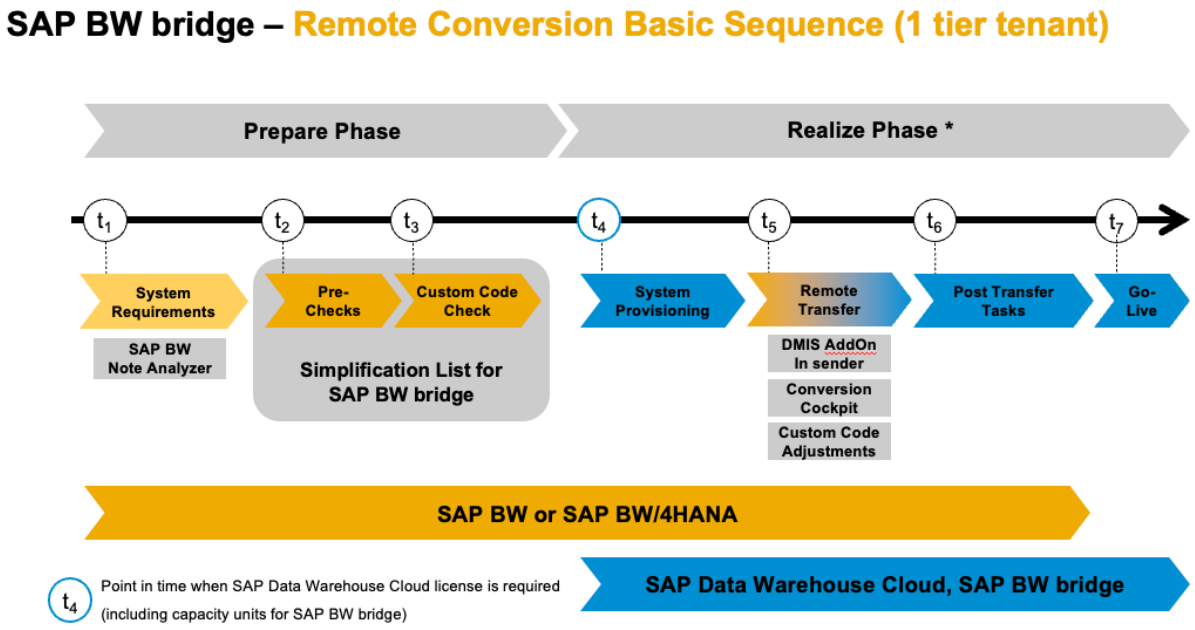


Figure 6: Remote Conversion sequence

There are some important documents which may serve as a reference for the planning and execution of your Remote Conversion to BW bridge:

Document	Available at	Comment
Getting Started with SAP Datasphere, SAP BW Bridge	Getting Started with BW bridge	Get onboarded to SAP Datasphere, SAP BW bridge and prepare your environment for acquiring data via the bridge
Information/ Restrictions Note for SAP Datasphere, SAP BW bridge	SAP Note 3117800	Provides information about restrictions in SAP Datasphere, SAP BW bridge
Conversion from SAP BW or SAP BW/4HANA to SAP Datasphere, SAP BW Bridge.	SAP Note 3141688	This SAP Note describes the different paths to get from SAP BW or SAP BW/4HANA to SAP Datasphere, SAP BW bridge.

Simplification List for SAP Datasphere, SAP BW bridge	SAP Note 3154420	Provides collection of all relevant SAP Notes related to Simplification List for SAP Datawarehouse Cloud, SAP BW Bridge
Datasphere, First Guidance: Development Guidelines and Naming Conventions	First Guidance	This document serves as a guideline for creating (reusable) data models for various source system connections, based on best practices and experience from existing projects about development as well as naming conventions.

2 Preparations for the Remote Conversion - Scope Collection Check

It is possible to check/simulate the scope collection of your scenario/data flow through a transfer check Task List (SAP_BW4_TRANSFER_CHECK_CLOUD_RMT), which can be created via the transaction STC01:

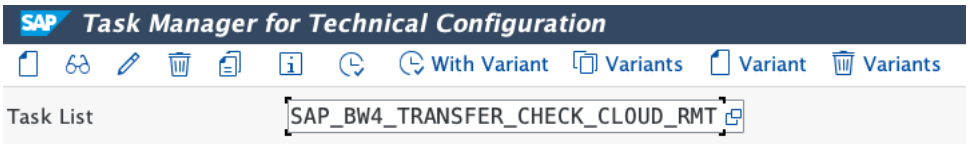


Figure 7: Task List creation (STC01)

It is beneficial to run this Task List beforehand as a preparation in order to anticipate/identify potential issues with the scope to be collected and work on its resolution without having to start the "real" Remote transfer Task List (SAP_BW4_TRANSFER_CLOUD_REMOTE). Besides, you can run it without the need to assign it to a Migration Package (transaction RSB4HCONV).

We can create a Task List run with type SAP_BW4_TRANSFER_CLOUD_REMOTE through the transaction STC01. Afterwards we can call the transaction STC02 and manually enter the BW object(s) to be transferred. A sample data flow was created in a BW 7.5 system for the purpose of this runbook and it will be used along all the activities. The data flow is an InfoCube loaded from an ADSO, with a DTP, Transformation and some InfoObjects.

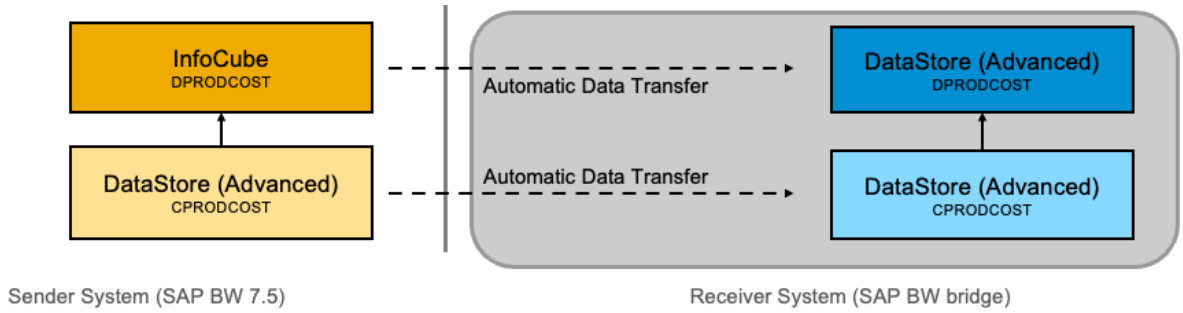


Figure 8: Data flow created for the runbook

SAP_BW4_TRANSFER_CHECK_CLOUD_RMT Task List has basically 2 tasks:

- Collect Scope for Transfer
- Define Object Mapping and Store Object List

2.1 Collect Scope for Transfer

When starting a transfer of objects into the object types which are supported by BW/4HANA and BW bridge, you must collect all objects which have associations to the object(s) that you want to transfer.

In a Remote Conversion, all dependent objects must be collected (based on your scope/data flow). This is because the objects are moved to the receiver system via RFC and can only be transferred and activated there if all dependent objects are available as well.

- Go to “Fill Parameters”.

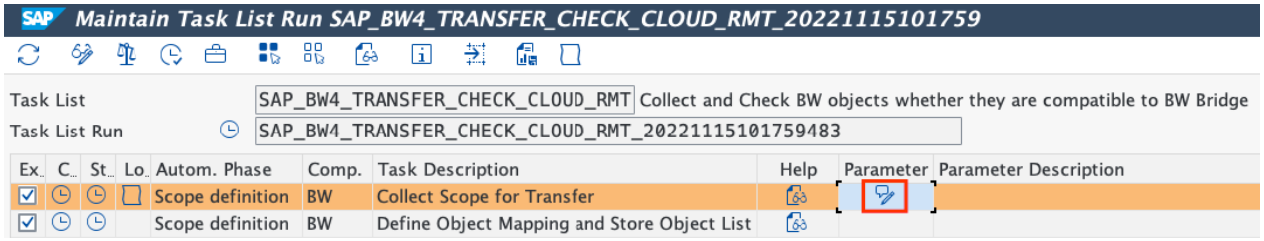


Figure 9: Collect Scope for Transfer - Parameter

- Enter the Object Type and Object Name. I’ve used the higher level one as a starting object - InfoCube DPRODCOST so that all dependent objects can be automatically collected as well.

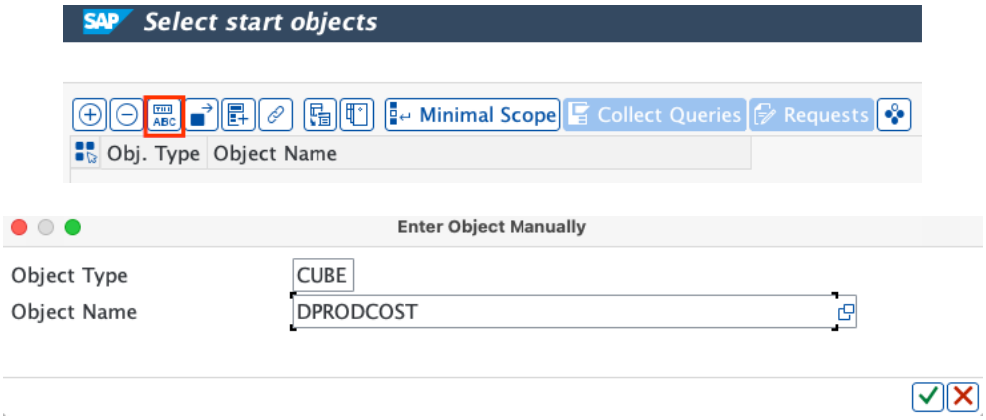


Figure 10: Collect Scope for Transfer - Enter Objects Manually

- Enter the RFC destination from your sender to your receiver system (if not created yet, to be created via transaction SM59), this will be needed independently if your receiver is a BW/4HANA or BW bridge system.

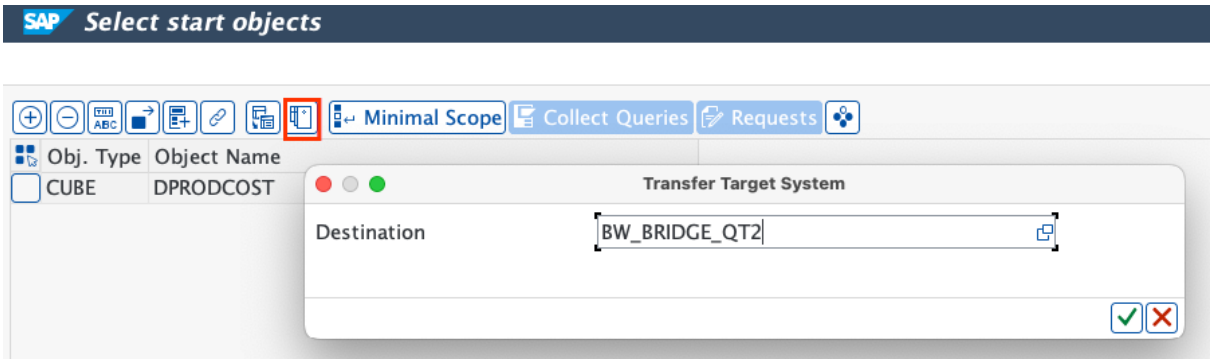


Figure 11: Collect Scope for Transfer - RFC destination

- Save it.

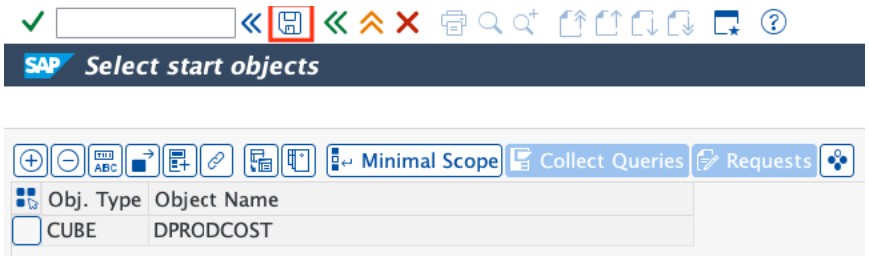


Figure 12: Collect Scope for Transfer - Selected scope

- You may run this task either in dialog or background (recommended for larger scopes).

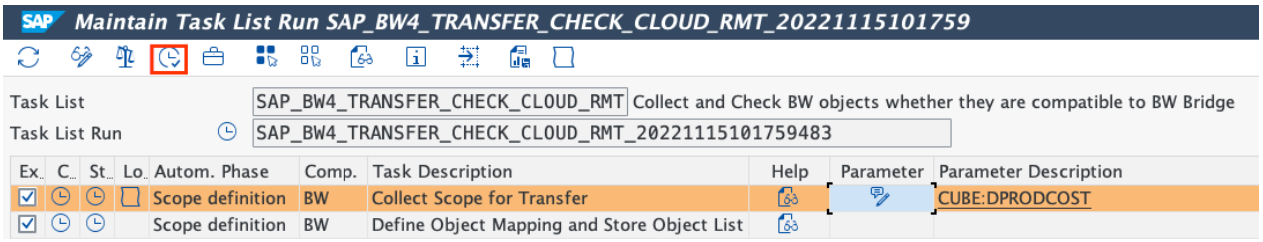


Figure 13: Collect Scope for Transfer – Before execution

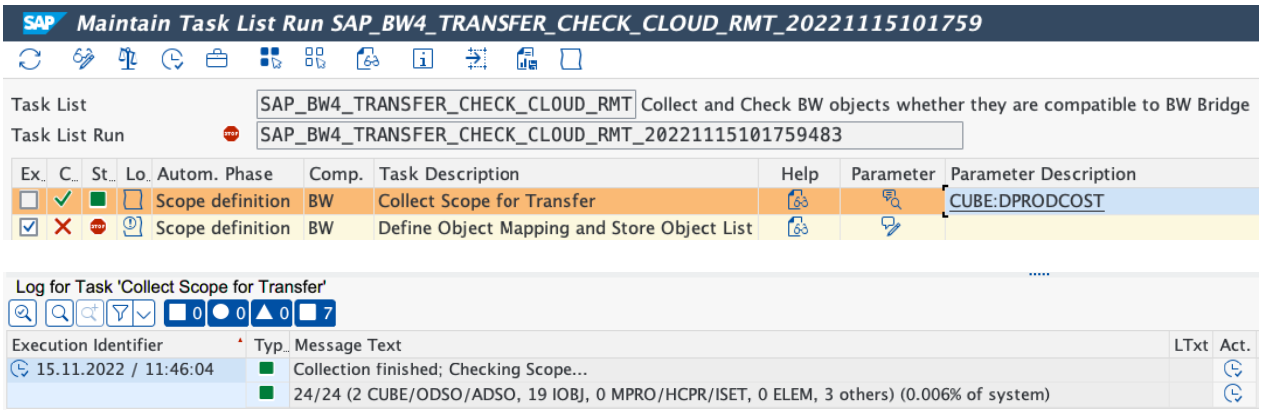


Figure 14: Collect Scope for Transfer – After execution

2.2 Define Object Mapping and Store Object List

This task offers the "main" user interface for the transfer tool. Here you select which objects were collected and are to be transferred (“Transfer Object” flag). The transfer tool will collect all dependent objects based on the “start objects”.

The following resources provide a good explanation about the collection behavior of the conversion approaches:

- BW/4HANA conversion: Scope Collection behavior
- SAP Note 2926288 - Options to make scope smaller for remote conversion

You may create filters to better visualize the object mappings, for example, the original object type (Original TLOGO), new object type (New TLOGO), the new object name and why an object was flagged for transfer (Required Objects).

It is possible to carve-out objects from scope, delete the scope (for a new execution) and download this list for an excel file (e.g., to validate the scope with your customer/analysts).

When using the SAP_BW4_TRANSFER_CHECK_CLOUD_RMT Task List type, the carve-out can be reverted by deleting the scope collection. When executing a “real” remote conversion (SAP_BW4_TRANSFER_CLOUD_REMOTE), the carve-out is a “final” carve-out. Objects which were carved out can’t be part of another scope transfer.

In this example, note that the InfoCube was mapped to a corresponding ADSO. This ADSO will be further imported/created in the receiver system and populated with the business data already existing in the original InfoCube.

- Go to “Fill Parameters” to check the objects that were collected based on the “start objects” (in our sample, InfoCube DPRODCOST).

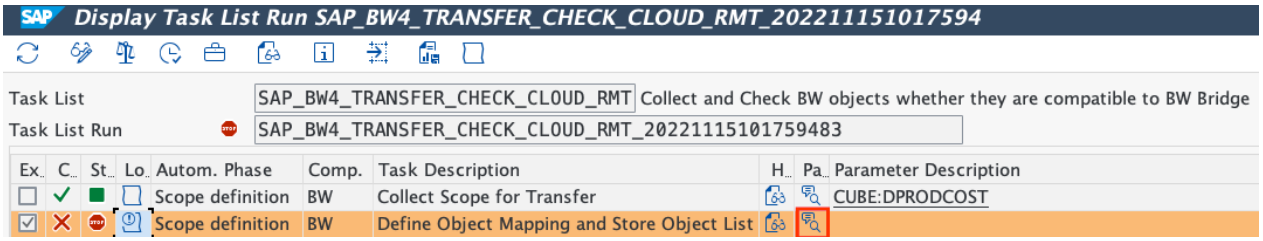


Figure 15: Define Object Mapping and Store Object List - Parameters

- In “Assign Receiver Package and Transport” (the “truck” button), you need to specify a Package and a Transport Request (created in the BW bridge receiver system – check SAP Note 3130759 for further information).

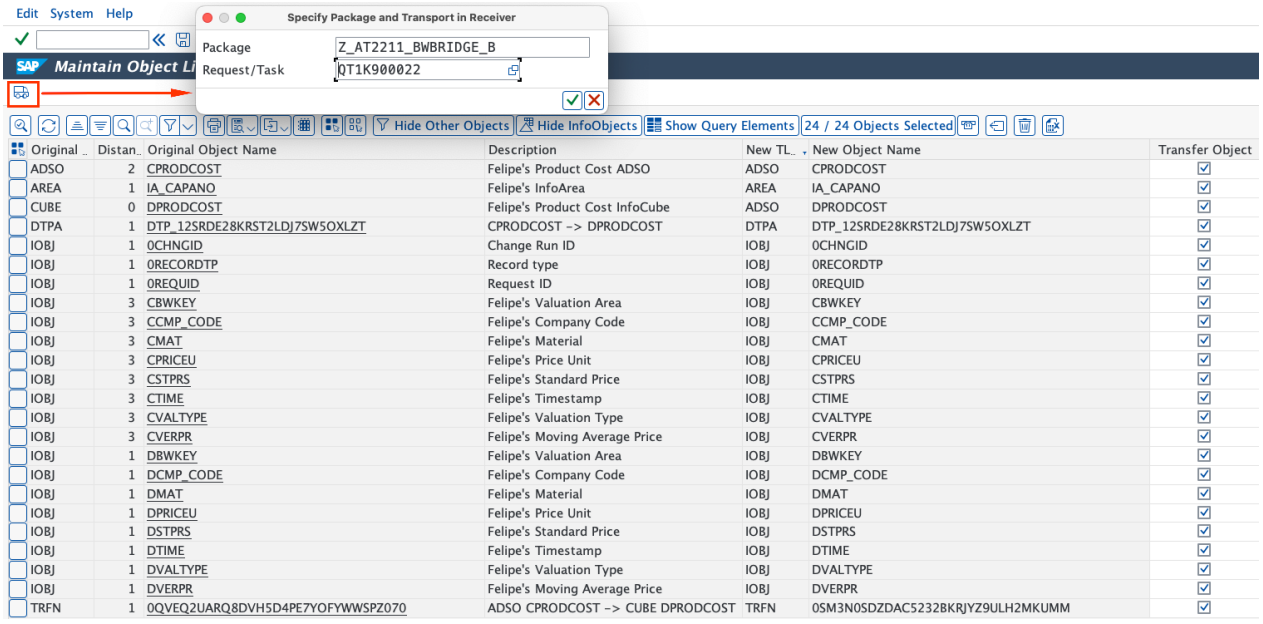


Figure 16: Define Object Mapping and Store Object List - Maintenance

- You should now save it and you will be automatically redirected to the Task List menu. Now you can start/resume the Task List run either in dialog or background.

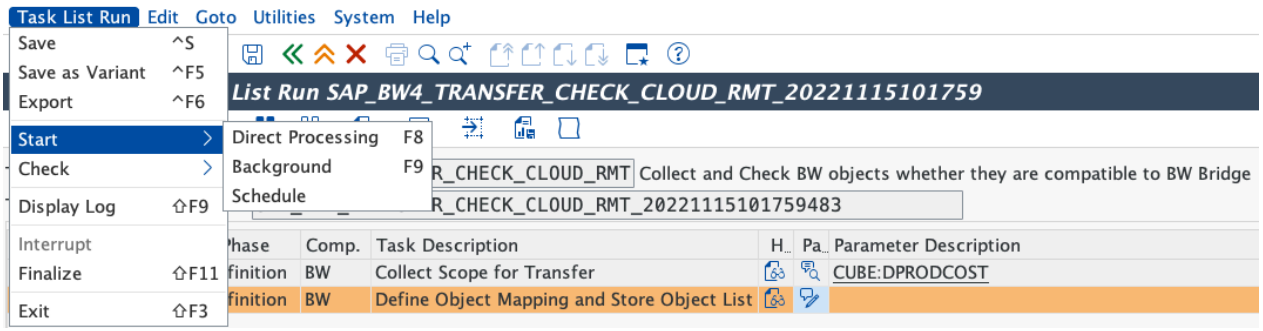


Figure 17: Define Object Mapping and Store Object List - Execution

Similar to the carve-out settings, no information is stored on the database or on the receiver system if the check Task List (SAP_BW4_TRANSFER_CHECK_CLOUD_RMT) is executed, since this is a simulation and everything is part of the current Task List run only. This information is stored when you run a "real" Task List run (SAP_BW4_TRANSFER_CLOUD_REMOTE).

The execution ended with a warning because there are requests which were not compressed in the InfoCube DPRODCOST. The compression is a mandatory activity/pre-requisite because of the request transfer task (to be executed at a later stage) as this will significantly improve the request transfer runtime. The compression takes effort/time, however we need to see it as an investment for having better data transfer performance.

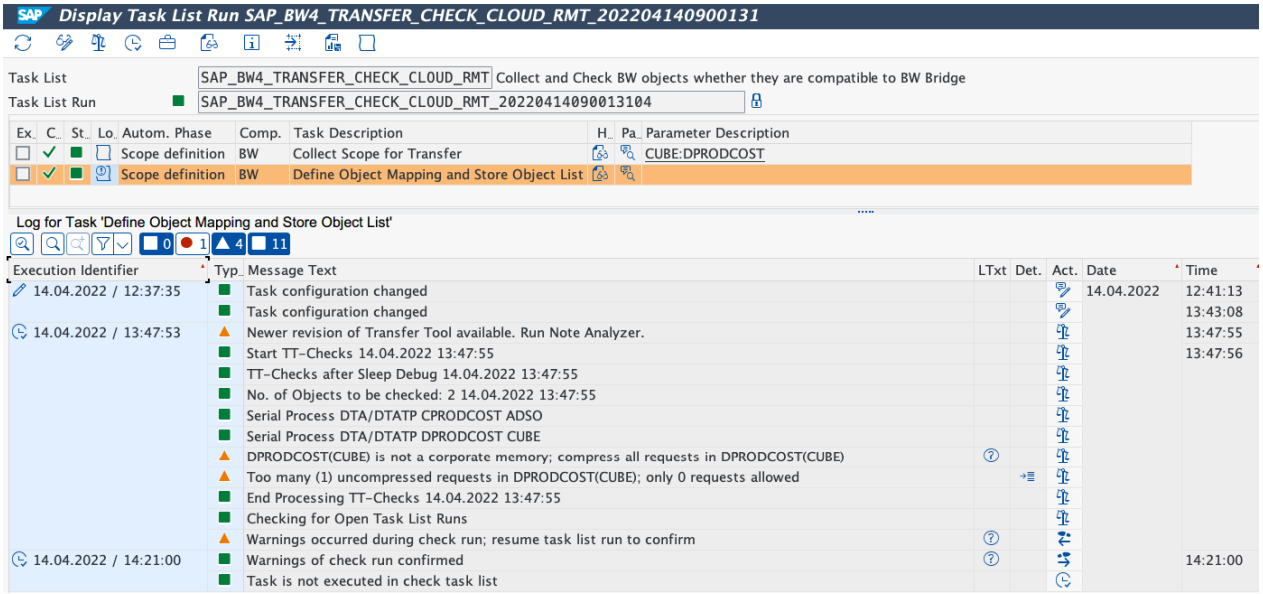


Figure 18: Define Object Mapping and Store Object List – Preliminary results

After compression:

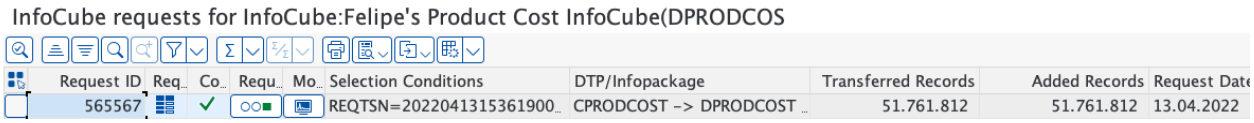


Figure 19: Compression of InfoCube DPRODCOST

- Repeat the execution of “Define Object Mapping and Store Object List”.

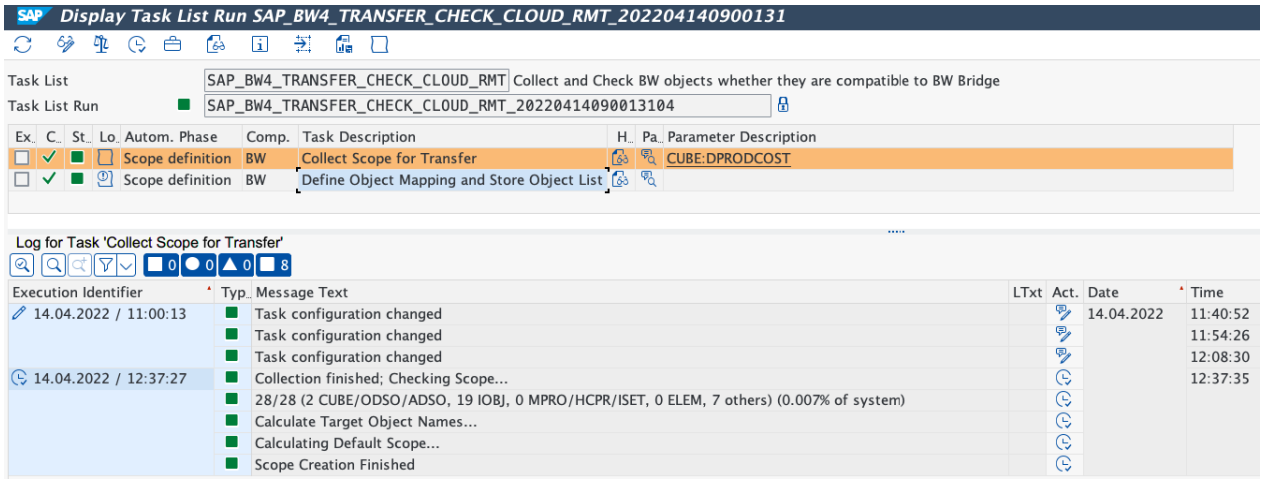


Figure 20: Define Object Mapping and Store Object List - Final results

SAP_BW4_TRANSFER_CHECK_CLOUD_RMT Task List finished successfully and we are ready to start a real/E2E Remote transfer Task List (SAP_BW4_TRANSFER_CLOUD_REMOTE).

Once the scope is collected, especially for a productive run, it’s recommended to define a development transport freeze:

Affected Objects:

All BW objects are affected by the freeze, such as: Queries, InfoObjects, InfoProviders, Transformations, Transformation-Routines and Formulas, DTPs, InfoPackages, DataSources, Process Chains. In addition, all custom code which is maintained in separated reports, functions, or classes, but called from a Transformation-Routine or an InfoObject-Routine - all Data Flows from the data acquisition to the query consumption.

Non affected Objects:

All custom code which is not used in a data load process (i.e., by Transformation- or InfoObject-Routines).

Types of change:

- Delete: This is strictly prohibited for the affected objects. Deleting objects which are in transfer scope will lead to a full stop of the transfer and the objects will have to be recreated.

- Create: This is allowed with restrictions: The new objects must not load data into (and ideally also not out of) objects which are in scope and these new objects will not be part of the transfer, therefore will be missing in the receiver system. An example of an allowed creation is a new report, which displays data of an existing provider (difference between “display” and “load” data).

- Change: This is allowed with strong restrictions. No changes are allowed which create new references to other objects, for instance adding a new InfoObject to an InfoProvider, because this will lead to inconsistencies in the receiver. Also no changes to flags that change the object behavior are allowed. In contrast to that, repairs which do not change references to other objects (like e.g., fixing custom routines) are allowed. However any such repairs must be active and released at the time when the metadata export takes place. They might also be done after the metadata export is done, but then they will be missing in the receiver. In case emergency changes are required, those cases should be communicated in advance to the project team and individually verified.

⇒ Tips:

  - The report RS_B4HANA_OBJECT_LOCKED does return the information, whether a given object is part of the active scope. In case of doubt, the project team should call the report prior to making any change.

  - a warning can also happen with this activity when there is a newer revision of the transfer tool. In our example it was not a showstopper, however it is extremely recommended to have the latest version implemented (check via the Note Analyzer report).

3 Realizing the Remote Conversion

In the transaction RSB4HCONV, select “Execute Scope Transfer”.

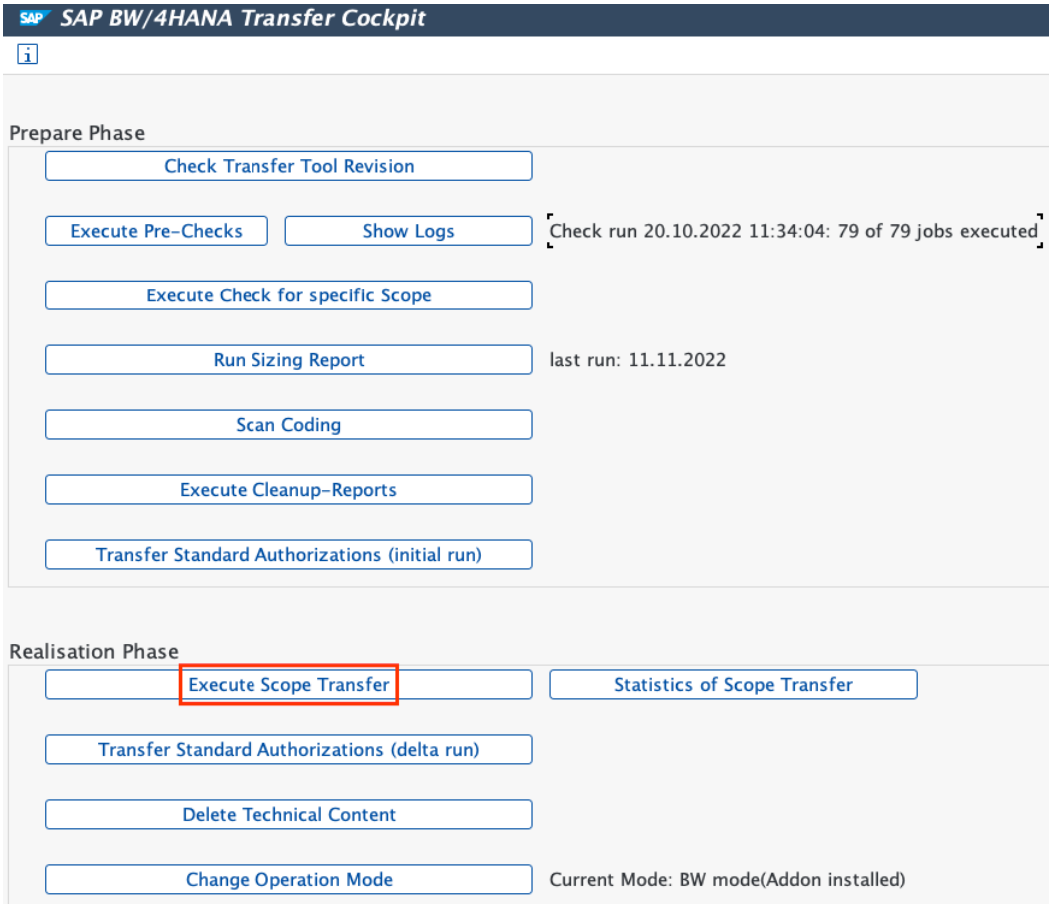


Figure 21: RSB4HCONV transaction

Select the receiver system type and the transfer option. For our sample, we will use a Remote transfer to BW bridge.

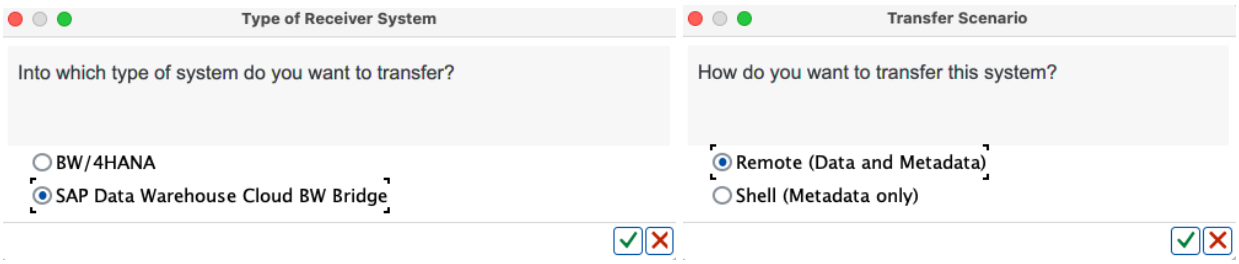


Figure 22: Type of Receiver System and Transfer Scenario selection

Select “New” to create a new package.

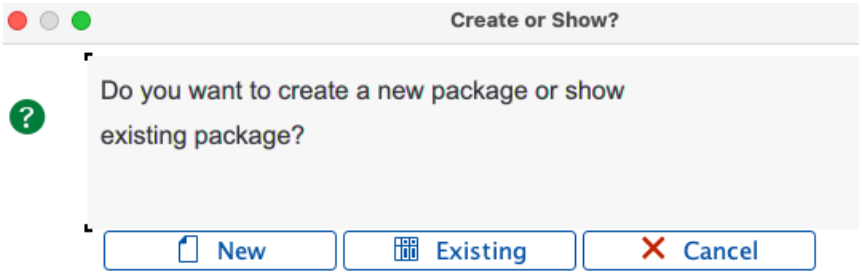


Figure 23: New package creation

Below is the transfer tool cockpit (also known as Conversion Cockpit). It contains the sequence of all activities considering the end-to-end transfer process to BW/4HANA or BW bridge. It can also be accessed via the transaction CNV_MBT_PCT.

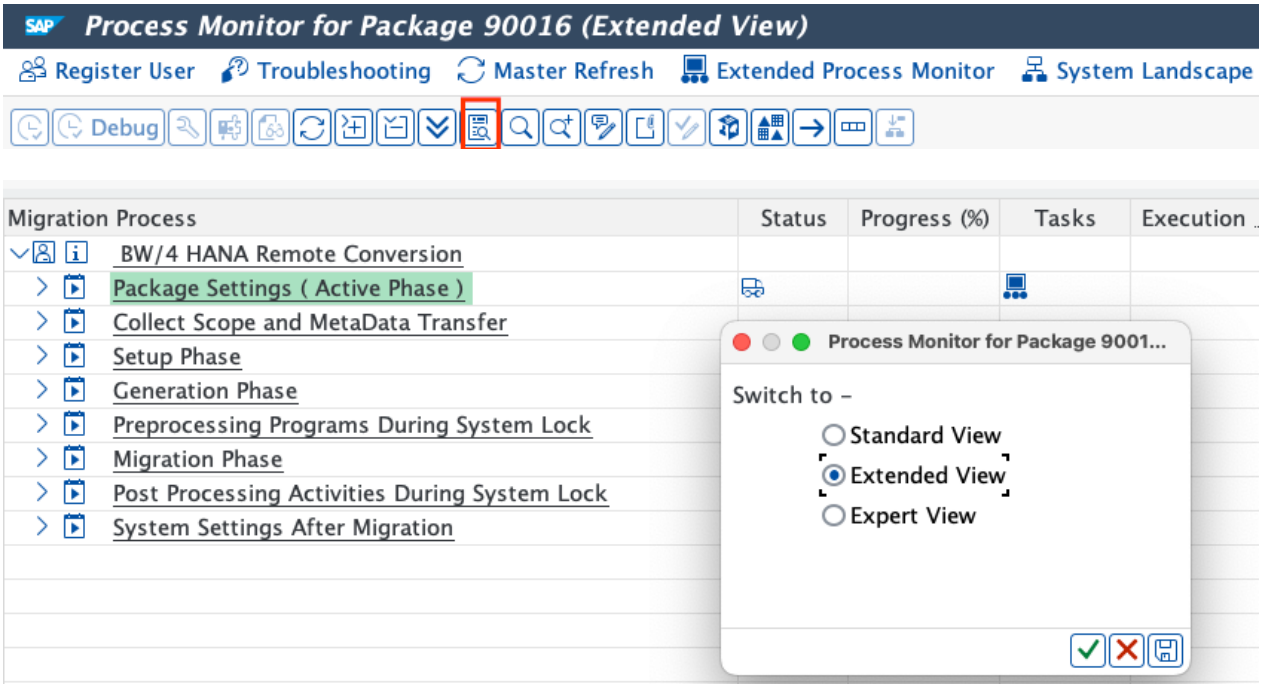


Figure 24: Conversion Cockpit (CNV_MBT_PCT)

There are 3 types of views to perform the transfer execution – Standard, Extended and Expert. We will use the “Extended View” option during this execution. The activity groups (also called phases) are:

1. Package Settings (Active Phase)
2. Collect Scope and MetaData Transfer
3. Setup Phase
4. Generation Phase
5. Preprocessing Programs During System Lock
6. Migration Phase
7. Post Processing Activities During System Lock
8. System Settings After Migration

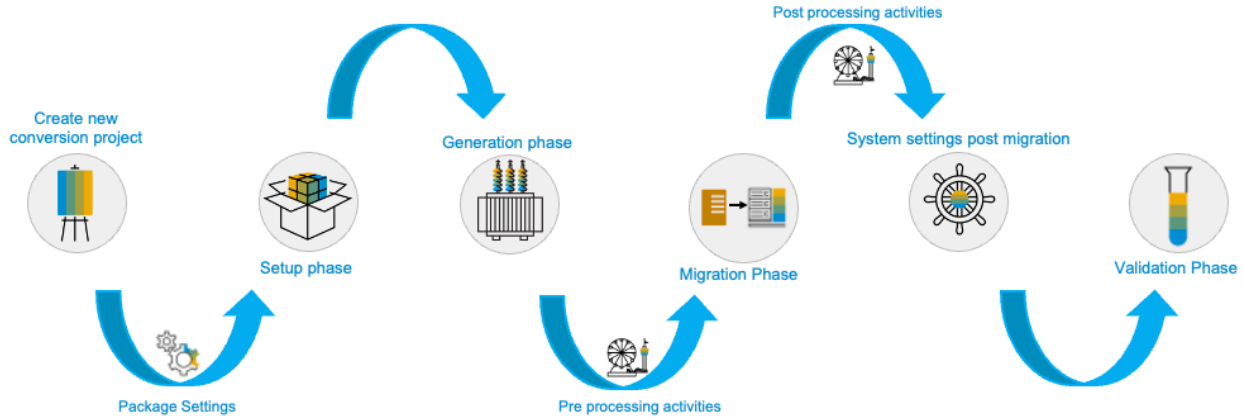


Figure 25: High level sequence of the Remote Conversion activities

In “Display Text” you find the description and detailed information for all activities of the Conversion Cockpit.

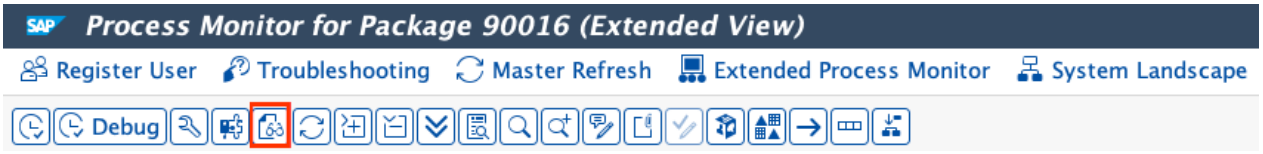


Figure 26: Display text

It is possible to include additional users to your package so that they can also work on /execute activities within your package: go to *“Register User” -> “Go to user” -> “Register”.*

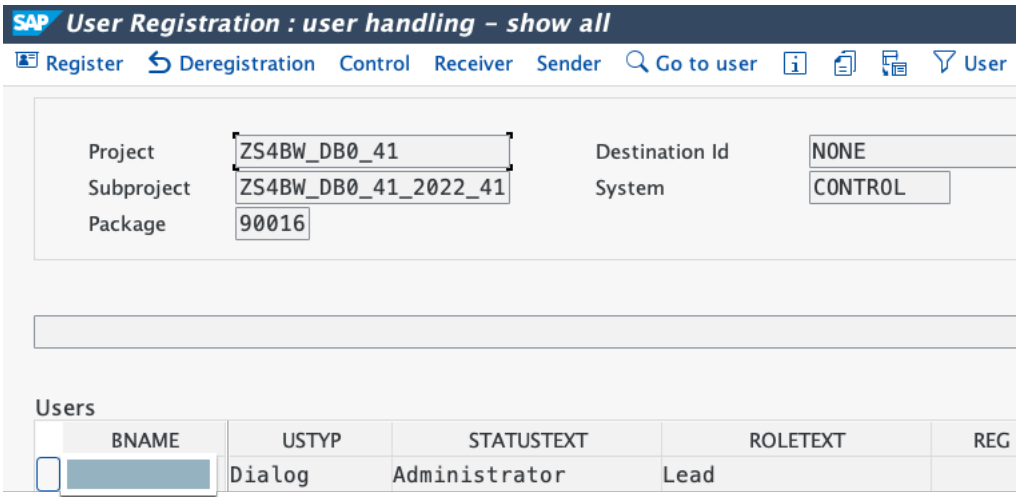


Figure 27: User registration

3.1 Package Settings (Active Phase)

This group of activities represents the initial settings for the Remote Conversion package.

3.1.1 Load Initial Package Settings

This activity is automatically set to green once the package is created.

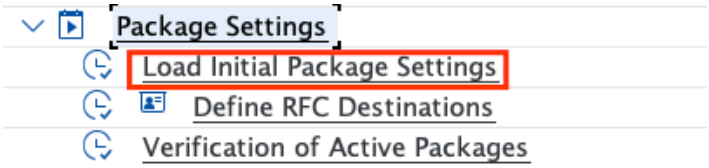


Figure 28: Load Initial Package Settings

3.1.2 Define RFC Destinations

With this activity you define the RFC which will be used for the transfer:

- *Sender system*: “NONE” by default as this is the sender/execution system.
- *Receiver system*: enter the RFC name that you created to your receiver system in SM59.



Figure 29: Conversion Cockpit – RFC maintenance

- Save it and return to the Transfer cockpit

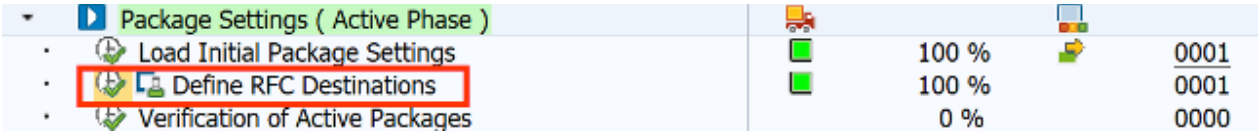


Figure 30: Define RFC Destinations

3.1.3 Verification of Active Packages

This activity verifies that only the current package is active in the system landscape. In case there are any other active packages, this activity stops and does not make the settings for the current package.

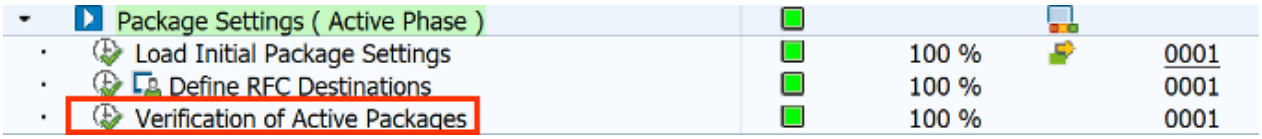


Figure 31: Verification of Active Packages

⇒ Tip: In case you are not able to activate your package because of an existing active package, there are 2 options:

a. Execute the troubleshooting associated with the activity “Verification of Active Packages”.

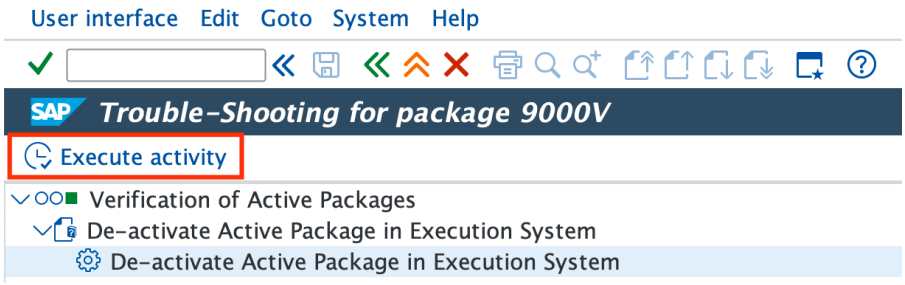


Figure 32: Troubleshooting for package deactivation

b. Or deactivate the already existing active package via the transaction CNV_MBT_SLOP.

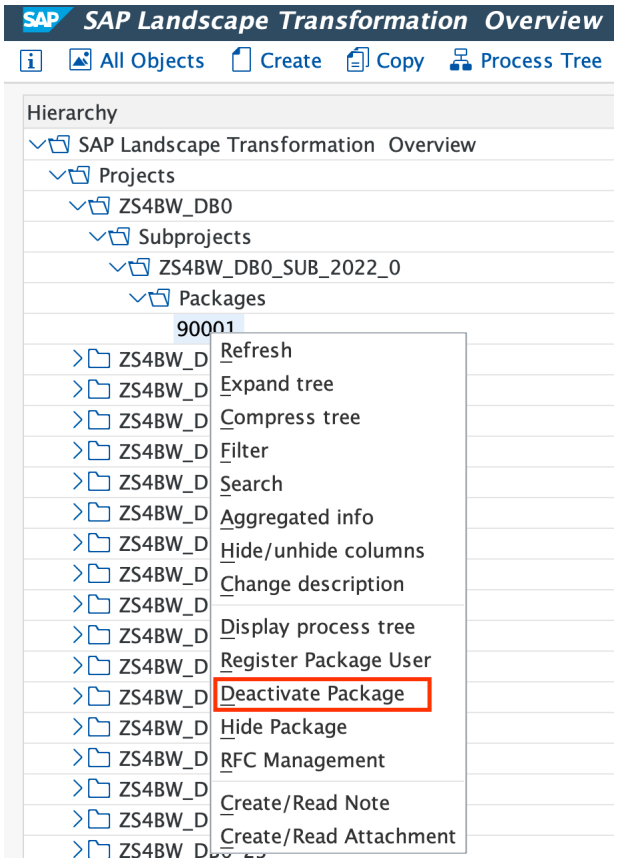


Figure 33: Transaction CNV_MBT_SLOP

Caution if older packages do not overlap with the collection tasks (objects in common), otherwise the new package will complain that there are open Task List runs.

3.2 Collect Scope and MetaData Transfer (Active Phase)

In order to start a transfer of objects into the object types which are supported by BW/4HANA and BW bridge, you have to collect all objects which have a certain association to the object(s) that you want to transfer. As a prerequisite, all collected objects should be active in the sender system.

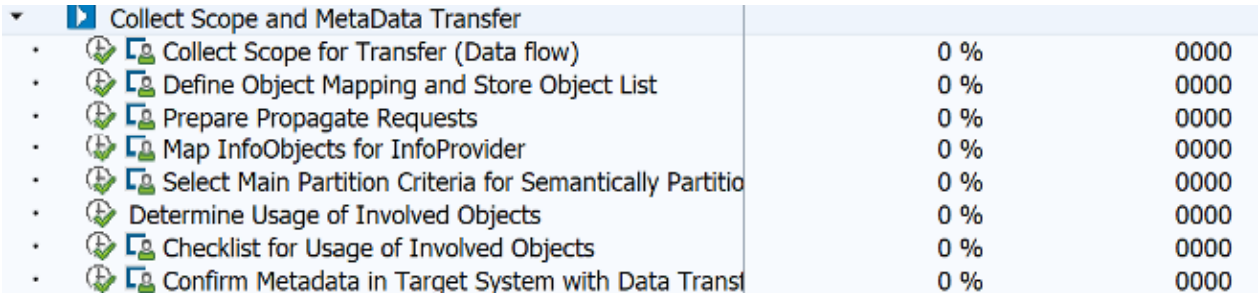


Figure 34: Collect Scope and MetaData Transfer

3.2.1 Collect Scope for Transfer (Data flow)

This activity is the same that was previously executed in the Preparation section. The difference is that now it is associated with a “real” Task List type (SAP_BW4_TRANSFER_CLOUD_REMOTE).

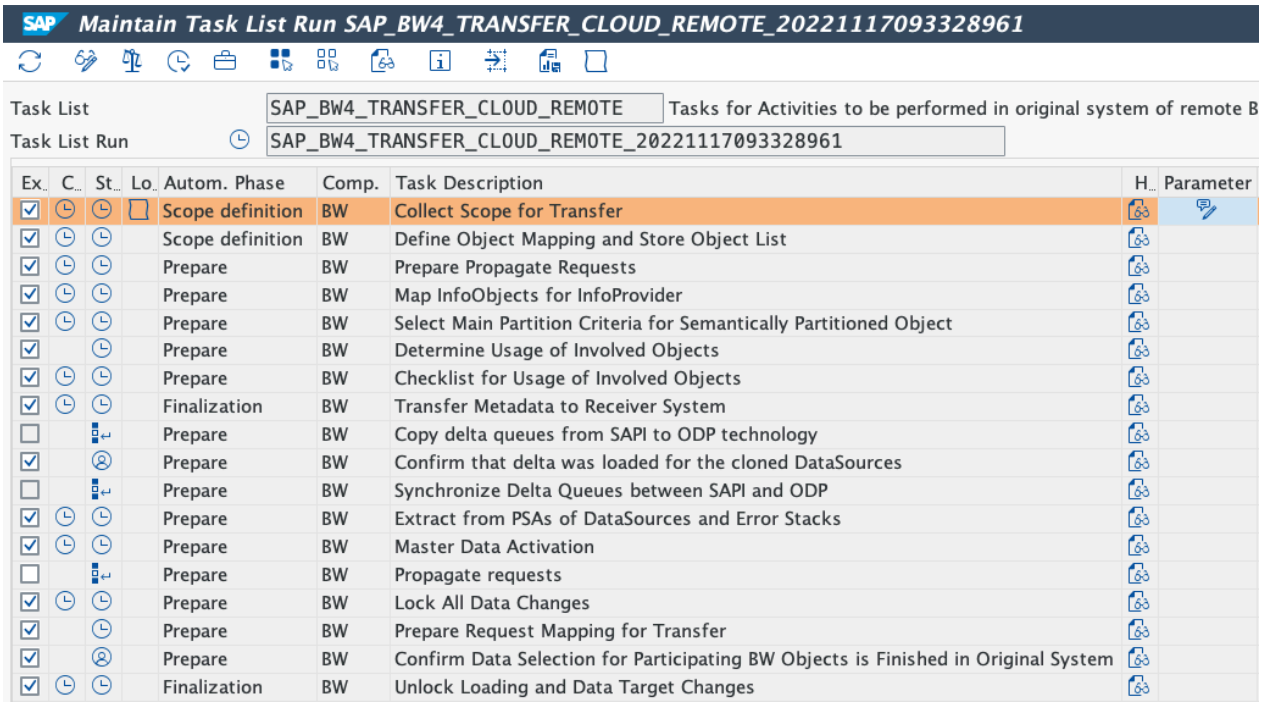


Figure 35: "Real" Task List type (SAP_BW4_TRANSFER_CLOUD_REMOTE)

This activity is a prerequisite for all other tasks of the transfer tool because it contains the objects collection (scope collection) of the transfer. You can maintain the scope and select or deselect objects for transfer in the activity "Define Object Mapping and Store Object List" only after the scope collection has finished.

This activity will take you to the corresponding task in the Task List monitor (STC02). Which has the same name “Collect Scope for Transfer (Data flow)”. You have the option to either use an existing Task List Run or to create a new one.

- Maintain the user interface of this activity and select the objects that you want to transfer. These objects are called "start objects" to the scope. The user interface allows you to search for your objects using the standard BW workbench search.

- You can add objects via the mentioned search or individually. InfoCube DPRODCOST was used as the start object as this is the top object in the sample data flow.

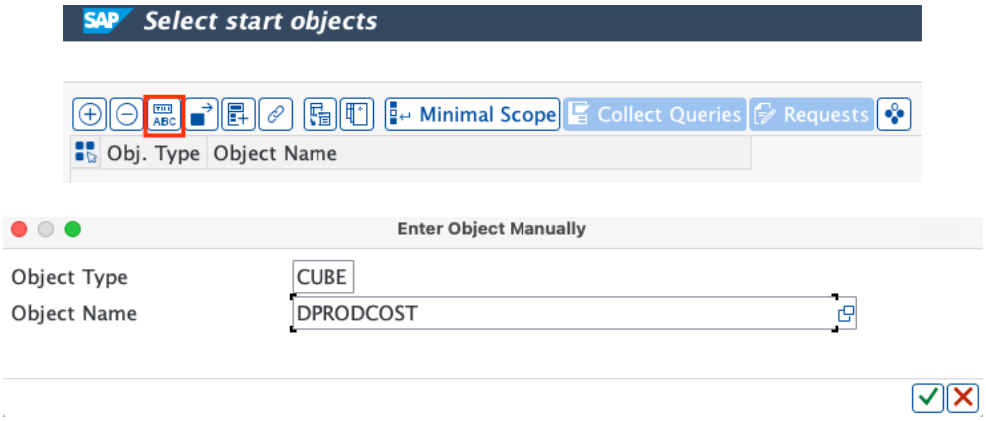


Figure 36: Collect Scope for Transfer - Enter Objects Manually

- You can also choose whether the collection should be a "minimal scope" or not. This option allows to separate the transfer of the reporting layer from the transfer of the staging layer. For example, transfer InfoCubes and DataStore objects without selecting the associated MultiProvider or queries. We will not use it for our sample as it does not have any queries or MultiProviders.

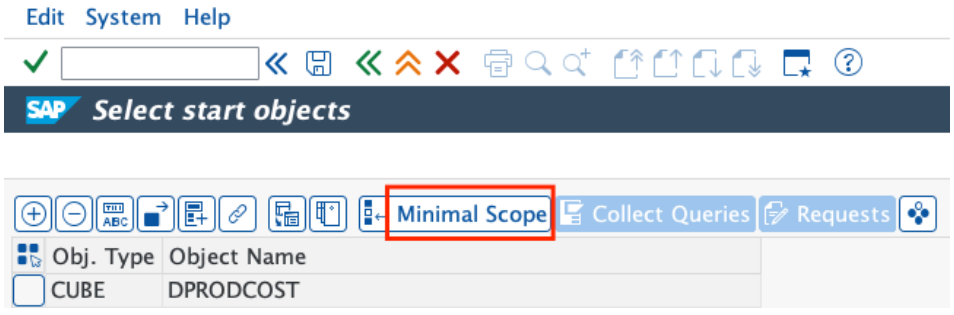


Figure 37: Collect Scope for Transfer - Minimal Scope

- Save it and execute it.

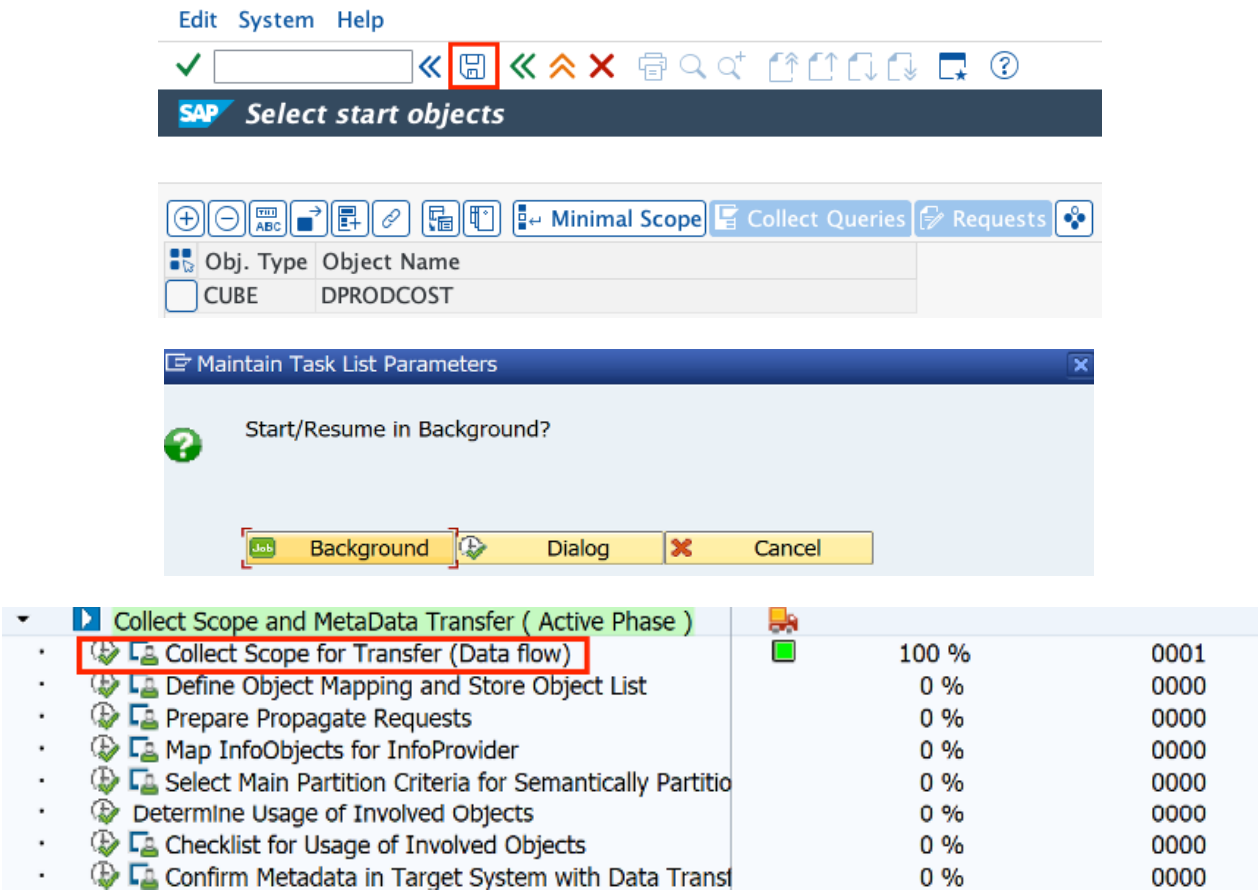


Figure 38: Collect Scope for Transfer - After execution

The above figure is from the Conversion Cockpit, but please note that these steps refer to and are identical to the steps of the Task List SAP_BW4_TRANSFER_CLOUD_REMOTE associated to the package created. The respective logs will be in sync and replicated accordingly.

If this task fails during the collection phase, you can try to re-execute it. If you get unspecific error messages like "Task failed", it means that an object could not be collected and most likely there is a short dump (check ST22). In all other cases where there is a specific error message, act like advised in the error message (for example, there is an inactive object which should be activated otherwise it can't be collected).

With most error and warning messages, a detail button is provided in the log. If the message is about a specific object, the button will lead you to a dependency graphic, which shows you why this object has been collected. This information is especially useful if the scope becomes surprisingly big and might help you to delete some objects or remove some unwanted dependencies before starting the collection again.

The main control interface and central point of access of the Remote Conversion end-to-end execution is the Conversion Cockpit (transaction CNV_MBT_PCT). Differently from the Shell Conversion, which process is fully controlled by the Task List (transaction STC02). The

integration of the Task List with the Conversion Cockpit ensures that there is a direct link between the metadata collection/transfer and the data collection/ transfer in a subsequent activity.

You can start creating your package and follow the sequence of activities indicated in the tree. At a certain point you will be automatically directed to the Task List. This will initially happen when you execute “Collect Scope for Transfer (Data flow)” – and you will have the option to either create a new task list or use an existing one. This task list will be assigned to your package.

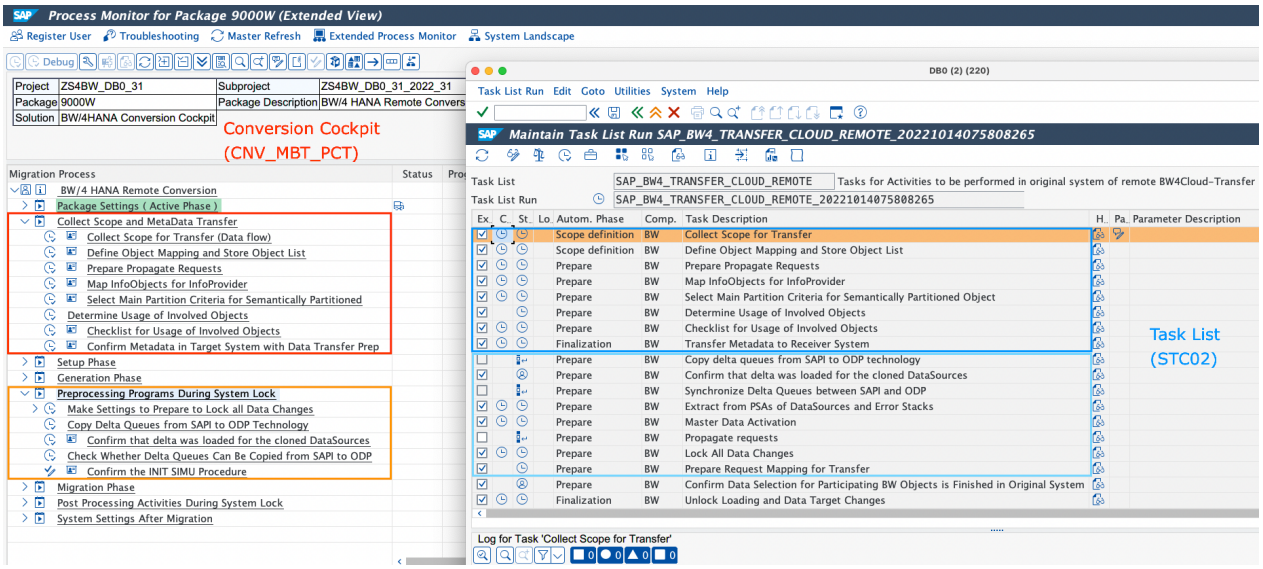


Figure 39: The group of activities under "Collect Scope and MetaData Transfer" and “Preprocessing Programs During System Lock” are executed within the Task List, through the Conversion Cockpit

⇒ Tips:

- When selecting start objects, the higher the object the less will be the effort to manually identify and include associated objects. For example, starting with a MultiProvider, all associated objects below will be automatically collected (including Process Chains).

- Queries can also be collected (by selecting “Collect Queries”) when the receiver system is BW/4HANA.

- When the receiver system is on BW/4HANA, another possibility to transfer the queries is to leave them out of the Remote Conversion scope collection and use a subsequent Shell Conversion Task List run only for query objects.

- When the receiver system is BW bridge, it is possible to import query objects to the BW bridge Dev Tenant however such objects are not functional (not accessible nor executable). Once they are imported to the BW bridge Dev Tenant, you need to migrate/adjust those query objects into Datasphere artifacts so that they can be used for reporting purposes. Queries built directly on top of InfoProviders with navigation attributes can’t be transferred. You may use the

report RS_B4HANA_RC_CHECK_QUERY to check whether a query is transfer-supported or not.

3.2.2 Define Object Mapping and Store Object List

This activity is the same that was executed in the Preparation section. The difference is that now it is associated with a “real” Task List.

With this task you can check which objects were collected and are to be transferred, based on the start object(s) defined in “Collect Scope for Transfer (Data flow)”. You can use the Hide/Show objects buttons to display the associated objects which were collected. You can also manually create filters for display.

Based on the start object DPRODCOST, the following objects were automatically collected as they have associations with the same data flow.

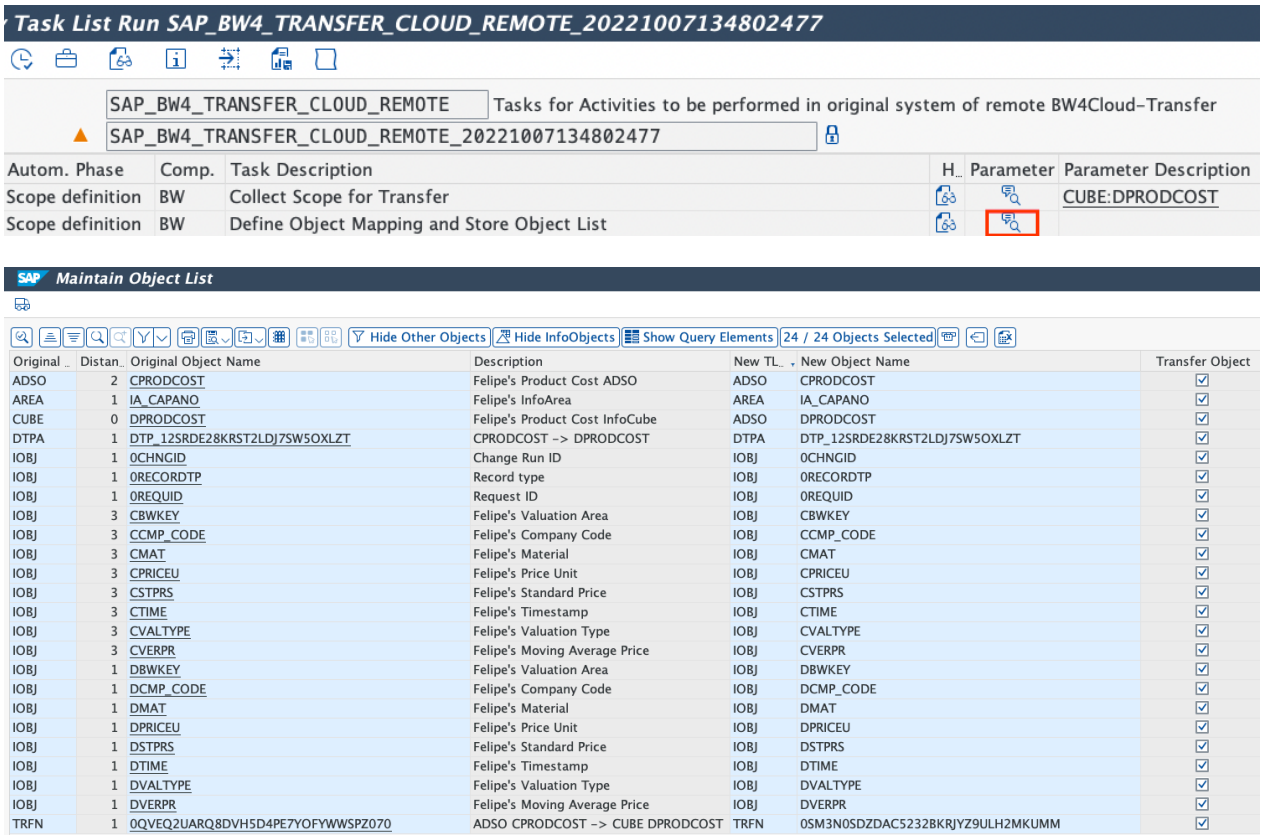


Figure 40: Define Object Mapping and Store Object List – Collected objects

As a prerequisite, you have created a BW bridge project, Software Component, ABAP Package and Transport Request in the BW bridge receiver system. Check Create a SAP BW Bridge Project and Create Software Components for further instructions.

⇒ Note: When the receiver system is on BW/4HANA (on-premise Remote Conversion), the Package and Transport in this case are created in the BW sender system instead (a transportable package and transport request(s) created by the tool).

- Select the ABAP Package and Transport Request and save it.

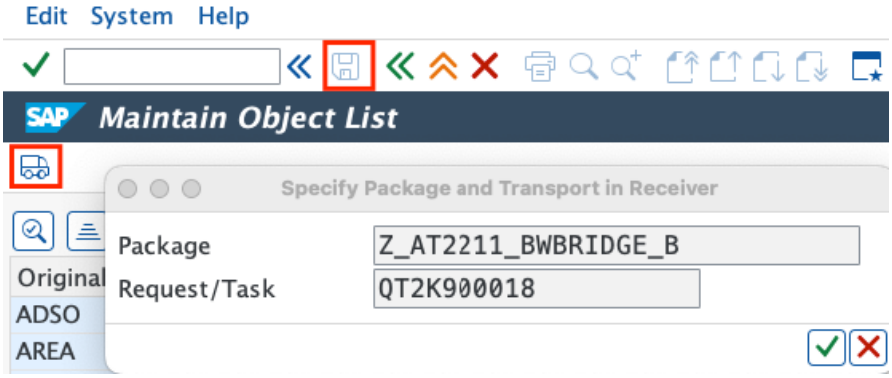


Figure 41: Define Object Mapping and Store Object List - Package and Transport assignment

- You will be returned to the Task List and can execute it.

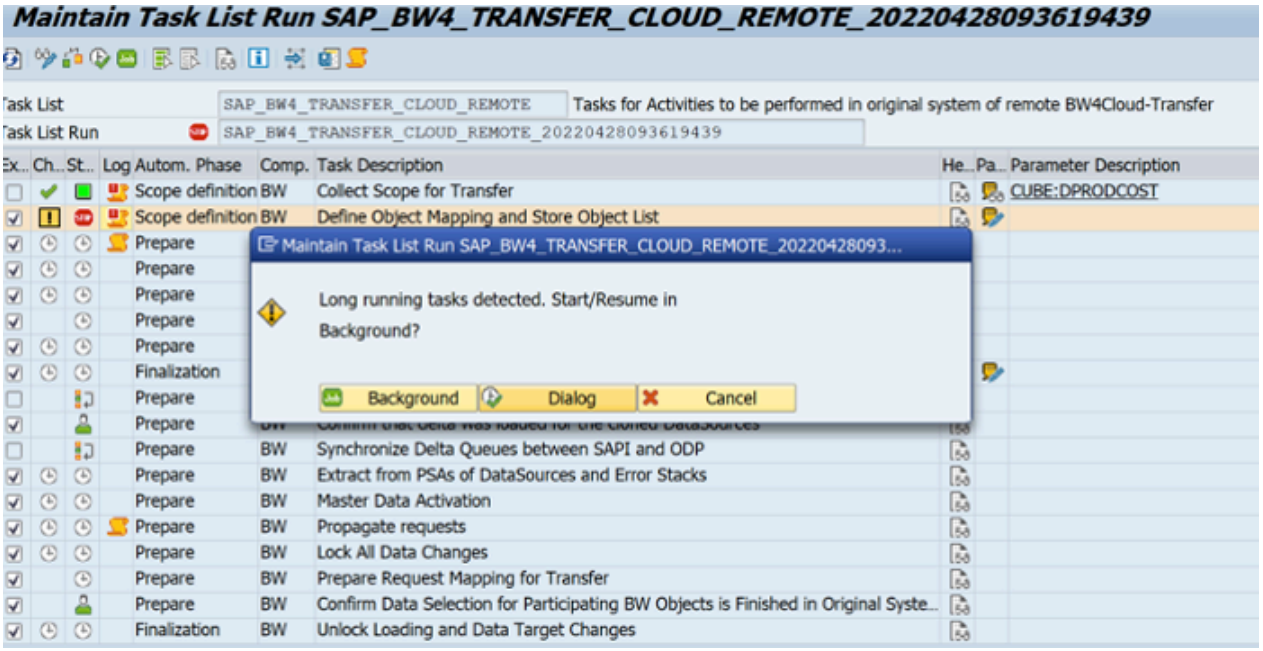


Figure 42: Define Object Mapping and Store Object List - Execution

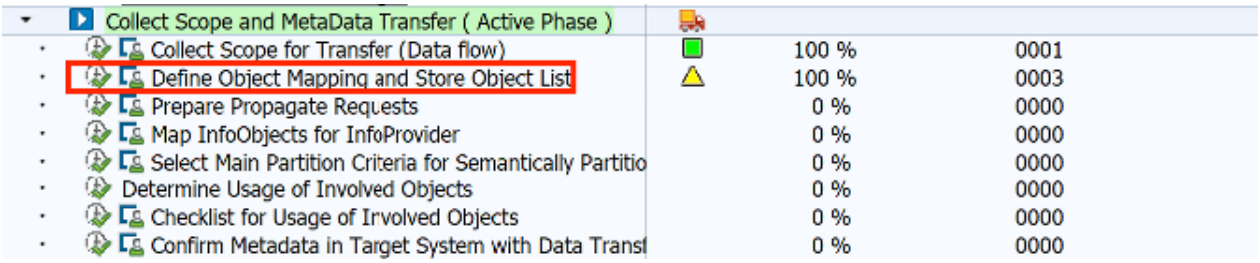


Figure 43: Define Object Mapping and Store Object List - After execution

⇒ Tip: With this activity, objects in scope are collected, mapped and transferred to the receiver system. The runtime/duration is directly proportional to the scope size. Review the scope and include what is necessary to be migrated. Applying data selectivity can also be an alternative, by carving-out objects that you don't need to be transferred.

3.2.3 Prepare Propagate Requests

The purpose of this activity is to automate the request propagation. For more details. see SAP Note 2982341 - Optimized Request Transfer and Automatic Propagation.

By clicking on the 'Change Parameters' button, all relevant DTPs will be listed in the UI. Default values are set for each of the DTPs (in maintenance mode only) and dependent information (e.g. Last Request) is derived. In our sample, there are no requests to be propagated therefore the parameter maintenance is not applied.

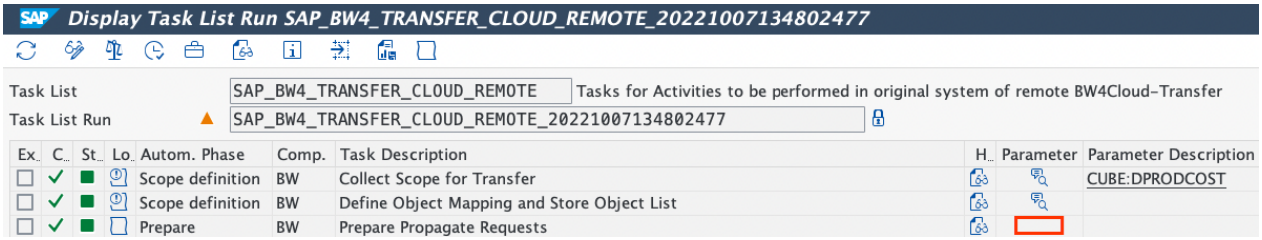


Figure 44: Prepare Propagate Requests

When applied, the following selections are then offered:

- *Confirm Execution*: Confirmation of the DTP for possible execution in the 'Propagate Requests' task. This is valid for delta DTPs. Full DTPs are ignored by this task.

- *Insert into Scope*: This flag can only be set if a DTP was identified which is not part of the scope collection, but the corresponding source of the DTP is part of the scope collection. If non-propagated requests exist, the DTP must be executed. By setting this flag, you must return to the scope collection and add the relevant DTP to the Scope. If you do not want to add the DTP to the scope, it must be logically deleted.

- *Logically Delete DTP*: This option provides the option of logically deleting the DTP directly from the UI. Please see SAP Note 1450242-P24: Removing DTP from delta admin.: Logical deletion for further details. After delta invalidation, the DTP will be ignored in the automatic request propagation and set to 'Skip' in the Scope Definition.

If a DTP is set to 'Skip' in the scope collection, but the corresponding source of the DTP is part of the scope collection, the DTP must be executed (if non-propagated requests exist). In this situation, the warning message 'DTP DTP_xxx has to be included in propagation' will appear in the UI. If you do not want to execute the DTP, it must be logically deleted.

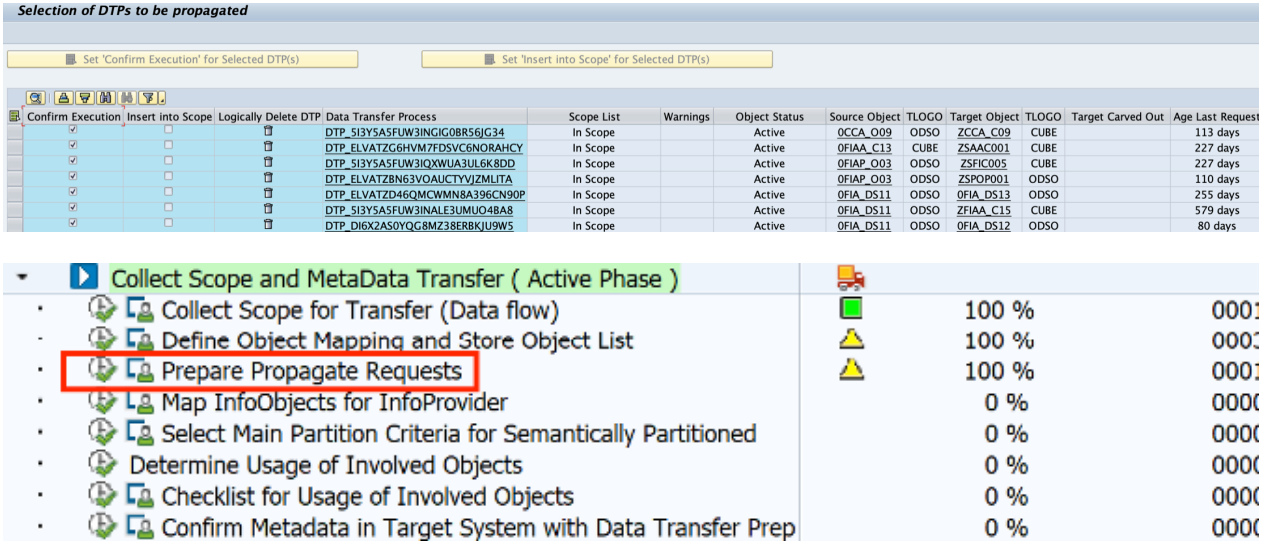


Figure 45: Prepare Propagate Requests - After execution

⇒ Tips:

- Logically deleted DTPs (report RSBKDTPDELETE) are neglected by 'Collect Scope' and their requests are transferred as non-DTP Request (managed by the tool). In general, the logical deletion 'cuts off' the delta mechanism so once a delta DTP is logically deleted, the delta pointer is invalidated. Hence any new delta DTP would then start an initialization after the first execution. Have in mind that you can have only one active delta DTP connected against the same target.

- The deletion is no issue for the request mapping nor for the 'Propagate Requests' task.

- In case you have inactive DTPs, which are not used anymore and identified to be collected for the transfer, a solution can be to activate such DTPs and afterwards logically delete them – before 'Collect Scope'.

- Change the delta DTP execution to 'init without data transfer' will not work because the 'Propagate Requests' task always executes DTPs with data.

3.2.4 Map InfoObjects for InfoProvider

This activity provides the option of performing the required InfoObjects mapping for the transfer of a MultiProvider or InfoSet to a CompositeProvider. This mapping is necessary, as the CompositeProvider requires a consistent mapping of navigation attributes. A navigation attribute therefore always requires a mapping, which matches the attribute-carrying characteristic.

For MultiProviders, the “Compatibility Mode” is automatically used and a proposal is generated. For InfoSets, it might be needed to create new InfoObjects to replace the navigation attributes.

⇒ Note: Not applied to the scope of our sample.

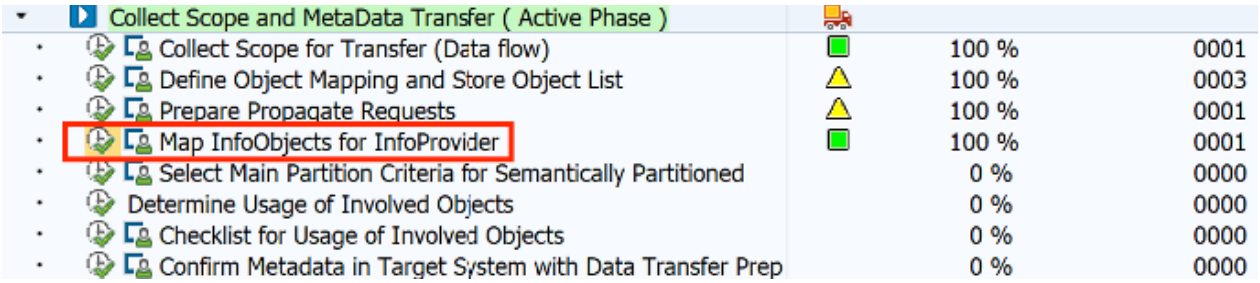


Figure 46: Map InfoObjects for InfoProvider

3.2.5 Select Main Partition Criteria for Semantically Partitioned

This activity is relevant in case SPOs (semantic partitioned objects, TLOGO=LPOA) are part of the scope.

It provides the option to select the main partition InfoObject, which should be used for the new DataStore (advanced) in the generated semantic group. As semantic groups support only one Partition InfoObject, this selection is necessary. The other partition InfoObject filters are nevertheless also added to the corresponding DataStore (advanced) as filter criteria.

⇒ Note: Not applied to the scope of our sample.

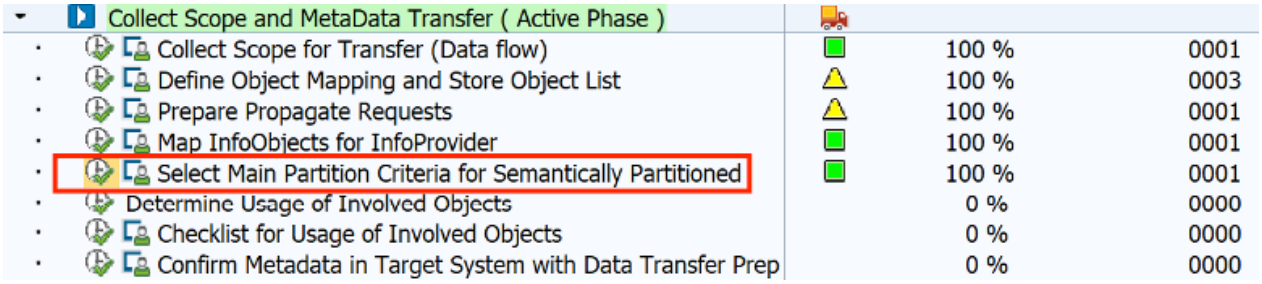


Figure 47: Select Main Partition Criteria for Semantically Partitioned

3.2.6 Determine Usage of Involved Objects

This activity shows a list of objects or findings, which should be checked or resolved. This includes usages of generated objects and of InfoProviders that are not there anymore after the transfer. Customer code also needs to be checked for changes that occur during the transfer, for example where InfoObjects are exchanged, or queries run on a CompositeProvider after the transfer instead of on top of a DataStore object/InfoCube.

Not all findings indicate actual issues. The findings are intended merely as an indication that coding should be revised (can be indicated by the ABAP Code Scan report).

For example, the 0requid information from InfoCubes will not be in the receiver InfoProvider anymore, so referencing 0requid or requid is a potential source of problems.

Findings in commented coding are also shown, as these comments might refer to dynamic coding that should be revised.

⇒ Note: Not applied to the scope of our sample.

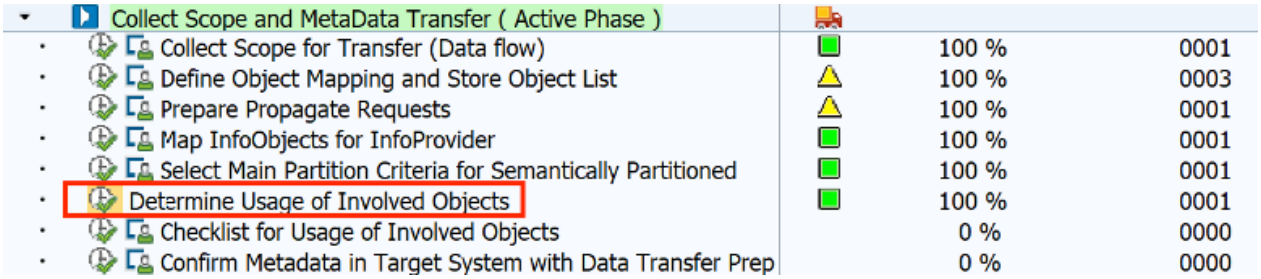


Figure 48: Determine Usage of Involved Objects

3.2.7 Checklist for Usage of Involved Objects

This activity represents the output from the previous one “Determine Usage of Involved Objects”.



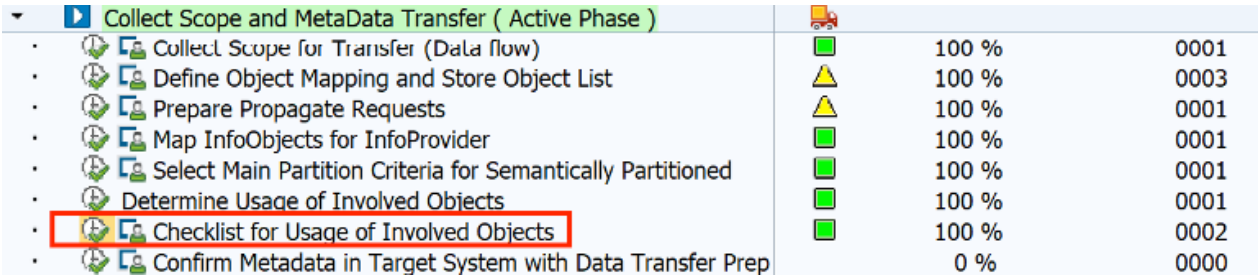


Figure 49: Checklist for Usage of Involved Objects

⇒ Note: Not applied to the scope of our sample.

3.2.8 Confirm Metadata in Target System with Data Transfer Prep

When reaching this activity, this means the metadata was transferred to the BW bridge system and you should check if those objects were successfully activated.

You also have the option to again maintain the parameters (ABAP Package and Transport Request), i.e. the original transport might have been released during the transfer and no longer suitable.

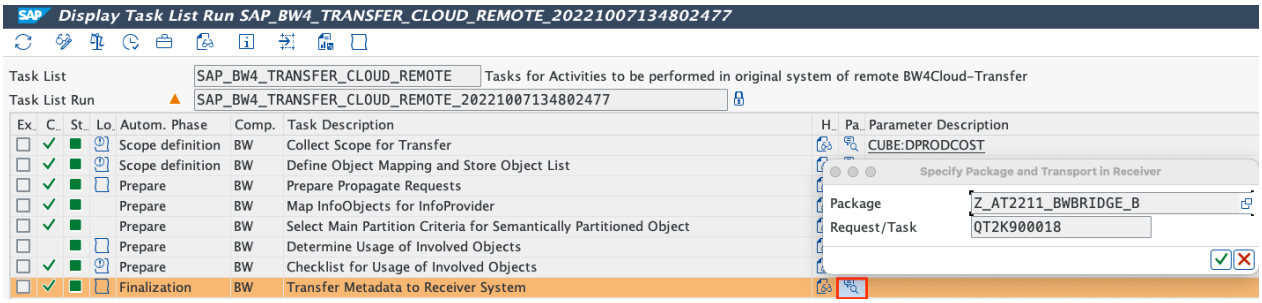


Figure 50: Task List - Transfer Metadata to Receiver System

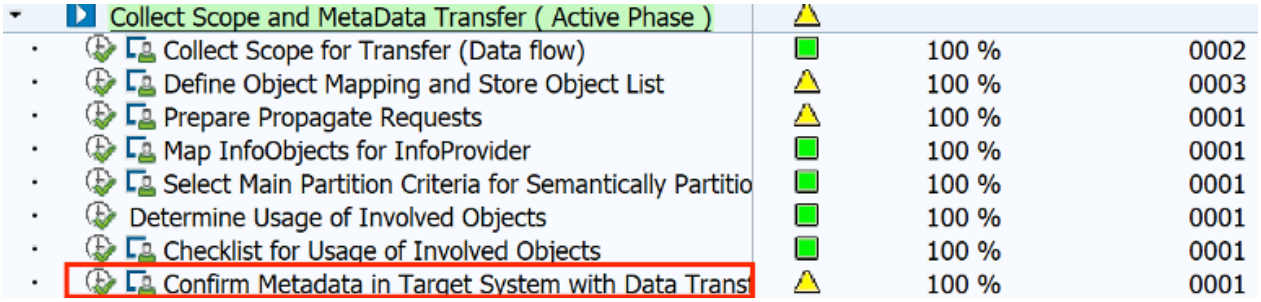


Figure 51: Conversion Cockpit - Confirm Metadata in Target System with Data Transfer Prep

Note that the activity “Confirm Metadata in Target System with Data Transfer Prep” in the Conversion Cockpit corresponds to the task “Transfer Metadata to Receiver System” in the Task List. The objects collected for our sample scope were transferred to BW bridge:

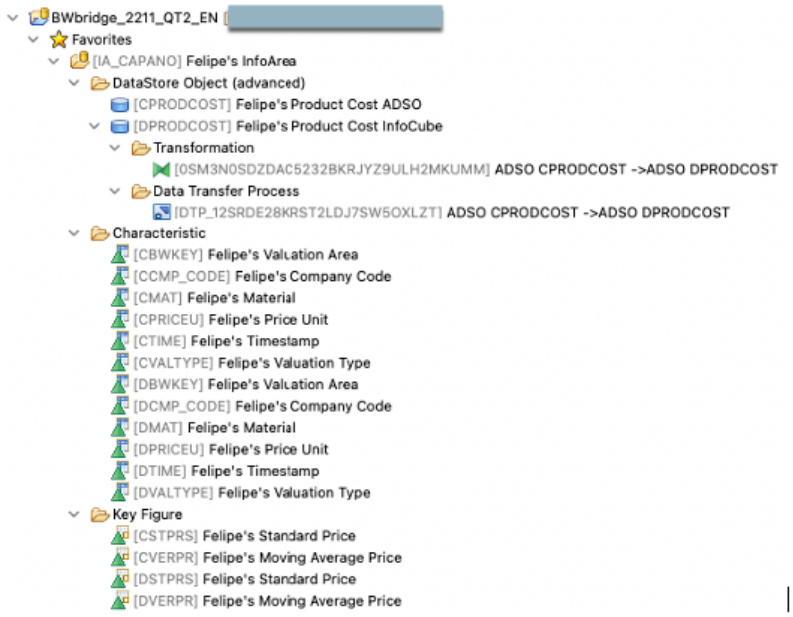


Figure 52: BW bridge project in BWMT

When the receiver system is on BW/4HANA, the metadata in scope is transferred via the standard transport mechanism.

When the receiver system is on BW bridge, there is no transport to bring the metadata into the bridge but a direct RFC. Consequently, the log can be seen as part of the Task List run, not in SE09. The package and transport request are required to assign the objects to the receiver system (because no objects must be brought into Steampunk without package and transport).

⇒ Tips:

- When the receiver system is on BW/4HANA (on-premise Remote Conversion), you can also the check the status of the transported objects via the table RSB4HTOBJECT in the receiver system. When an object is successfully transferred, the METADATA field will be equal to “X”. As this check requires SAP GUI access, it would not be possible when the receiver system is BW bridge.

- For further information about BW bridge limitations, see:
  - SAP Note 3117800 - Information/Restrictions Note for SAP Datasphere, SAP BW Bridge

3.3 Setup Phase

The metadata collection (scope collection) and transfer activities were finished. This phase consists in a group of activities which are relevant for the business data transfer.

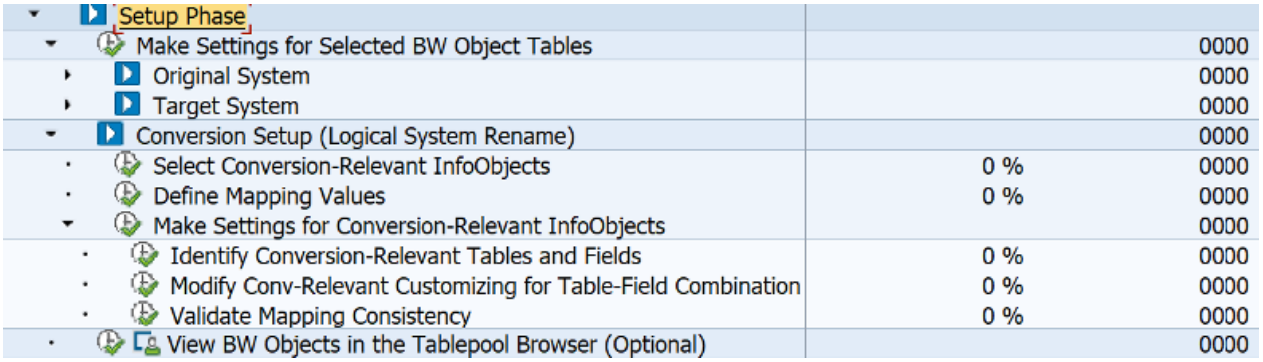


Figure 53: Setup phase

3.3.1 Make Settings for Selected BW Object Tables

This activities group starts a sequence of activities that run in the background. Since each activity automatically starts its successor activity, no further user interaction is required. You also have the option to individually execute each activity through the Extended View.

The following activities are executed in the sender system:

- Read Object Mappings from the BW Transfer Tool
- Identify BW Table Pools from Data Migration
- Collect and Transfer Package Settings Information

The following activities are executed in the receiver system (BW bridge or BW/4HANA):

- Identify Semantic Differences between InfoCubes and ADSOs
- Identify InfoCubes for Migration
- Generate Dimension and Fact Tables for InfoCubes
- Build Table Mapping between Mapped Objects
- Display DDIC Differences Between Mapped Objects
- Configure MIGSID Conversion Rule for Master Data ID Fields
- Make Settings to Facilitate Program Generation

⇒ Tip: If you are using the Standard View and the activities group terminates or ends with errors or warnings, activate the Extended View of the process tree. You can now view all activities that belong to the group activity and their status.

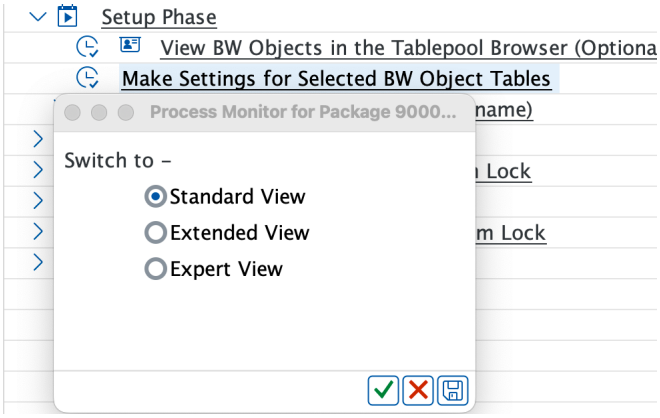


Figure 54: Switch views

3.3.1.1 Original System: Read Object Mappings

This activity reads object mappings (old object name/type -> new object name/type) and stores the data in the CNVLTRLBW4OBJMAP control table in preparation for data migration.

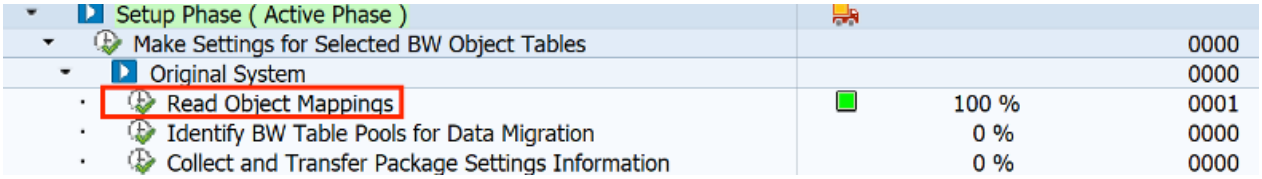


Figure 55: Read Object Mappings

3.3.1.2 Original System: Identify BW Table Pools for Data Migration

This activity identifies the list of transfer relevant BW objects (so called table pool) from the sender system to the receiver system.

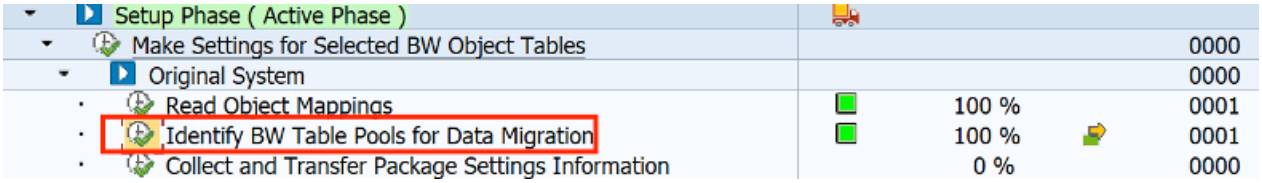


Figure 56: Identify BW Table Pools for Data Migration

3.3.1.3 Original System: Collect and Transfer Package Settings Information

This activity collects and transfers package settings information from the sender) system to the receiver system.

It determines the DDIC information of table pools identified by the predecessor activity “Identify BW Table Pools from Data Migration”. The DDIC information is stored in the control table CNVLTRL_DDIC in the sender system.

In addition, the activity also sends information about BW objects to the receiver system with regards to the following tables:

- CNVLTRL_OBJTAB - Contains a list of tables to be transferred from the sender system to the receiver system.
- CNVLTRL_DDIC - Contains technical details of the tables in the table pool so that semantic checks can be performed. This information is also used to generate the programs in the receiver system.
- CNVLTRLBW4OBJMAP - Contains mappings between the BW objects of the sender and receiver systems.
- CNVLTRL_BW4_CUBE - Contains information about participating InfoCubes so that the required InfoCube-related tables can be generated in the receiver system.

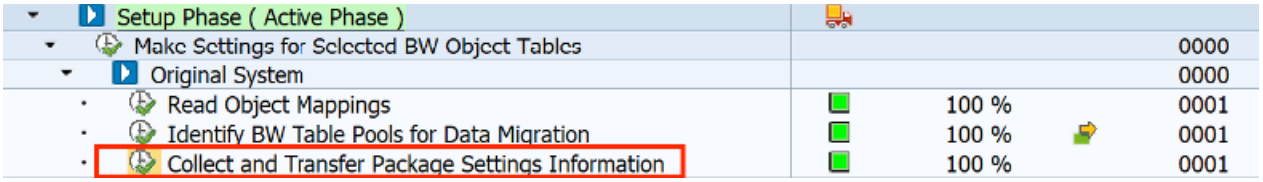


Figure 57: Collect and Transfer Package Settings Information

3.3.1.4 Target System: Identify Semantic Differences between InfoCubes and ADSOs

This activity identifies structural and sequential differences between InfoCubes and mapped ADSOs.

a. If an InfoObject/field present in the InfoCube in the sender system is missing in the mapped ADSO in the receiver system, an error log is generated (red status).

b. If an InfoObject/field present in the ADSO in the receiver system is missing in the mapped InfoCube in the sender system, a warning log is generated (yellow status). For example: An InfoCube in scope was modified (e.g. a characteristic or key figure removed) while the Remote Conversion package is running. If that happens after the scope collection and metadata transfer have finished, this difference will be identified and displayed by this activity.

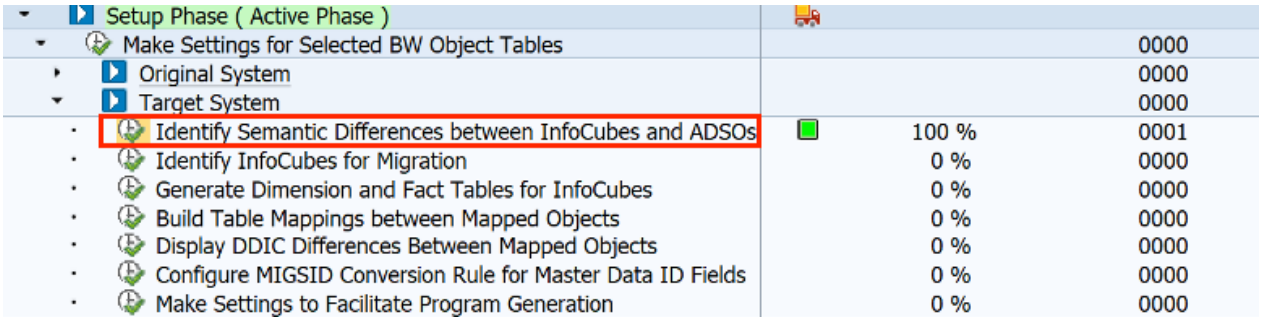


Figure 58: Identify Semantic Differences between InfoCubes and ADSOs

Neither a nor b were applied to our sample.

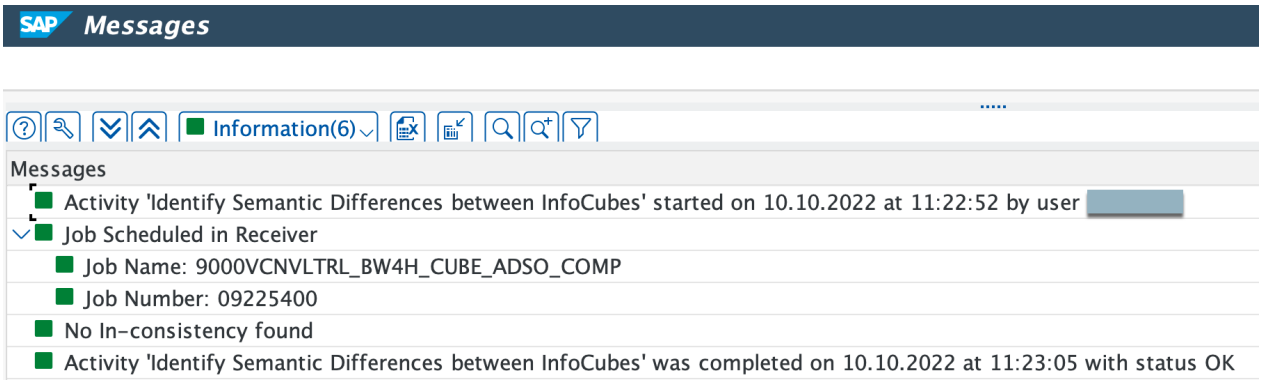


Figure 59: Identify Semantic Differences between InfoCubes and ADSOs - Logs

3.3.1.5 Target System: Identify InfoCubes for Migration

This activity identifies the relevant InfoCubes in the sender system that must be converted into ADSO objects in the receiver system.

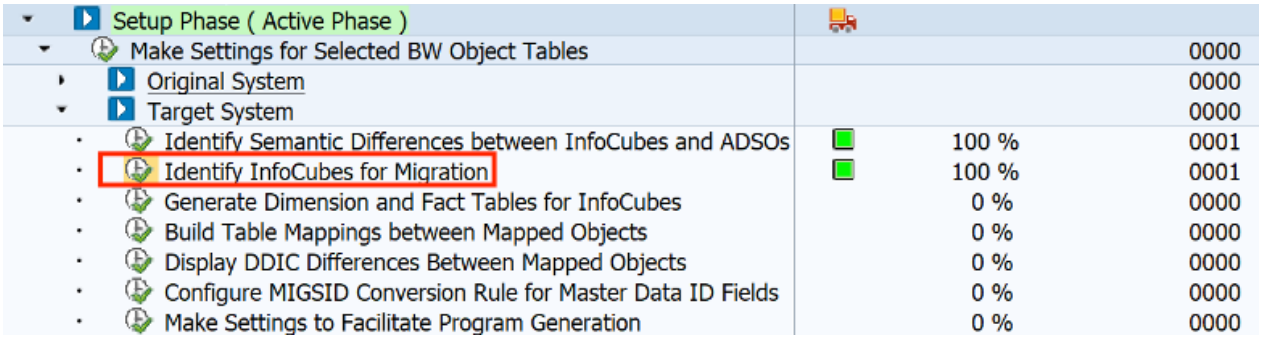


Figure 60: Identify InfoCubes for Migration

DPRODCOST is the InfoCube in our sample.

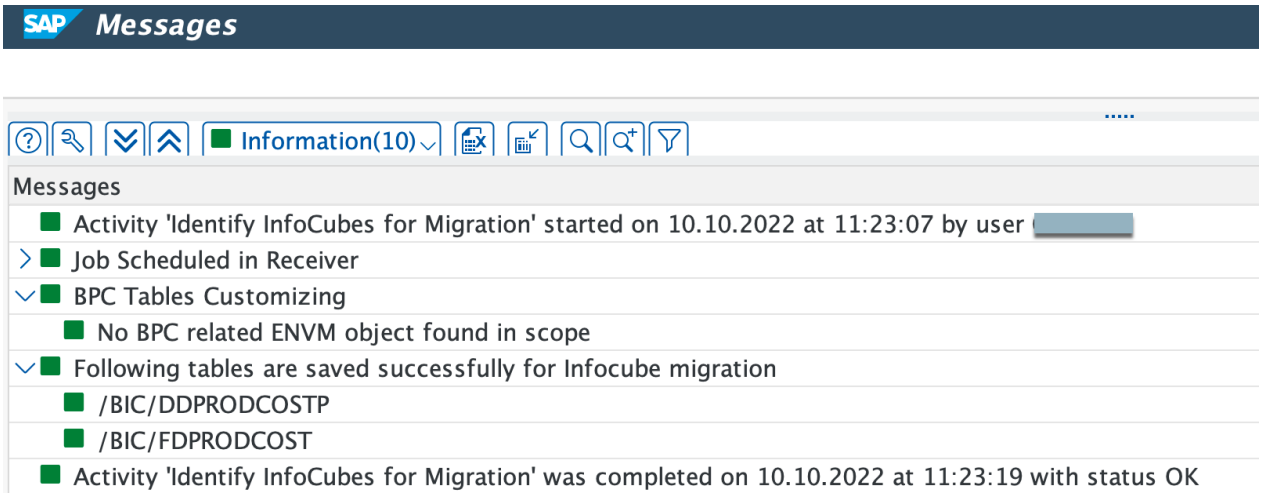


Figure 61: Identify InfoCubes for Migration - Logs

3.3.1.6 Target System: Generate Dimension and Fact Tables for InfoCubes

This activity generates the dimension and fact tables in the receiver system for the list of InfoCubes identified in the predecessor activity. This activity is not relevant in case the sender system is on BW/4HANA.

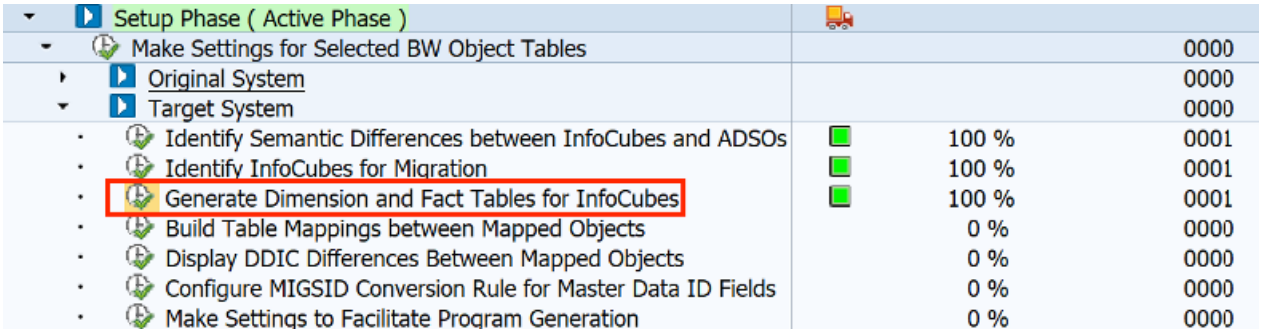


Figure 62: Generate Dimension and Fact Tables for InfoCubes

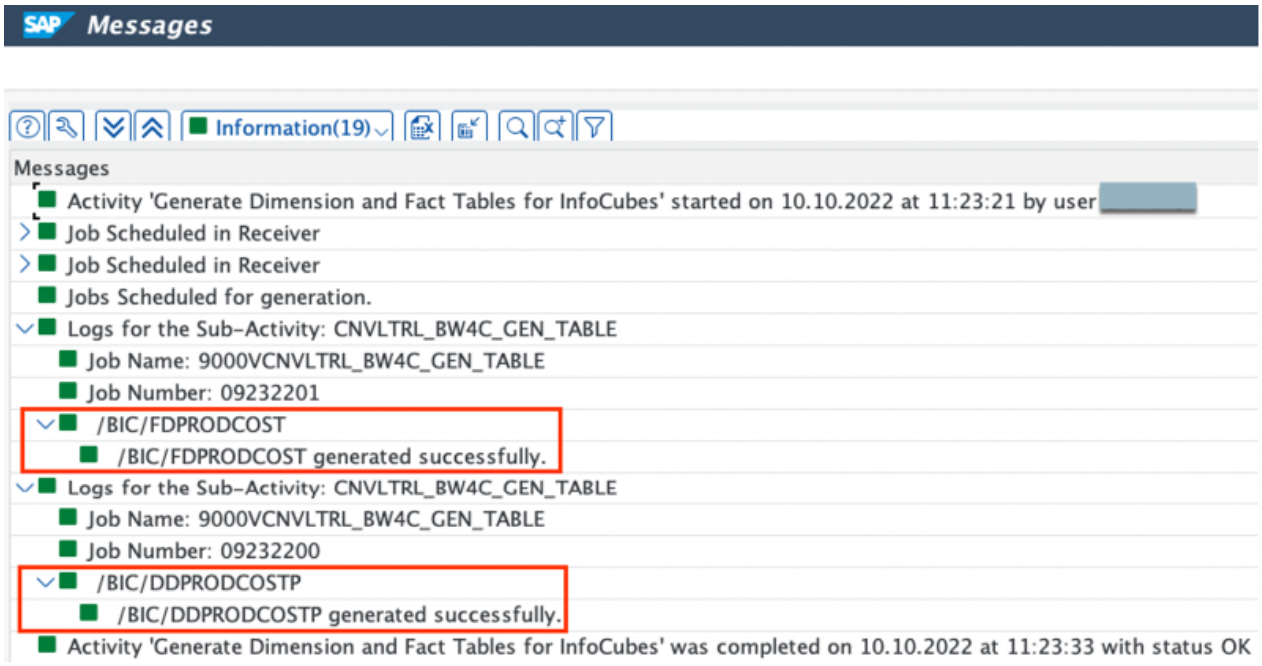


Figure 63: Generate Dimension and Fact Tables for InfoCubes - Logs

3.3.1.7 Target System: Build Table Mappings between Mapped Objects

This activity maps BW objects which were selected for transfer in the sender system to corresponding BW objects in the receiver system.

Here are some examples of such mappings:

- ODSOs to ADSOs
- InfoCubes to ADSOs
- ADSOs to ADSOs
- PSAs to ADSOs
- Open Hub Destinations to Open Hub Destinations

This activity carries out the following checks:

- If the receiver object tables exist in the receiver system.
- If the mapped tables in the sender and receiver systems have any DDIC differences.

If the mappings and checks are successful, the activity saves the results in the CNVLTRL_TABMAP control table (required for programs generation, activity which will be executed at a later stage).

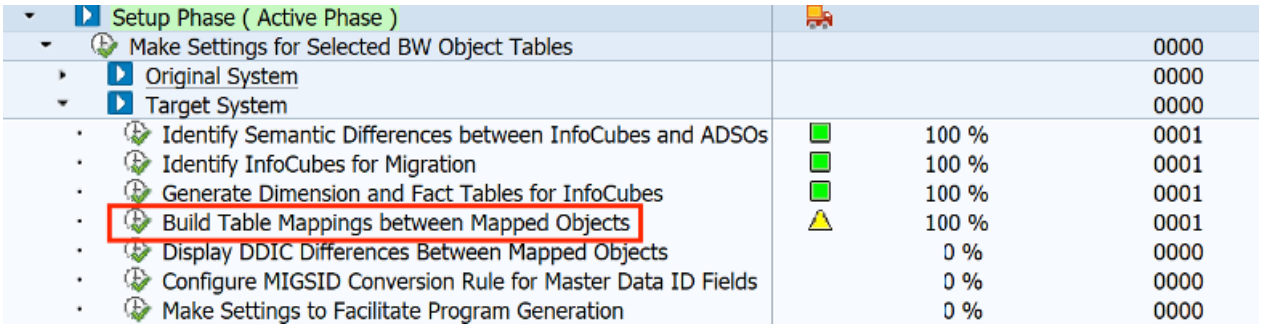


Figure 64: Build Table Mappings between Mapped Objects

The warning happened for our sample because the standard receiver tables RSB4H_ODSACTREQ and RSB4H_ICCONT contain an additional field (REDUCED) in the receiver system. This is not an issue/showstopper and the transfer process may proceed.

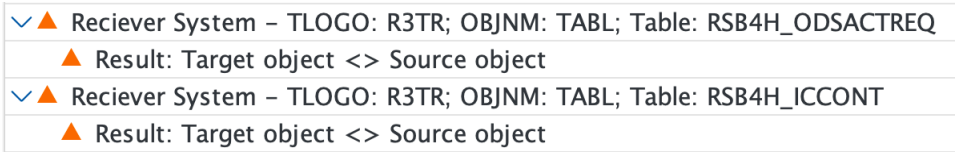


Figure 65: Build Table Mappings between Mapped Objects – Logs

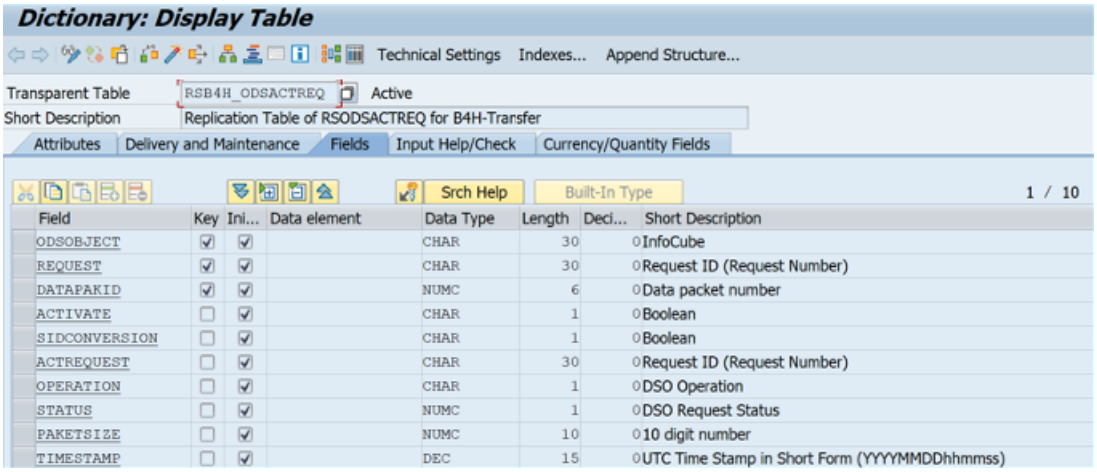


Figure 66: Table RSB4H_ODSACTREQ in the sender system

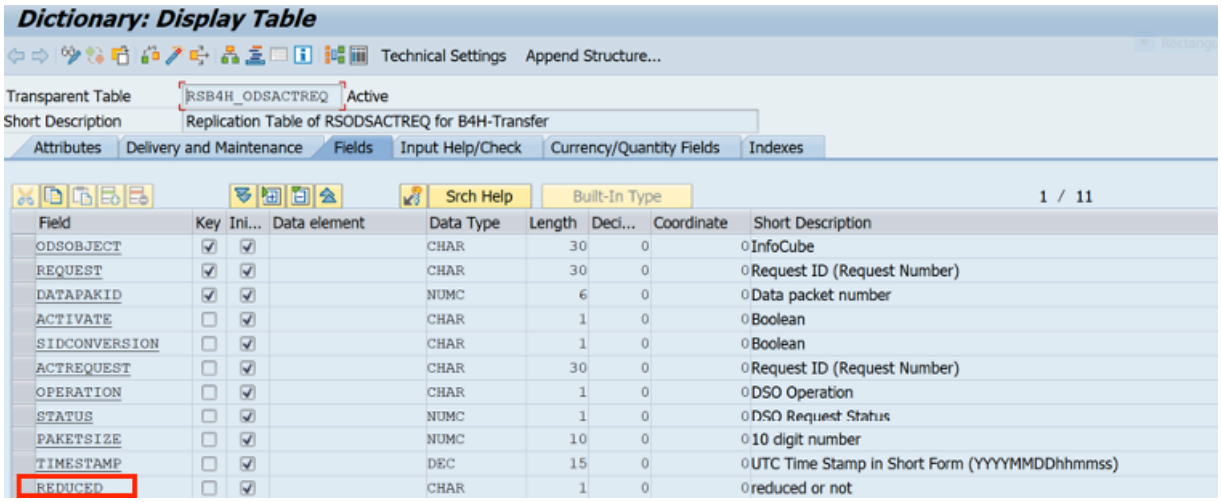


Figure 67: Table RSB4H_ODSACTREQ in the receiver system

3.3.1.8 Target System: Display DDIC Differences Between Mapped Objects

This activity compares the table and field level information for the mapped objects in both sender and receiver systems. If there are any DDIC differences in the mapped objects, the activity generates error logs.

This activity carries out the following checks:

- If any field of a table in the receiver system is missing in the sender system, the activity generates a warning of yellow severity.
- If a field of a table in the sender system is missing in the receiver system, the activity generates an error of red severity.
- If the mapped tables have the same field name but different data types or lengths, the activity generates an error of red severity.

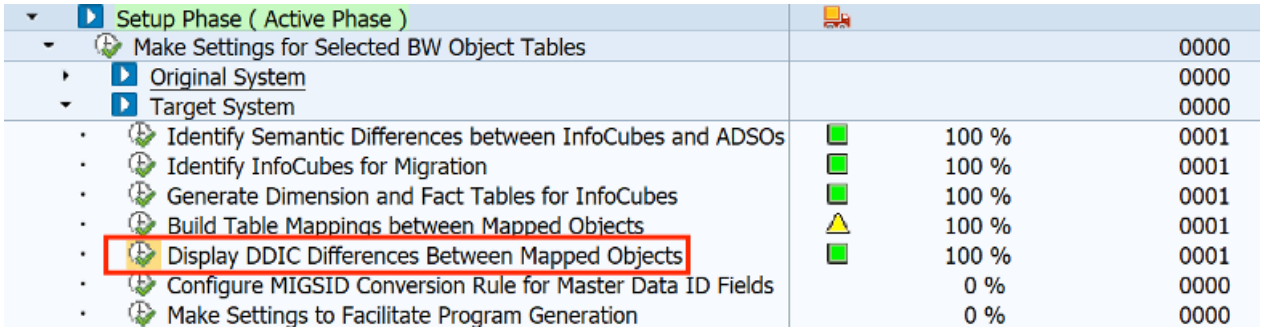


Figure 68: Display DDIC Differences Between Mapped Objects

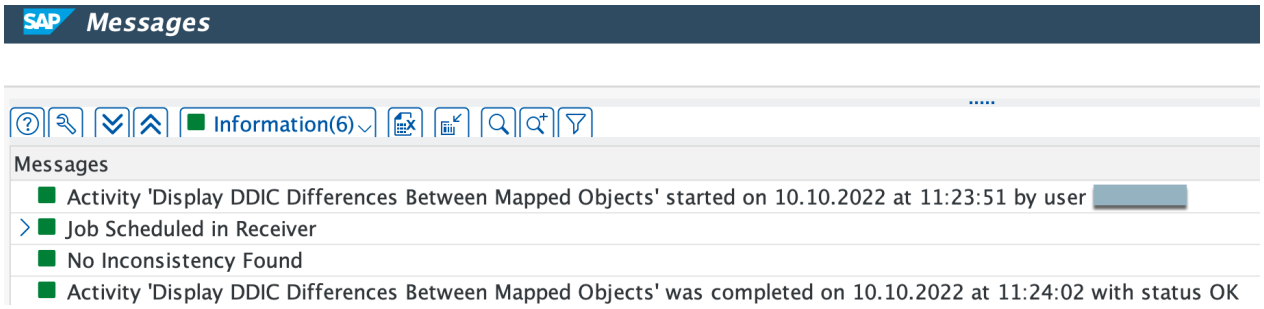


Figure 69: Display DDIC Differences Between Mapped Objects – Logs

This activity does not carry out checks for RS* (BW administration) tables.

⇒ Tip: When an activity returns a warning and you nevertheless decide to proceed, you may create a note for it to describe the reason to be used for further reference.

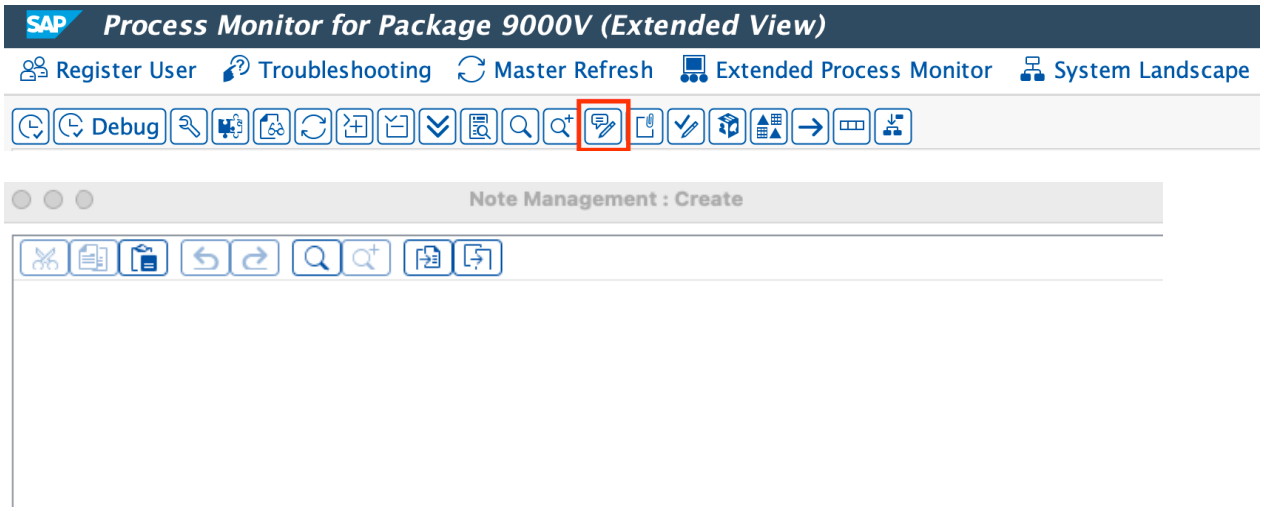


Figure 70: "Manage Notes" in the Conversion Cockpit

3.3.1.9 Target System: Configure MIGSID Conversion Rule for Master Data ID Fields

This activity verifies whether the SID tables (master data IDs) and the RS* tables (BW administration tables) containing master data, already contain data in the target (receiver) system. If data is present, this activity uses the MIGSID conversion rule for the Master Data ID (SID) fields to prevent conflicts in the receiver system. The MIGSID rule is available in the CNVLTRL_OBJTAB control table.

When a SID table in the receiver system already contains master data, merges are expected and a conversion task of the SID table will use the MIGSID conversion rule to generate SID mappings. The SID mappings are used by the conversion tasks of InfoProviders to migrate data consistently across systems. For more information, see the CNVLTRL_SID table in the control system.

This activity scans and adds the SID field and assigns the MIGSID conversion rule to the field. The assignment of rules to the SID fields in the SID tables is stored in the CNVLTRL_OBJTAB control table.

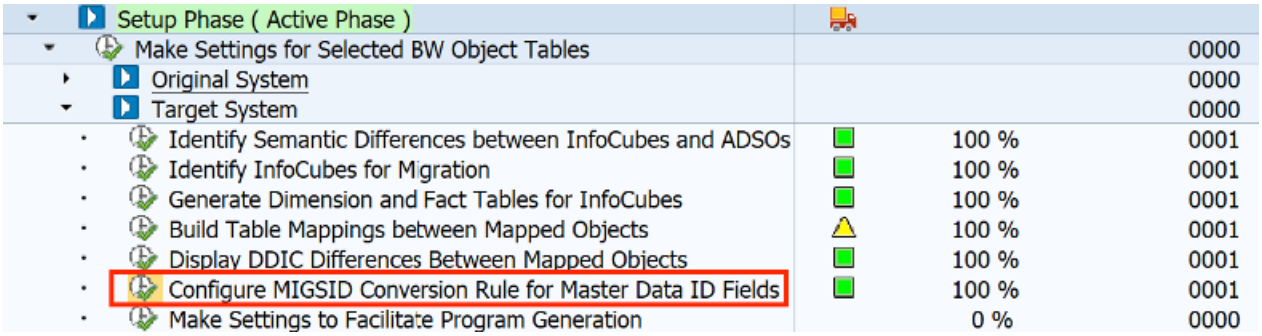


Figure 71: Configure MIGSID Conversion Rule for Master Data ID Fields

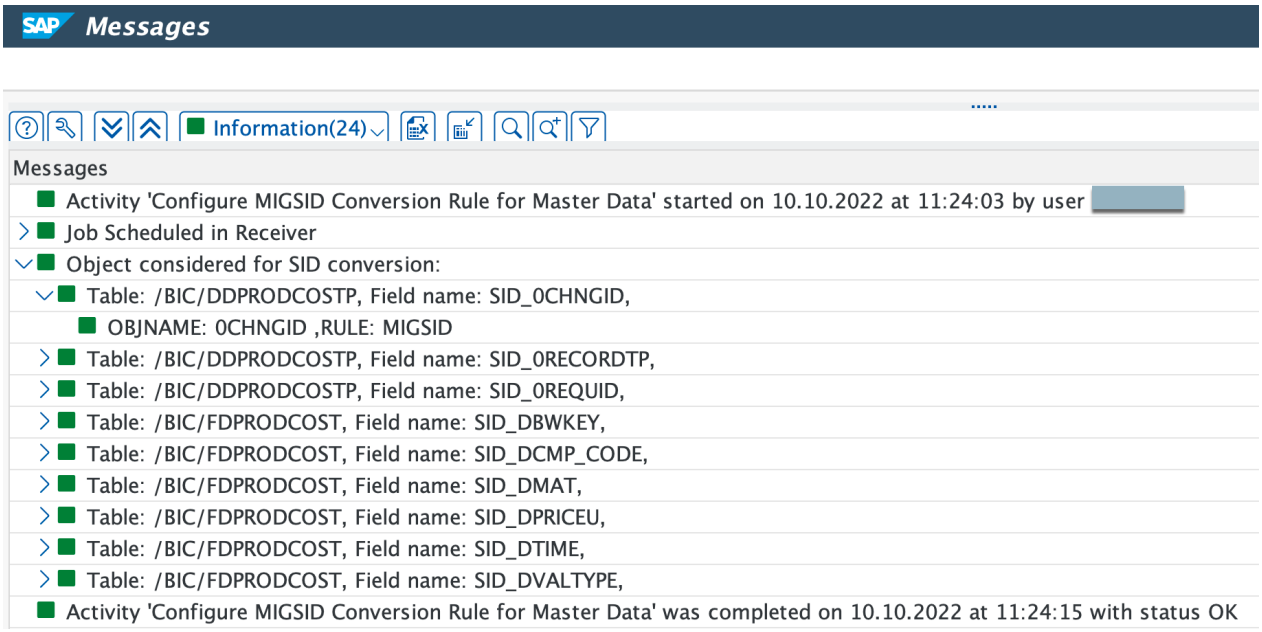


Figure 72: Configure MIGSID Conversion Rule for Master Data ID Fields - Logs

3.3.1.10 Target System: Make Settings to Facilitate Program Generation

This activity schedules sub-activities to rectify inconsistencies in the CNVLTRL_OBJTAB table in the receiver system and to assign conversion rules to handle the renaming of request IDs to TSN values. It validates the consistency of the configurations made in predecessor activities for BW objects that are in the transfer scope in the CNVLTRL_OBJTAB control table. In case any inconsistencies are found, it deactivates the BW objects from the transfer.

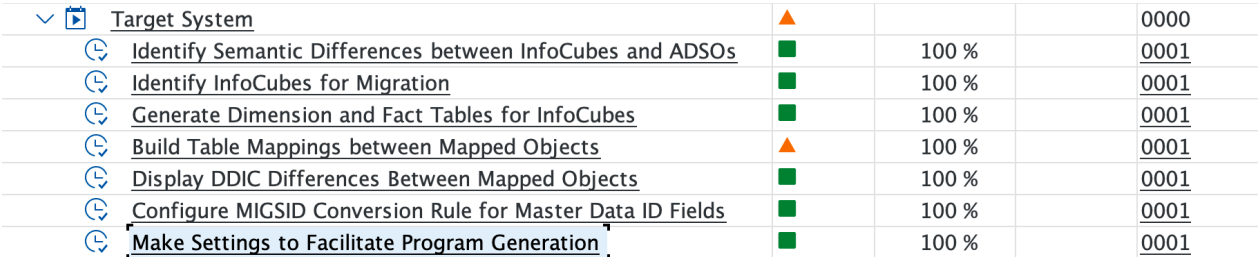


Figure 73: Make Settings to Facilitate Program Generation

⇒ Note: The request ID generation concept is changed with the BW/4HANA, BW bridge and BW 7.5 versions. RSSM-based requests are replaced with TSNs. Hence, during the data transfer from classic objects to HANA-optimized objects in the BW/4HANA or BW bridge systems, old SID-based requests need to be changed to TSN values.

3.3.2 Conversion Setup (Logical System Rename)

This activities group includes the following activities:

- Select Conversion-Relevant InfoObjects
- Define Mapping Values
- Identify Conversion-Relevant Tables and Fields
- Modify Conv-Relevant Customizing for Table-Field Combination
- Validate Mapping Consistency

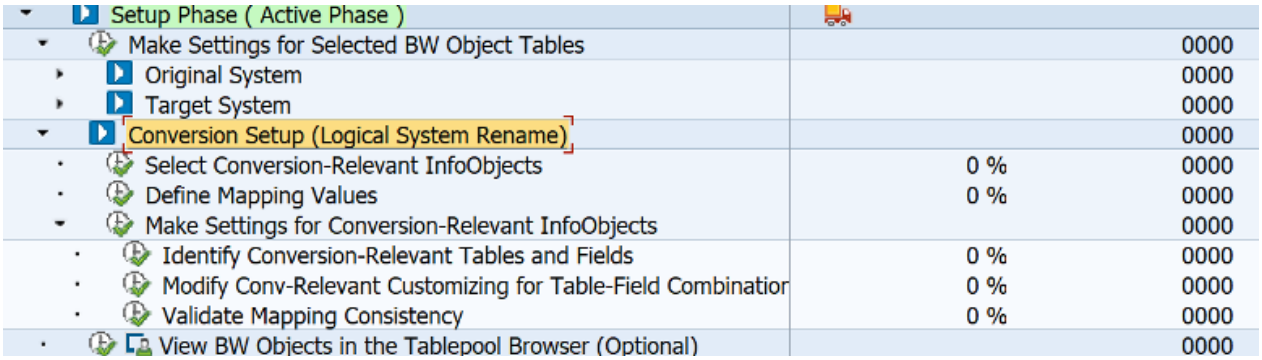


Figure 74: Conversion Setup (Logical System Rename)

3.3.2.1 Select Conversion-Relevant InfoObjects

This activity is relevant for conversion scenarios when you need to customize certain InfoObjects.

When you select a characteristic InfoObject, you can view a proposal of referenced (R) and compounded (C) InfoObjects set for selection. You can choose to include the referenced and compounded InfoObjects or leave them out of the transformation. The activity saves the data in the CNVLTRL_OBJ control table.

When you select an InfoObject for conversion, you are required to assign a rule and rule version to the InfoObject. The rule definitions are stored in the CNVLTRL_RULETP control table. You can use the same rule to convert multiple InfoObjects.

Conversion mappings include a combination of rule and rule version to ensure that each mapping is unique. If the InfoObjects require conversion with different mappings, you can define the same rule for them but with different rule versions.

For example, you want to convert InfoObject 0ACCOUNT and InfoObject 0CHRT_ACCTS using the rule MERGE. Both objects have different mappings for conversion. Hence, you can use Rule MERGE Version '100' for 0ACCOUNT and Version '200' for 0CHRT_ACCTS.

⇒ Note: Not applied to the scope of our sample.

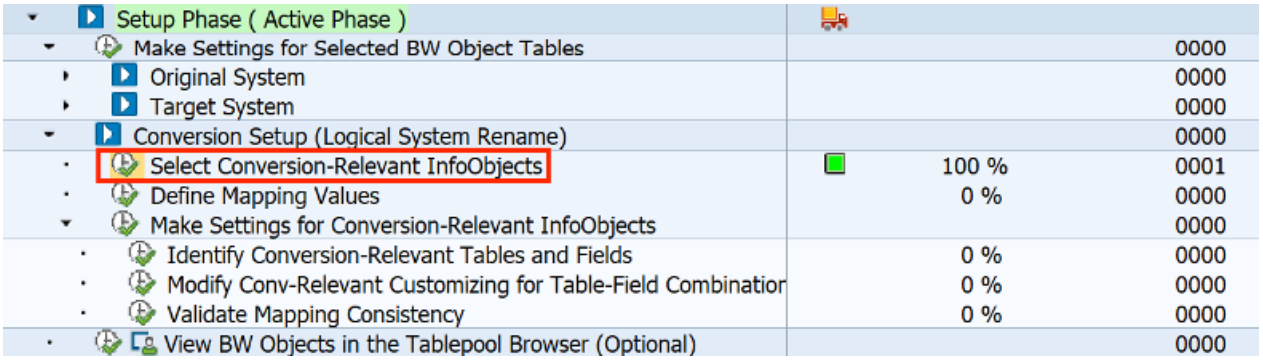


Figure 75: Select Conversion-Relevant InfoObjects

3.3.2.2 Define Mapping Values

In this activity, you can define new values for existing InfoObject values by entering a unique combination of conversion rule and rule version.

By default, rule "LOGSYS_NEW", version '100' is used for the logical system rename.

⇒ Note: Our sample does not have any additional specific mapping combination.

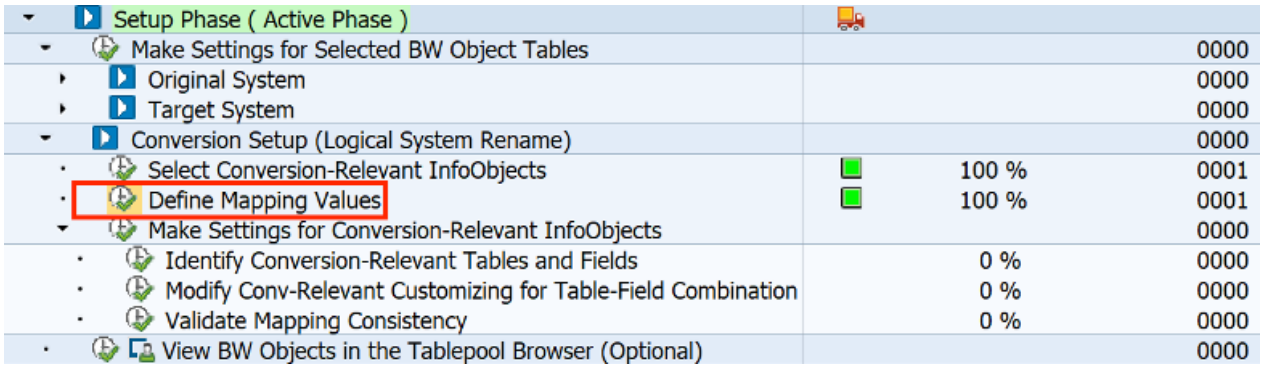


Figure 76: Define Mapping Values

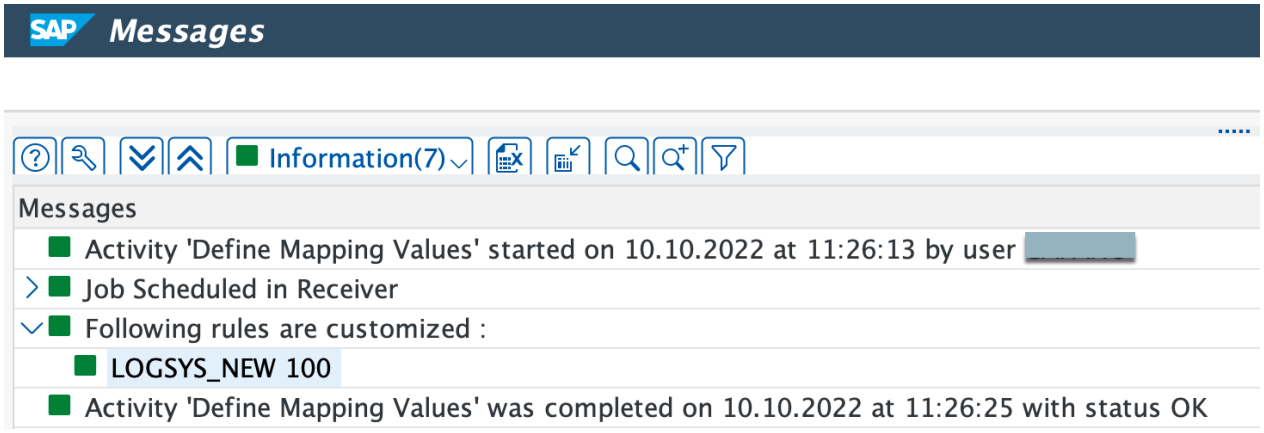
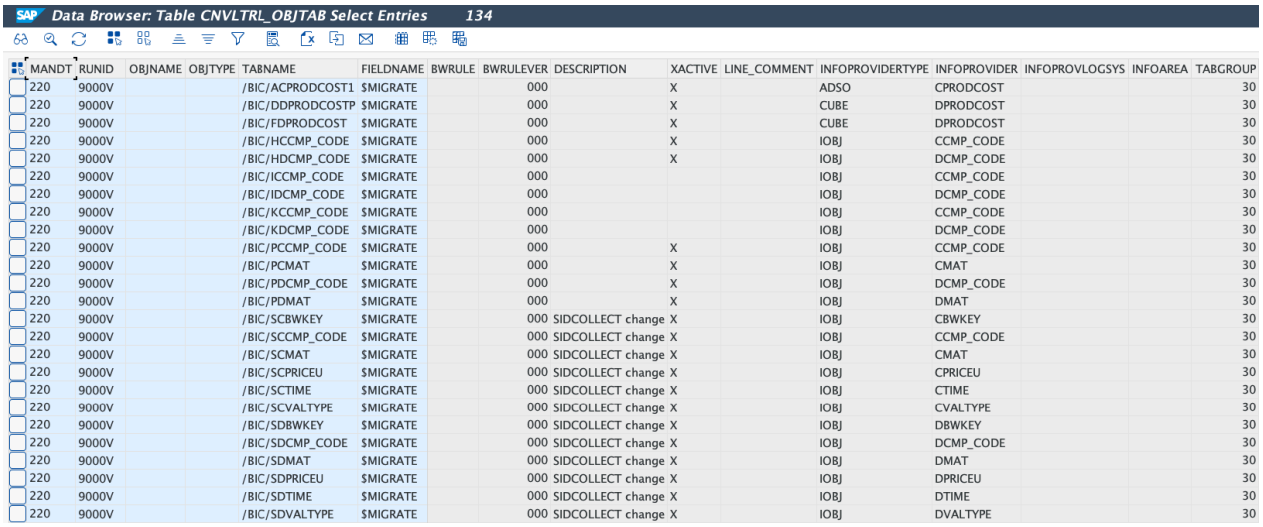


Figure 77: Define Mapping Values - Logs

3.3.2.3 Make Settings: Identify Conversion-Relevant Tables and Fields

This activity looks up the where-used list for the InfoObjects already selected for the transfer. For the list of InfoProviders found, the activity builds a table pool (stored in the CNVLTRL_OBJTAB control table). The table pool contains the list of tables which are relevant for the transfer.



Data Browser: Table CNVLTRL_OBJTAB Select Entries 134

MANDT	RUNID	OBJNAME	OBJTYPE	TABNAME	FIELDNAME	BWRULE	BWRULEVER	DESCRIPTION	XACTIVE	LINE_COMMENT	INFOPROVIDERTYPE	INFOPROVIDER	INFOPROVLOGSYS	INFOAREA	TABGROUP
220	9000V			/BIC/ACPRODCOST1	$MIGRATE		000		X		ADSO	CPRODCOST			30
220	9000V			/BIC/DDPRODCOSTP	$MIGRATE		000		X		CUBE	DPRODCOST			30
220	9000V			/BIC/FDPRODCOST	$MIGRATE		000		X		CUBE	DPRODCOST			30
220	9000V			/BIC/HCCMP_CODE	$MIGRATE		000		X		IOBJ	CCMP_CODE			30
220	9000V			/BIC/HDCMP_CODE	$MIGRATE		000		X		IOBJ	DCMP_CODE			30
220	9000V			/BIC/ICCMP_CODE	$MIGRATE		000				IOBJ	CCMP_CODE			30
220	9000V			/BIC/IDCMP_CODE	$MIGRATE		000				IOBJ	DCMP_CODE			30
220	9000V			/BIC/KCCMP_CODE	$MIGRATE		000				IOBJ	CCMP_CODE			30
220	9000V			/BIC/KDCMP_CODE	$MIGRATE		000				IOBJ	DCMP_CODE			30
220	9000V			/BIC/PCCMP_CODE	$MIGRATE		000		X		IOBJ	CCMP_CODE			30
220	9000V			/BIC/PCMAT	$MIGRATE		000		X		IOBJ	CMAT			30
220	9000V			/BIC/PDCMP_CODE	$MIGRATE		000		X		IOBJ	DCMP_CODE			30
220	9000V			/BIC/PDMAT	$MIGRATE		000		X		IOBJ	DMAT			30
220	9000V			/BIC/SCBWKEY	$MIGRATE		000	SIDCOLLECT change	X		IOBJ	CBWKEY			30
220	9000V			/BIC/SCCMP_CODE	$MIGRATE		000	SIDCOLLECT change	X		IOBJ	CCMP_CODE			30
220	9000V			/BIC/SCMAT	$MIGRATE		000	SIDCOLLECT change	X		IOBJ	CMAT			30
220	9000V			/BIC/SCPRICEU	$MIGRATE		000	SIDCOLLECT change	X		IOBJ	CPRICEU			30
220	9000V			/BIC/SCTIME	$MIGRATE		000	SIDCOLLECT change	X		IOBJ	CTIME			30
220	9000V			/BIC/SCVALTYPE	$MIGRATE		000	SIDCOLLECT change	X		IOBJ	CVALTYPE			30
220	9000V			/BIC/SDBWKEY	$MIGRATE		000	SIDCOLLECT change	X		IOBJ	DBWKEY			30
220	9000V			/BIC/SDCMP_CODE	$MIGRATE		000	SIDCOLLECT change	X		IOBJ	DCMP_CODE			30
220	9000V			/BIC/SDMAT	$MIGRATE		000	SIDCOLLECT change	X		IOBJ	DMAT			30
220	9000V			/BIC/SDPRICEU	$MIGRATE		000	SIDCOLLECT change	X		IOBJ	DPRICEU			30
220	9000V			/BIC/SDTIME	$MIGRATE		000	SIDCOLLECT change	X		IOBJ	DTIME			30
220	9000V			/BIC/SDVALTYPE	$MIGRATE		000	SIDCOLLECT change	X		IOBJ	DVALTYPE			30

Figure 78: List of relevant tables (table pool) in CNVLTRL_OBJTAB control table

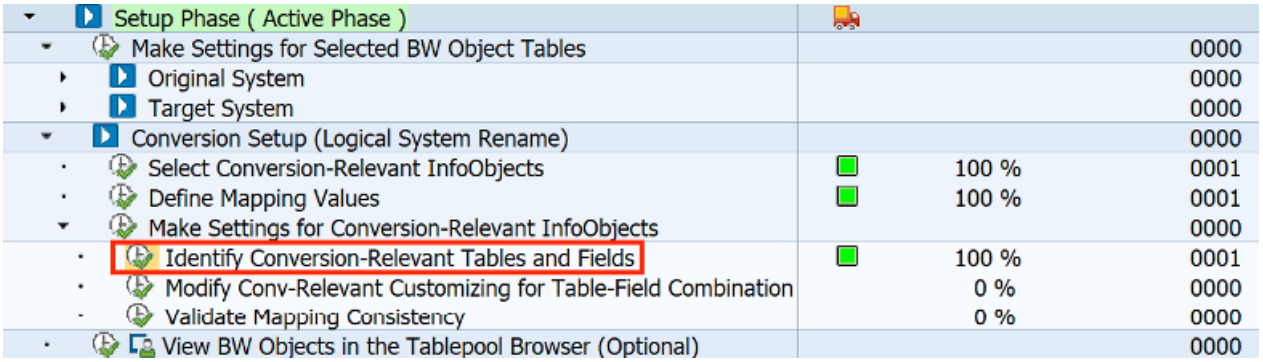


Figure 79: Identify Conversion-Relevant Tables and Fields

3.3.2.4 Make Settings: Modify Conv-Relevant Customizing for Table-Field Combination

This activity uses the data identified by the activity in the predecessor step “Identify Conversion-relevant Tables and Field” and modifies the data further with table-level information in the sender system.

It updates the table names in the CNVLTRL_OBJTAB control table with the names of the sender tables in the receiver system by using the following mappings:

- Table mappings present in the control table CNVLTRL_TABMAP.
- Table-Field mappings present in the control table CNVLTRL_FLDMAP.

⇒ Note: Not relevant for the scope of our sample.

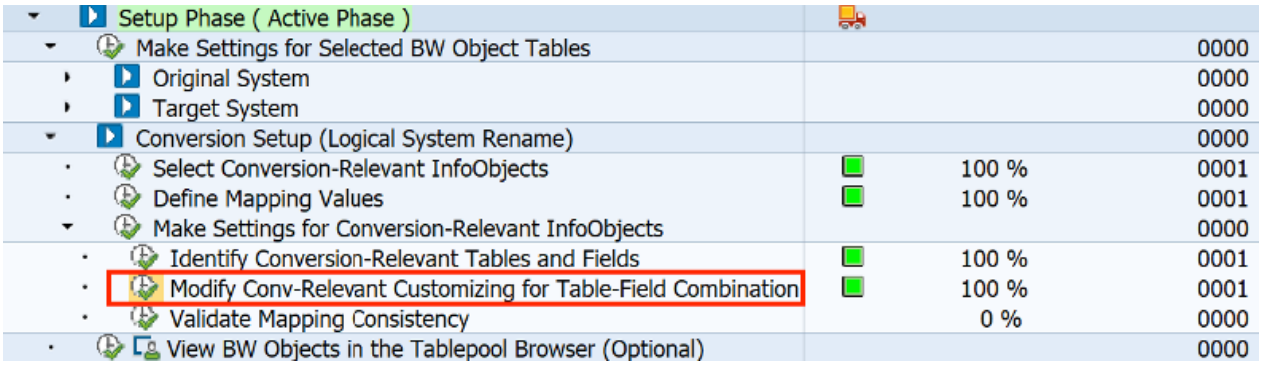


Figure 80: Modify Conv-Relevant Customizing for Table-Field Combination

3.3.2.5 Make Settings: Validate Mapping Consistency

This activity carries out various checks to validate the mappings defined in the predecessor activity “Define Mapping Values”.

The following checks are made:

- *Old Value Existence Check*: All the old values that you maintained in the conversion mapping should exist in the master data tables, otherwise the activity ends in error.
- *Swap Check*: If you change old values to new values, and you change the new value again to a different new value. Example: If you want to convert A->B and B->C and both A & B exist in the Master Data, the activity ends in error.
- *Implicit Check*: If you want to convert an old value to a new value and the new value already exists. Example: If you want to convert A->B and B already exists in the master data. In this case:
    - In a 1:1 conversion, it is not allowed that source is changed to target and the target itself is not getting changed: Error message.
    - In an N: 1 conversion, you should check if you want the existing customer record to be overwritten: Warning message.

⇒ Note: Not relevant for the scope of our sample.

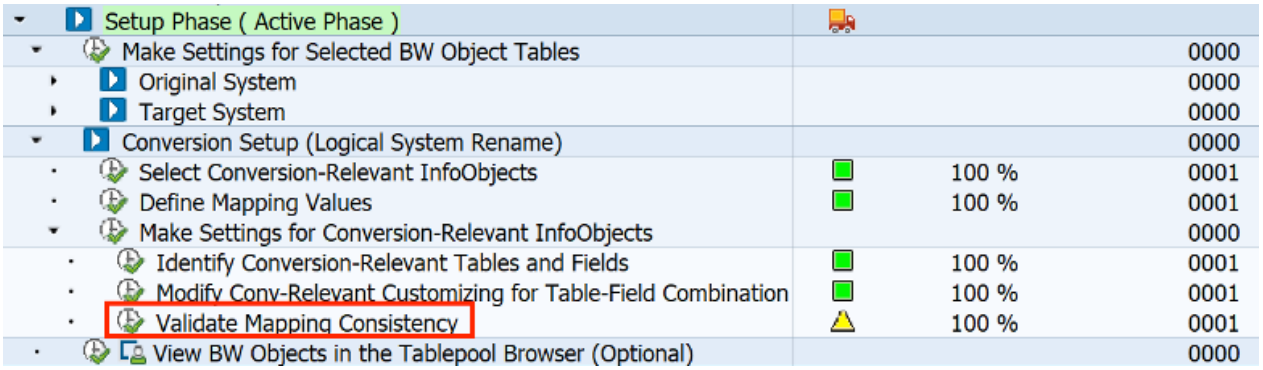


Figure 81: Validate Mapping Consistency

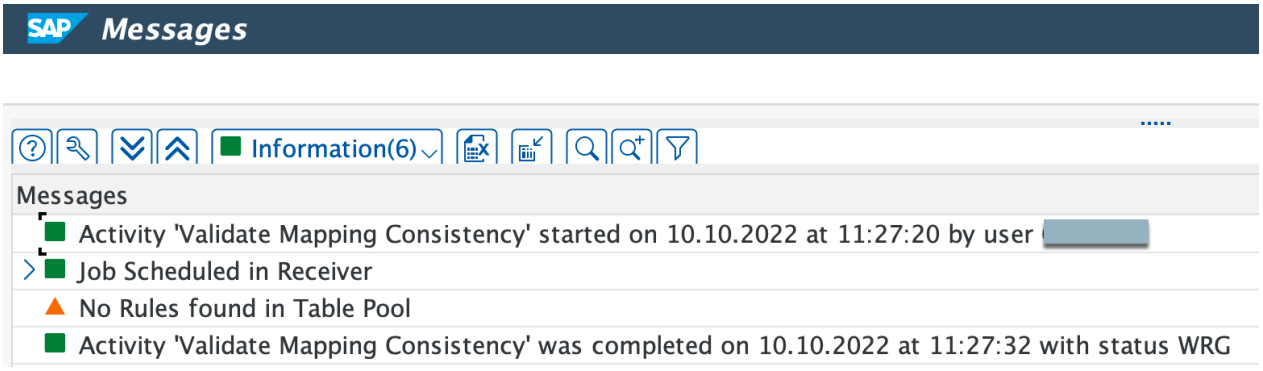


Figure 82: Validate Mapping Consistency - Logs

3.3.3 View BW Objects in the Tablepool Browser (Optional)

In this optional activity, you can get a complete view of all relevant BW objects selected for the transfer process.

- *InfoProvider view:* Displays all BW InfoProviders selected.
- *InfoObject view*: Displays all InfoObjects selected.
- *Rule & Rule Version view*: Displays tables that have assigned rules and rule versions as well as tables that have no assigned rules.
- *Tables view*: Displays tables selected.

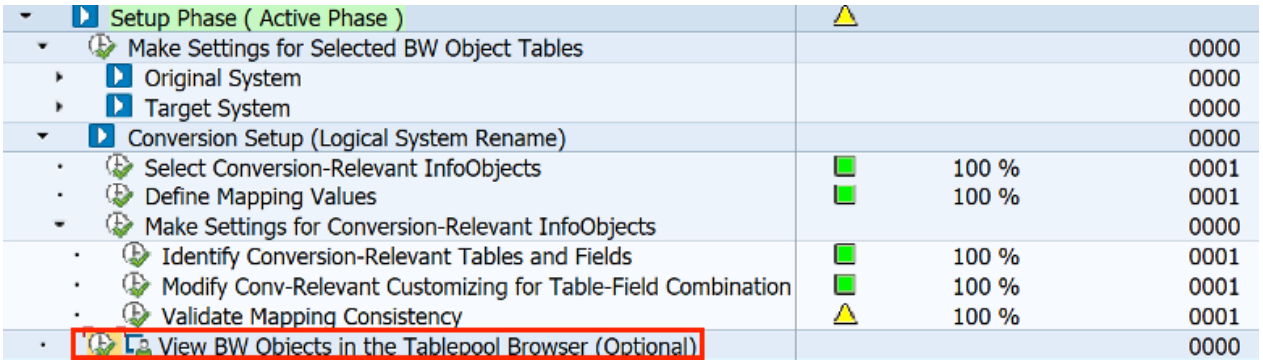


Figure 83: View BW Objects in the Tablepool Browser

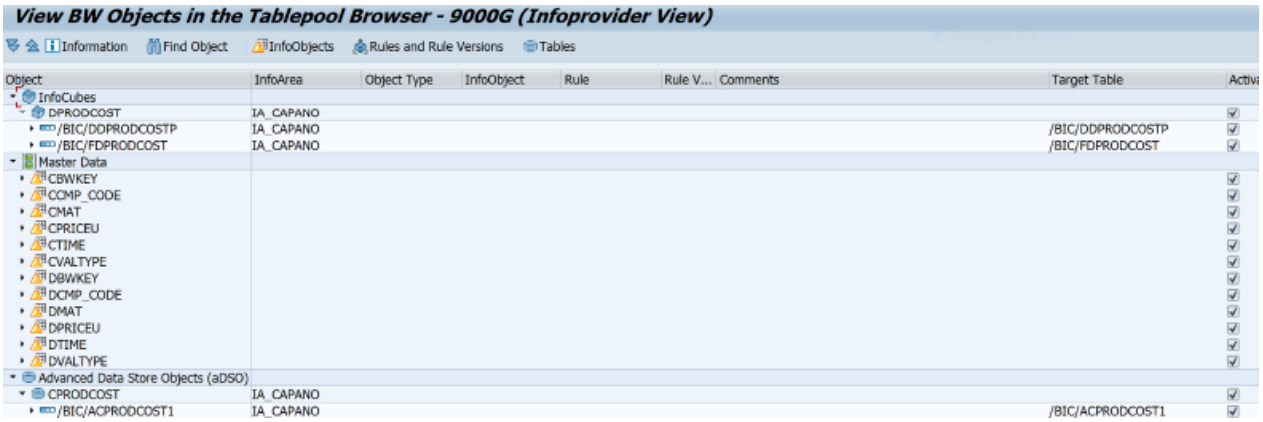


Figure 84: View BW Objects in the Tablepool Browser - InfoProvider view

3.4 Generation Phase

In this phase, the programs for the relevant tables in the transfer process are generated.

One (1) program is generated for each phase/step of a table processing:

1. *Read* (selection of tables).
2. *Move* (transfer of the data from the sender system to the receiver system).
3. *Convert* (insertion of the data into the corresponding tables into the receiver system).

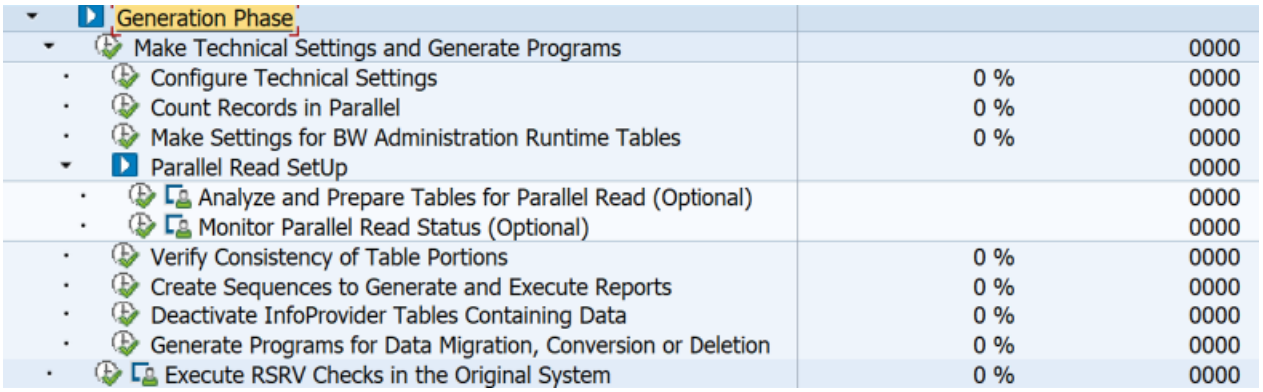


Figure 85: Generation Phase

The next activity is an activities group and you have the option to either run it all together (mass execution) or each activity individually. If you want to execute any of the dialog activities, go to the Extended View and execute each activity sequentially. If you use the mass execution, the dialog activities under this activities group are skipped.

If the activities group terminates or ends with errors or warnings, activate the Extended view of the process tree. You can now view all activities that belong to the group activity and their status.

3.4.1 Make Technical Settings and Generate Programs: Configure Technical Settings

This activity determines how the tooling should handle tables present in the CNVLTRL_OBJTAB control table during the transfer process.

It analyzes each relevant table and makes configurations such as:

- *Count Rows*: Number of rows in the table.
- *Key Change:* Whether the key field of a table is configured for a conversion.
- *Megabytes per Package*: The size of the data in each package before commits are done (by default, the data size for each package is set to 20 MB).
- *Insert Mode of Table*: The type of the table (insert, modify or update).

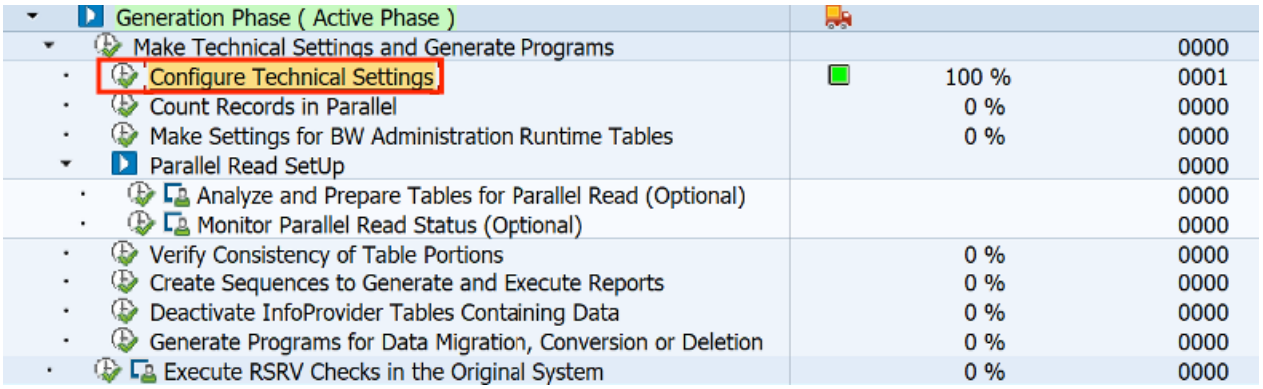


Figure 86: Configure Technical Settings

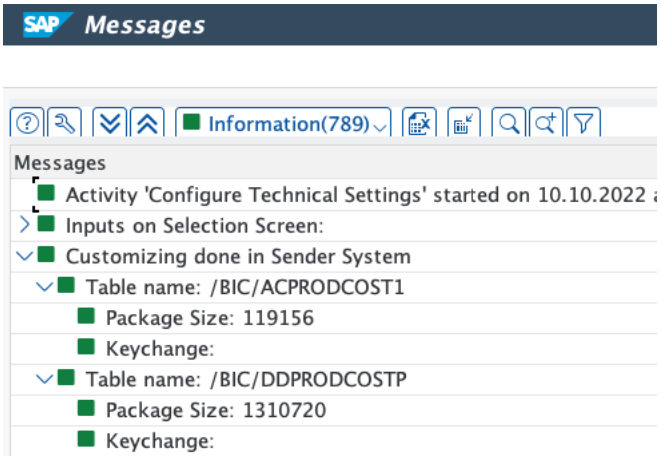


Figure 87: Configure Technical Settings – Logs

After making the configurations, this activity stores them in the CNVLTRL_TEC control table.

3.4.2 Make Technical Settings and Generate Programs: Count Records in Parallel

This activity counts the number of records for the list of relevant tables (table pool), and stores the results in the CNVLTRL_OBJTAB and CNVLTRL_TEC control tables.

It carries out the rows count in parallel using the number of background jobs set by default in the standard configuration. You can also change the default number of background jobs in the process monitor.

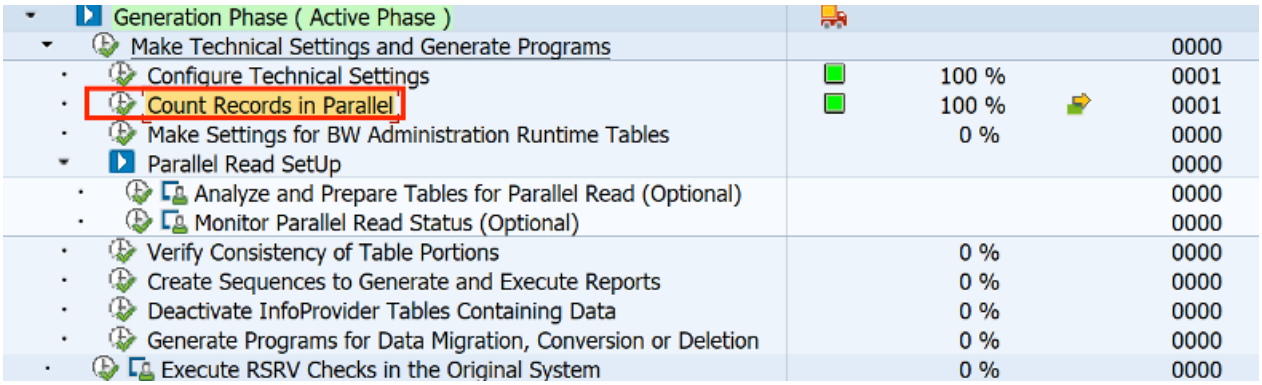


Figure 88: Count Records in Parallel

In our sample, we used 5 background jobs.

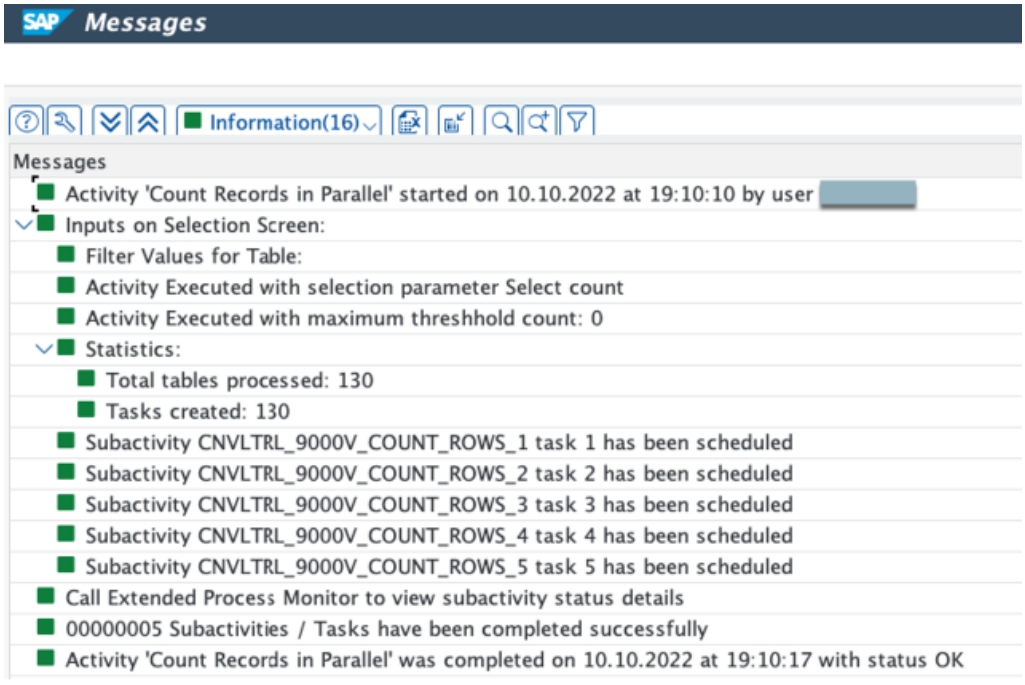


Figure 89: Count Records in Parallel - Logs

Data Browser: Table CNVLTRL_TEC Select Entries 130

RUNID	TABNAME	XKEYCHANGE	PACKSIZE	TABGROUP	PRIO_GROUP	COUNT_ROWS	TABSIZE
9000V	/BIC/ACPRODCOST1		119156	30	00	00000000000051761812	00000000000008896561
9000V	/BIC/DDPRODCOSTP		1310720	30	00	00000000000000000003	00000000000000000000
9000V	/BIC/FDPRODCOST	X	455903	30	00	00000000000005239810	00000000000000235382
9000V	/BIC/HCCMP_CODE		110376	30	00	00000000000000000000	00000000000000000000
9000V	/BIC/HDCMP_CODE		110376	30	00	00000000000000000000	00000000000000000000
9000V	/BIC/PCCMP_CODE		1747627	30	00	00000000000000000008	00000000000000000000
9000V	/BIC/PCMAT		524288	30	00	00000000000000073547	00000000000000002873
9000V	/BIC/PDCMP_CODE		1747627	30	00	00000000000000000008	00000000000000000000
9000V	/BIC/PDMAT		524288	30	00	00000000000000073547	00000000000000002873
9000V	/BIC/SCBWKEY		1165084	30	00	00000000000000001594	00000000000000000028
9000V	/BIC/SCCMP_CODE		1165084	30	00	00000000000000000008	00000000000000000000
9000V	/BIC/SCMAT		455903	30	00	00000000000000073547	00000000000000003304
9000V	/BIC/SCPRICEU		1310720	30	00	00000000000000000006	00000000000000000000
9000V	/BIC/SCTIME		806597	30	00	00000000000000000002	00000000000000000000
9000V	/BIC/SCVALTYPE		699051	30	00	00000000000000000003	00000000000000000000
9000V	/BIC/SDBWKEY		1165084	30	00	00000000000000001594	00000000000000000028
9000V	/BIC/SDCMP_CODE		1165084	30	00	00000000000000000008	00000000000000000000
9000V	/BIC/SDMAT		455903	30	00	00000000000000073547	00000000000000003304
9000V	/BIC/SDPRICEU		1310720	30	00	00000000000000000006	00000000000000000000
9000V	/BIC/SDTIME		806597	30	00	00000000000000000002	00000000000000000000
9000V	/BIC/SDVALTYPE		699051	30	00	00000000000000000003	00000000000000000000
9000V	/BIC/TCCMP_CODE		238313	30	00	00000000000000000000	00000000000000000000
9000V	/BIC/TCMAT		275941	30	00	00000000000000000000	00000000000000000000
9000V	/BIC/TDCMP_CODE		238313	30	00	00000000000000000000	00000000000000000000
9000V	/BIC/TDMAT		275941	30	00	00000000000000000000	00000000000000000000
9000V	RSAPPL		113976	30	00	00000000000000000977	00000000000000000176

Figure 90: COUNT_ROWS is updated in the CNVLTRL_TEC control table

3.4.3 Make Technical Settings and Generate Programs: Make Settings for BW Administration Tables

This activity harmonizes release versions in the sender and receiver systems.

It assigns the RSBKREQUEST_V database view to the RSBKREQUEST table in the sender system and afterwards stores the assignment information in the CNVLTRL_TEC table in the sender system.

Finally, this activity modifies the data of the RSBKREQUEST table in the CNVLTRL_DDIC table in the receiver system to handle structural differences between the sender and receiver tables.

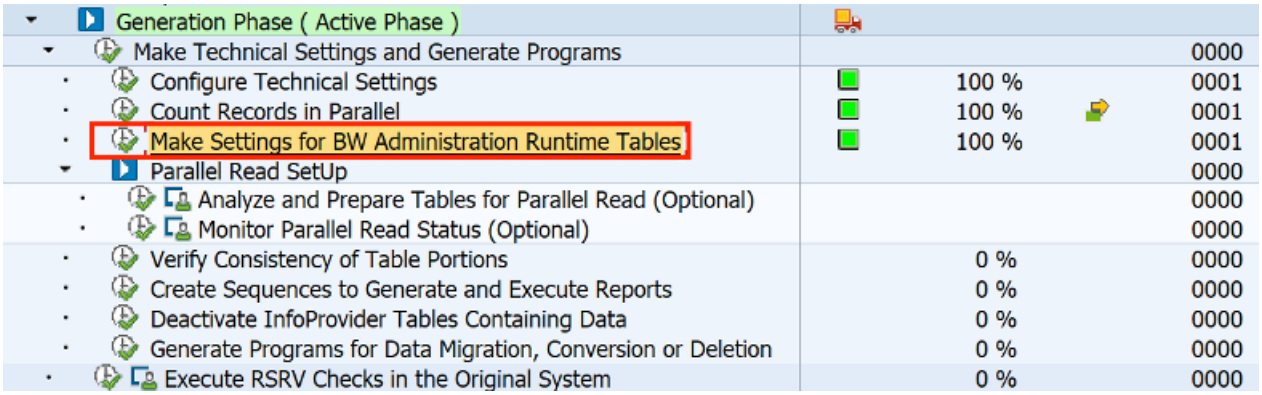


Figure 91: Make Settings for BW Administration Tables

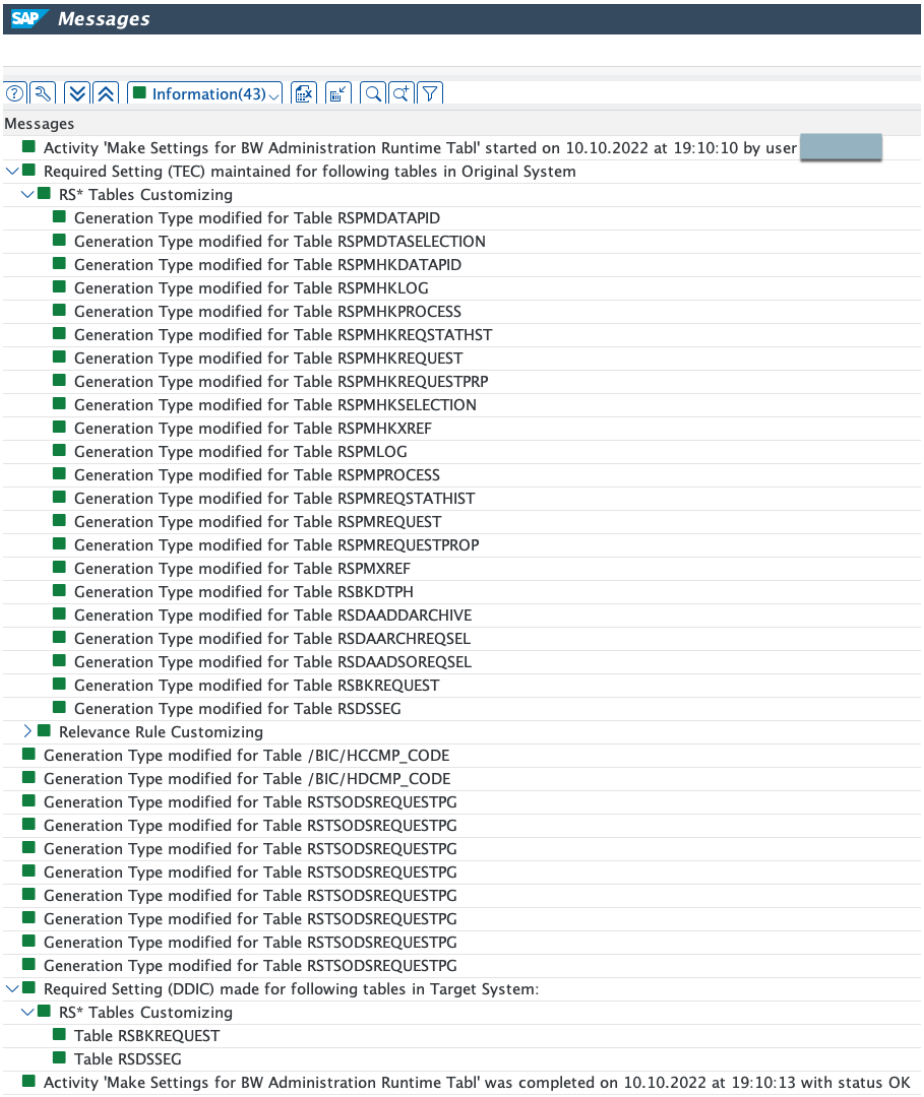


Figure 92: Make Settings for BW Administration Tables - Logs

3.4.4 Make Technical Settings and Generate Programs: Parallel Read Setup

In this activity, you can analyze and prepare the transfer relevant tables to enable a parallel reading of the data to be processed in the "Migration phase". This activity creates portions for those tables which allow parallelization of the read tasks (data selection). One portion is read per each read task.

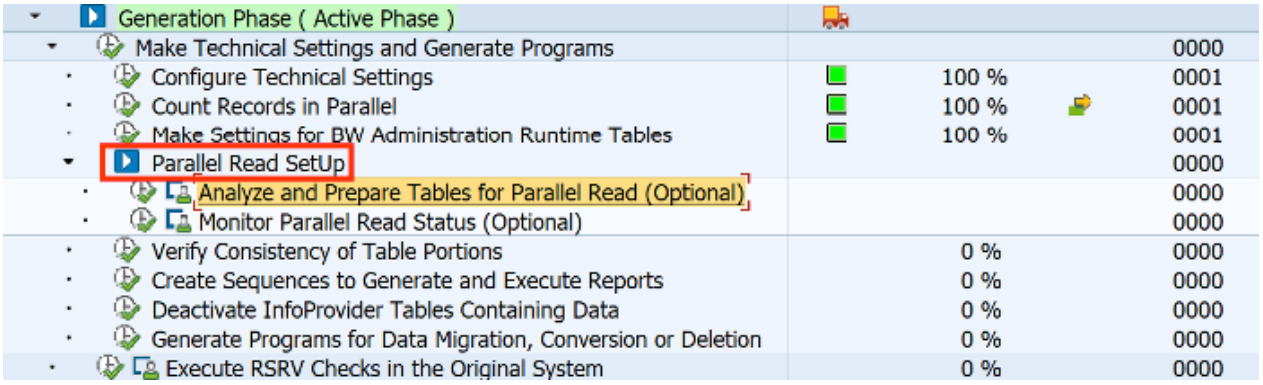


Figure 93: Parallel Read Setup

3.4.4.1 Analyze and Prepare Tables for Parallel Read (Optional)

The selection screen for the parallel read settings contains:

- *Table Name*: You can select the tables based on the list of relevant ones (table pool).
- *Threshold Value*: All tables with rows count above this value will be selected for the parallel read settings. If you enter 100 million, for example, all tables with more than 100 million records are selected for parallel read.
- *Default Portion Size*: Determines the number of portions that are created for parallel read for each table. For example, if a table has 100 million records and the portion size is 20 million then this activity creates 5 (five) portions for parallel read. This will determine the default number of portions for each table (it can be modified later on).

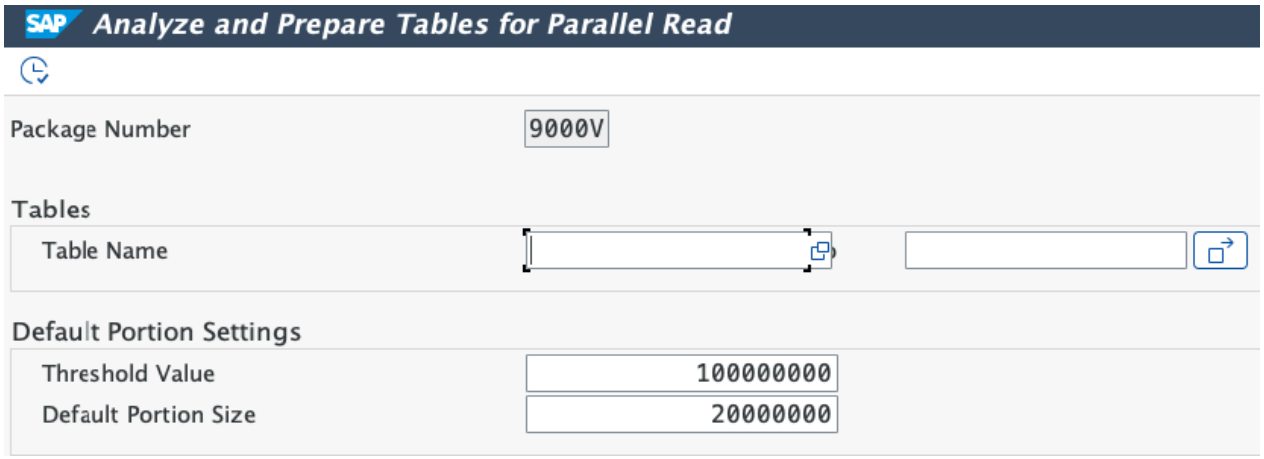


Figure 94: Analyze and Prepare Tables for Parallel Read (1)

After executing this activity, the monitor will display all tables with default parallel read field and portion number if the parallel read settings are not already maintained. Tables with rows count above threshold value are selected.

Our sample did not use the parallel read settings. Just to give an overview, these are columns displayed:

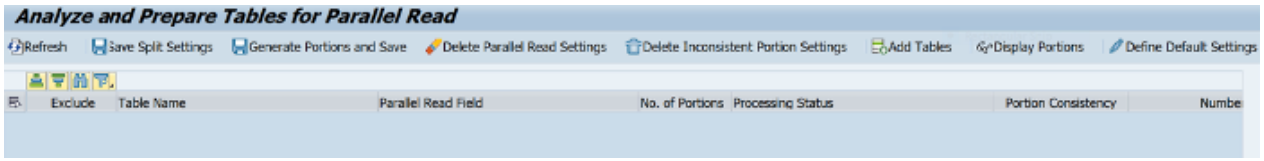


Figure 95: Analyze and Prepare Tables for Parallel Read (2)

- *Exclude checkbox*: Select this checkbox to exclude an entry from the selected ones for processing while performing a parallel read settings execution. For example, if you select all entries then rows with “exclude checkbox” set will not be processed.
- *Parallel Read field:* Portions of the table are generated based on this split field. The tool will bring a proposal for this field but it can also be modified.
- *No. of Portions*: You can modify the number of portions to be created. If portions are already created, then this column displays the number of portions created. This field can also be modified. The actual number of portions created can be different from the number you have maintained as it depends on the number of records in a table.
- *Processing Status*: Displays the status of the parallel read setup for a given table.

- *Portion Consistency*: Displays the consistency of the portions created for each table.
- *Number of Entries*: Displays the number of records in the table (rows count).

a. After executing “Save Split Settings”, this activity saves the parallel read settings in the CNVLTRL_TEC control table with parallel read SPLIT type. These settings will be used to generate portions for parallel read. “Processing Status” is then updated to ‘Portions do not exist’. “Portion Consistency” will be red as parallel read settings are saved but portions are not generated yet. If you want to delete the saved parallel read settings of a table, then select the table entry (s) and execute “Delete Parallel Read Settings”.

b. Afterwards, execute “Generate Portions and Save” to generate the portions for the selected tables (you may select the number of background jobs to be used for this processing).

c. “Processing Status” is updated to ‘Generation running’ if the portions generation is not finished yet.

d. “Processing Status” is updated to ‘Portions exist’ once after the portions are generated successfully. “Portion Consistency” displays the consistency status of the portions.

e. The parallel read setup is complete for a table if “Processing Status” indicates that ‘Portions exist’ and “Portion Consistency” status is not red.

    - If “Portion Consistency status “of a table is red, you can choose a different parallel read field from the dropdown list and repeat steps b and c for this table. If consistent portions cannot be generated using any field, then continue with the next step.

f. Execute “Delete Inconsistent Portions Settings” to delete parallel read settings for all tables that have inconsistent portions. This option is used if consistent portions cannot be generated for a table with any of the given parallel read fields which are available in the dropdown. In this case, the parallel read settings need to be deleted (“Processing Status” is updated to ‘Split criteria not saved’ for such tables).

⇒ Tips:

- When defining the *Parallel Read field* for the split criteria, a good practice is to select a field with the higher number of distinct values in the table. For example, a table which contains 5 distinct values for the field ‘Company Code’ and 100 distinct values for ‘Material’ -> by selecting ‘Material’ the split criteria will be more efficient for the number of portions to be created and consequently more background jobs will be possible to be used for the parallelization of this table.

- Parallel read setting is a powerful feature and extremely recommended to be used. It can have significant impact for runtime optimization.

3.4.4.2 Monitor Parallel Read Status (Optional)

In this activity, you can monitor the execution of the parallel read settings created in the previous activity. It is display-only, therefore if you need to change any settings (i.e. the field for the parallel read split which resulted into inconsistencies), it has to be done through "Analyze and Prepare Tables for Parallel Read (Optional)".

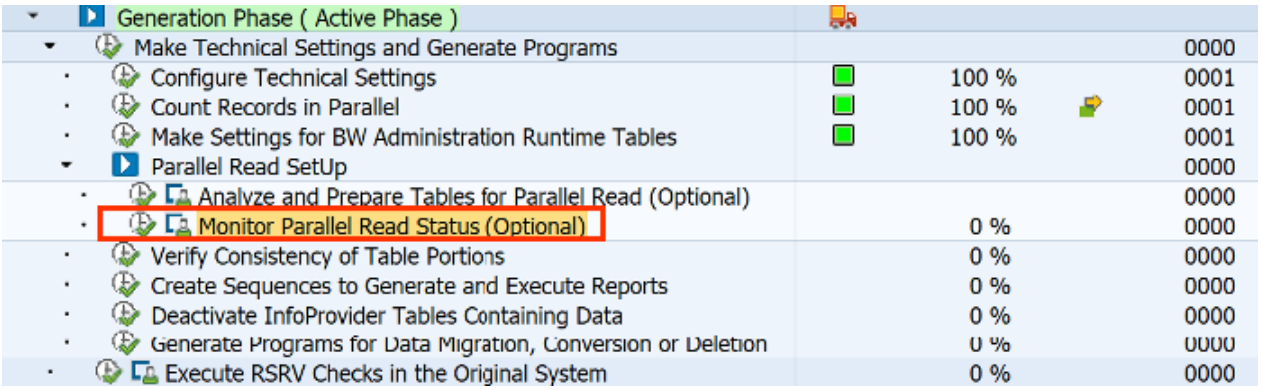


Figure 96: Monitor Parallel Read Status (Optional)

3.4.5 Verify Consistency of Table Portions

This activity verifies the consistency of the table portions set for parallel read in the previous activity by checking the data stored in the CNVLTRL_PORTION control table.

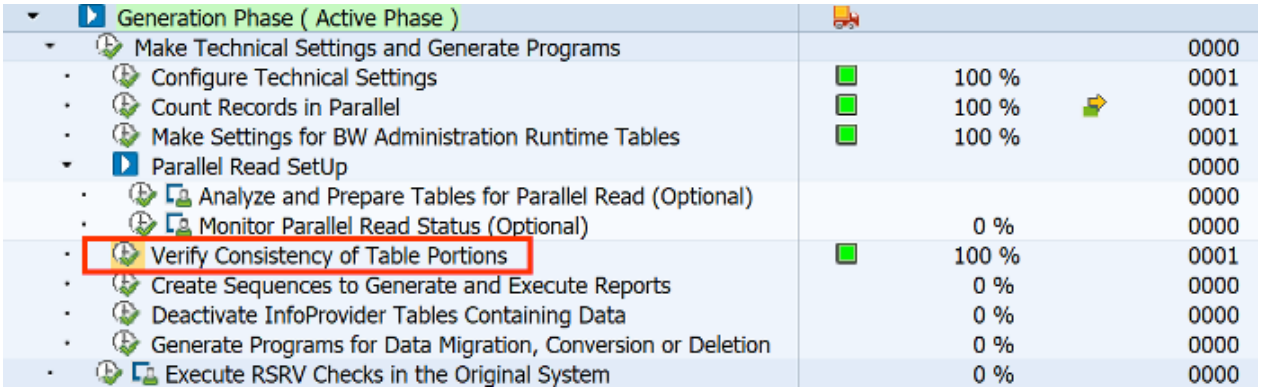
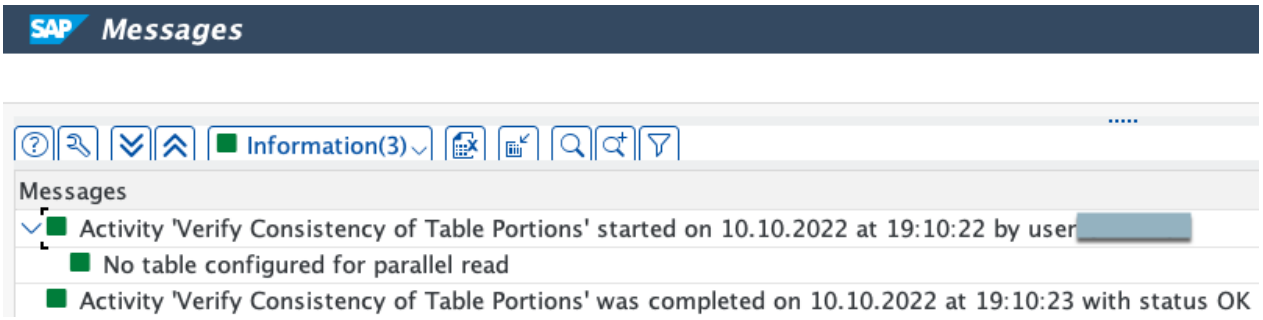


Figure 97: Verify Consistency of Table Portions



There were no portions created as we did not use parallel read settings in our sample.

3.4.6 Create Sequences to Generate and Execute Reports

This activity creates the sequences and task numbers for the generation and execution of the Remote Conversion reports - Read (in the sender system), Move (in the sender system) and Insert (in the receiver system).

The sequence information is stored in the CNVLTRL_SEQ and CNVLTRL_TASK control tables in the sender and receiver systems.

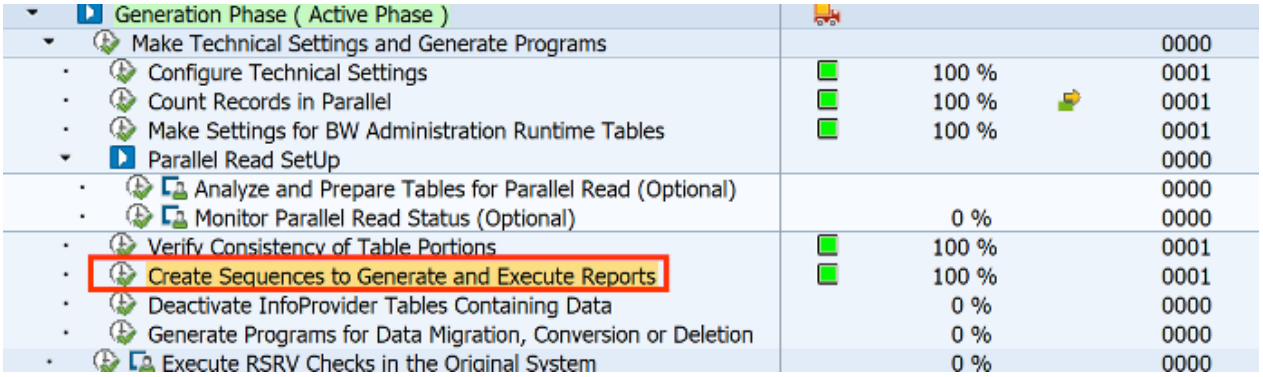


Figure 98: Create Sequences to Generate and Execute Reports

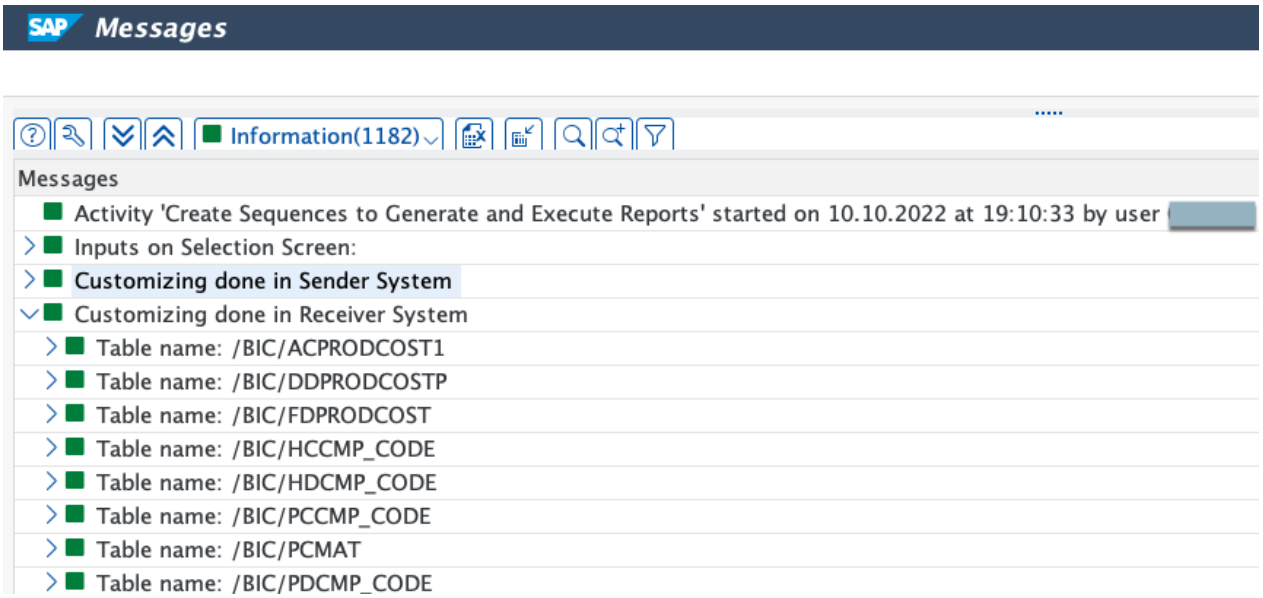
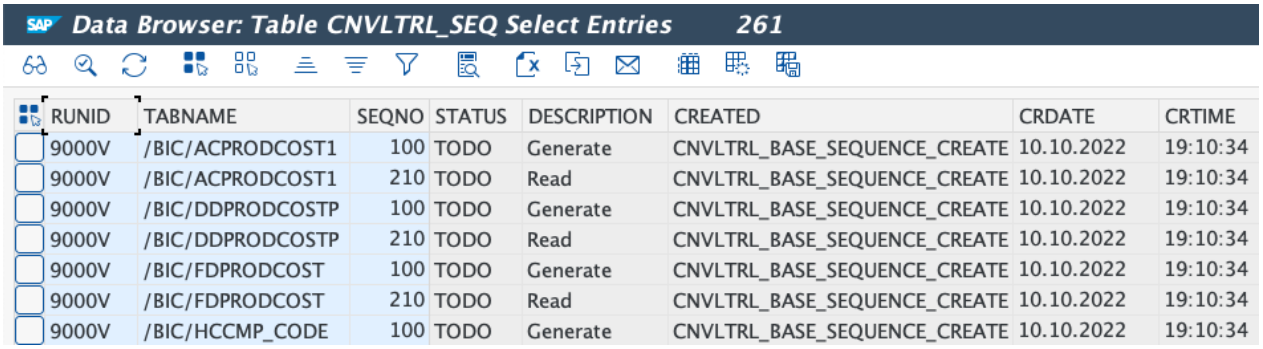


Figure 99: Create Sequences to Generate and Execute Reports – Logs



RUNID	TABNAME	SEQNO	STATUS	DESCRIPTION	CREATED	CRDATE	CRTIME
9000V	/BIC/ACPRODCOST1	100	TODO	Generate	CNVLTRL_BASE_SEQUENCE_CREATE	10.10.2022	19:10:34
9000V	/BIC/ACPRODCOST1	210	TODO	Read	CNVLTRL_BASE_SEQUENCE_CREATE	10.10.2022	19:10:34
9000V	/BIC/DDPRODCOSTP	100	TODO	Generate	CNVLTRL_BASE_SEQUENCE_CREATE	10.10.2022	19:10:34
9000V	/BIC/DDPRODCOSTP	210	TODO	Read	CNVLTRL_BASE_SEQUENCE_CREATE	10.10.2022	19:10:34
9000V	/BIC/FDPRODCOST	100	TODO	Generate	CNVLTRL_BASE_SEQUENCE_CREATE	10.10.2022	19:10:34
9000V	/BIC/FDPRODCOST	210	TODO	Read	CNVLTRL_BASE_SEQUENCE_CREATE	10.10.2022	19:10:34
9000V	/BIC/HCCMP_CODE	100	TODO	Generate	CNVLTRL_BASE_SEQUENCE_CREATE	10.10.2022	19:10:34

Figure 100: Conversion sequence stored in the CNVLTRL_SEQ control table in the sender system

3.4.7 Deactivate InfoProvider Tables Containing Data

This activity makes settings to avoid multiple writing of data into InfoProviders in order to ensure consistency during the data transfer. It checks if InfoProviders tables in the receiver system already contain data.

If an InfoProvider contains data, the activity deactivates the respective table in the CNVLTRL_OBJTAB control table and deletes the tasks from the CNVLTRL_TASK control table. This activity also truncates the BW administration runtime tables (RS* tables) in the receiver system that contain data.

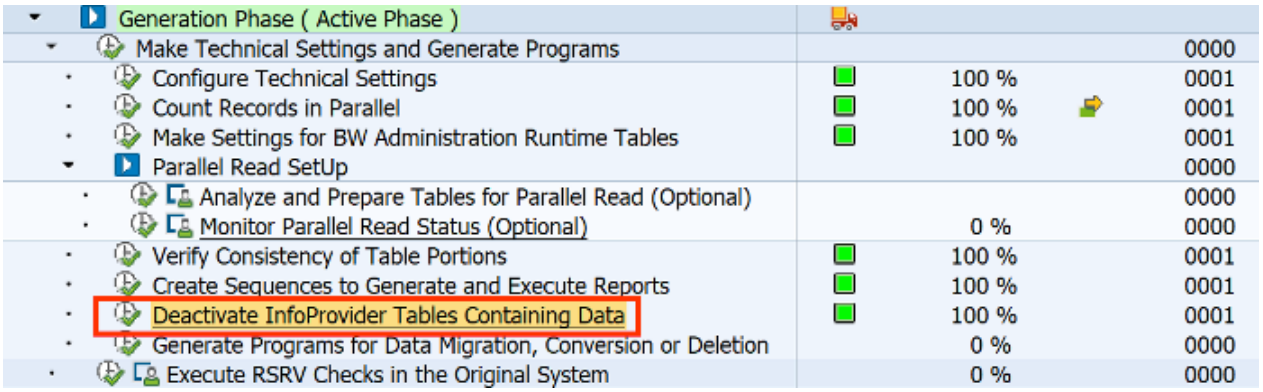


Figure 101: Deactivate InfoProvider Tables Containing Data

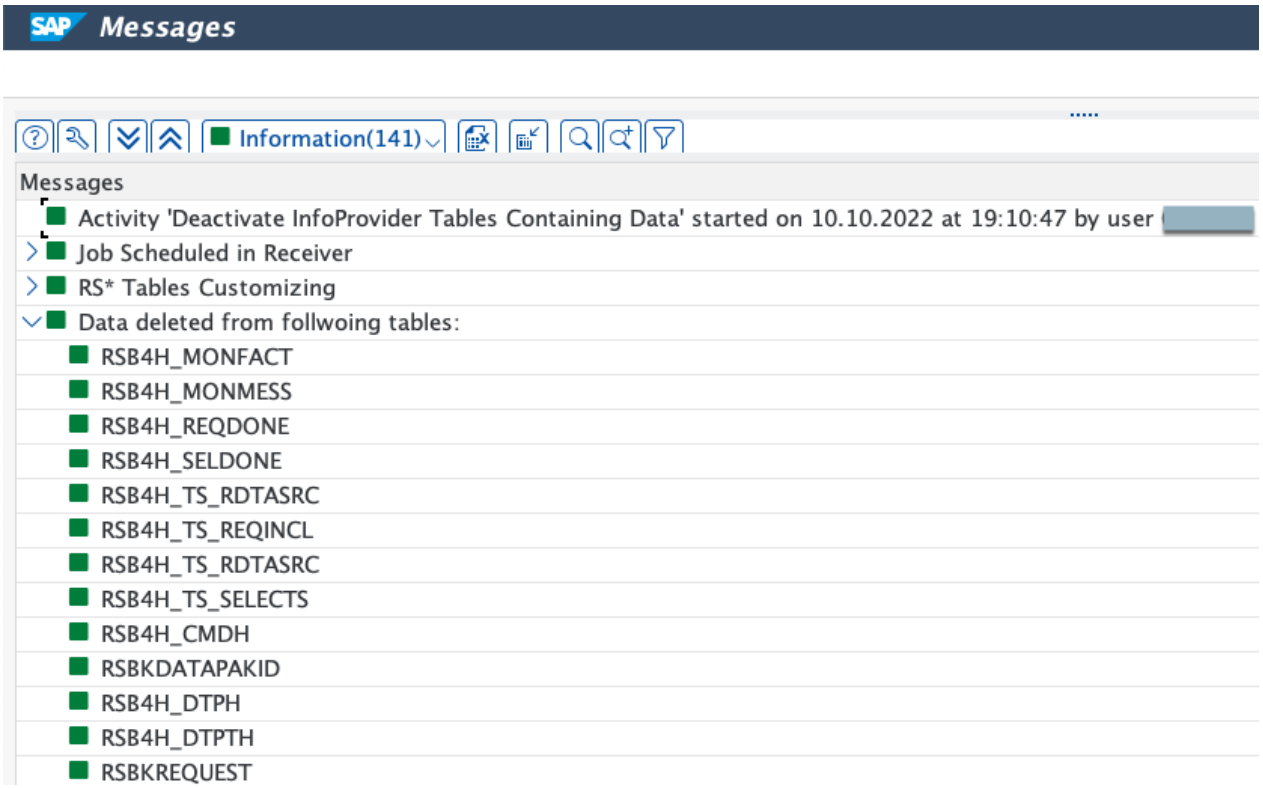


Figure 102: Deactivate InfoProvider Tables Containing Data - Logs

3.4.8 Generate Programs for Data Migration, Conversion or Deletion

This activity verifies the settings from all predecessor preparation activities and automatically generates ABAP programs to enable the data transfer for all relevant tables. This activity generates the relevant programs in the /1LT/BW_GEN_OBJ development class in the /1LT/BW namespace.

It collects the relevant technical settings stored in the CNVLTRL_TEC control table and generates the programs for each phase of the transfer process. It also carries out a syntax check of the generated programs.

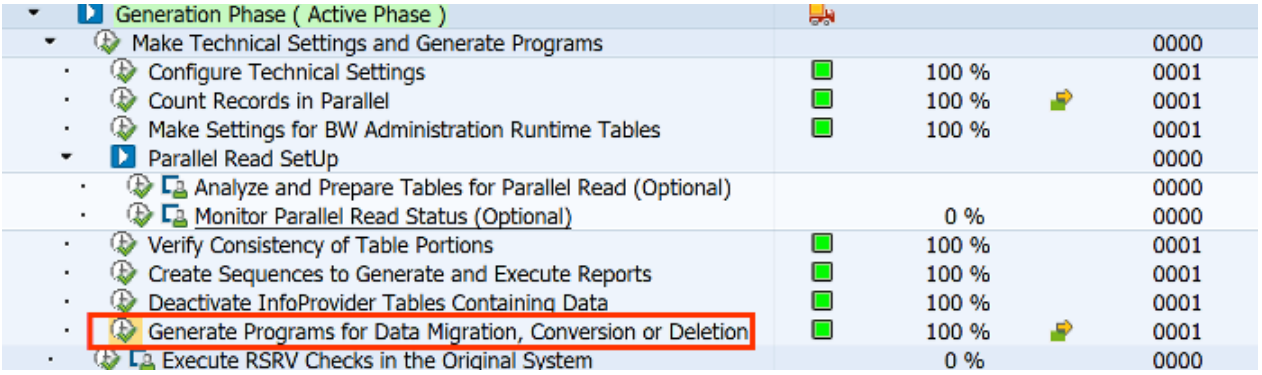


Figure 103: Generate Programs for Data Migration, Conversion or Deletion

One (1) program is generated for every task type of a table in scope. Those programs are sequentially executed by the tool considering that the data should be:

1. *Selected* in the sender system (read phase).
2. *Transferred* from the sender to the receiver system clusters (Move phase).
3. *Inserted* into the corresponding tables in the receiver system (Conversion phase).

Task types created in the sender system:

- **I_GEN_READ**: Data read generation programs.
- **I_EXC_READ:** Data read execution programs.
- **I_GENTASK-TRANSF**: Data transfer generation programs.
- **CLU_BEAM**: Data transfer execution programs.

Task types created in the receiver system:

- **I_GEN_CNV:** Data insertion generation programs.
- **CLU_CONVERT**: Data insertion execution programs.
- **CRT_TAB_REDIST**: Data redistribution across active nodes (for optimization).
- **DELTA_MERGE**: HANA operation to transfer data to the main storage to optimize memory consumption and performance.
- **MOVE_TO_ADSO:** Data transfer from the shadow InfoCube P Dimension tables (populated through the CLU_CONVERT associated tasks) in the receiver system into corresponding ADSOs tables. Only InfoCube P Dimension tables will have this task type associated.

- **MOVE_RSSM_RSPM**: Request information conversion (Request ID to TSN). Only the table RSBKCMDH will have this task type associated. This is the last table and task type to be executed in the transfer process.

You can check the reports generated for each table in scope in the CNVLTRL_TASK control table (existing in both sender and receiver systems). The REPID field contains the name of the generated program and the INTTYPE field indicates the task type associated with the given generated program.

An example for one of our sample tables - /BIC/ACPRODCOST1:

Sender system

Data Browser: Table CNVLTRL_TASK Select Entries 438

RUNID	TABNAME	SEQNO	TASKNO	STATUS	XPAR	CLUNAME	INTTYPE	DESCRIPTION	REPID
9000V	/BIC/ACPRODCOST1	100	000100	DONE			I_GEN_READ	Generate Read Program	/1LT/BW9000VR_BIC_ACPRODCOST1
9000V	/BIC/ACPRODCOST1	210	000100	DONE			I_EXC_READ	Execute Read Program	/1LT/BW9000VR_BIC_ACPRODCOST1
9000V	/BIC/ACPRODCOST1	210	000200	DONE			I_GENTASK-TRANSF	Generate Transfer Tasks	
9000V	/BIC/ACPRODCOST1	220	000001	DONE	X	9000V_.000000./BIC/ACPRODCOST1_00000	CLU_BEAM	Data transfer	

Figure 104: CNVLTRL_TASK control table (sender system)

Receiver system

Data Browser: Table CNVLTRL_TASK Select Entries 438(Customer Connection Productive)

RUNID	TABNAME	SEQNO	TASKNO	STATUS	XPAR	CLUNAME	INTTYPE	DESCRIPTION	REPID	COUNT_READ_ROW
9000V	/BIC/ACPRODCOST1	100	000300	DONE			I_GEN_CNV	Generate Conversion Program	/1LT/BW9000VC_BIC_ACPRODCOST1	00000000000000000000
9000V	/BIC/ACPRODCOST1	219	000001	DONE			CRT_TAB_REDIST	Table Re-Distribution		00000000000051761812
9000V	/BIC/ACPRODCOST1	220	000001	DONE	X	9000V_.000000./BIC/ACPRODCOST1_00000	CLU_CONVERT	unpack cluster	/1LT/BW9000VC_BIC_ACPRODCOST1	00000000000000119157

Figure 105: CNVLTRL_TASK control table (receiver system)

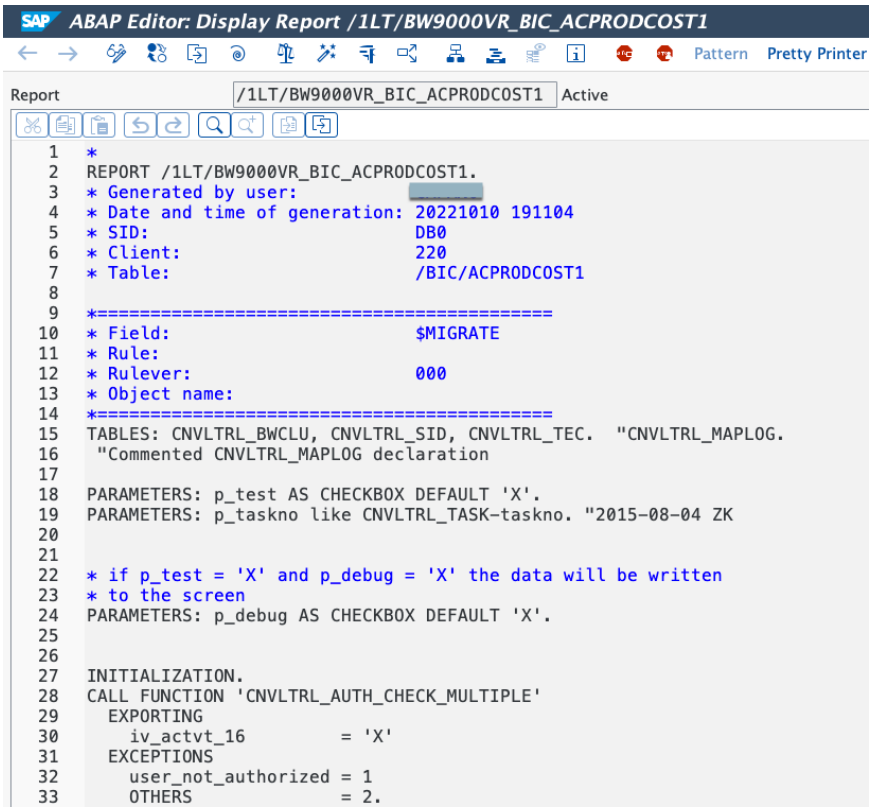


Figure 106: Read execution program for the table /BIC/ACPRODCOST1

⇒ Tip: For the read execution task type (I_EXC_READ) of a certain table, the number of tasks generated for it will depend on if the parallel read settings were used for that particular table. For example, if table A was set for parallel read and two (2) portions were created for it, this means there will be two (2) tasks for the read execution program of table A and you will have the option to process the read of this table with 2 background jobs in parallel.

3.4.9 Execute RSRV Checks in the Original System

This activity creates, executes or saves an Analysis and Repair of BW Objects (RSRV) analysis package in the standard RSRV transaction. You can load and execute the RSRV analysis package in the RSRV transaction to validate the consistency of the BW objects included in the current run.

Those checks (before and after images) are important to compare the status of the objects in the sender and receiver systems:

a. If there were inconsistencies in the sender system prior to the Remote Conversion. This is quite relevant for master data and InfoCubes. Already existing inconsistencies (i.e. missing SIDs in where-used master data or missing key dimensions) will be a showstopper for the MOVE_TO_ADSO task in the receiver system.

b. If there are inconsistencies in the receiver system which did not exist before (i.e. the receiver system already contained some master data). This would mean the inconsistency was caused by transfer and should be repaired.

When the receiver system already contains master data, the Remote Conversion toolset will need to manage the creation of SIDs to map and harmonize the insertion of this new master data (so called SID mapping). In this case, it is expected that there will be merges of values and the affected master data will be inconsistent with the RSRV check after the data transfer.

⇒ Tips:

- The checks can include master data, hierarchy, line item, dimension, unused DIMIDs and duplicates in dimensions. It is recommended to only select master data as here is where the most common inconsistencies happen. Doing the checks for other tables other than master data (like InfoProviders) may take very long and this can negatively impact the overall runtimes. Therefore, from time/effort perspective, it is recommended to save/execute the RSRV packages only for master data and in case inconsistent InfoProviders are identified after the Remote Conversion (this will pop up during the MOVE_TO_ADSO tasks for the P dimension table) you can run RSRV on demand for those InfoProviders.

- Master data inconsistencies can be repaired through the RSRV transaction or the standard program RSDMD_CHECKPRG_ALL.

- Select “Save package without execution” and “Master data”. The activity will finish in green and afterwards you can load/execute the package.

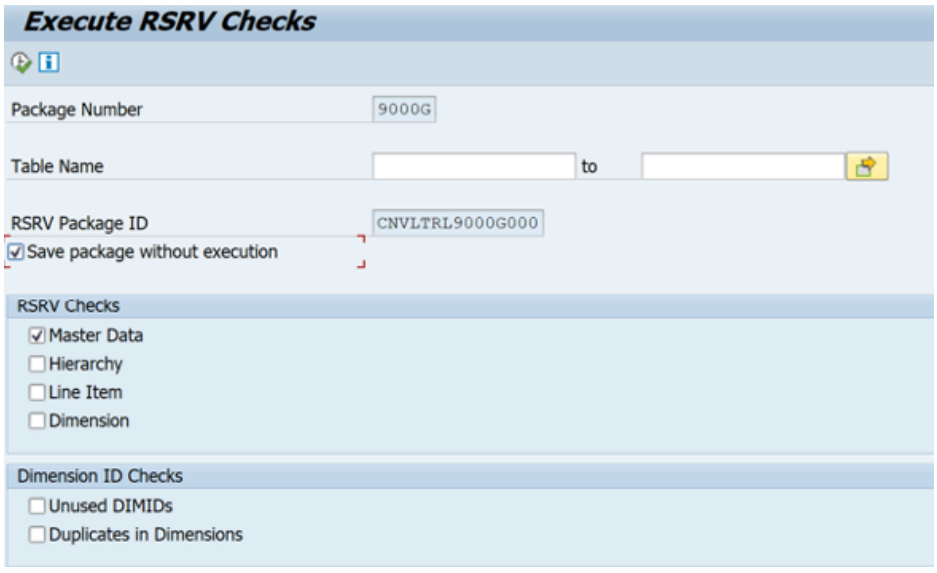


Figure 107: Execute RSRV Checks in the Original System - Parameters

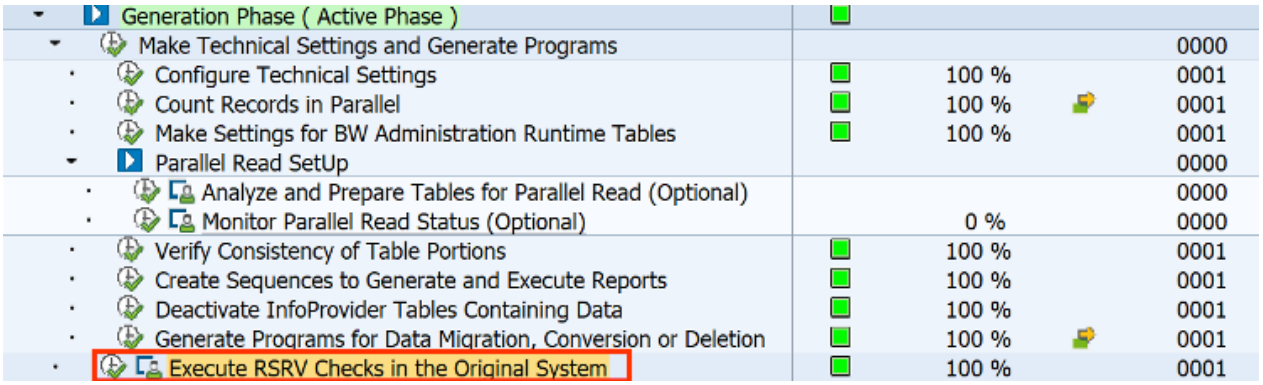


Figure 108: Execute RSRV Checks in the Original System

- Go to transaction RSRV in the sender system -> Test Package -> Load.

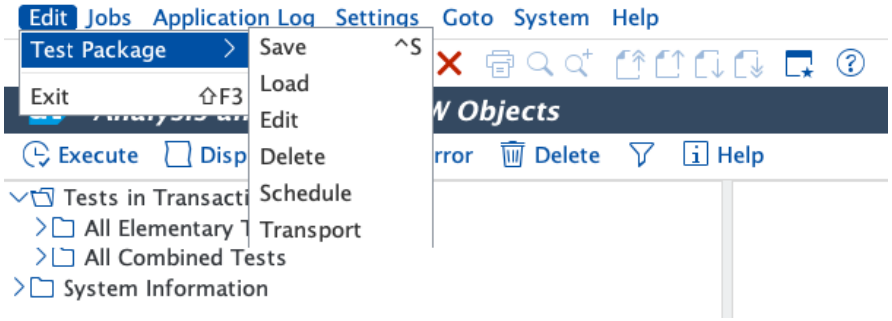


Figure 109: RSRV - Load Test Package in the sender system (1)

- Select the package created in the previous step (CNVLTRL9000G000).

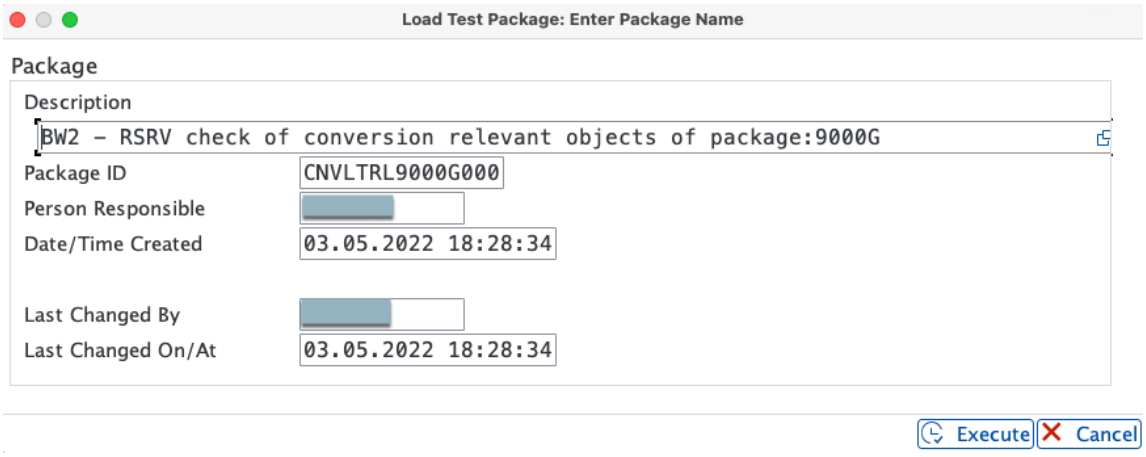


Figure 110: RSRV - Load Test Package in the sender system (2)

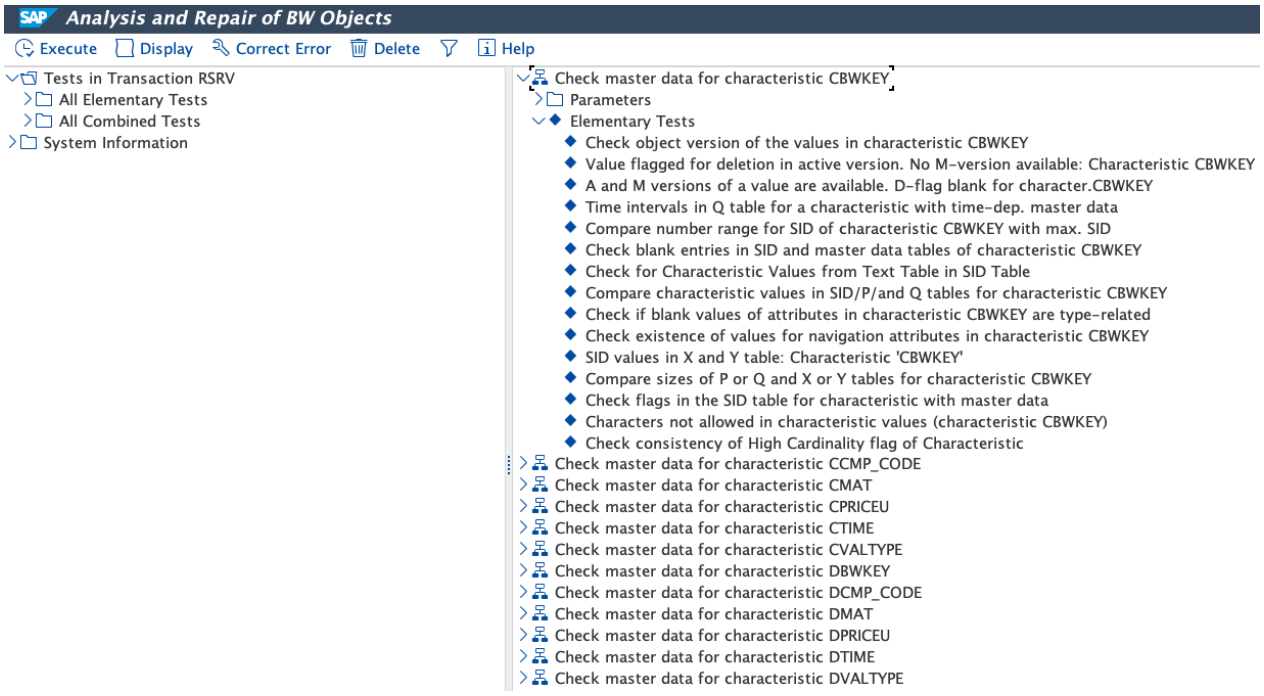


Figure 111: RSRV - Execute Test Package in the sender system (1)

- Execute the package. The following master data are the InfoObjects used in our sample.

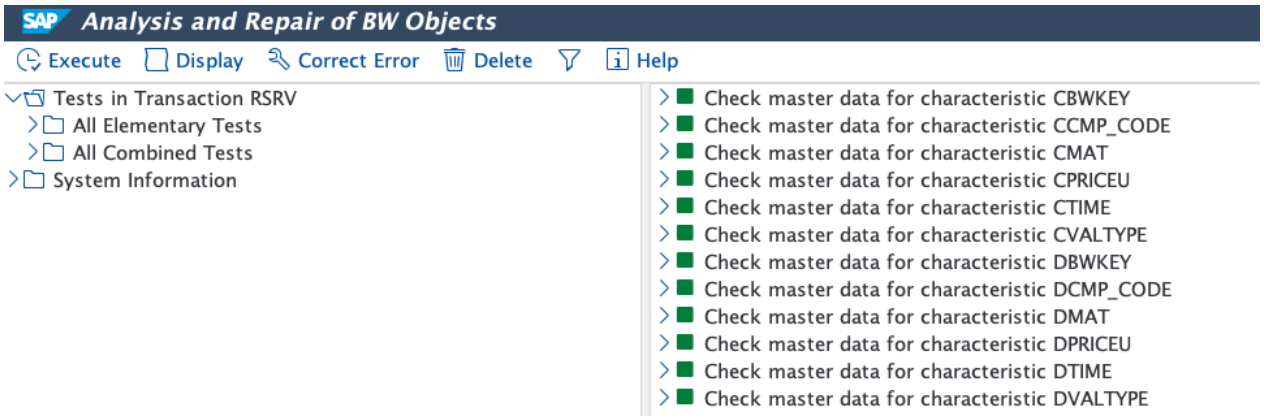


Figure 112: RSRV - Execute Test Package in the sender system (2)

3.5 Preprocessing Programs During System Lock

This group of activities is the last one before the Migration phase starts.

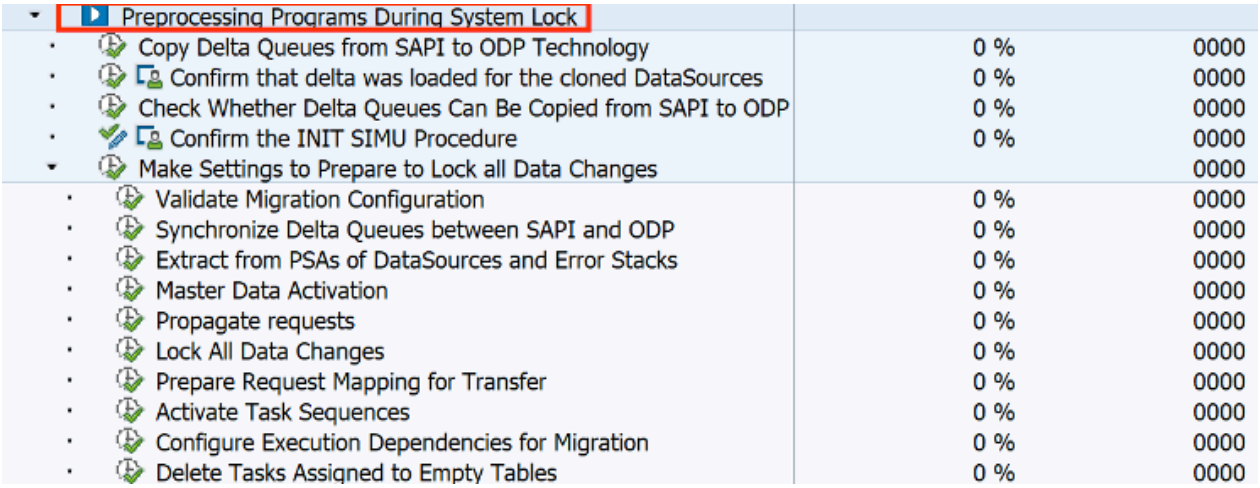


Figure 113: Preprocessing Programs During System Lock

3.5.1 Copy Delta Queues from SAPI to ODP Technology

This activity clones the delta queues in the connected sender system in order to switch to ODP (Operational Data Provisioning) technology for data extraction.

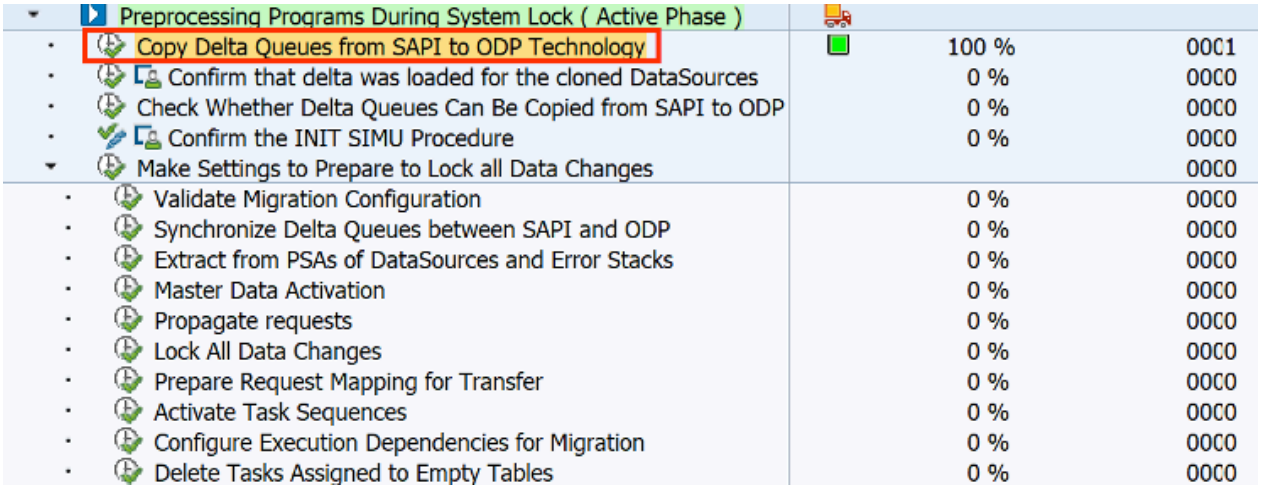


Figure 114: Copy Delta Queues from SAPI to ODP Technology

An ODP Delta Queue is called "subscription". A subscription has a "subscriber", which is usually a BW system. A subscription has a "subscriber process", which is either a BW DataSource and Source System (for InfoPackage loads) or a DTP.

- Original Object is an **InfoPackage for a SAPI** (Service API) Source System: The SAPI Delta Queue is copied into an ODP Subscription with the new subscriber.

- Original Object is an **InfoPackage for an ODP** Source System: The ODP-subscription for the target process "DataSource and Source System" is copied into an ODP-Subscription for the target Process "DTP" with the new subscriber.

- Original Object is a **DTP for an ODP** source system: The ODP-subscription for the target process "DTP" is copied into an ODP-Subscription for the target Process "DTP" with the new subscriber.

After you have executed this activity, you have to request new delta for all DataSources in scope and you also have to synchronize the delta queues in the connected BW sender system (subsequent step "Synchronize Delta Queues between SAPI and ODP"). This next activity will check whether you have indeed loaded the deltas for all cloned DataSources.

This activity locks the creation of new delta initialization requests until the Task List has been completed. This means the BW sender system will not be able to create a delta initialization request until the activity "Unlock Loading and Data Target Changes" has been executed.

Our sample does not make use of any delta DataSources.

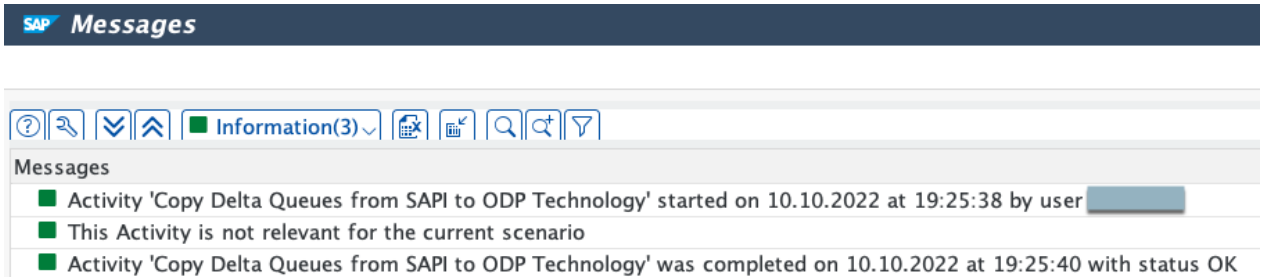


Figure 115: Copy Delta Queues from SAPI to ODP Technology – Logs

⇒ Tips:

- Technically it is possible to execute this activity with the BW sender system being operational in terms of data loads, however it is recommended to have a data load freeze period at this point because this can speed up the duration of the copy and synchronization of the delta queues. For a productive run, this activity is usually executed during a weekend or a day before, having the data load freeze window being previously agreed within the project.

- The most frequent issues are related to old data loads (ended in error or unfinished). You should repair them as usual for normal operations. You can also eventually delete unnecessary delta initializations.

3.5.2 Confirm that delta was loaded for the cloned DataSources

This activity is a placeholder to trigger delta requests for all cloned delta queues. Once a new delta request has been created for all cloned delta queues, you can confirm this activity.

The delta requests can be triggered by manually executing the corresponding InfoPackages and DTPs or by using process chains (this is not managed by the tool). Please not that these last deltas are needed only up to the PSAs – no needed to trigger deltas to upward layers/data marts.

The completeness of this activity will only be successfully if delta requests for all relevant DataSources have been processed correctly in the source systems.

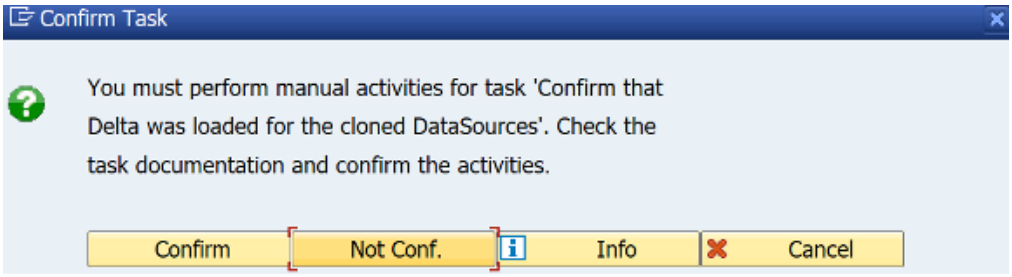


Figure 116: Confirm that delta was loaded for the cloned DataSources (1)

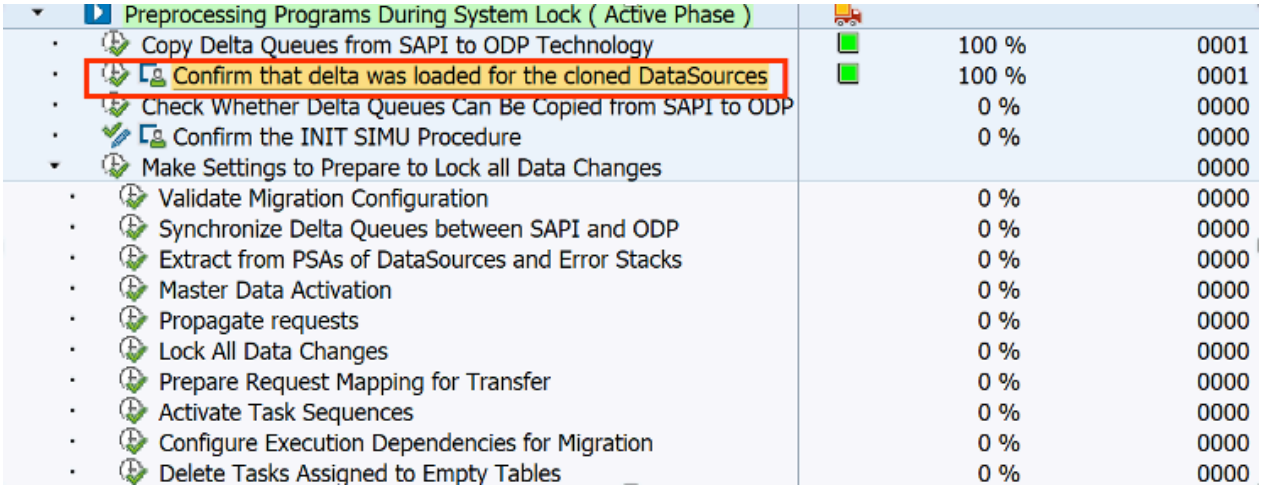


Figure 117: Confirm that delta was loaded for the cloned DataSources (2)

⇒ Note: This activity was not relevant for our sample.

3.5.3 Confirm the INIT SIMU Procedure

In case the delta queue cloning is not possible for a DataSource, this will be indicated in the predecessor activities of this phase. For all such DataSources, you need to manually simulate the initialization of the delta procedure without actual data transfer.

Execute the activity and confirm it:

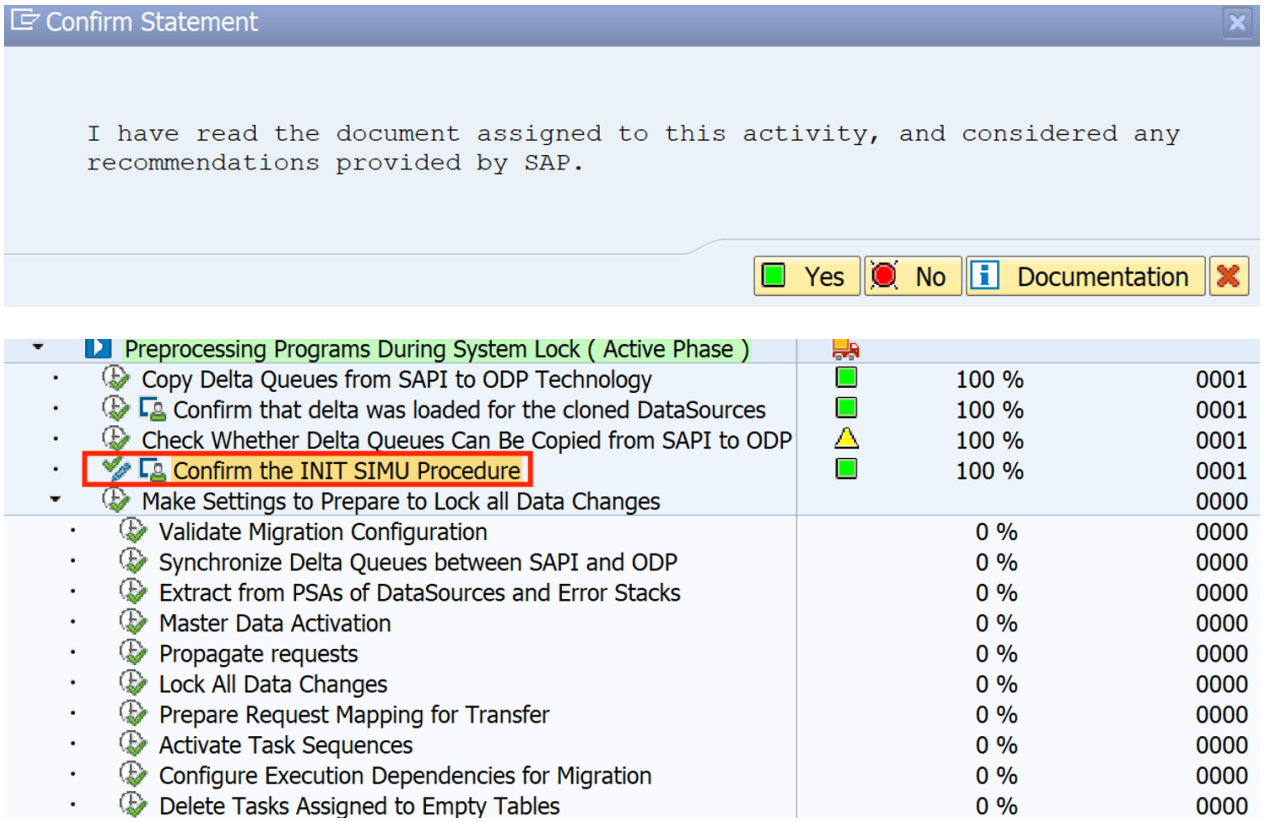


Figure 118: Confirm the INIT SIMU Procedure

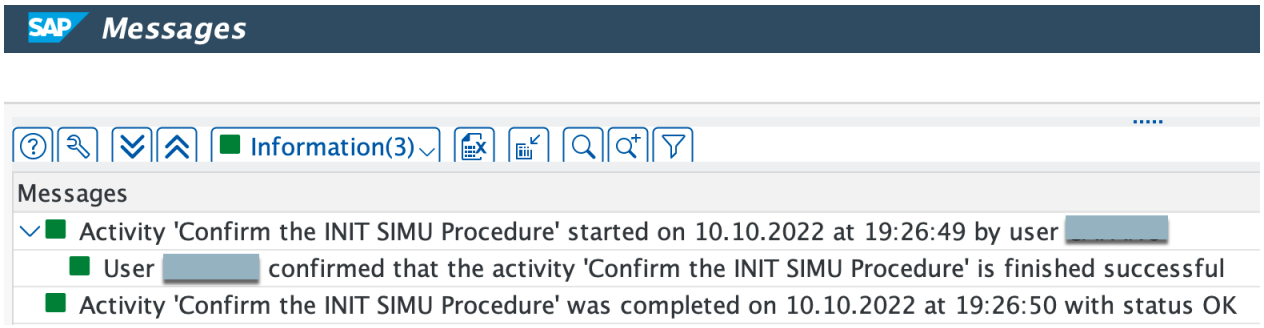


Figure 119: Confirm the INIT SIMU Procedure – Logs

⇒ Note: This activity was not relevant for our sample.

3.5.4 Make Settings to Prepare to Lock all Data Changes

This is an activities group and you have the option to either run it all together (mass execution) or each activity individually (in this case you should execute the activities in the define sequence). When using the mass execution, each activity automatically starts its successor activity, no further user interaction is required.

If you want to execute any of the dialog activities, go to the extended view and execute each activity sequentially. If you use the mass execution, the dialog activities under this activities group are skipped.

If the activities group terminates or ends with errors or warnings, activate the extended view of the process tree. You can now view all activities that belong to the group activity and their statuses.

3.5.4.1 Validate Migration Configuration

This activity checks if the configuration of the BW objects in scope is consistent – by validating whether each participating BW object in the sender system is correctly mapped to a corresponding object in the receiver system.

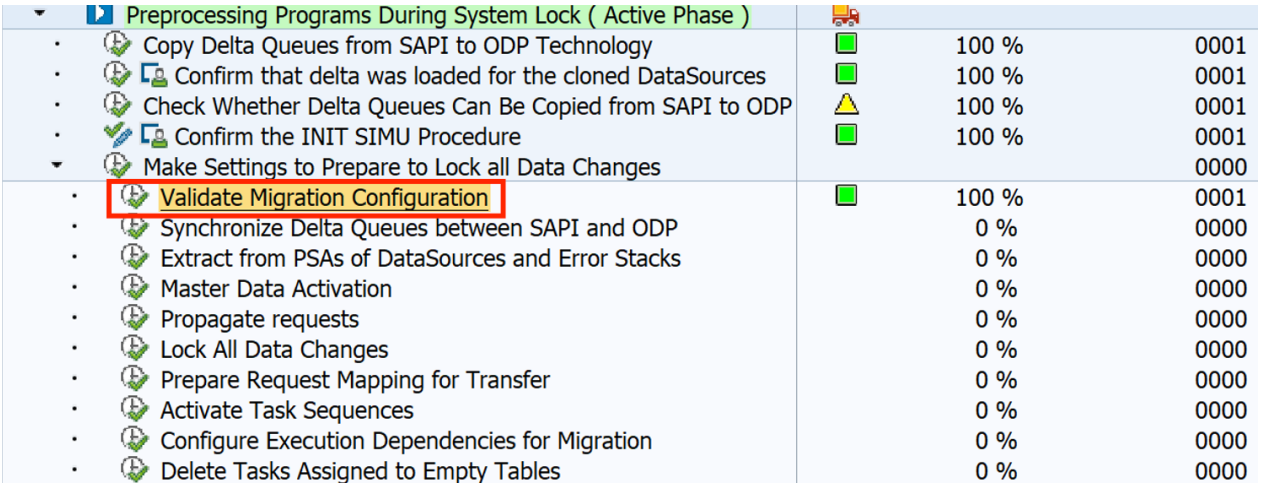


Figure 120: Validate Migration Configuration

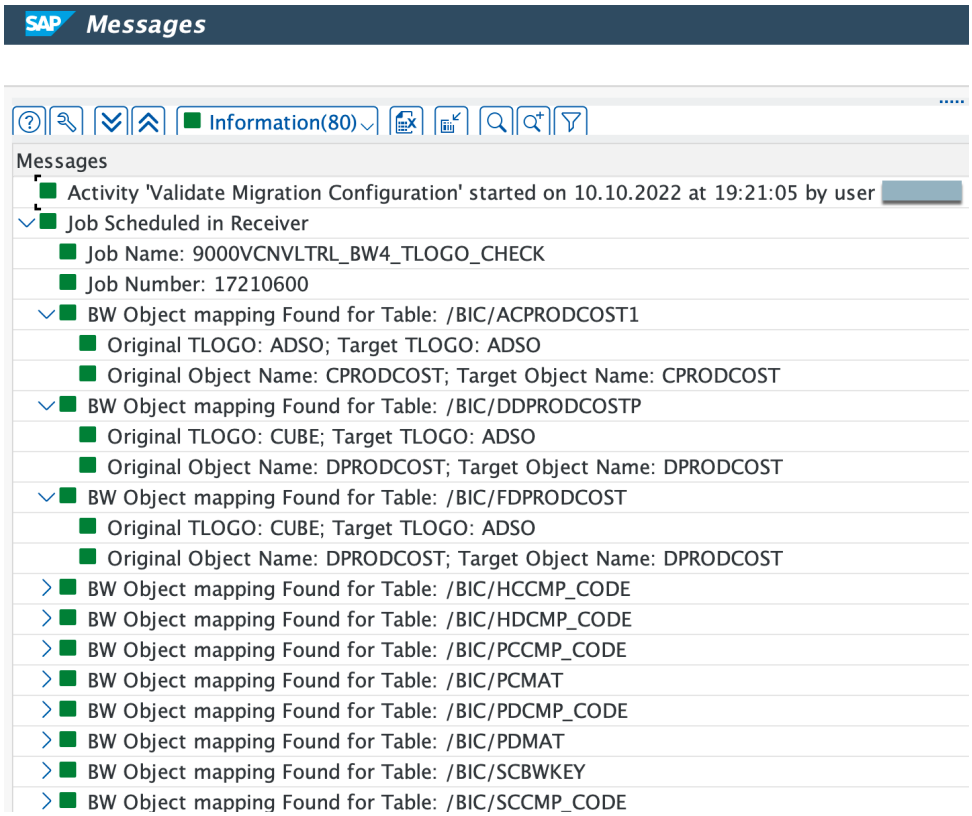


Figure 121: Validate Migration Configuration - Logs

3.5.4.2 Synchronize Delta Queues between SAPI and ODP

This activity is used to synchronize the cloned delta queues in the connected BW sender system.

The creation of delta and repeat requests are locked. This means it is no longer possible to load deltas for the DataSources in scope until “Unlock Loading and Data Target Changes” (last task in the Task List Run).

A check is performed afterwards to determine whether delta requests have been triggered and completed. If this is not the case, an error is raised. When it happens, you need to check the request status in transaction RSRQ and create additional delta requests where necessary (or repeat requests) for the DataSources mentioned in the error message.

Finally, the queues of the original and new delta queues and subscriptions are synchronized (for SAPI to ODP cloning). This means that the data which has been loaded into the “old DataSource” after cloning is marked as read for the “new DataSource” and discarded so that it is not loaded twice via the new DataSource. This is necessary in order to have a sharp cut between data loaded and data not yet loaded into the receiver system.

For ODP to ODP, the cloning is done within this activity (as it works on TSNs and the sharp cut between loaded data and data that has not been loaded yet can also be guaranteed without intermediate loading).

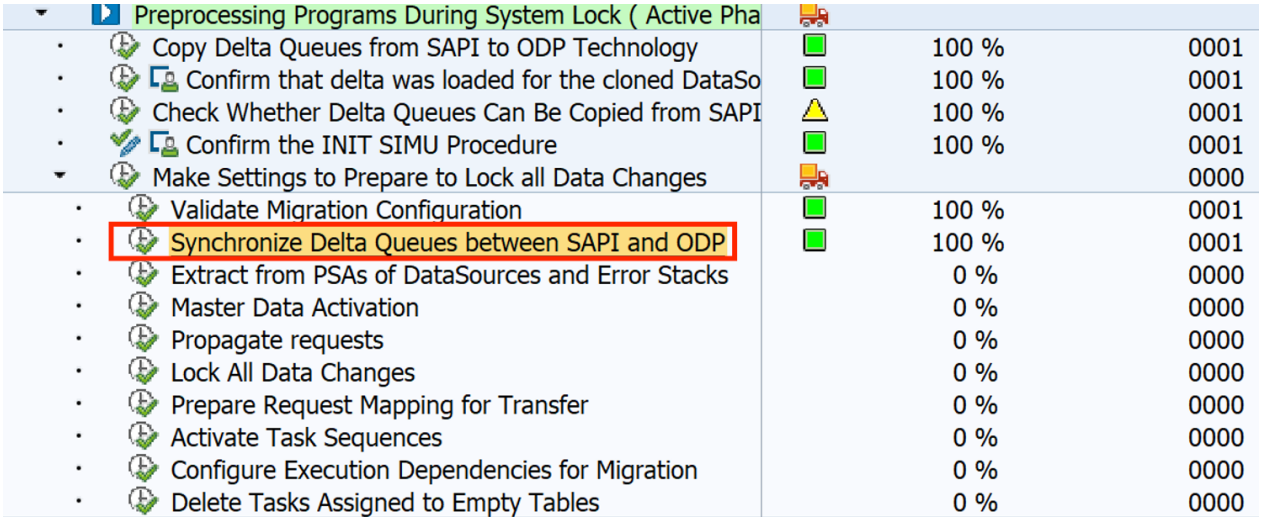


Figure 122: Synchronize Delta Queues between SAPI and ODP

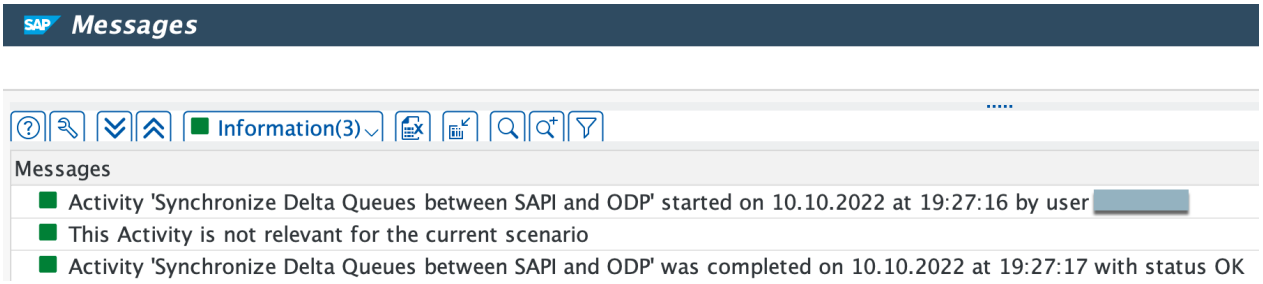


Figure 123: Synchronize Delta Queues between SAPI and ODP – Logs

⇒ Note: This activity was not relevant for our sample.

3.5.4.3 Extract from PSAs of DataSources and Error Stacks

This activity performs the necessary preparations for the InfoProviders in scope:

a. Execute all delta DTPs in scope by extracting from the PSA of a DataSource if this PSA will not be transferred as a field-based ADSO, this means the PSA-ADSOs without the 'transfer flag' in 'Define Object Mapping and Store Object List'. In this case, all requests from the PSA must be propagated into InfoProviders by delta DTPs.

b. Execute error DTPs (create if necessary) for all non-empty error stacks. Even if the error DTP requests fail, the error stack will be emptied. The error stacks must be empty, as they are not transferred.

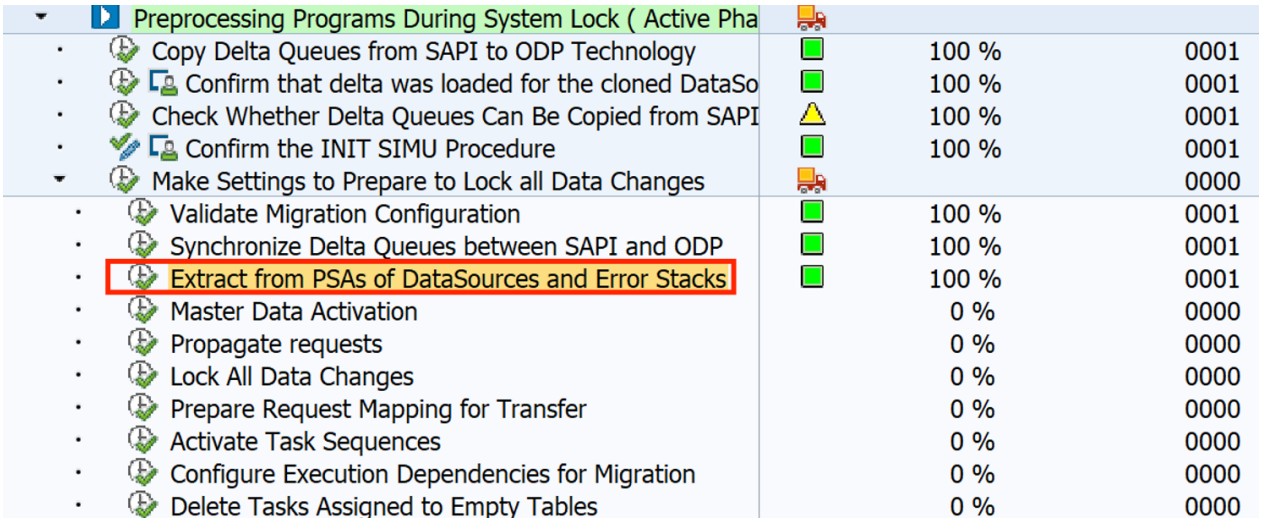


Figure 124: Extract from PSAs of DataSources and Error Stacks

⇒ Note: This activity was not relevant for our sample.

3.5.4.4 Master Data Activation

This task will do the necessary preparations for master data activation.

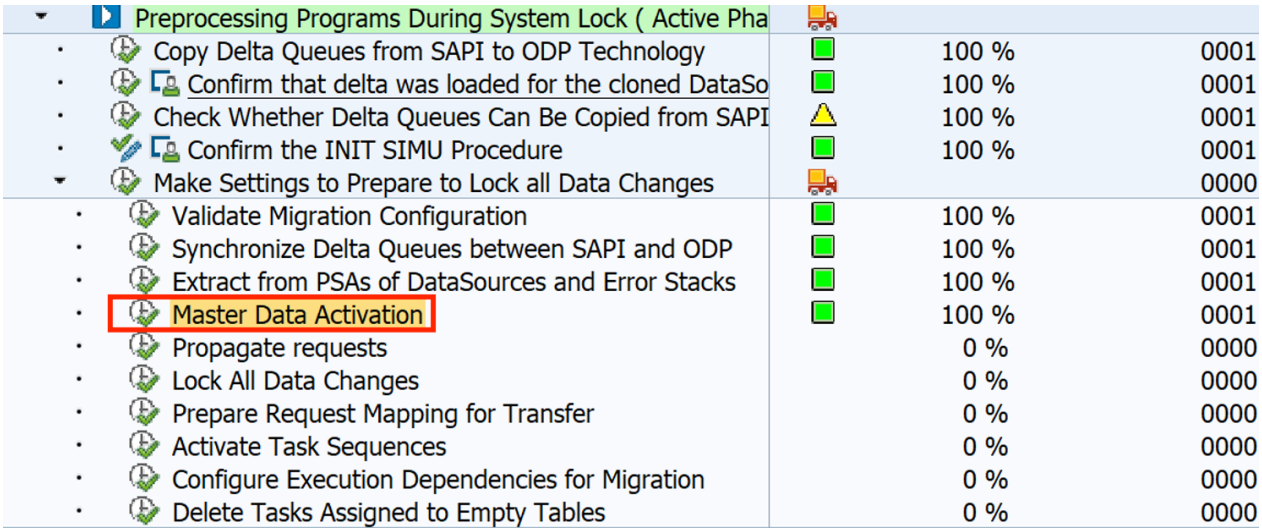


Figure 125: Master Data Activation

3.5.4.5 Propagate requests

This activity performs request propagation with the following logic:

a. Collection of relevant DTPs from the scope (the following conditions must be met: 1. Delta DTPs & Delta Status Active; 2. Source Object of the DTP is part of the scope list; 3. Supported object type (TLOGO) for the source/sender object: CUBE, MPRO, LPOA, ODSO).

b. Creation of temporary error DTPs for all relevant DTPs (only if the DTP has set the error handling to on, and no error DTPs exist).

c. Activation of non-activated requests of standard ODSOs prior to the request propagation.

d. Automatic request propagation (execution of the relevant DTPs until no more new data is available).

e. Roll-up and compression of InfoCubes.

f. Cleanup activities (e.g. deletion of temporarily created error DTPs).

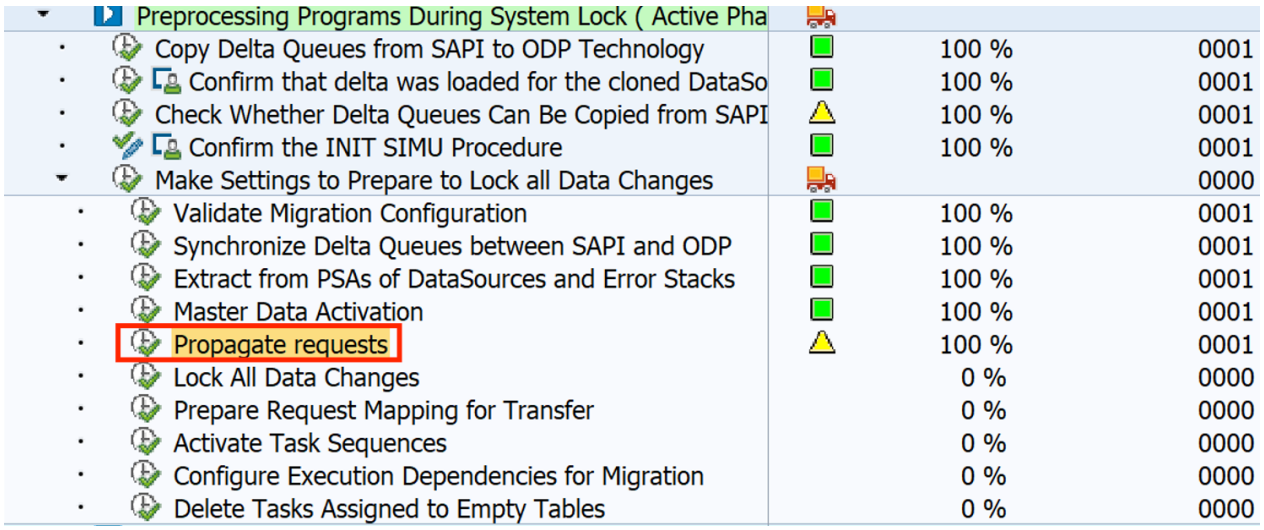


Figure 126: Propagate Requests

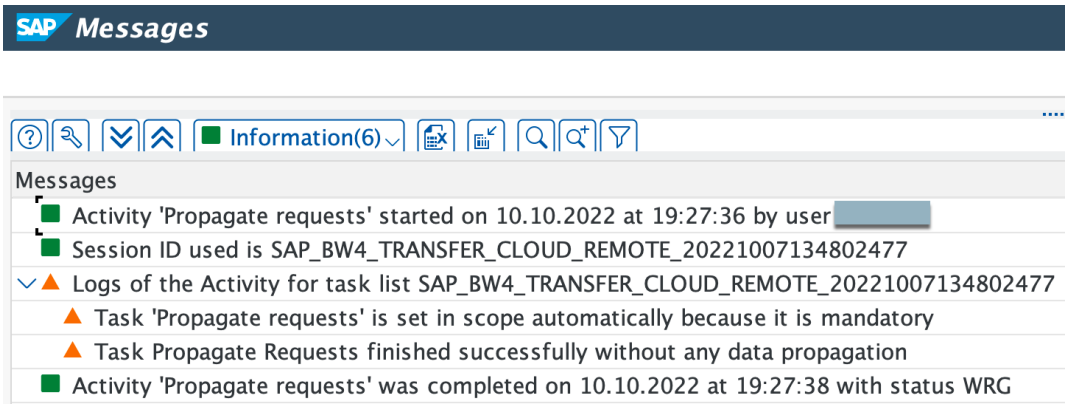


Figure 127: Propagate Requests – Logs

⇒ Note: This activity was not relevant for our sample.

⇒ Tips:

- Check SAP Note 2982341. By increasing the parameter 'WAIT' for the table RSNEWTTADMIN, it is possible to parallelize the 'Propagate request' task with more than 1 work process. This should speed up the completeness of the task. The note indicates '10' as the max parallel degree but it is possible to set a higher number if there are available BTC work process in the system.

- Roll-up and Compression of InfoCubes – the 'Propagate requests' task will automatically manage the compression of InfoCubes, for the ones which were not compressed earlier. All InfoCubes must be compressed, only the ones having DTPs with a DataSource as a source can have up to 20 uncompressed requests.

- It's recommended to have the InfoCubes compressions done at earlier stages to speed up the runtime of this task. In case table RSNEWTTADMIN is already maintained (as described above), the parallel degree is taken from the value maintained, otherwise the parallel degree is always calculated automatically.

3.5.4.6 Lock All Data Changes

This activity locks all InfoProviders in scope against any data changes. The data in the InfoProviders is then ready for copy into the new ADSO objects (this is a prerequisite for the MOVE_TO_ADSO tasks).

The request mapping from RSSM (Request ID) to RSPM (TSN) request management also requires that no data and request operations are performed anymore, which is ensured by this activity.

- It checks that the data of all DataStore objects is activated
- It checks that no InfoCube in scope is in planning mode
- It checks that all archiving runs in scope are finished

For ADSOs, data changes are locked via the RSPM framework (new request management).

For all other objects (including DataSources), locking is ensured by entering corresponding entries in the table RSMIGROBJ in the sender system.

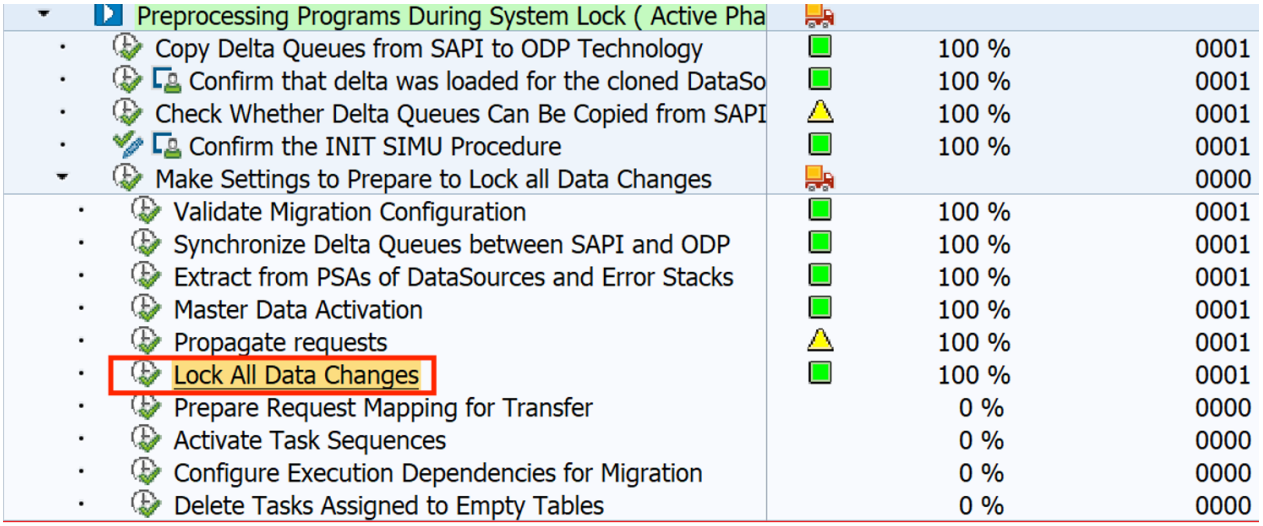


Figure 128: Lock All Data Changes

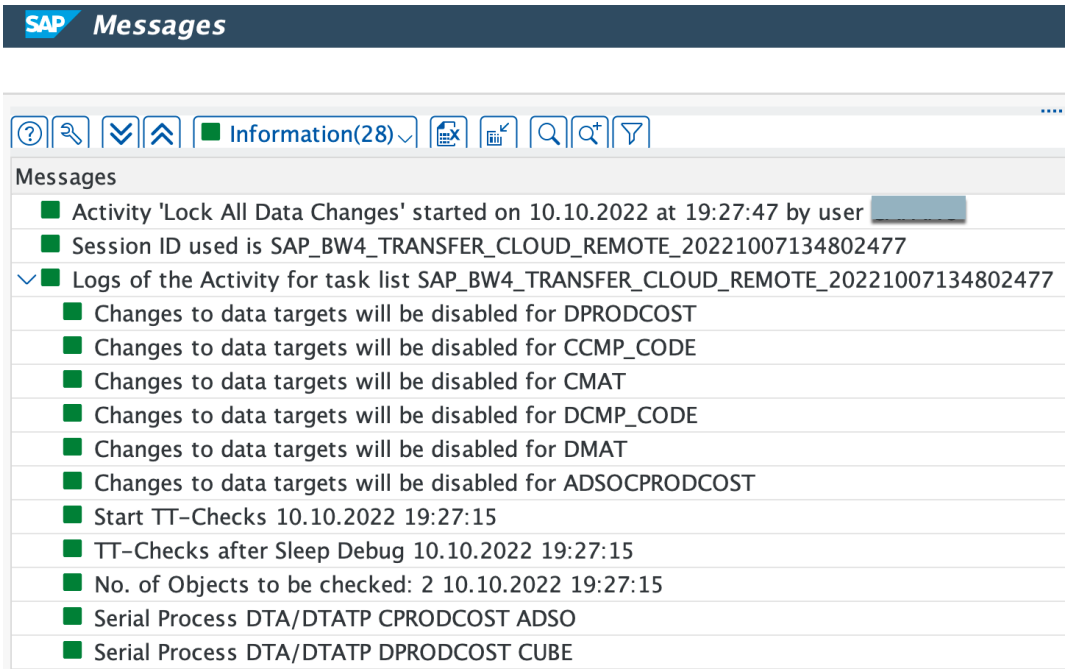


Figure 129: Lock All Data Changes – Logs

⇒ Tip: In case of error messages, it might not be possible to retrieve the lock because other processes are currently active. In this case, wait until the other process is finished, and

make sure that the data is activated in the corresponding InfoProviders. Then resume this task.

3.5.4.7 Prepare Request Mapping for Transfer

This activity fills the mapping table RSB4HPREQUESTMAP between Request GUID, Request SID and TSN for all request types of an InfoProvider. The request mapping table is used later for data transfer as well as for creating the new TSN requests in the RSPM request management.

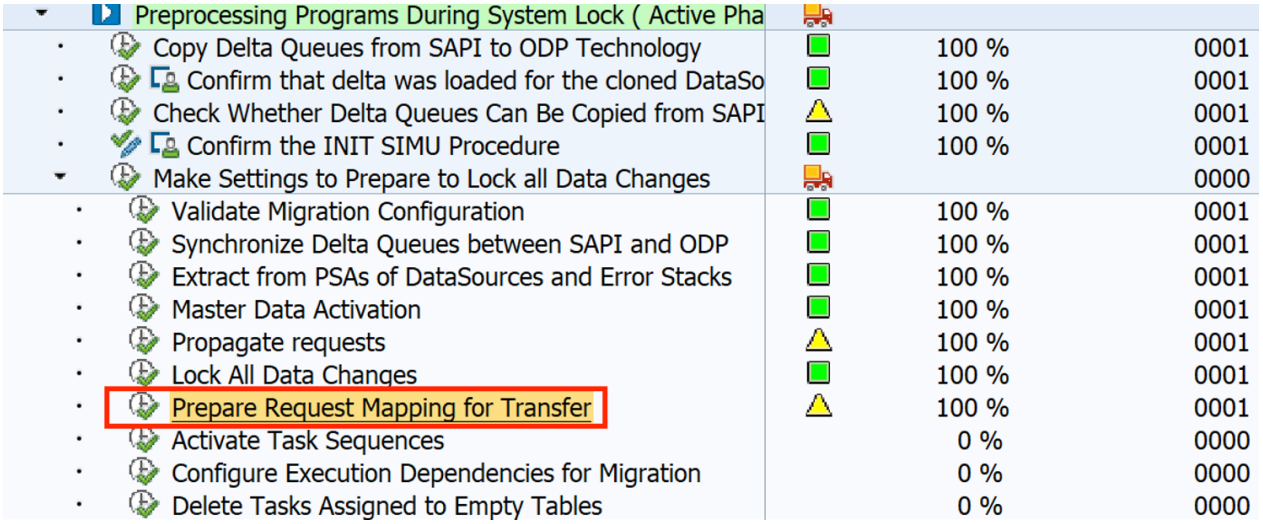


Figure 130: Prepare Request Mapping for Transfer

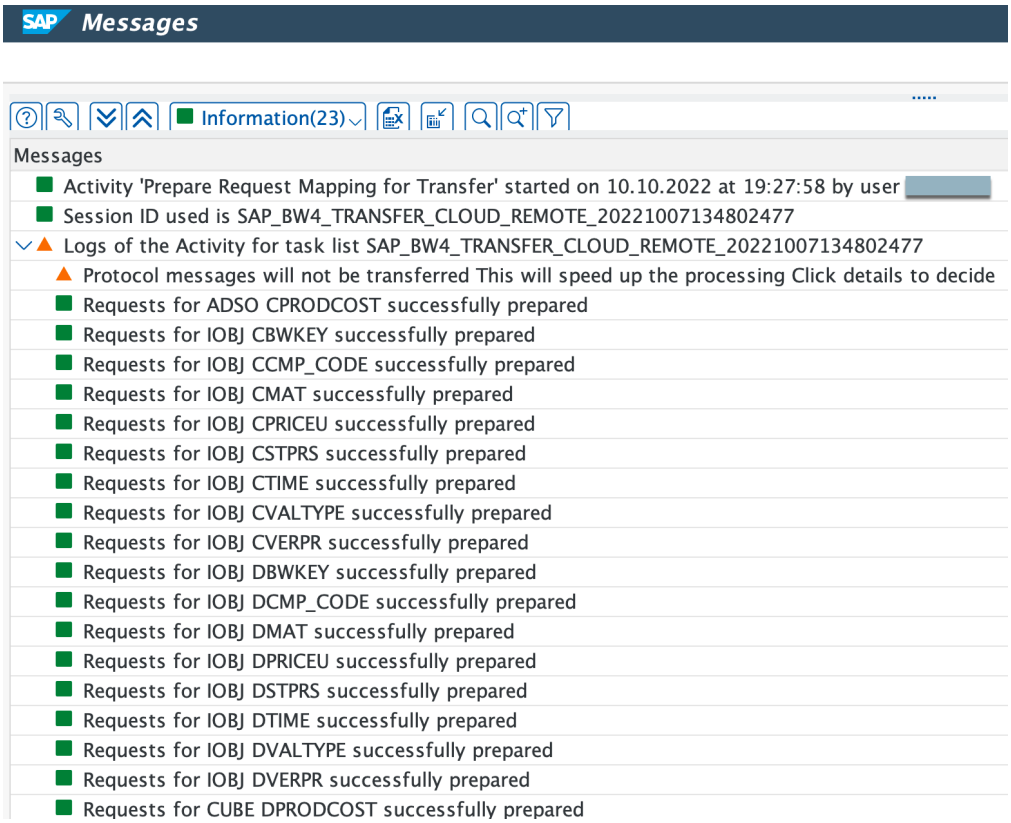


Figure 131: Prepare Request Mapping for Transfer - Logs

Logic: For each InfoProvider and DataSource in scope, a parallel process is started, which determines all requests of this object and creates a dummy TSN for it. This mapping between old and new requests is stored in table RSB4HPREQUESTMAP. The logic of how the TSN is composed is documented in comments of the function module RS_B4HTRF_API_REQUEST_MAP. For most requests, the TSN starts with 19000101000000 + (request SID as seconds). This will lead to TSNs between 1900 and 1970, thus clearly distinguishing them from requests created natively in the new RSPM request management.

3.5.4.8 Activate Task Sequences

This activity activates the task sequences before data migration to avoid accidental initiation of scheduled tasks.

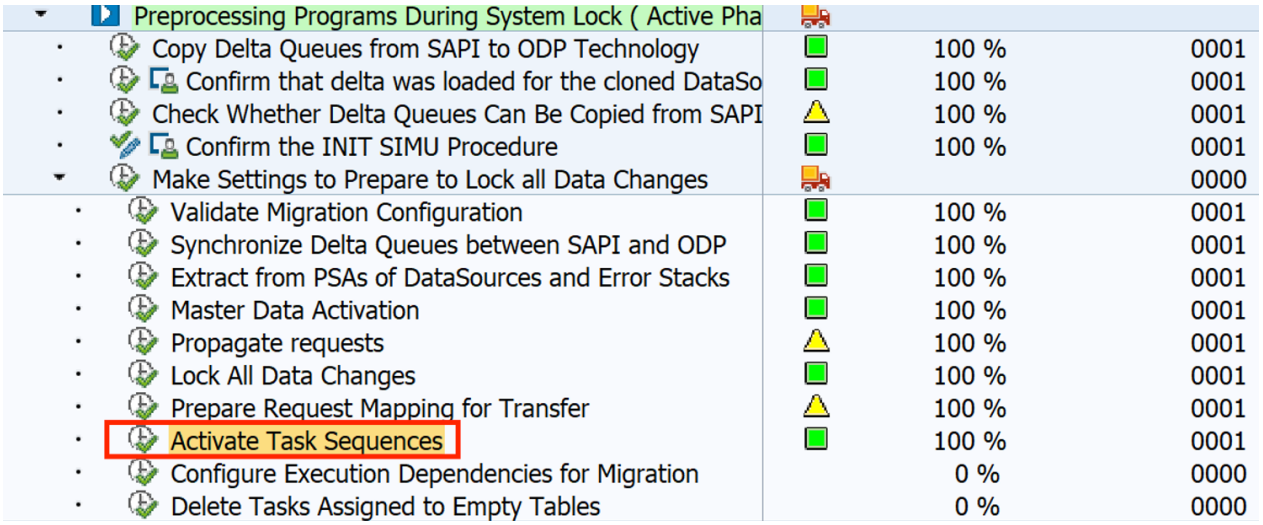


Figure 132: Activate Task Sequences

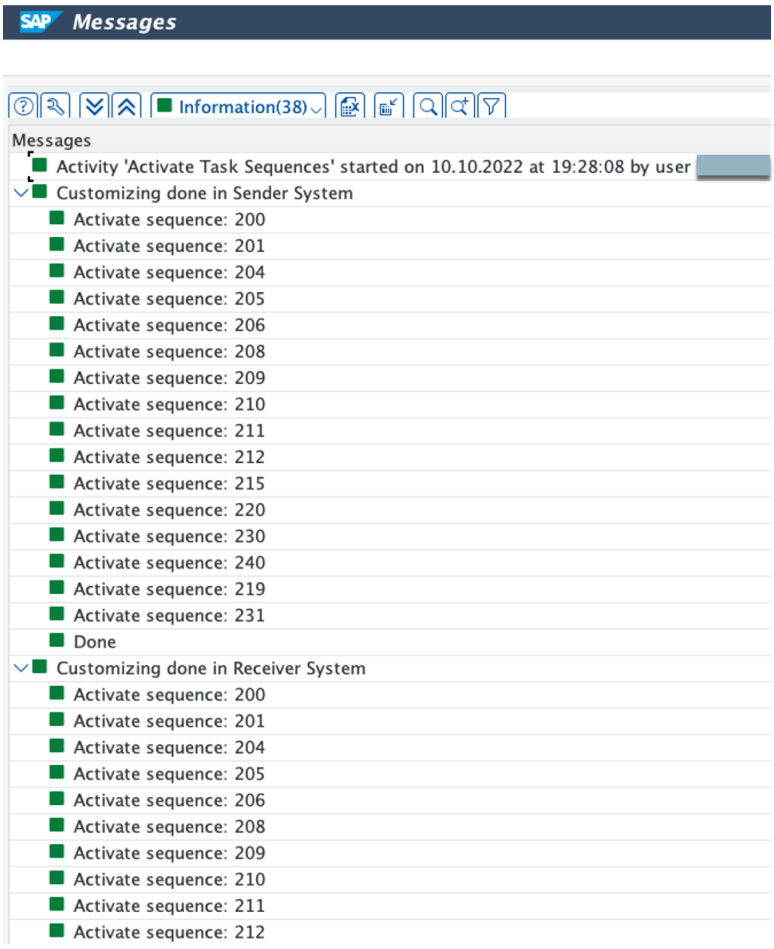


Figure 133: Activate Task Sequences - Logs

3.5.4.9 Configure Execution Dependencies for Migration

This activity configures tables, tasks, and sequences dependencies for the data transfer.

It generates the dependency information and saves the results in the CNVLTRL_TAB_SEQ control table in both sender and receiver systems.

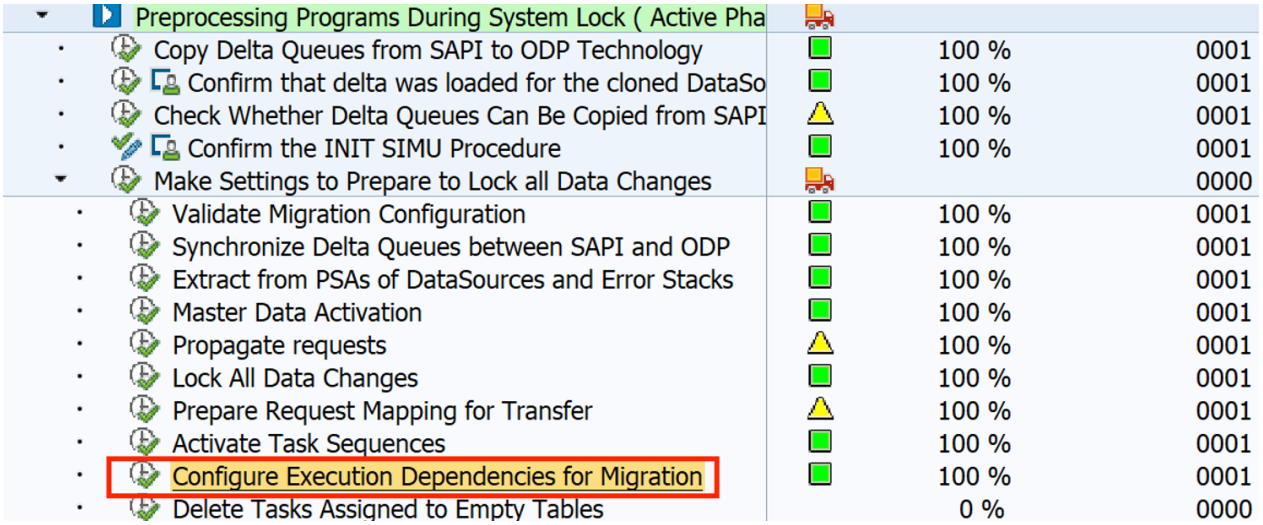


Figure 134: Configure Execution Dependencies for Migration

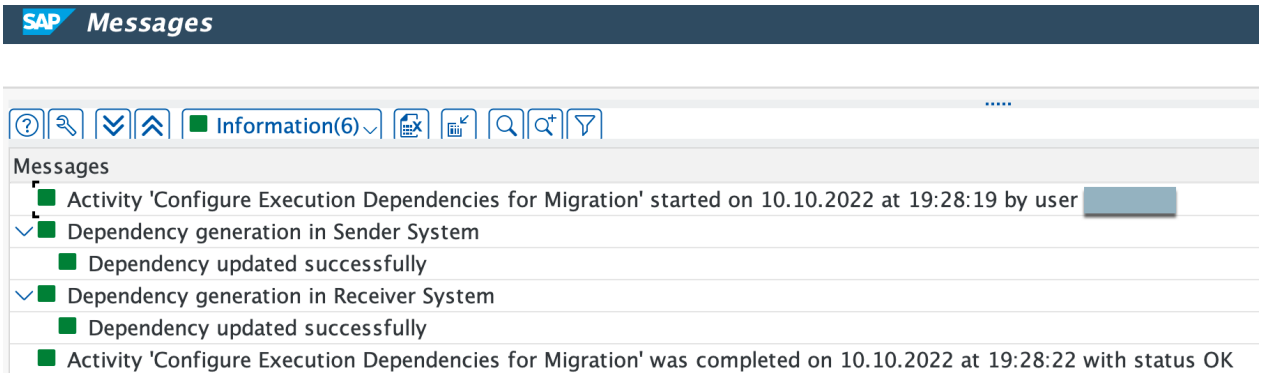


Figure 135: Configure Execution Dependencies for Migration - Logs

3.5.4.10 Delete Tasks Assigned to Empty Tables

This activity identifies and deletes the tasks of empty tables in the sender system (CNVLTRL_TASK control table) and tables that are set as inactive in the receiver system.

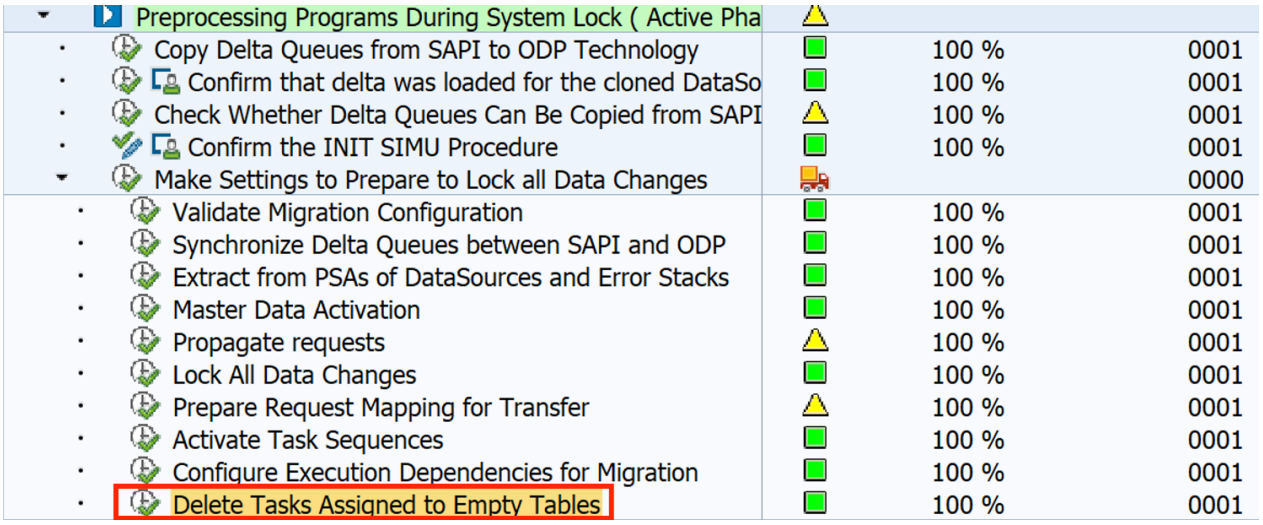


Figure 136: Delete Tasks Assigned to Empty Tables

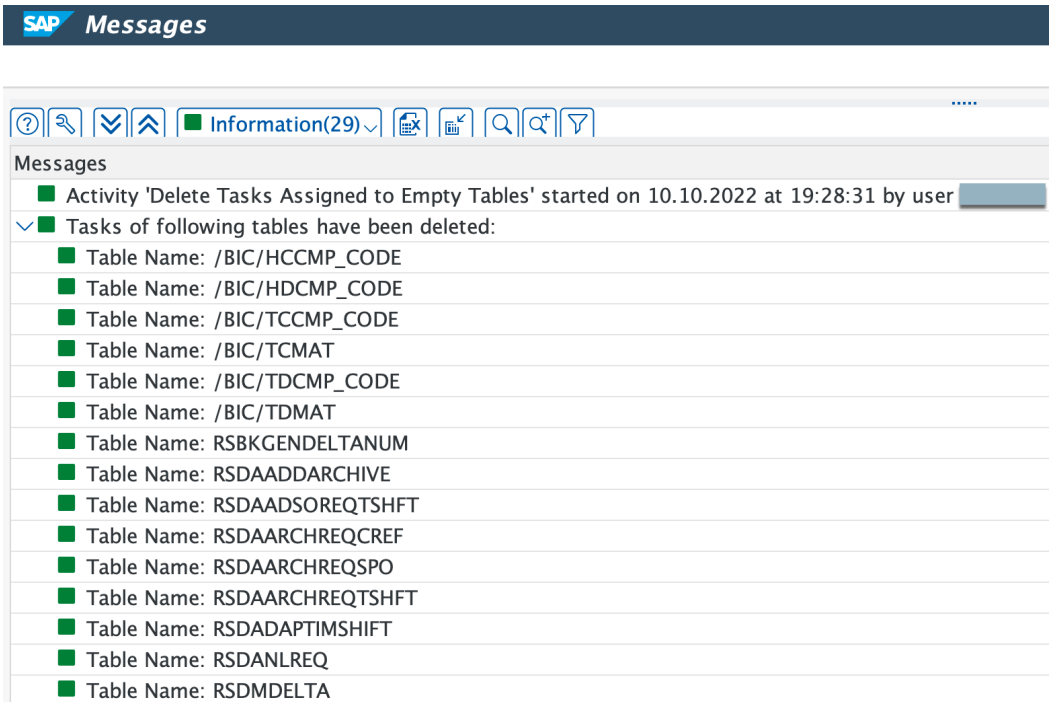


Figure 137: Delete Tasks Assigned to Empty Tables – Logs

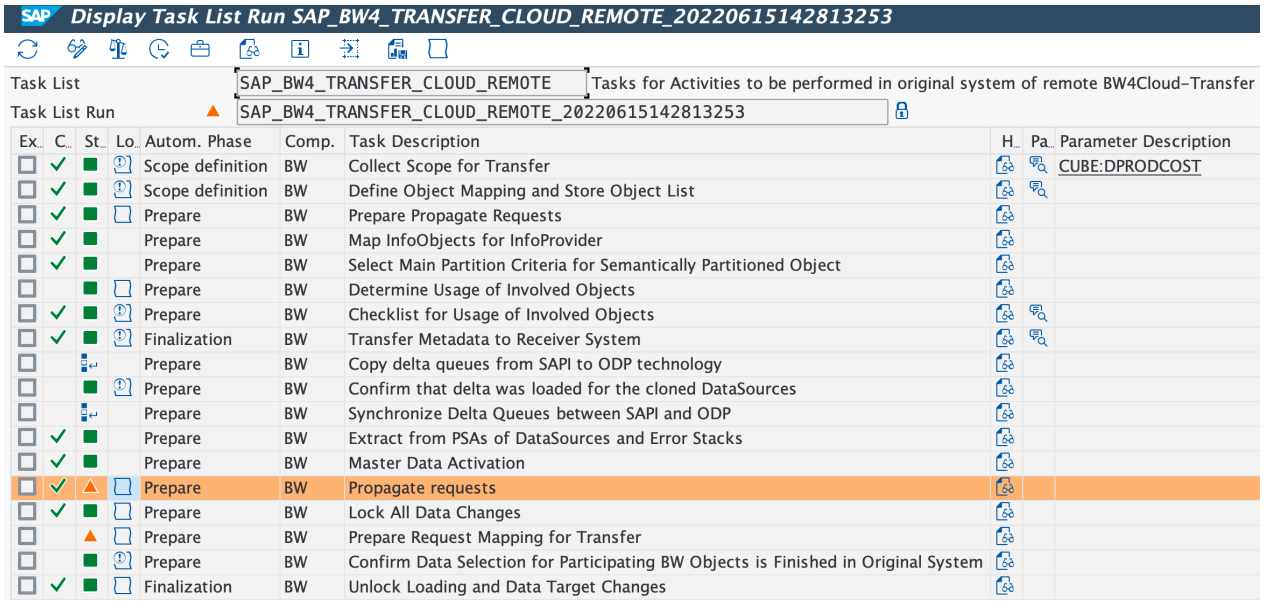
⇒ Tip: it can happen that a certain table is empty at the moment the data transfer preparations has started and receives data loads during the process (after the rows count in the Generation phase). It is recommended to manually run a new rows count prior to this activity to ensure that such tables will not be skipped from the data transfer process.

3.5.5 Lock Data Changes and Prepare SID Mappings

This activity is related to SEM-BCS and does not apply to BW bridge.

It is valid if you have SEM-BCS and running a migration to BCS/4HANA. This activity automatically locks the legacy SEM-BCS system to prevent the data changes. This locking is necessary to ensure the completeness and consistency of the transferred data.

All preparations for the data transfer are now finished and the Migration phase can start.



Ex.	C.	St.	Lo.	Autom. Phase	Comp.	Task Description	H.	Pa.	Parameter Description
☐	✓	■		Scope definition	BW	Collect Scope for Transfer			CUBE:DPRODCOST
☐	✓	■		Scope definition	BW	Define Object Mapping and Store Object List			
☐	✓	■		Prepare	BW	Prepare Propagate Requests			
☐	✓	■		Prepare	BW	Map InfoObjects for InfoProvider			
☐	✓	■		Prepare	BW	Select Main Partition Criteria for Semantically Partitioned Object			
☐		■		Prepare	BW	Determine Usage of Involved Objects			
☐	✓	■		Prepare	BW	Checklist for Usage of Involved Objects			
☐	✓	■		Finalization	BW	Transfer Metadata to Receiver System			
☐		■		Prepare	BW	Copy delta queues from SAPI to ODP technology			
☐		■		Prepare	BW	Confirm that delta was loaded for the cloned DataSources			
☐		■		Prepare	BW	Synchronize Delta Queues between SAPI and ODP			
☐	✓	■		Prepare	BW	Extract from PSAs of DataSources and Error Stacks			
☐	✓	■		Prepare	BW	Master Data Activation			
☐	✓	▲		Prepare	BW	Propagate requests			
☐	✓	■		Prepare	BW	Lock All Data Changes			
☐		▲		Prepare	BW	Prepare Request Mapping for Transfer			
☐		■		Prepare	BW	Confirm Data Selection for Participating BW Objects is Finished in Original System			
☐	✓	■		Finalization	BW	Unlock Loading and Data Target Changes			

Figure 138: Task List Run status prior to the start of the Migration phase

3.6 Migration Phase

There are basically two (2) options of running the data transfer (Migration Phase):

1. "*Start Migration*" option: available for the "Standard" and "Extended" views.

    a. Pros: it automatically controls the transfer sequence/scheduling (read –> move –> insert for each table).

    b. Cons: Although it controls the transfer sequence, this type of parallelization may create some idle time in both sender and receiver systems, the reason why you see the systems not running in full capacity at certain times. It may also create delays for the completeness of the tasks in the sender system (usually the critical path as the sender system is operational and its release/handover to business users would take longer).

2. "*Start Transformation of Identified Tables (Optional)*" option: available with the "Expert View"

a. Pros: It is possible to manually define/control the sequence and the n of jobs for table types, critical tables, etc., by for example, focusing on the read/move tasks in the sender system (to finish it faster and then focus on the receiver tasks afterwards).

b. It is not automated like the “Start Migration” option, it requires massive manual work/intervention and deeper knowledge about the data transfer technology. This manual work is needed to optimize the sequences and keep the systems as busy as possible.

- Option 1 will be used for the execution of our sample and usually fits for most of the customers.

- Option 2 can optimize the migration process – but would require additional work. Customers who decide for it usually require SAP involvement in the project.

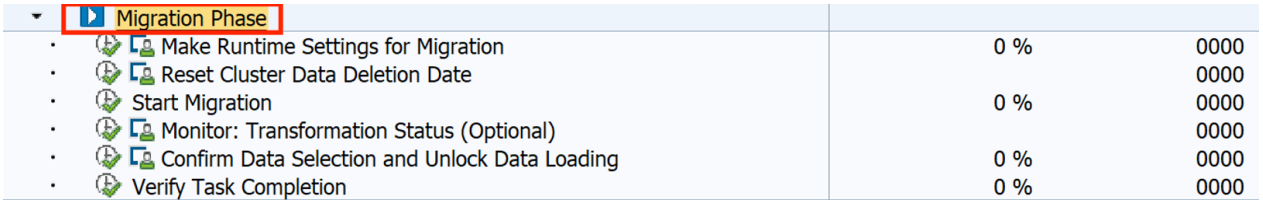


Figure 139: Migration Phase

3.6.1 Make Runtime Settings for Migration

In this activity, you can customize migration (data transfer) settings by defining the number of jobs (background processes) and the application server name for any table in the sender and receiver systems.

You can define the retention days by entering the number of days you want to retain the cluster data.

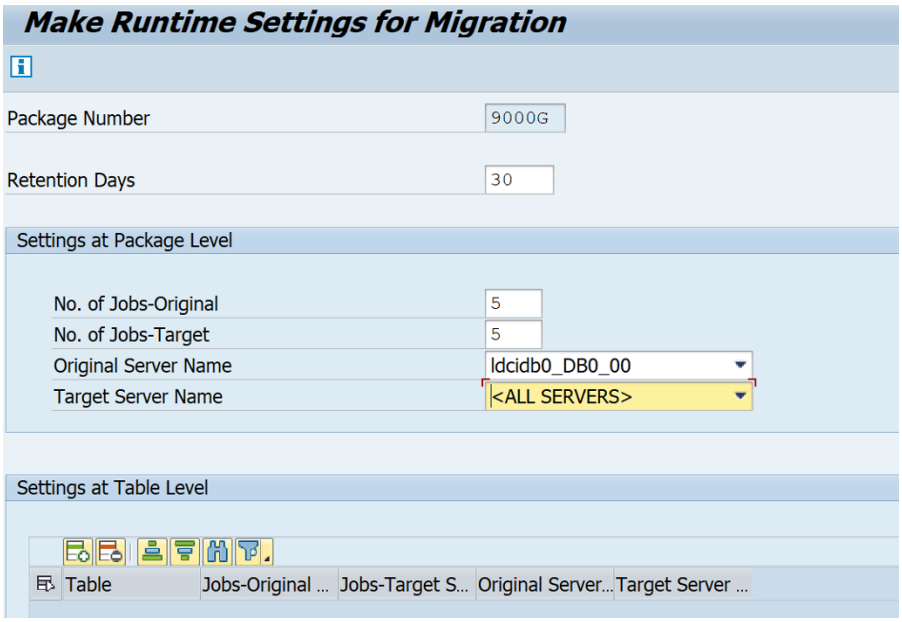


Figure 140: Make Runtime Settings for Migration

Package-level settings:

- Enter the default number of jobs to be used in the sender system and in the receiver system. Example: if you enter 10 for the sender system and 5 for the receiver system, then 10 jobs run simultaneously in the sender system, while 5 jobs run in the receiver system.

- Enter the application server name of the sender system and the receiver system in which you want the jobs to run (applicable when there is more than 1 application server in use).

Table-level settings:

- You can enter a specific number of jobs for a single table by choosing the Insert Row pushbutton. You can also enter other relevant values for this table. Example: you specified 10 jobs for the sender system at the package level and 3 jobs at the table level, then this activity uses no more than 3 jobs at a time for the table.

You can check in the transaction SM50 the number of jobs which are available:

Sender system

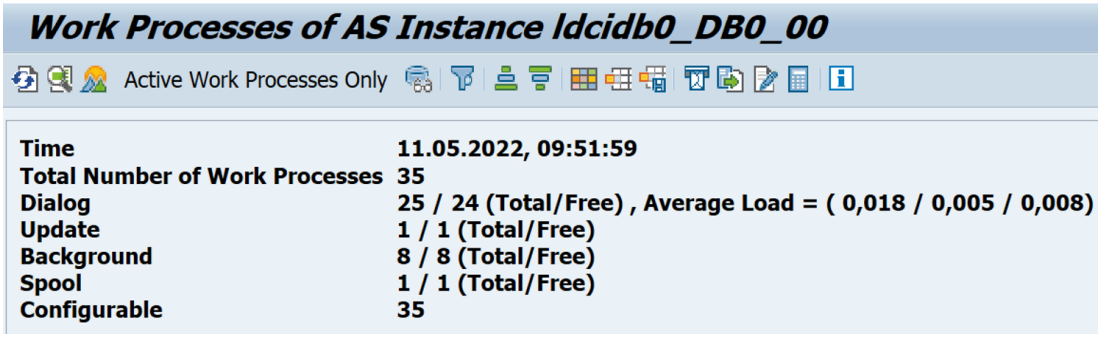


Figure 141: Number of work processes in the sender system

Receiver system

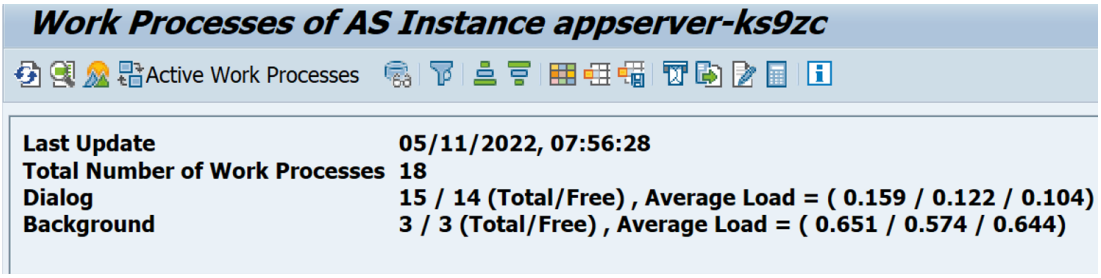


Figure 142: Number of work processes in the receiver system

For our sample, it was possible to use up to 8 jobs for the sender system and 3 jobs for the receiver system.

The balance settings between dialog and background jobs can be reconfigured via the transaction RZ04. It is normally a good practice to define a balance with the higher number of background jobs as possible for the Migration phase. This gives more power and flexibility for the execution and the settings can be reverted to the originals when the data transfer is finished.

Our sample ADSO CPRODCOST contains 51.761.812 records in the sender system and this same data is expected to be inserted into the corresponding ADSO in the receiver system:

- Table BIC/ADPRODCOST1 (sender table) -> Table BIC/ADPRODCOST1 (receiver table).

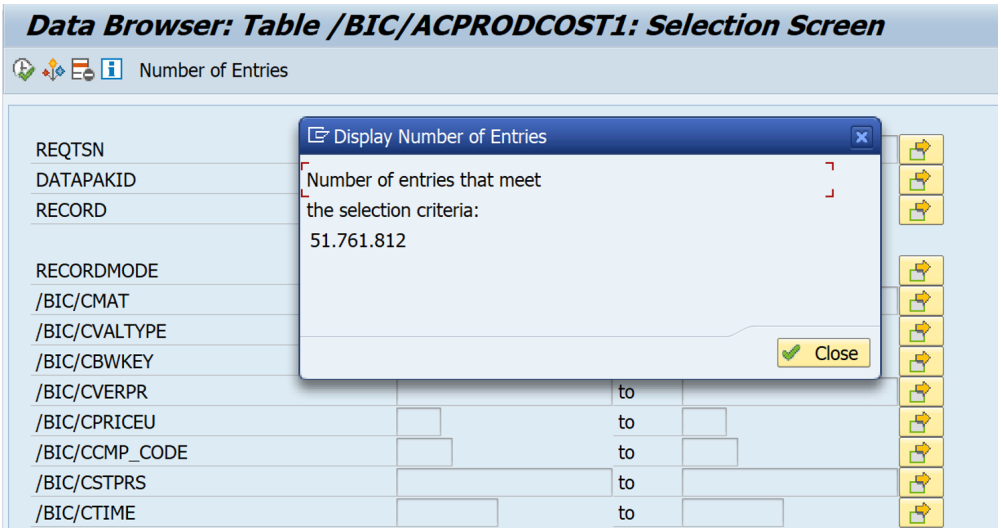


Figure 143: Table BIC/ADPRODCOST1 (sender system)

Our sample InfoCube DPRODCOST also contains 51.761.812 records in the sender system and the same data is expected to be inserted into the corresponding ADSO in the receiver system:

- Table /BIC/FDPRODCOST (sender table) -> Table /BIC/FDPRODCOST (receiver shadow) -> Table /BIC/ADPRODCOST1 (receiver table).

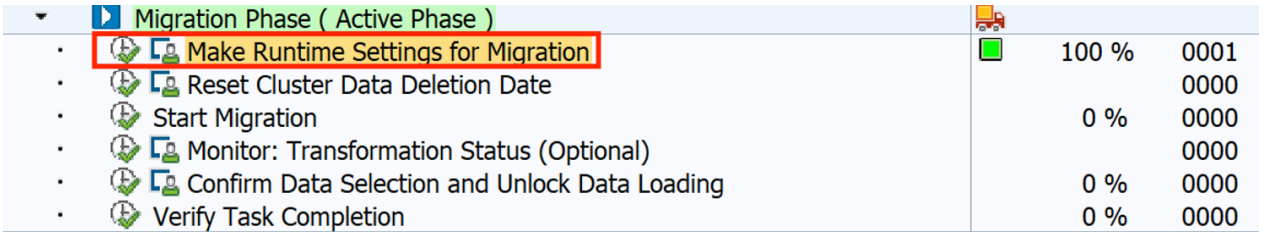


Figure 144: Make Runtime Settings for Migration

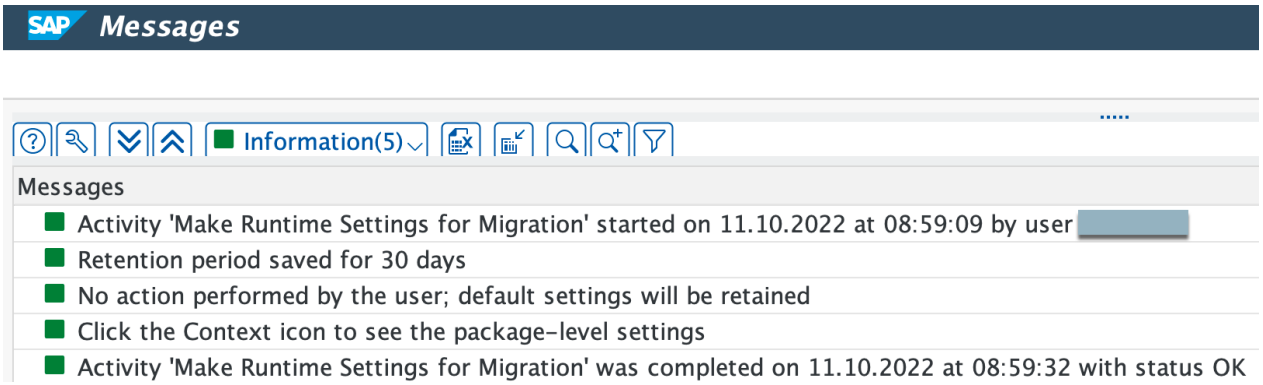


Figure 145: Make Runtime Settings for Migration – Logs

⇒ Tip: InfoCubes in the sender system are converted to ADSOs in the receiver system, nevertheless the original InfoCubes tables (F/E/D) also exist in the receiver system (so called “shadow” InfoCube tables”). They have the same structure as in the sender system and are used to “move” the data into the corresponding ADSOs (executed by the MOVE_TO_ADSO tasks). This data can be dropped after the overall data transfer process is finished. It is recommended to keep this data until the data transfer results are validated.

3.6.2 Reset Cluster Data Deletion Date

In this activity, you can set a date to delete the data from the cluster tables in the sender) and the receiver systems. The default retention period is 30 days and can be extended.

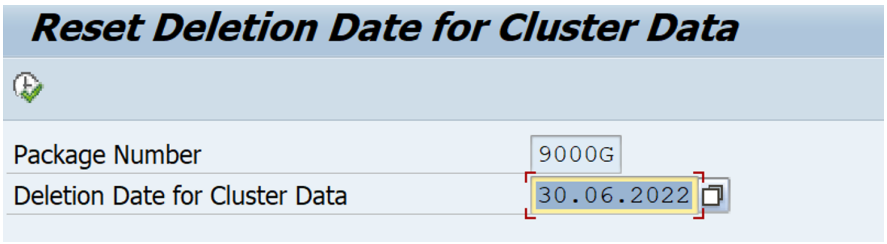


Figure 146: Reset Cluster Data Deletion Date – Parameters

These tables are used to manage the data transfer (CLU_BEAM tasks) and exist in both sender and receiver systems. They can be useful as a backup of the transferred data and it is recommended to also keep them until the data transfer results are validated.

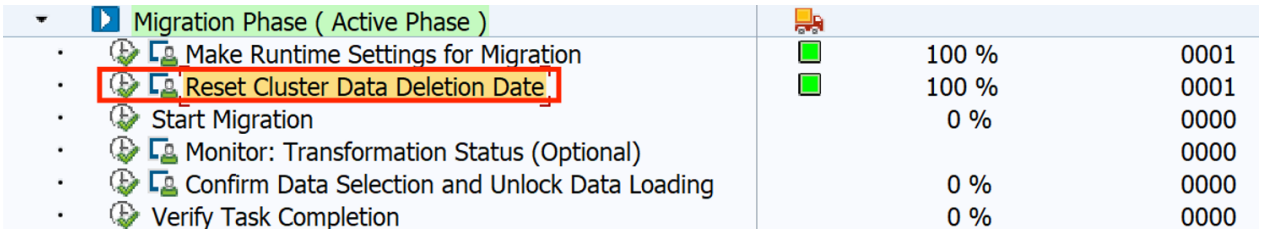


Figure 147: Reset Cluster Data Deletion Date

3.6.3 Start Migration

This activity automatically starts the read tasks (so called I_EXC_READ tasks) and cluster transfer tasks (so called Move or CLU_BEAM tasks) in the sender system, and insertion tasks (so called CLU_CONVERT tasks) in the receiver system.

This sequence is critical for a successful execution of this activity therefore do not start any tasks from the Program Execution transactions (SA38 or SE38).

Always start the tasks by executing this activity:

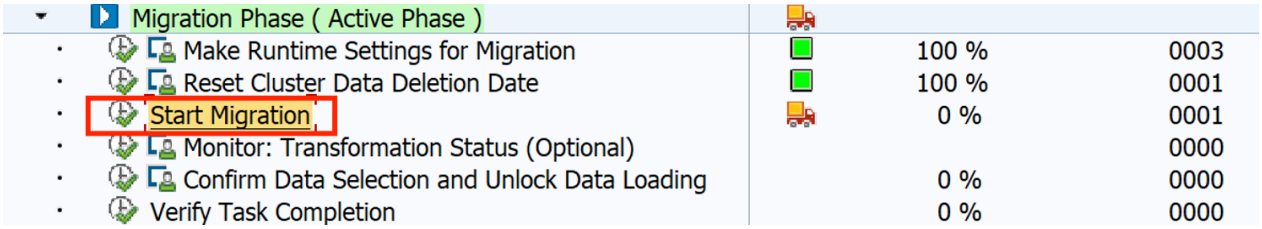


Figure 148: Start Migration

You may monitor the execution via the next activity “Monitor: Transformation Status (Optional)” or via the transaction CNVLTRL_MONITOR in the sender system.

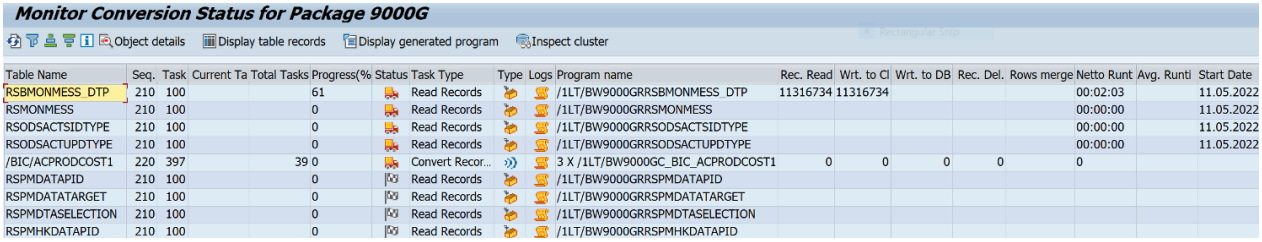


Figure 149: Conversion Monitor

3.6.4 Monitor: Transformation Status (Optional)

In this activity, you can monitor the status of the overall data transfer for the BW objects in scope.

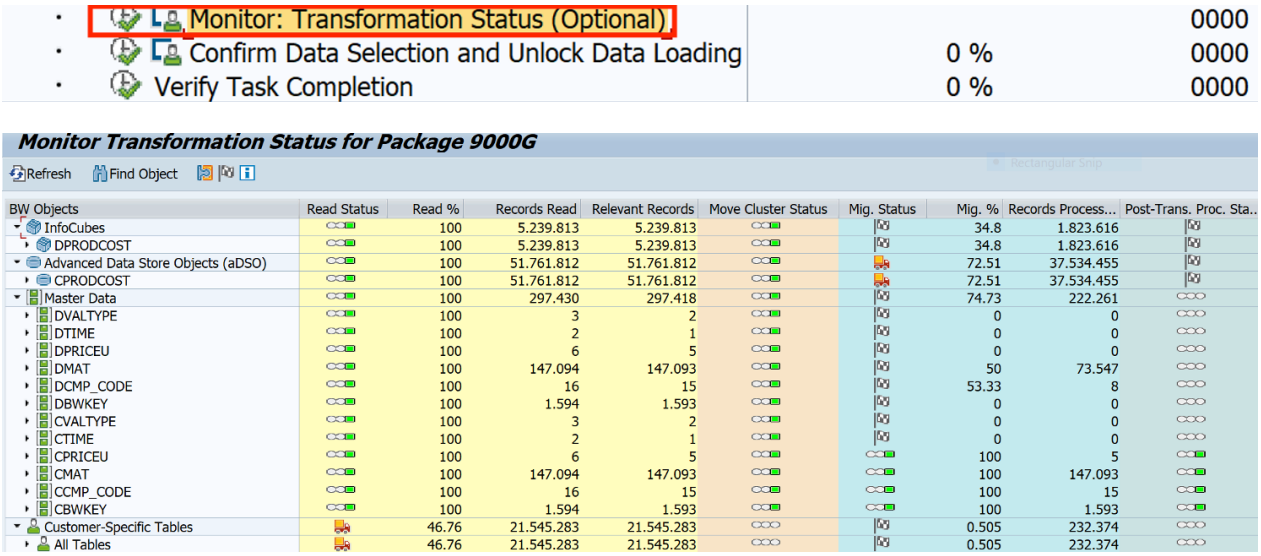


Figure 150: Monitor: Transformation Status (Optional) – data transfer in progress

This is the final result of our sample data transfer:

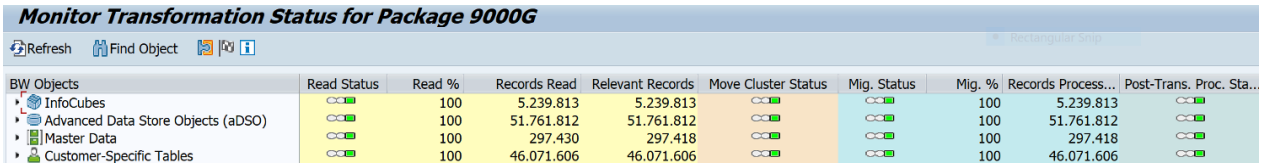


Figure 151: Monitor: Transformation Status (Optional) – data transfer finished

Note that the number of records under “InfoCubes” represents the F/E/D tables which were transferred from the original InfoCubes tables in the sender system to the shadow InfoCubes tables in the receiver system.

51.761.812 records were inserted into CPRODCOST ADSO in the receiver system as expected. This can be checked via the BW bridge cockpit:

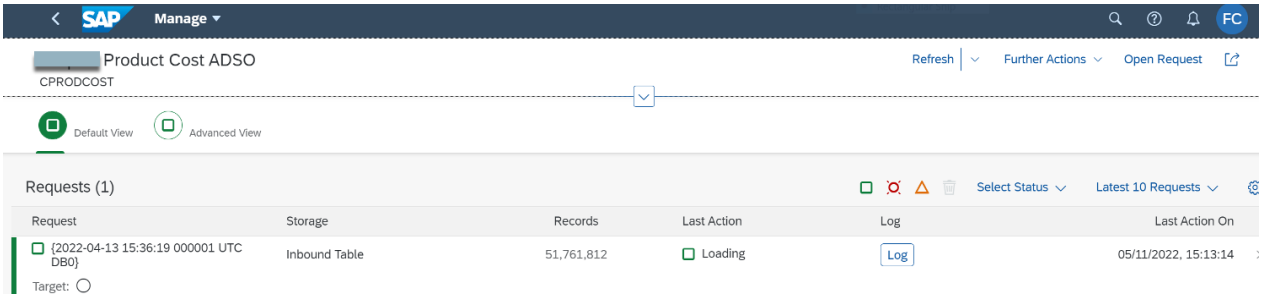


Figure 152: BW bridge cockpit – CPRODCOST ADSO (1)

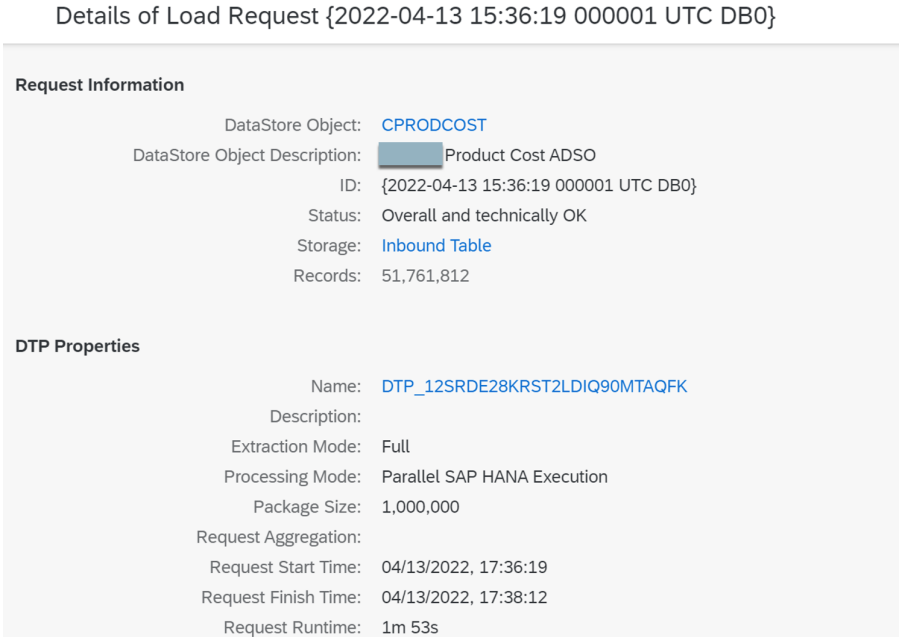


Figure 153: BW bridge cockpit – CPRODCOST ADSO (2)

51.761.812 records were inserted into DPRODCOST ADSO in the receiver system as expected. The “InfoCube” description came from the original InfoCube DPRODCOST description in the sender system and can be modified.

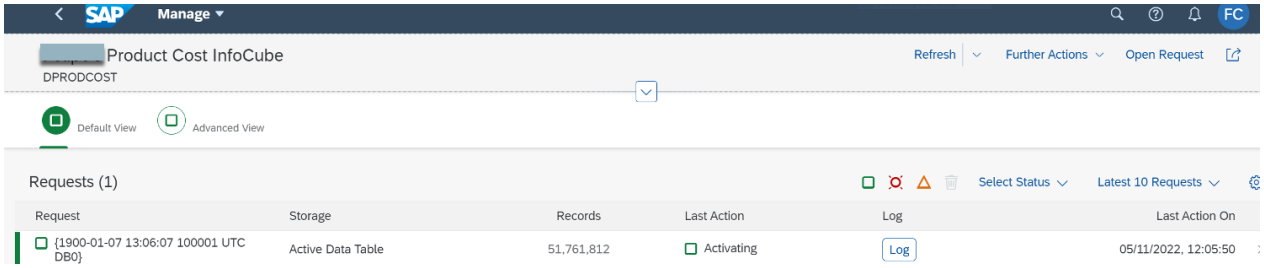


Figure 154: BW bridge cockpit – DPRODCOST ADSO (1)

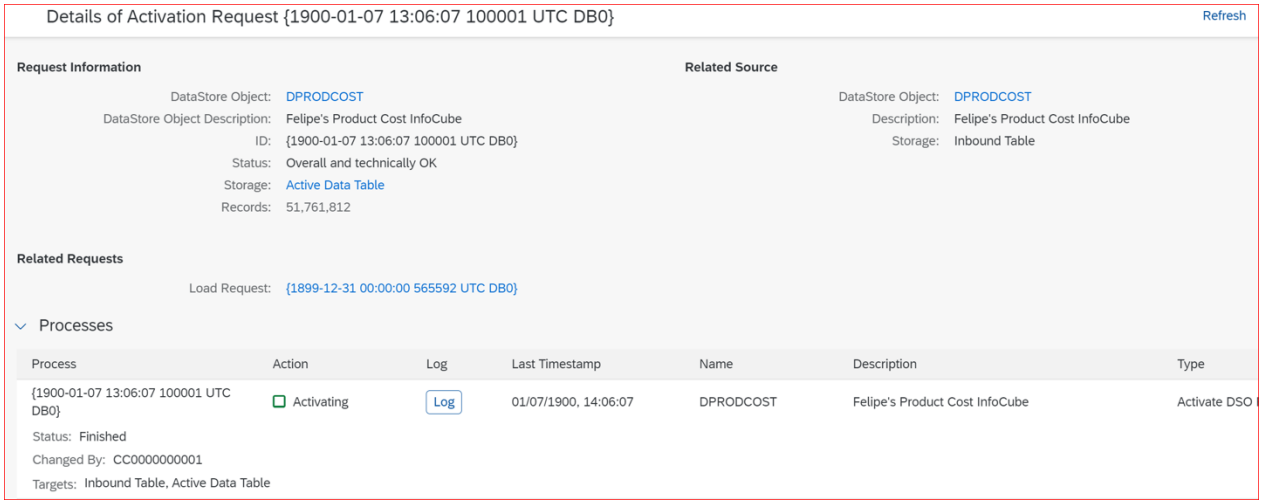


Figure 155: BW bridge cockpit – DPRODCOST ADSO (2)

3.6.5 Confirm Data Selection and Unlock Data Loading

With this activity, you can confirm the data selected in the sender system and unlock the system.

It confirms that the read tasks in the migration phase have been successfully completed in the sender system and removes the locks on DataSources and InfoProviders included in scope, allowing the operations for BW objects in the sender system to be resumed.

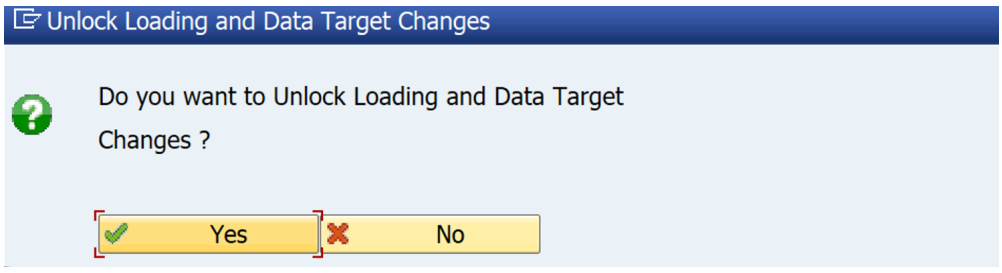


Figure 156: Confirm Data Selection and Unlock Data Loading - Unlock confirmation

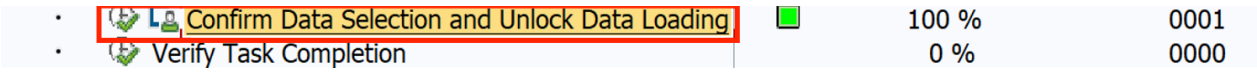


Figure 157: Confirm Data Selection and Unlock Data Loading

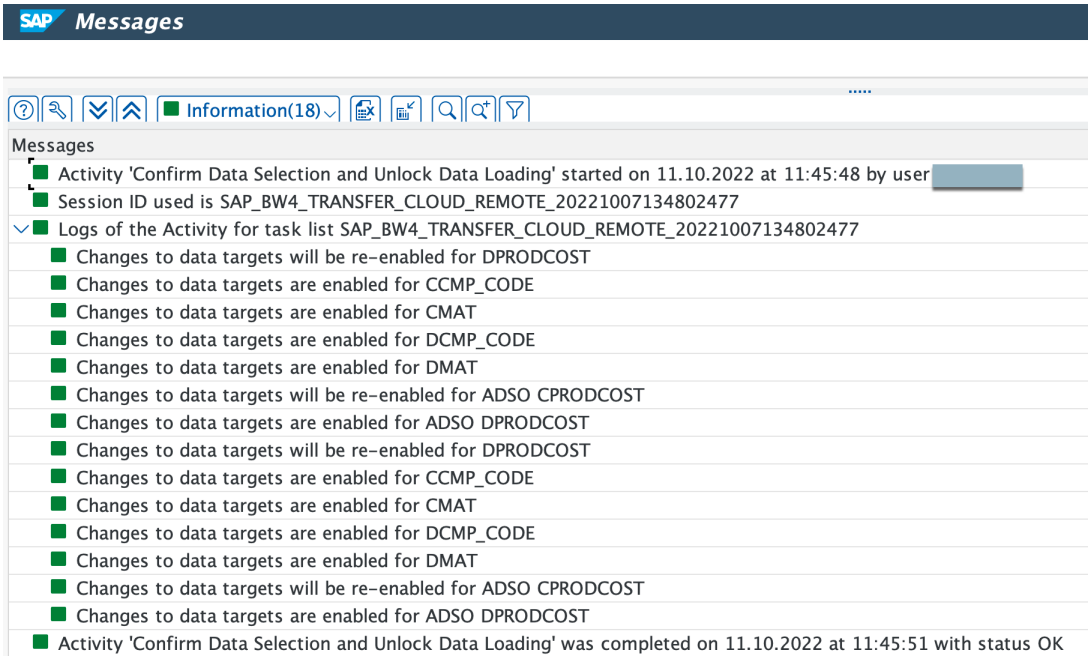


Figure 158: Confirm Data Selection and Unlock Data Loading – Logs

⇒ Tip: When the receiver system is on BW/4HANA, the confirmation of this activity unlocks the SEM-BCS system. In order to prevent the overwriting of number ranges and ensure that the feature works smoothly, ensure that there are no SEM BCS number range intervals in the receiver system. If there are number range intervals in the receive system, you need to:

- Delete the number range intervals before executing this activity or
- Adjust the number ranges manually in the receiver system

The Migration phase was successfully finished.

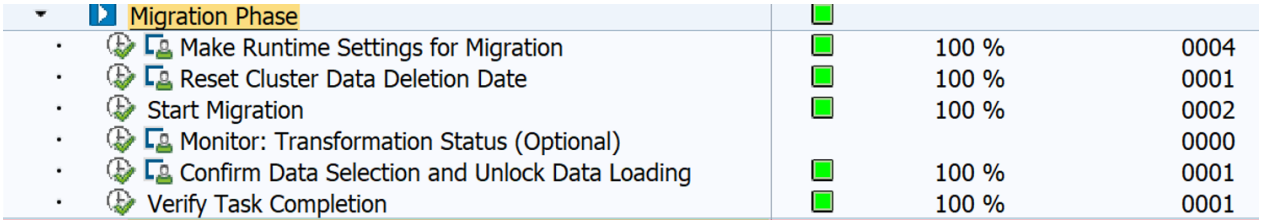


Figure 159: Migration Phase was finished

3.7 Post Processing Activities During System Lock

With the completeness of the Migration phase, you can now run the Post Processing activities.

3.7.1 Rebuild Hierarchy and Reset Number Ranges

This is an activities group and you have the option to either run it all together (mass execution) or each activity individually. If you want to execute any of the dialog activities, go to the extended view and execute each activity sequentially. If you use the mass execution, the dialog activities under this activities group are skipped.

If the activities group terminates or ends with errors or warnings, activate the extended view of the process tree. You can now view all activities that belong to the group activity and their statuses.

3.7.1.1 Rebuild Hierarchy tables (/BI+/H*, /BI+/I*)

This activity determines the relevant InfoObjects that require the rebuilding of hierarchy tables after the data transfer:

- /BI+/H*
- /BI+/J*
- RSHIEDIR
- RSMHIERNODE
- RSTHIERNODE

These tables are not relevant for our sample.

Post Processing Activities During System Lock (Active Phase		
Rebuild Hierarchy and Reset Number Ranges		0000
Rebuild Hierarchy tables (/BI+/H*, /BI+/I*)	100 %	0001
Reset Number Ranges	0 %	0000
Set All Queries for Generation	0 %	0000
Execute RSRV Checks in the Target System	0 %	0000
Execute Validations in Target and Check Results (Expert)		0000
View Time-Related Migration Statistics (Optional)		0000

Figure 160: Rebuild Hierarchy tables (/BI+/H, /BI+/I*)*

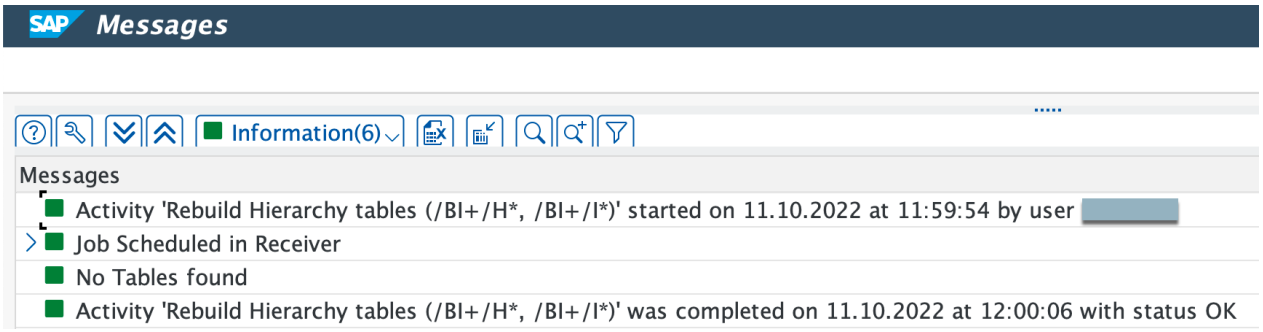


Figure 161: Rebuild Hierarchy tables (/BI+/H, /BI+/I*) – Logs*

3.7.1.2 Reset Number Ranges

This activity checks and repairs the number ranges of the BW objects (NRIV objects) after the data transfer.

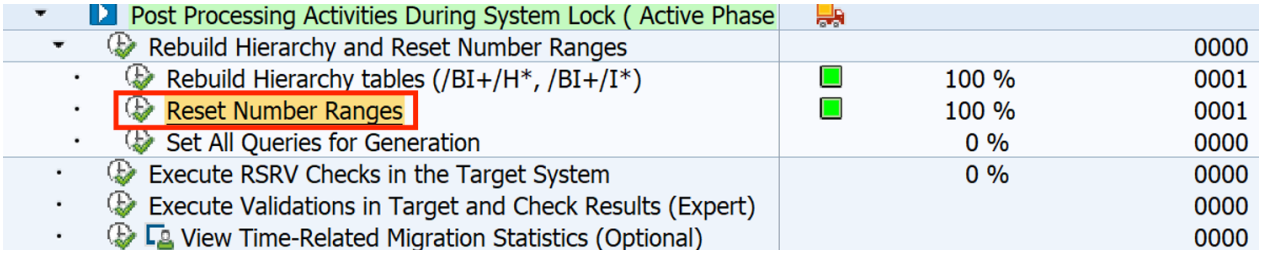


Figure 162: Reset Number Ranges

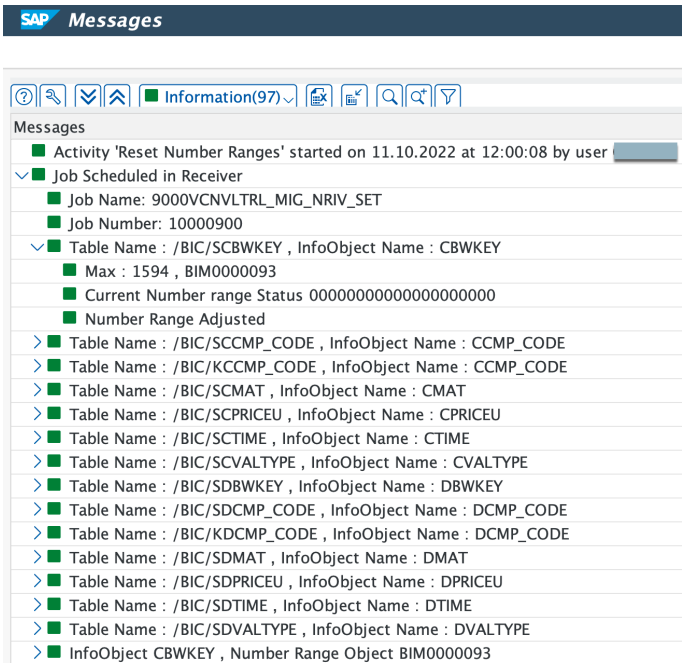


Figure 163: Reset Number Ranges – Logs

3.7.1.3 Set All Queries for Generation

This activity sets the BW queries to be generated once again before the next call from the Business Explorer Analyzer (BEx analyzer).

Note this is valid when the receiver system is on BW/4HANA.

This activity is not applied when the receive system is on BW bridge and not relevant to our sample as there are no queries in scope.

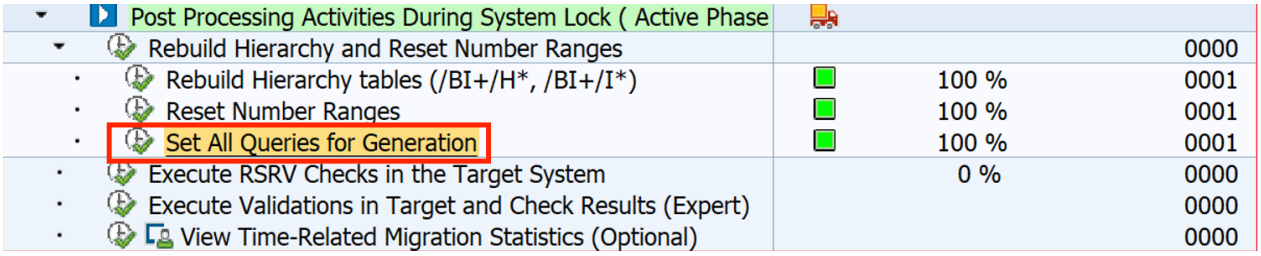


Figure 164: Set All Queries for Generation

3.7.2 Execute RSRV Checks in the Target System

This activity will run the RSRV checks in the receiver system in order you can compare them with the RSRV checks results which were previously made for the sender system.

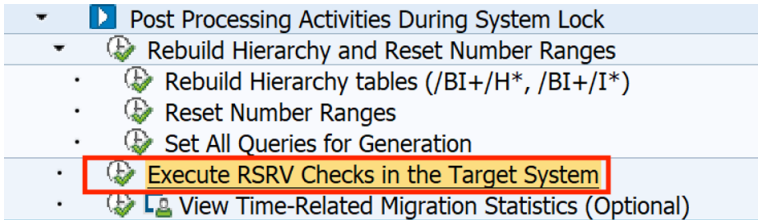


Figure 165: Execute RSRV Checks in the Target System

- Load the same package which was earlier created (CNVLTRL9000G000)

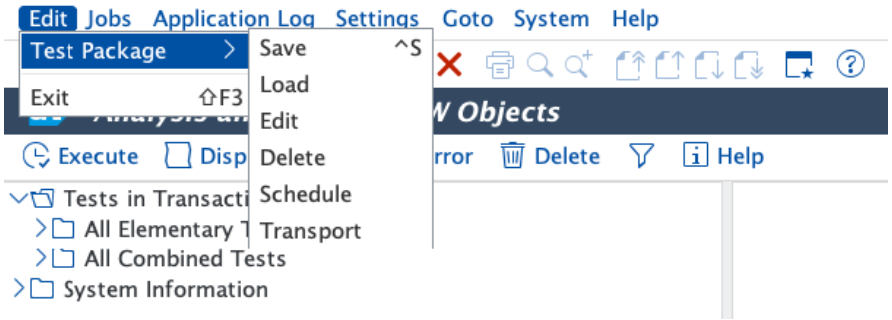


Figure 166: RSRV - Load Test Package in the receiver system (1)

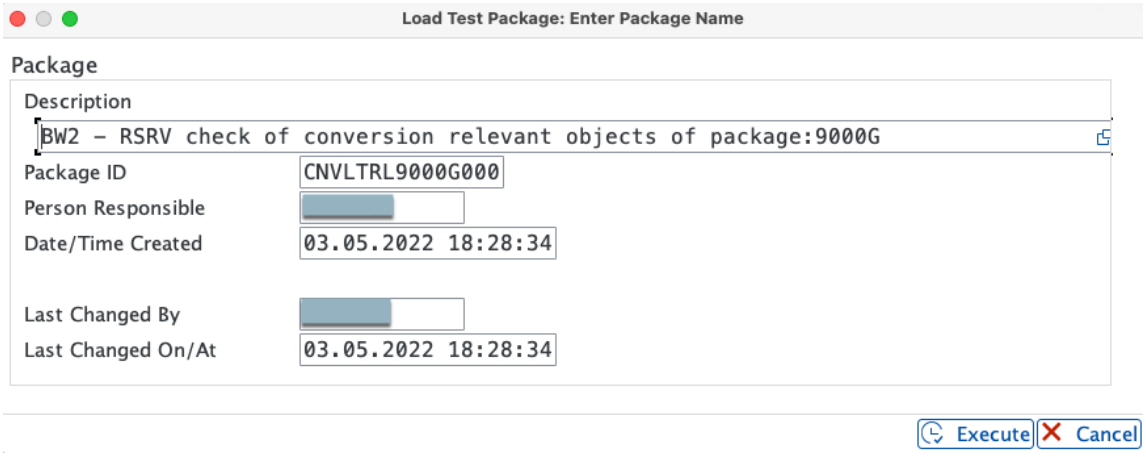


Figure 167: RSRV - Load Test Package in the receiver system (2)

- Execute the test package.

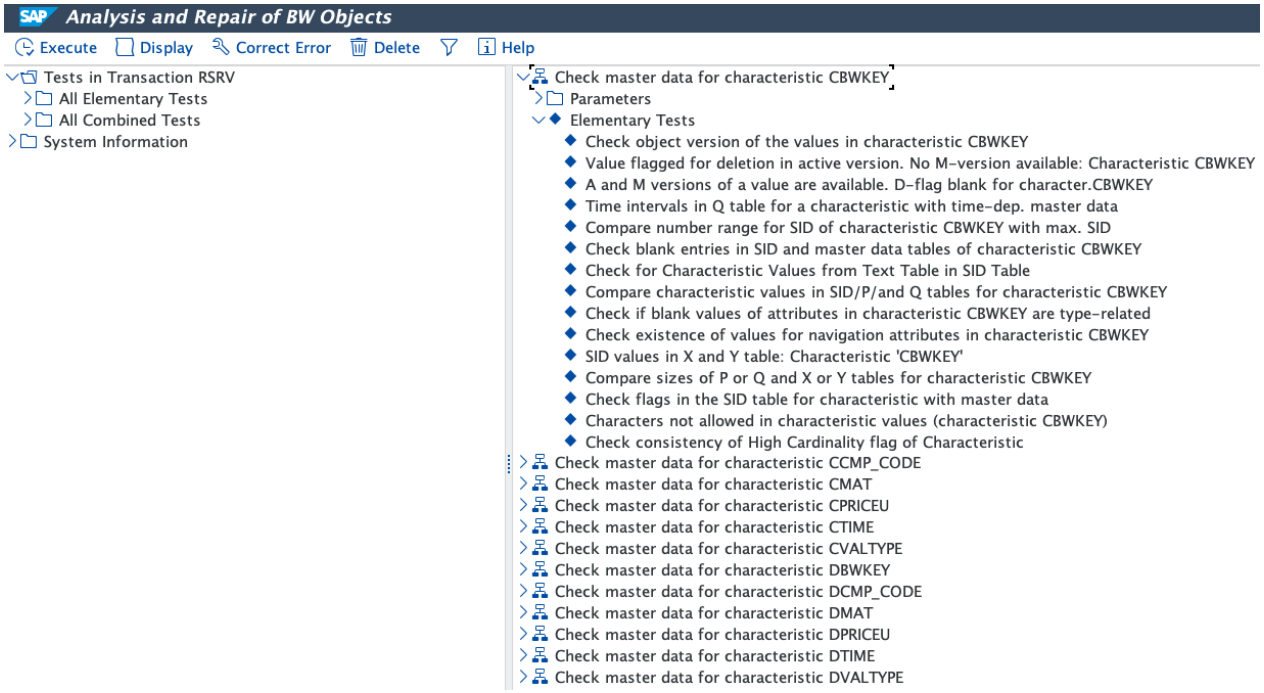


Figure 168: Execute Test Package in the receiver system (1)

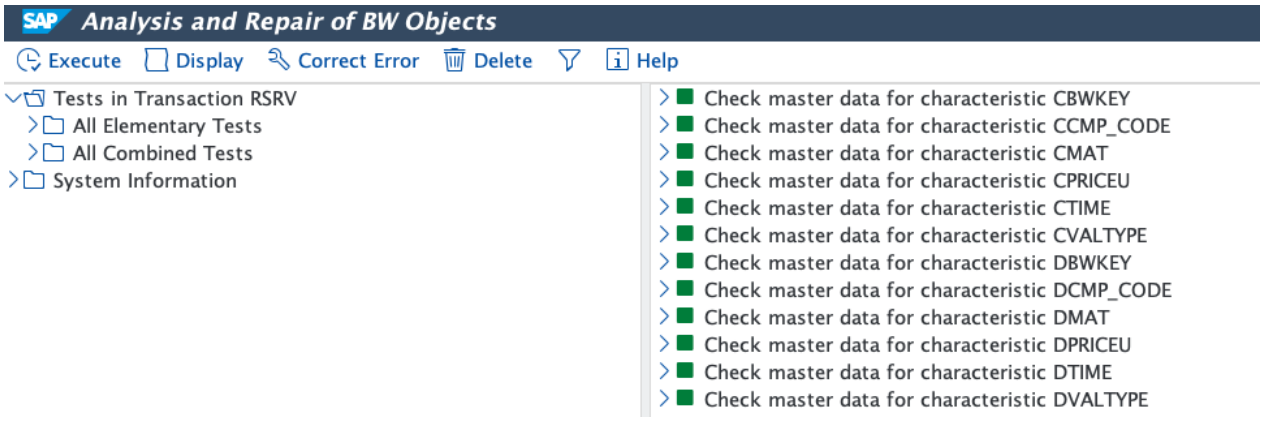


Figure 169: Execute Test Package in the receiver system (2)

For our sample, no inconsistencies were found after the data transfer.

For further information and tips, check with the previous activity Execute RSRV Checks in the Original System.

Note the backend access to BW bridge via SAP GUI is not possible for partners and customers. Contact SAP by opening an incident under the components BW-B4H-CNV-RMT or CA-LT-BW4 for executing this activity.

3.7.3 View Time-Related Migration Statistics (Optional)

In this activity, you can view the runtimes of the data transfer for BW relevant objects (InfoProvider and table view) from the sender system to the receiver system.

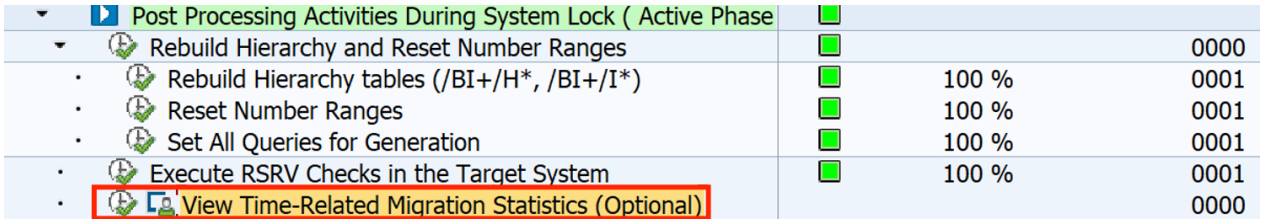


Figure 170: View Time-Related Migration Statistics (Optional) – (1)

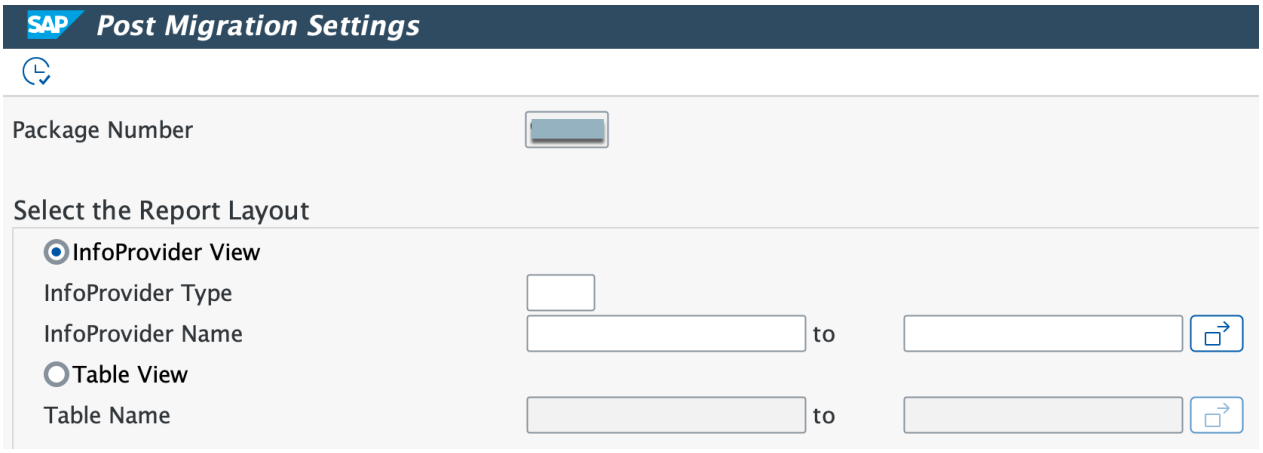
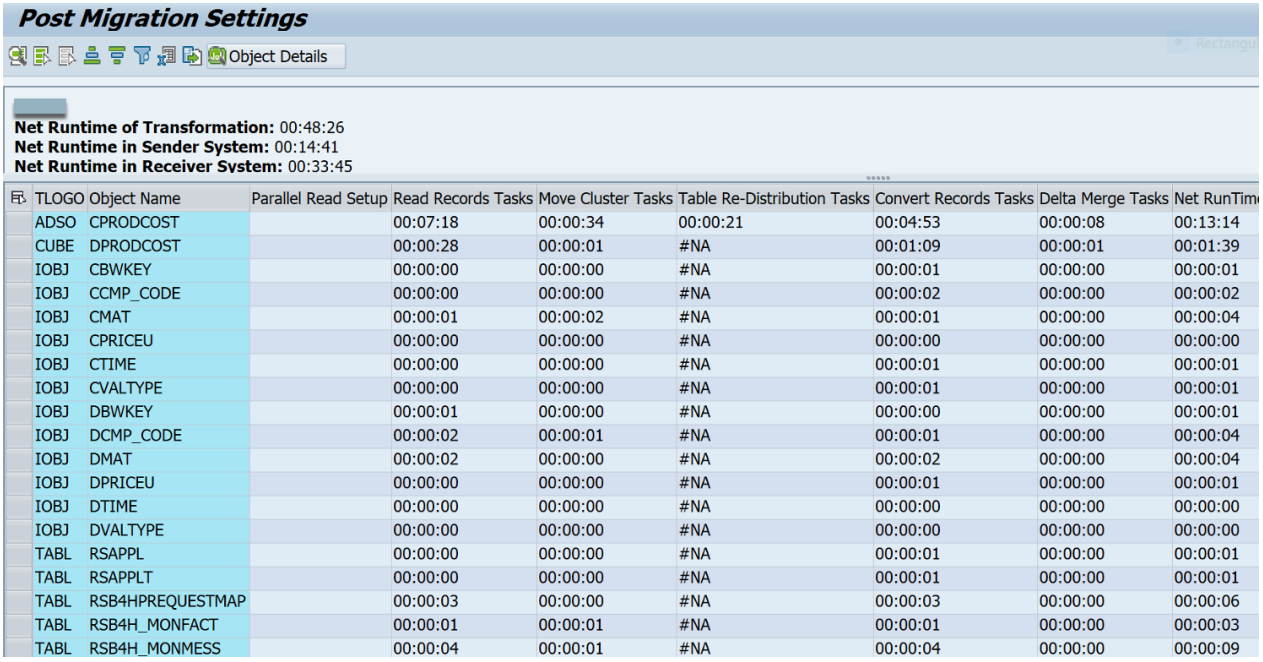


Figure 171: View Time-Related Migration Statistics (Optional) - View types

Our sample data transfer took approximately 48 minutes to finish:

- 15 min with tasks in the sender system (read and move).
- 33 min with tasks in the receiver system (insertion).



TLOGO	Object Name	Parallel Read Setup	Read Records Tasks	Move Cluster Tasks	Table Re-Distribution Tasks	Convert Records Tasks	Delta Merge Tasks	Net RunTim
ADSO	CPRODCOST		00:07:18	00:00:34	00:00:21	00:04:53	00:00:08	00:13:14
CUBE	DPRODCOST		00:00:28	00:00:01	#NA	00:01:09	00:00:01	00:01:39
IOBJ	CBWKEY		00:00:00	00:00:00	#NA	00:00:01	00:00:00	00:00:01
IOBJ	CCMP_CODE		00:00:00	00:00:00	#NA	00:00:02	00:00:00	00:00:02
IOBJ	CMAT		00:00:01	00:00:02	#NA	00:00:01	00:00:00	00:00:04
IOBJ	CPRICEU		00:00:00	00:00:00	#NA	00:00:00	00:00:00	00:00:00
IOBJ	CTIME		00:00:00	00:00:00	#NA	00:00:01	00:00:00	00:00:01
IOBJ	CVALTYPE		00:00:00	00:00:00	#NA	00:00:01	00:00:00	00:00:01
IOBJ	DBWKEY		00:00:01	00:00:00	#NA	00:00:00	00:00:00	00:00:01
IOBJ	DCMP_CODE		00:00:02	00:00:01	#NA	00:00:01	00:00:00	00:00:04
IOBJ	DMAT		00:00:02	00:00:00	#NA	00:00:02	00:00:00	00:00:04
IOBJ	DPRICEU		00:00:00	00:00:00	#NA	00:00:01	00:00:00	00:00:01
IOBJ	DTIME		00:00:00	00:00:00	#NA	00:00:00	00:00:00	00:00:00
IOBJ	DVALTYPE		00:00:00	00:00:00	#NA	00:00:00	00:00:00	00:00:00
TABL	RSAPPL		00:00:00	00:00:00	#NA	00:00:01	00:00:00	00:00:01
TABL	RSAPPLT		00:00:00	00:00:00	#NA	00:00:01	00:00:00	00:00:01
TABL	RSB4HPREQUESTMAP		00:00:03	00:00:00	#NA	00:00:03	00:00:00	00:00:06
TABL	RSB4H_MONFACT		00:00:01	00:00:01	#NA	00:00:01	00:00:00	00:00:03
TABL	RSB4H_MONMESS		00:00:04	00:00:01	#NA	00:00:04	00:00:00	00:00:09

Figure 172: View Time-Related Migration Statistics (Optional) - InfoProvider view

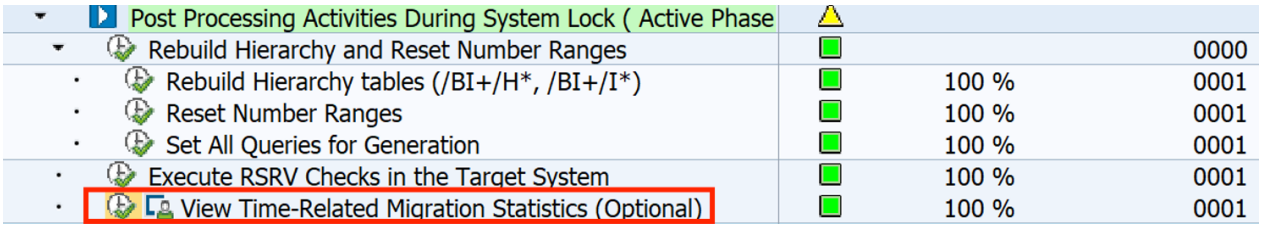


Figure 173: View Time-Related Migration Statistics (Optional) – (2)

3.8 System Settings After Migration

This is the last phase before closing the Remote Conversion package. It contains optional housekeeping activities which will drop InfoCubes shadow tables and cluster tables. As previously mentioned, it is recommended to execute them only after the data transfer is 100% validated (technical validations and business users' validation). You have ensured that data was moved consistently from the InfoCubes to the corresponding ADSOs.

The reason for keeping the content of those tables during the validation period is because in case of any issues, the data can be reprocessed in a faster way, avoiding having to read (select) the data again from the sender system.

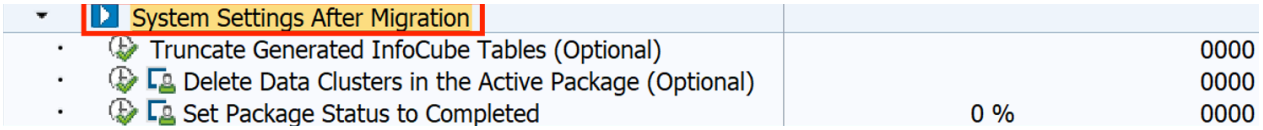


Figure 174: System Settings After Migration

For our sample, note the InfoCubes shadow tables have 5.239.813 records (F + D) as indicated in the monitor.

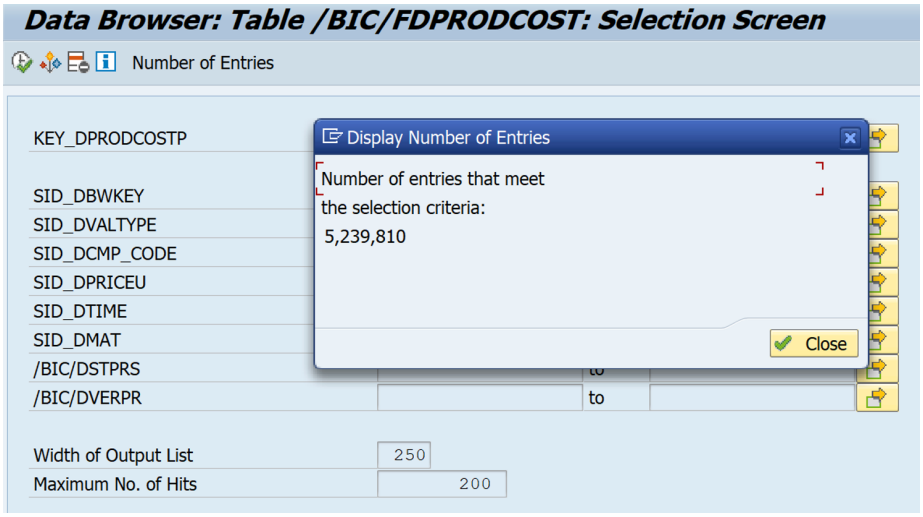


Figure 175: Table /BIC/FPRODCOST in the receiver system (1)

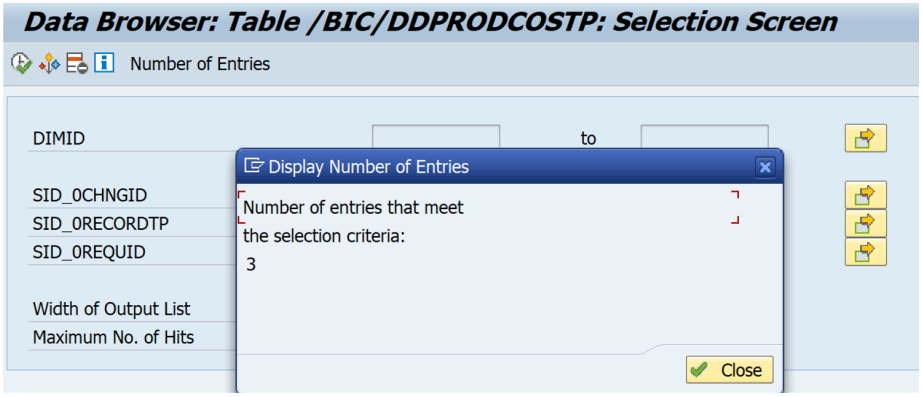


Figure 176: Table /BIC/DDPRODCOSTP in the receiver system (1)

3.8.1 Truncate Generated InfoCube Tables (Optional)

This activity truncates the data of the following temporary tables:

- Dimension and fact tables for InfoCubes
- RS* tables used for BW Administration Runtime tables

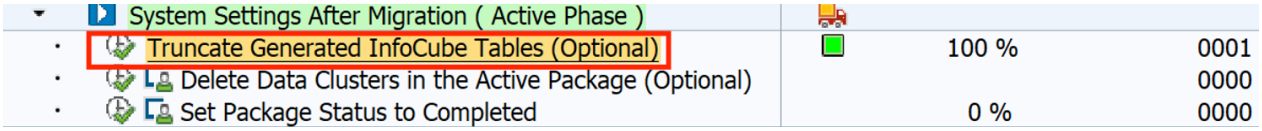


Figure 177: Truncate Generated InfoCube Tables (Optional)

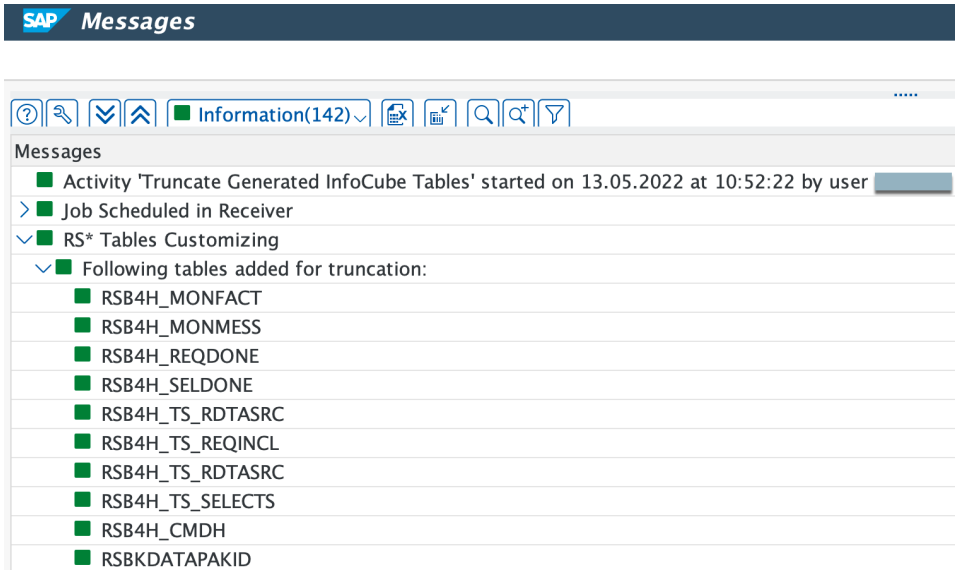


Figure 178: Truncate Generated InfoCube Tables (Optional) - Logs

After truncating:

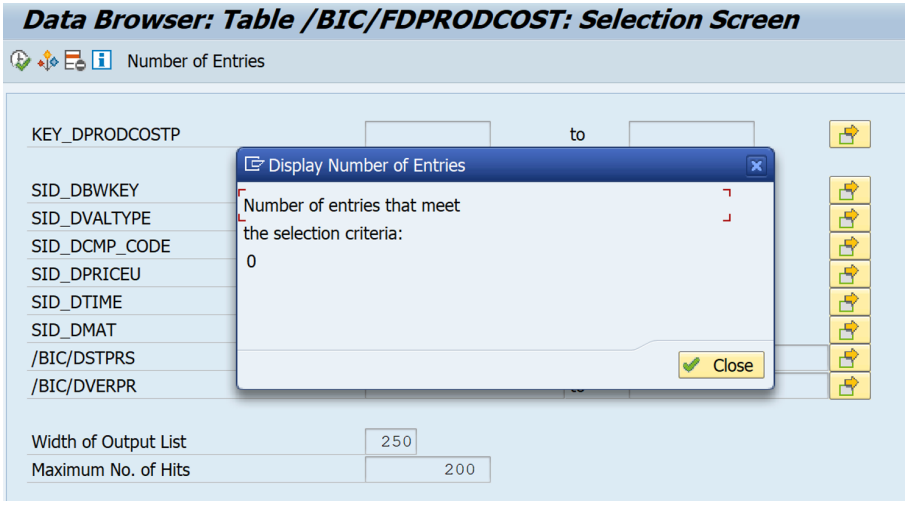


Figure 179: Table /BIC/FPRODCOST in the receiver system (2)

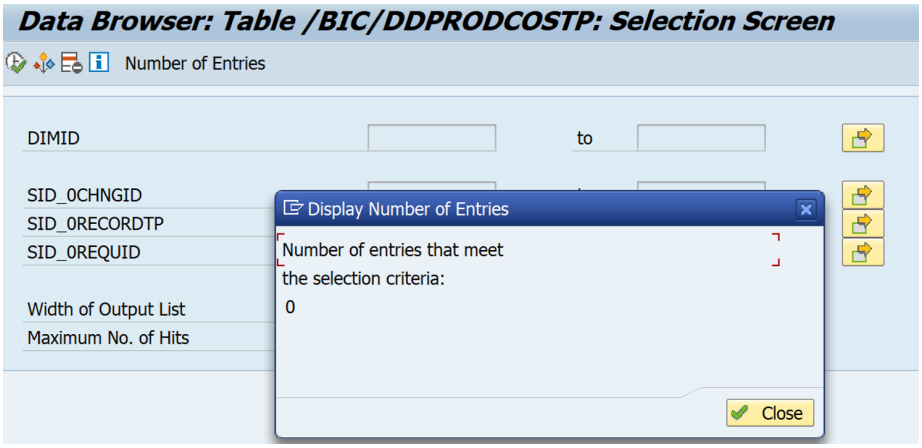


Figure 180: Table /BIC/DDPRODCOSTP in the receiver system (2)

3.8.2 Delete Data Clusters in the Active Package (Optional)

This activity refreshes the CNVLTRL_BWCLU control table or any generated cluster tables in the sender and receive systems.

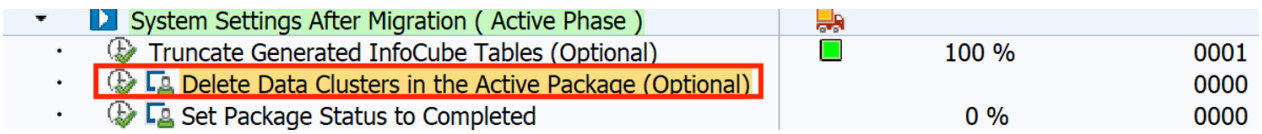


Figure 181: Delete Data Clusters in the Active Package (Optional) – (1)

With that, the cluster data in the tables are deleted in both sender and receiver systems.

You can specify a date for the deletion or choose to delete the cluster data immediately.

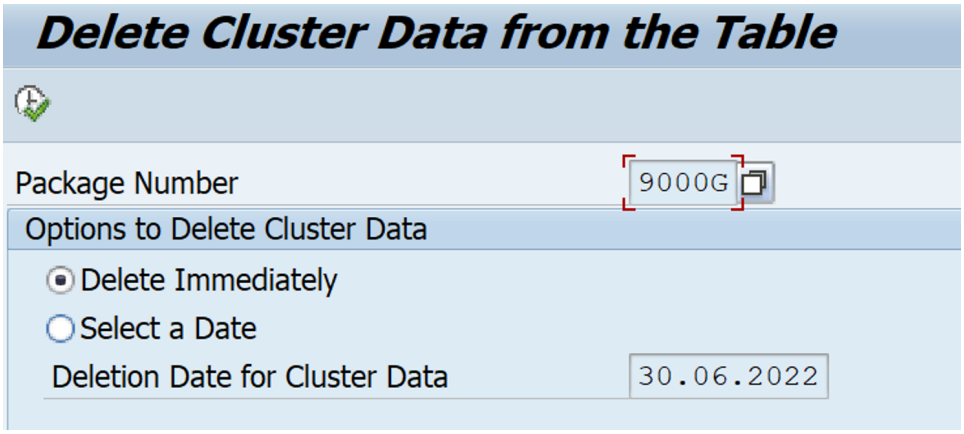


Figure 182: Delete Data Clusters in the Active Package (Optional) - Deletion options

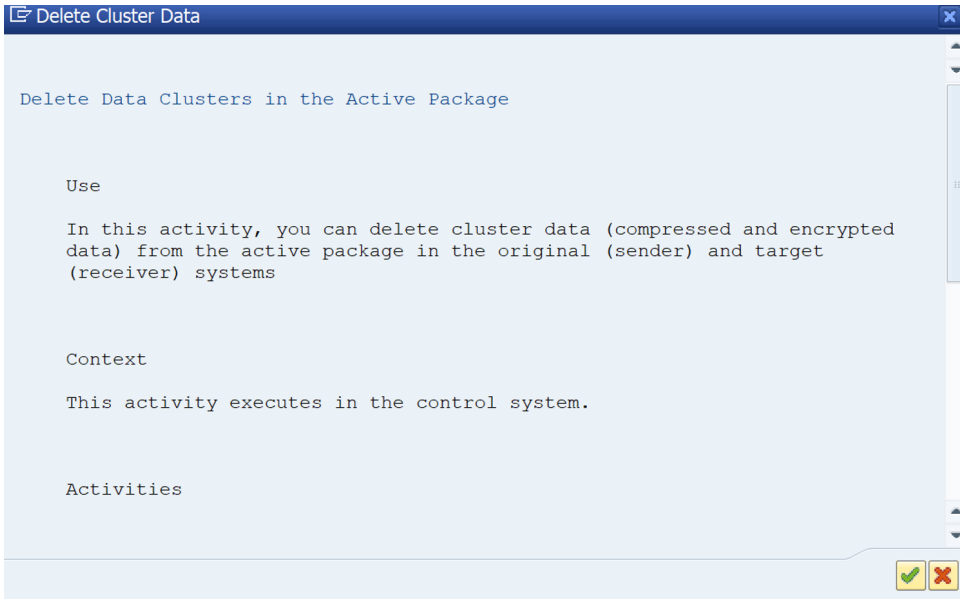


Figure 183: Delete Data Clusters in the Active Package (Optional) - Deletion confirmation

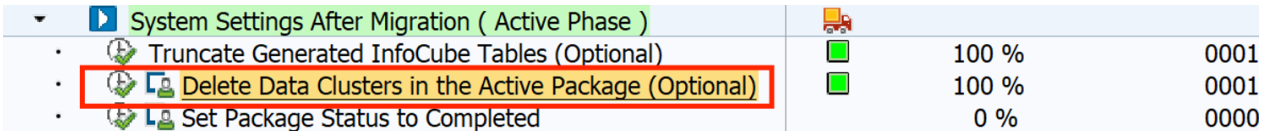


Figure 184: Delete Data Clusters in the Active Package (Optional) – (2)

3.8.3 Set Package Status to Completed

Having completed all activities in the package as well as the corresponding application tests, you can close the package. When you do so, the system sets the final status for the package and deactivates it.

By closing a package for a test conversion, you confirm that the data was consistently transferred and state that the transfer of the productive system (or the next test conversion) can start. By completing a package for a productive conversion, you confirm that live operation in the system can be resumed.

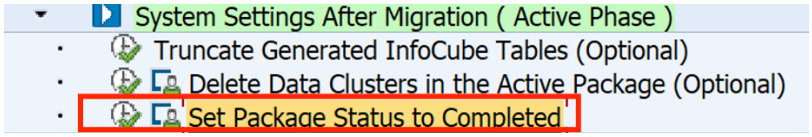


Figure 185: Set Package Status to Completed (1)

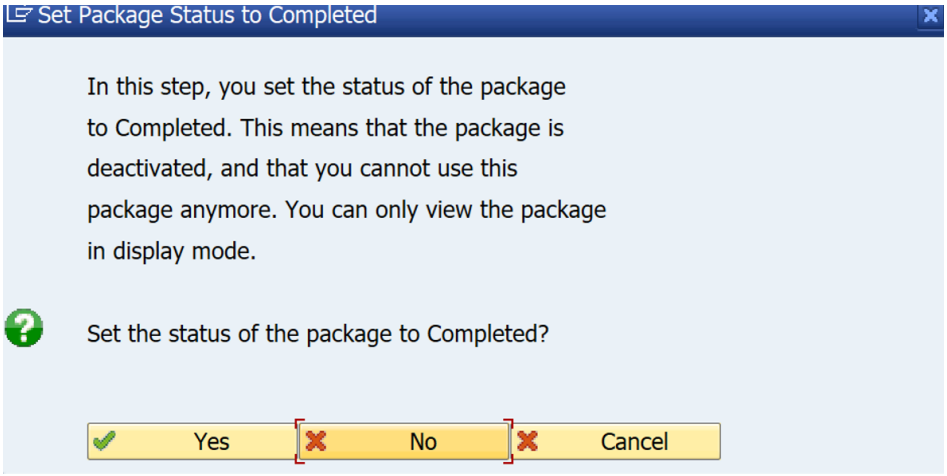


Figure 186: Set Package Status to Completed – Confirmation

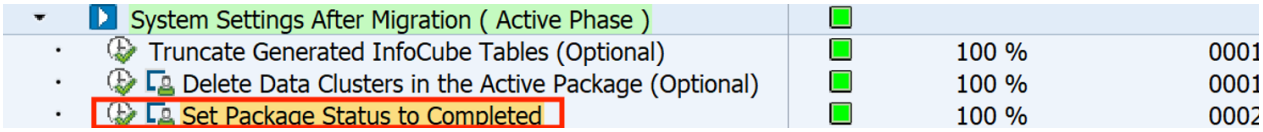


Figure 187: Set Package Status to Completed (2)

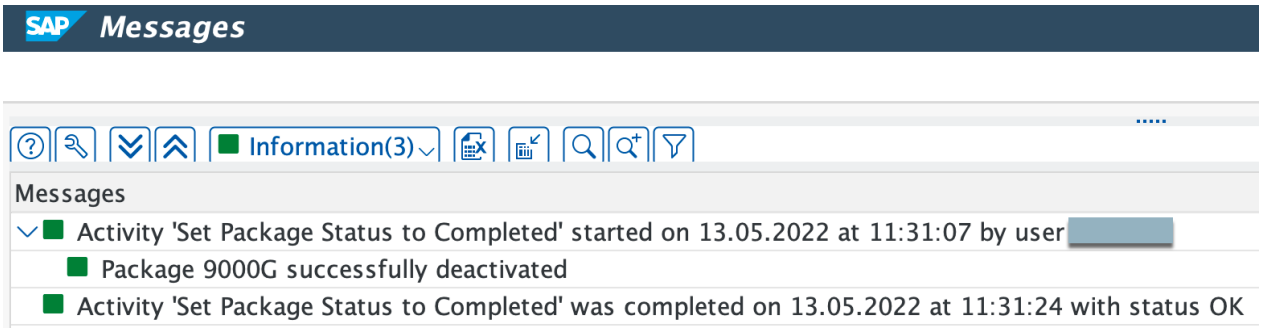


Figure 188: Set Package Status to Completed – Logs

The package and Task List run were successfully finished:

- Transaction CNV_MBT_PCT:

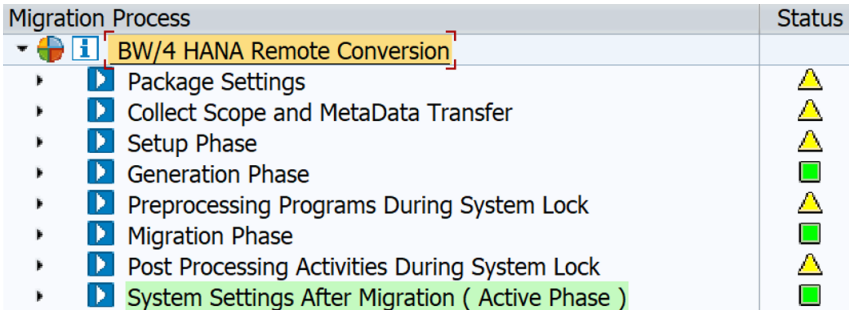


Figure 189: Package completeness in the transaction CNV_MBT_PCT

- Transaction STC02:

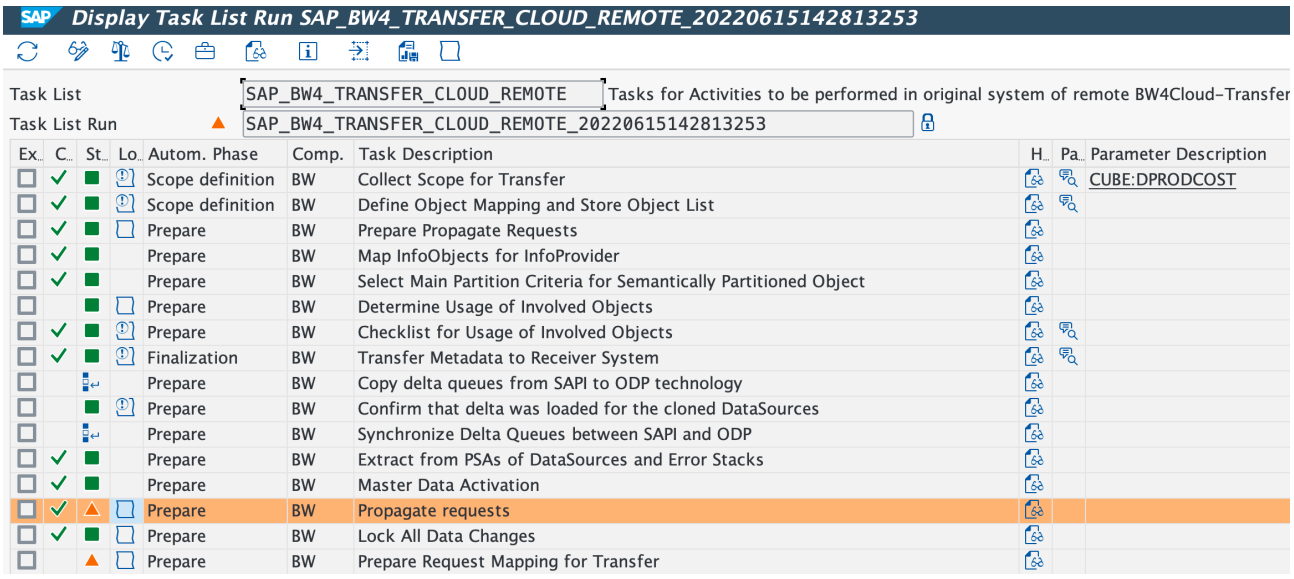


Figure 190: Task List completeness in the transaction STC02

⇒ Note: After the Remote Conversion is done, migration validated, and you want to start transporting to BW/4HANA PROD from DEV, you need to run the report RS_B4HANA_TRANSFER_REM_END from the sender system to remove the 'ACTIVE' flag in the table RSB4HSYSMAP. Otherwise each import of a transferred object will be considered a new transfer. Please take note of this action and align with the customer.

4 Datasphere Integration

When using the Remote Conversion for moving to BW bridge, the BW objects and corresponding business data which already exist in the BW sender are transferred to BW bridge and imported as remote tables to Datasphere (in the BW bridge space).

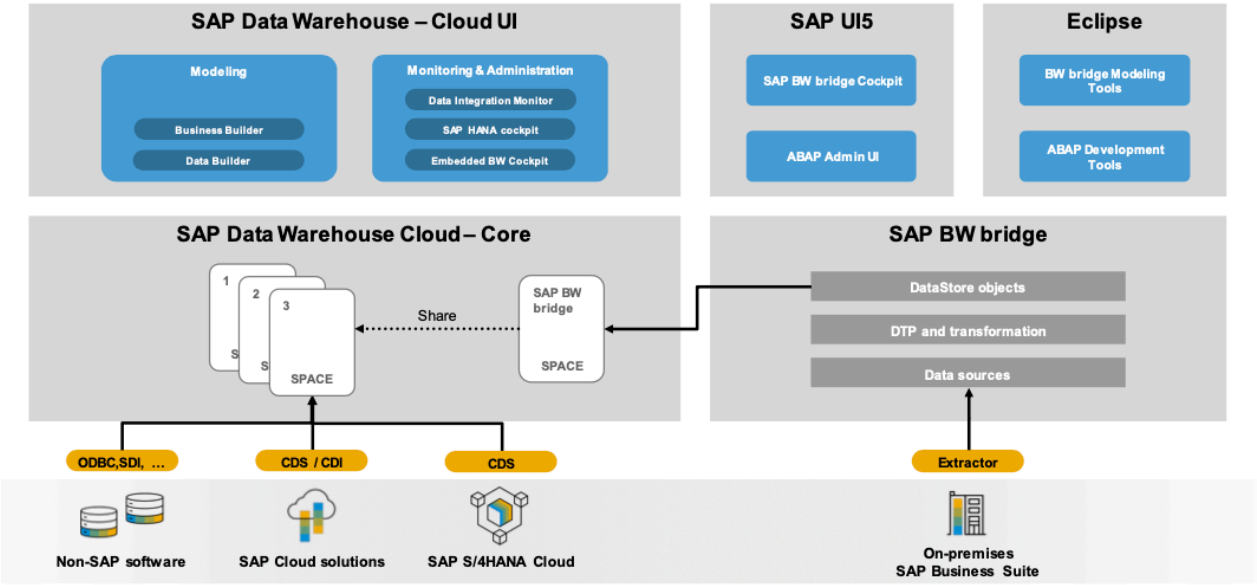


Figure 191: BW bridge, Datasphere integration

You can create as many spaces as desired in Datasphere (Space Type "SAP Datasphere"), however there can be only one BW bridge space (Space Type "SAP BW bridge").

As a feature of Datasphere, BW bridge enables the integration between the ABAP and cloud "worlds", by creating new objects directly in the BW bridge system or transferring them from on-premise BW or BW/4HANA systems.

Check our SAP Datasphere, SAP BW Bridge: Overview and Technical Deep Dive blog for further information.

4.1 Space Management Overview

In the Datasphere tenant, we have created two (2) spaces for our sample.

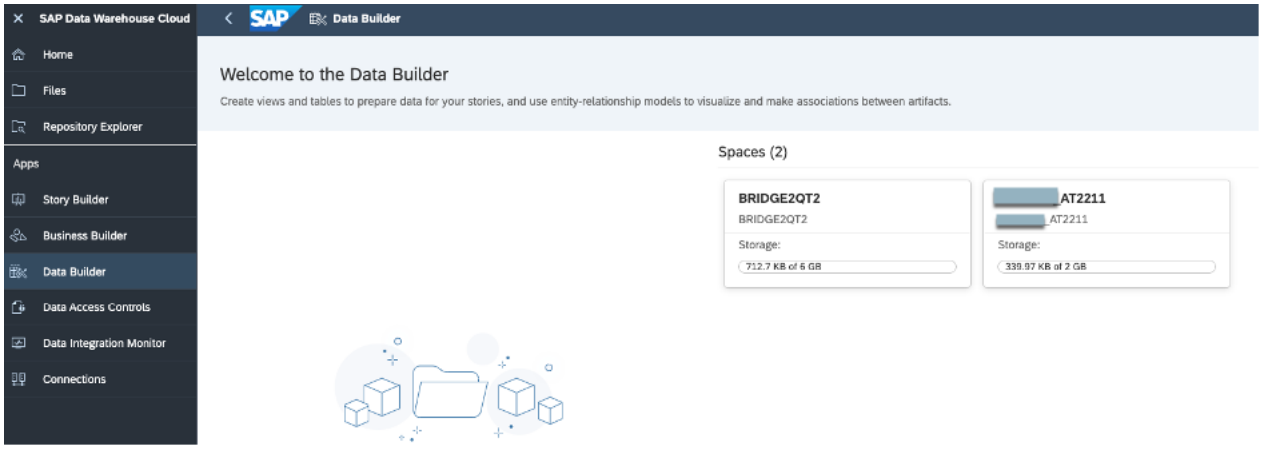


Figure 192: Datasphere - Data Builder

Space Management: BW bridge space:

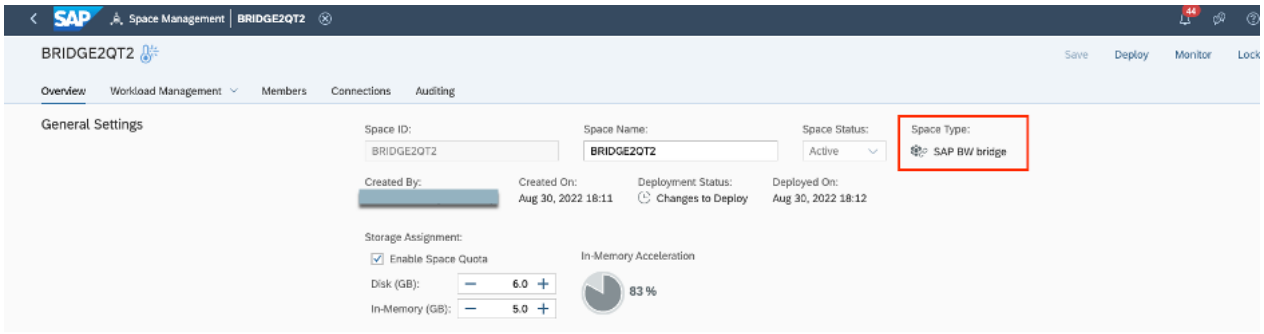


Figure 193: Datasphere - BW bridge space

Space Management: Datasphere space:

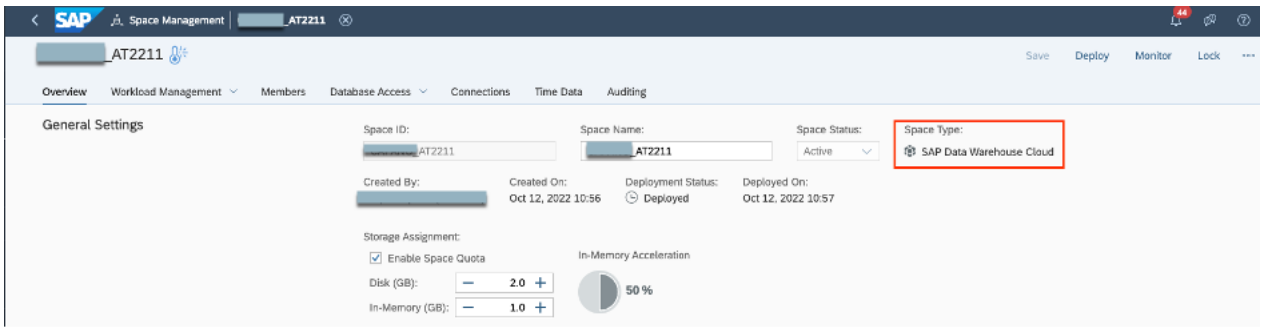


Figure 194: Datasphere - Datasphere space

4.2 Data Integration Monitor: BW bridge Cockpit

The SAP BW bridge cockpit can be accessed either through the SAP Data Integration Monitor of Datasphere or the BW Modeling Tools (BWMT) in Eclipse/HANA Studio.

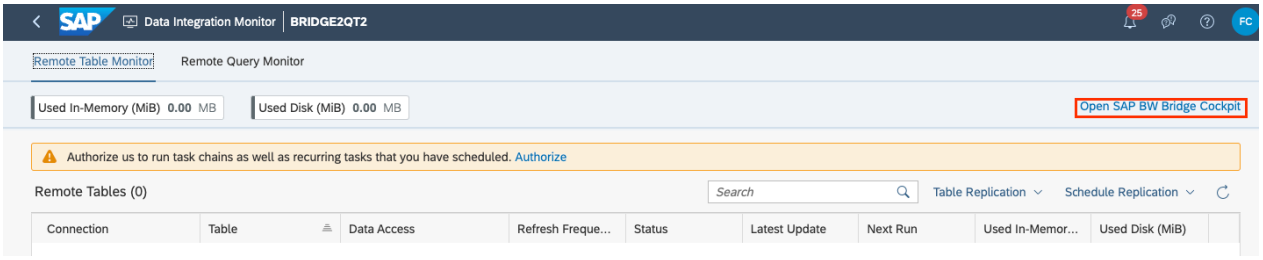


Figure 195: Datasphere - Data Integration Monitor

Under “DataStore Objects” you can see the two (2) InfoProviders which were transferred with our sample (CPRODCOST and DPRODCOST).

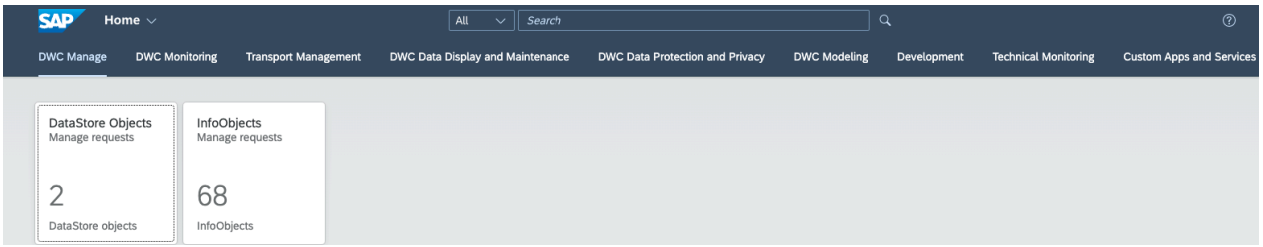


Figure 196: BW bridge cockpit

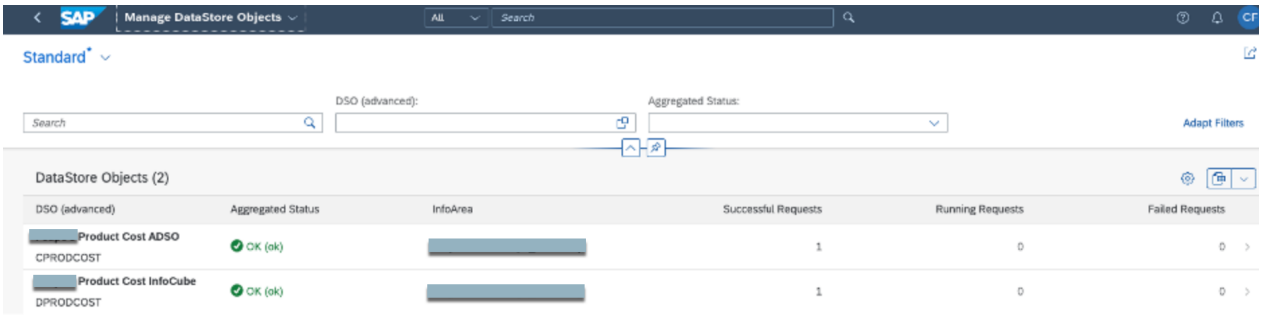


Figure 197: BW bridge cockpit - DataStore Objects

4.3 Data Builder: Import Entities and Space Sharing

- Go to the Data Builder and select “Import Entities”.

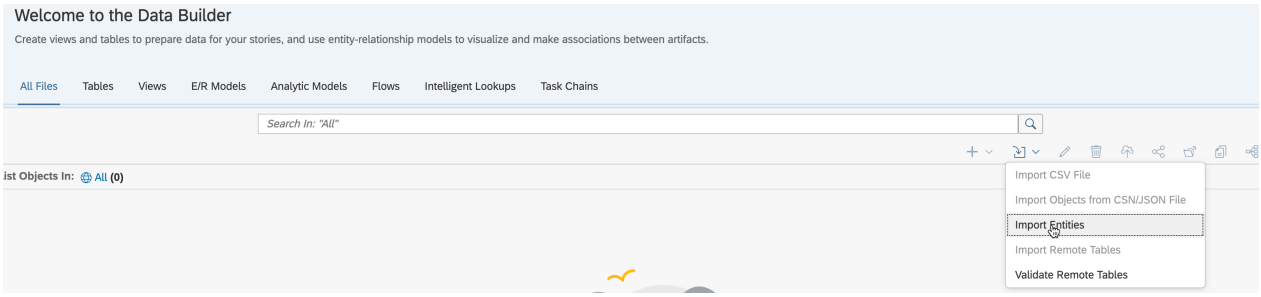
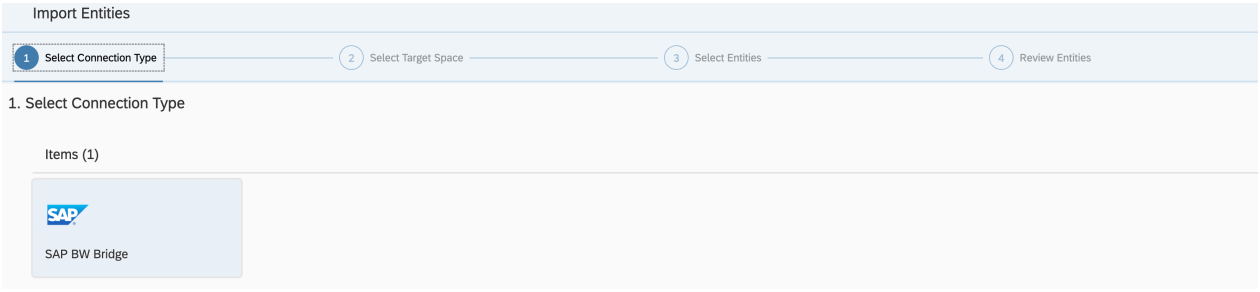


Figure 198: Datasphere - Import Entities in Data Builder (1)



The following DataStore objects (ADSOs) and InfoObject tables are the ones used in our sample and transferred to BW bridge.

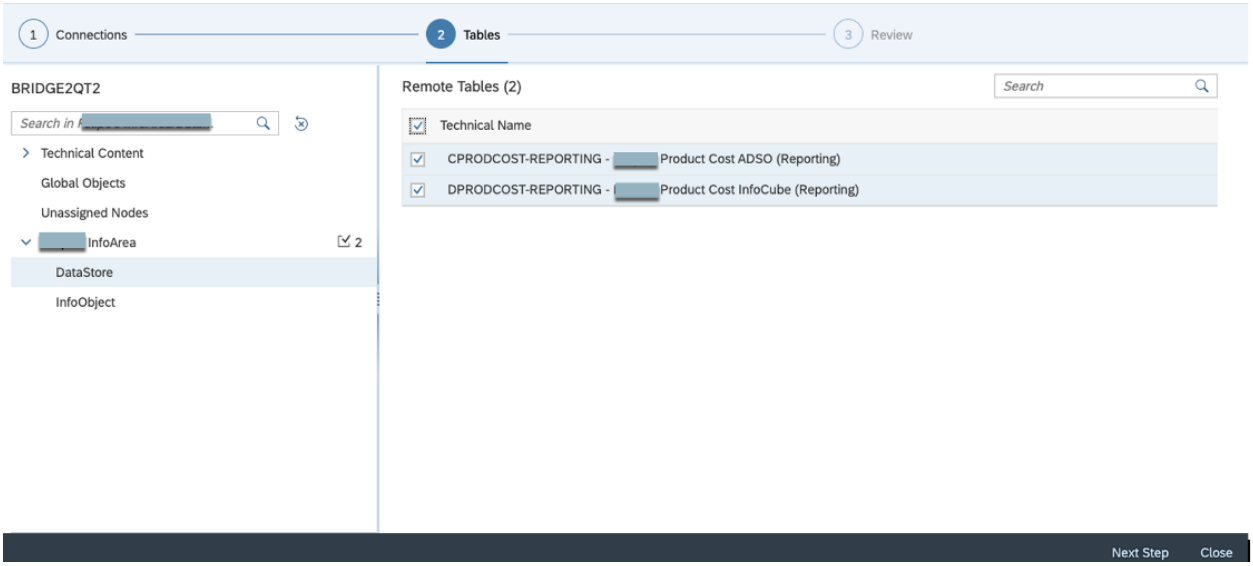


Figure 199: Datasphere - Import Entities in Data Builder (2)

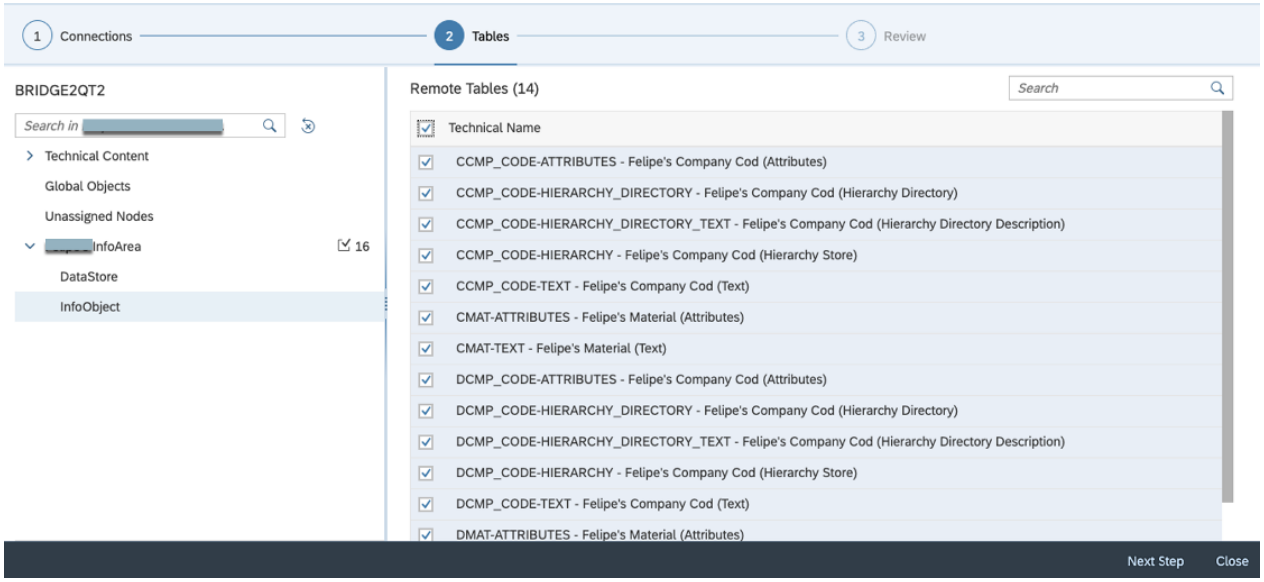


Figure 200: Datasphere - Import Entities in Data Builder (3)

It is possible to maintain the “Business Name” and “Technical Name” for the tables to be imported.

Select “Import and Deploy”:

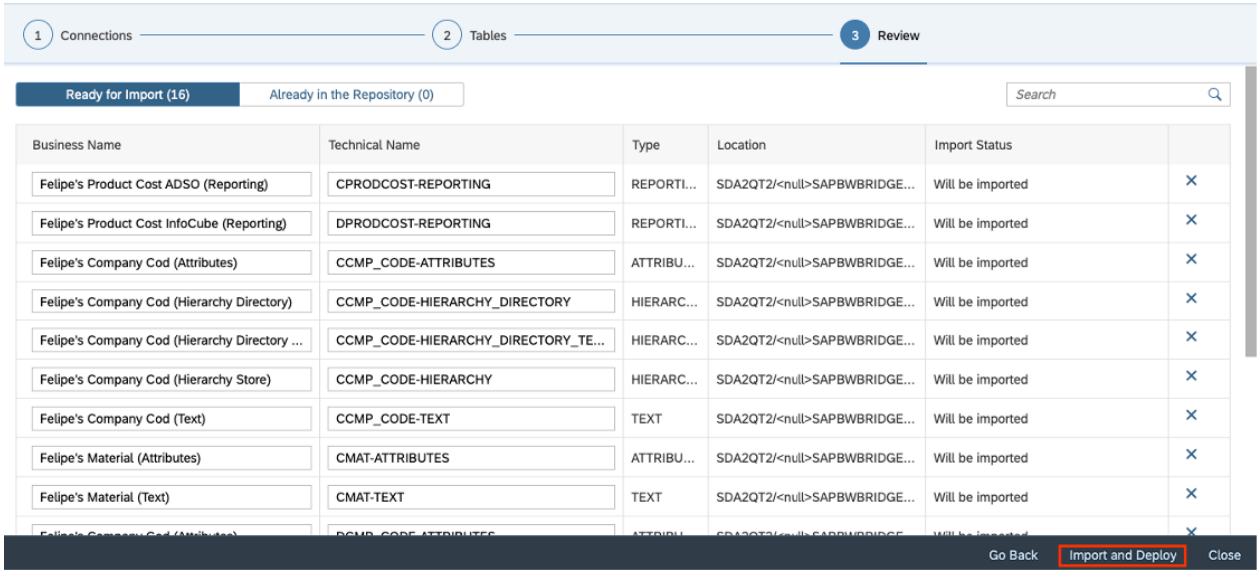


Figure 201: Datasphere - Import Entities in Data Builder (4)

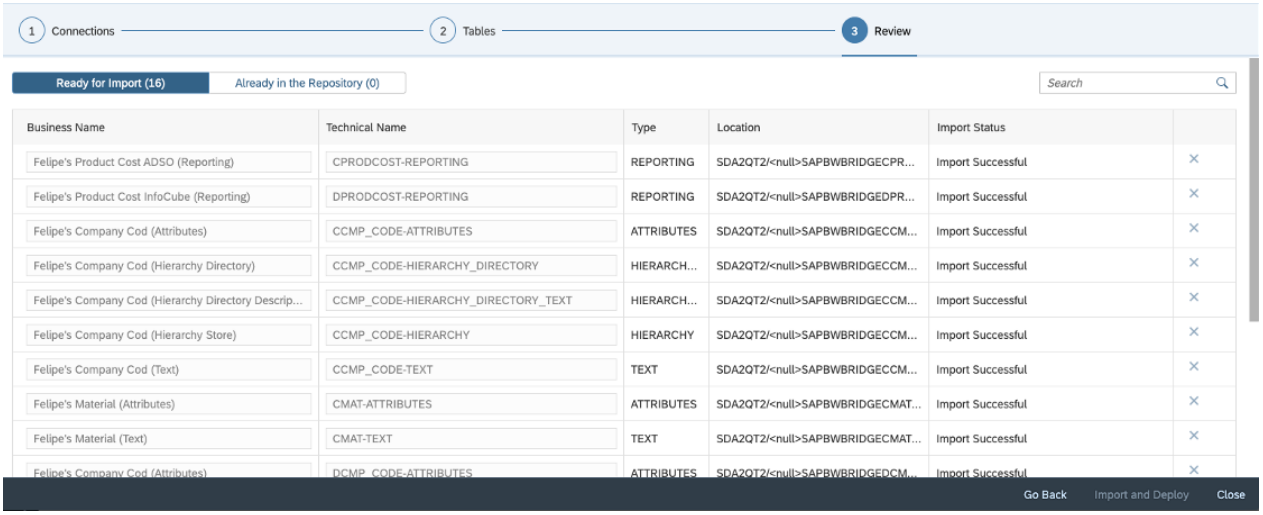


Figure 202: Datasphere - Import Entities in Data Builder (5)

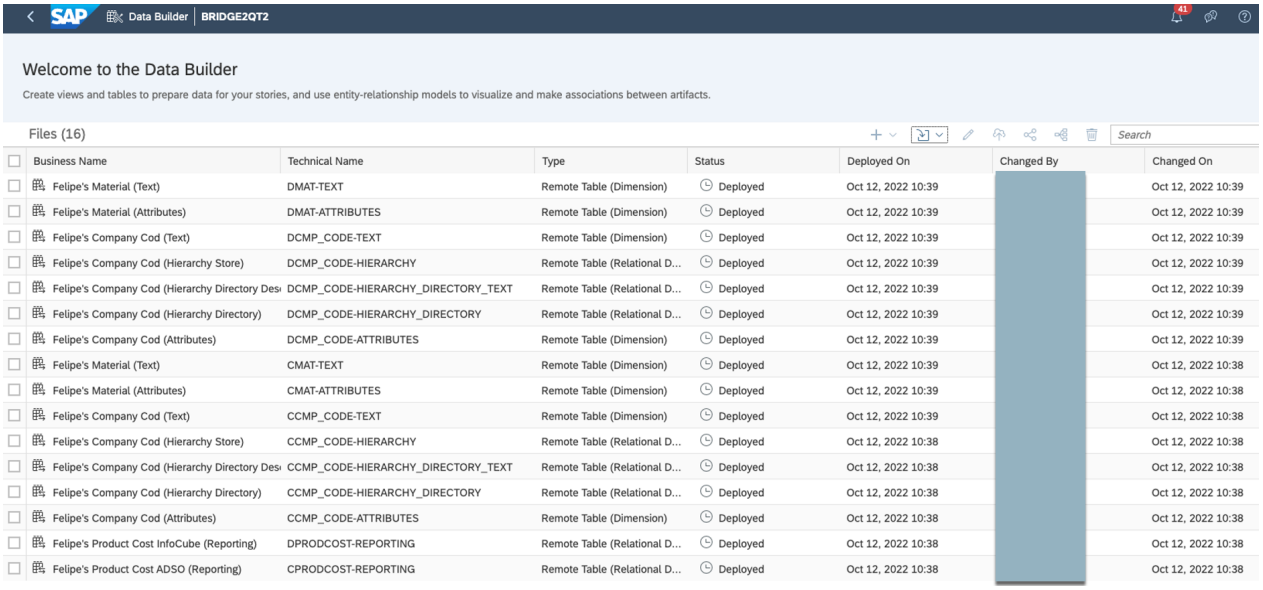


Figure 203: Datasphere - Import Entities in Data Builder (6)

You can display the transferred data under “View”:

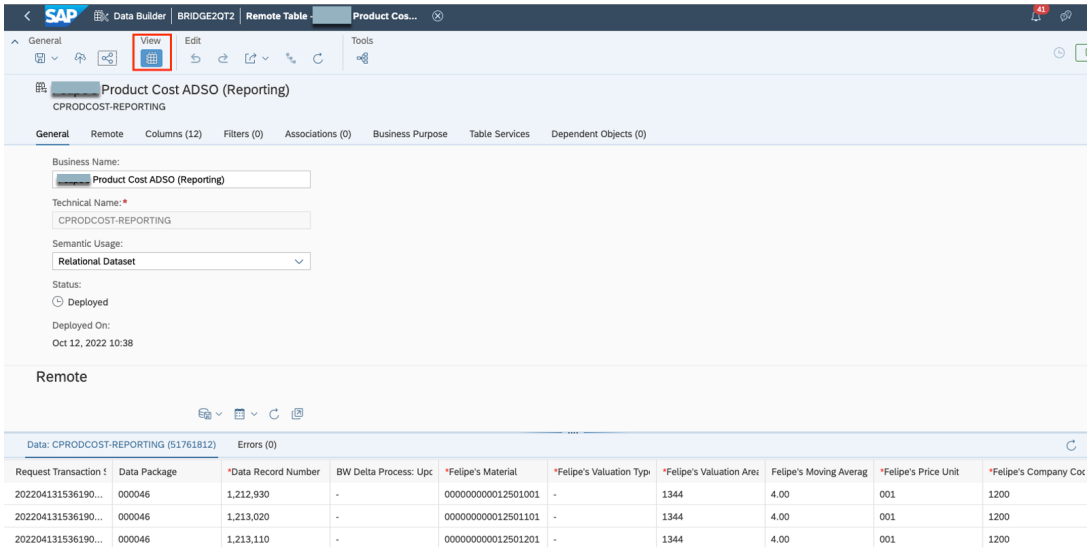


Figure 204: Datasphere - View data

After the deployment, the imported remote tables can be shared with other spaces. Below we are sharing them with the Datasphere space which was created for this sample.

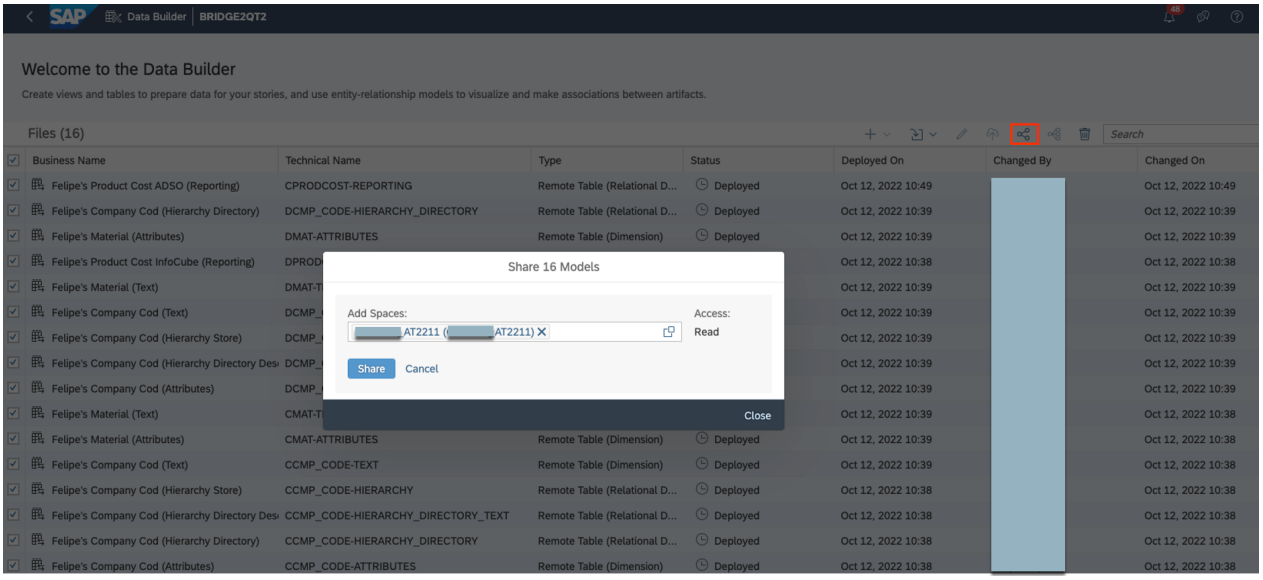


Figure 205: Datasphere - Space sharing

Now it is possible, for example, to create graphical views to be consumed in the Datasphere space.

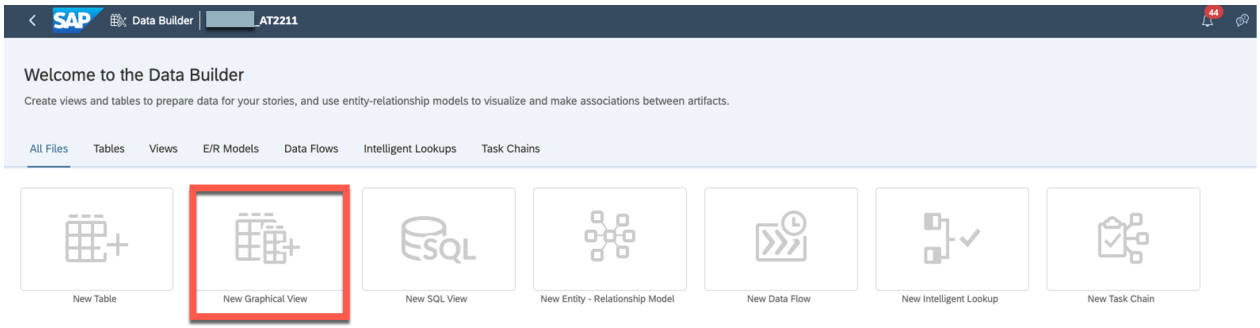


Figure 206: Datasphere - New Graphical View in Data Builder (1)

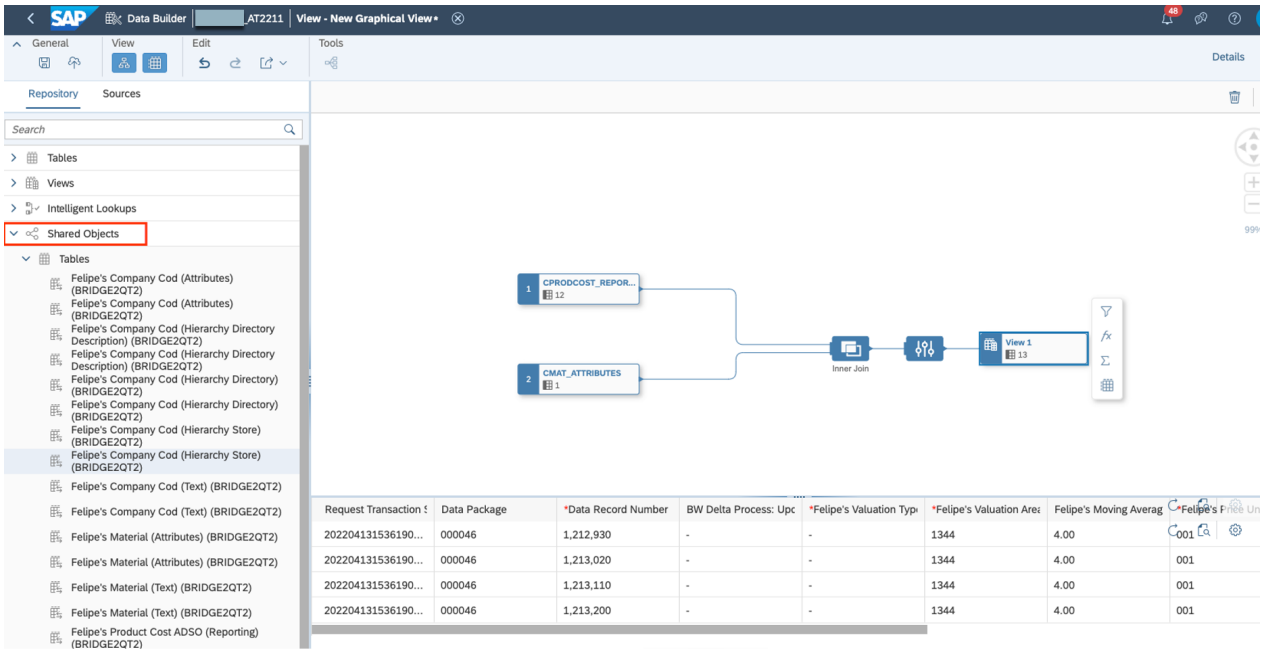


Figure 207: Datasphere - New Graphical View in Data Builder (2)

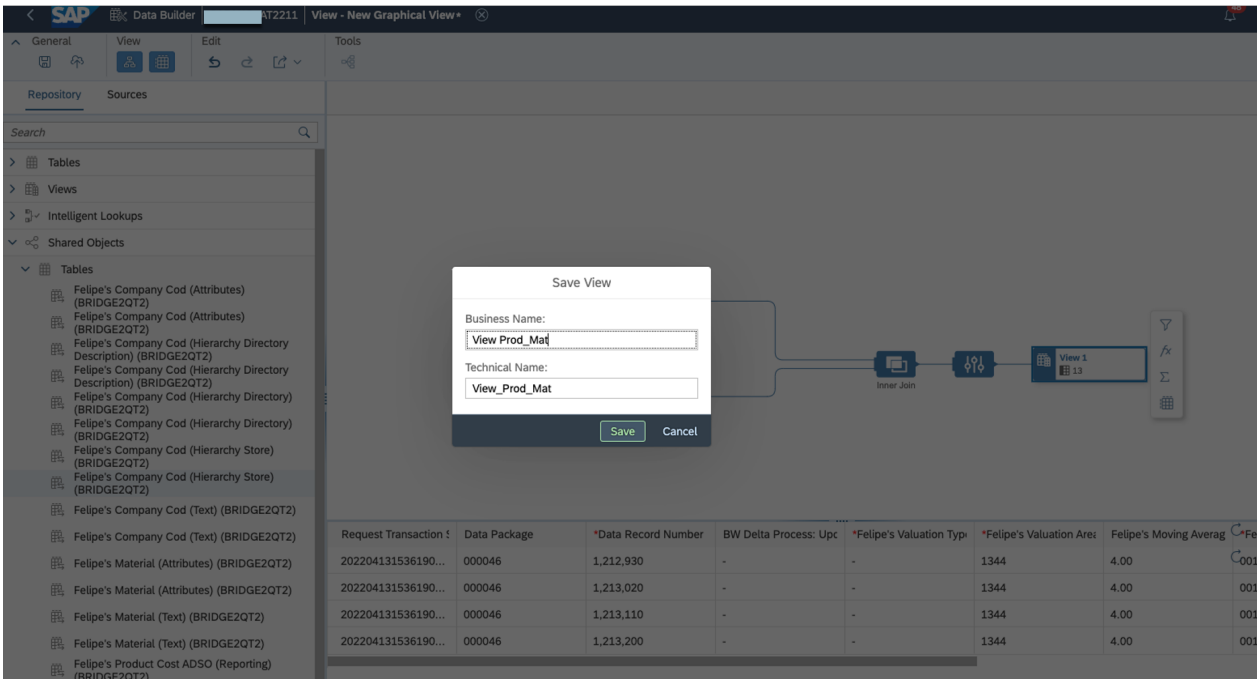


Figure 208: Datasphere - New Graphical View in Data Builder (3)

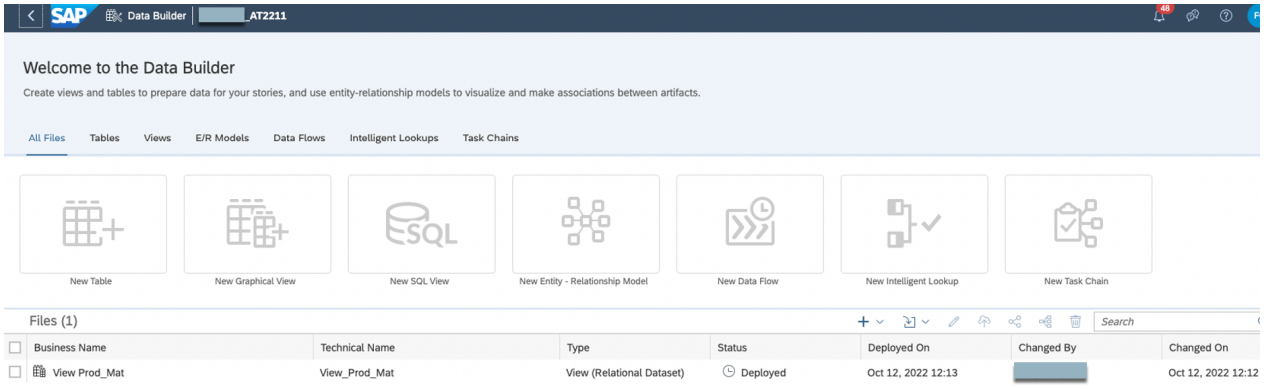


Figure 209: Datasphere - New Graphical View in Data Builder (4)

Check our Webinar, which contains a demo about the Remote Conversion to BW bridge.

5 Data Transfer to a 2-tier Tenant Landscape

When having a SAP Datasphere, BW bridge landscape with one development and one productive tenant, it is assumed that you wish to populate this productive tenant with the same data from your SAP BW productive system. For that, it is required that the SAP BW productive system is used as the sender system for the data transfer process.

The data transfer process to a 2-tier tenant landscape basically consists on 3 main steps:

1. Transfer the metadata from the SAP BW productive system (sender) to the SAP BW bridge development system (receiver).

2. Transport the metadata from the SAP Datasphere, BW bridge development system to the SAP Datasphere, BW bridge productive system via git-based CTS (gCTS).

3. Transfer the data from the SAP BW productive system (sender) to the SAP BW bridge productive receiver system (receiver).

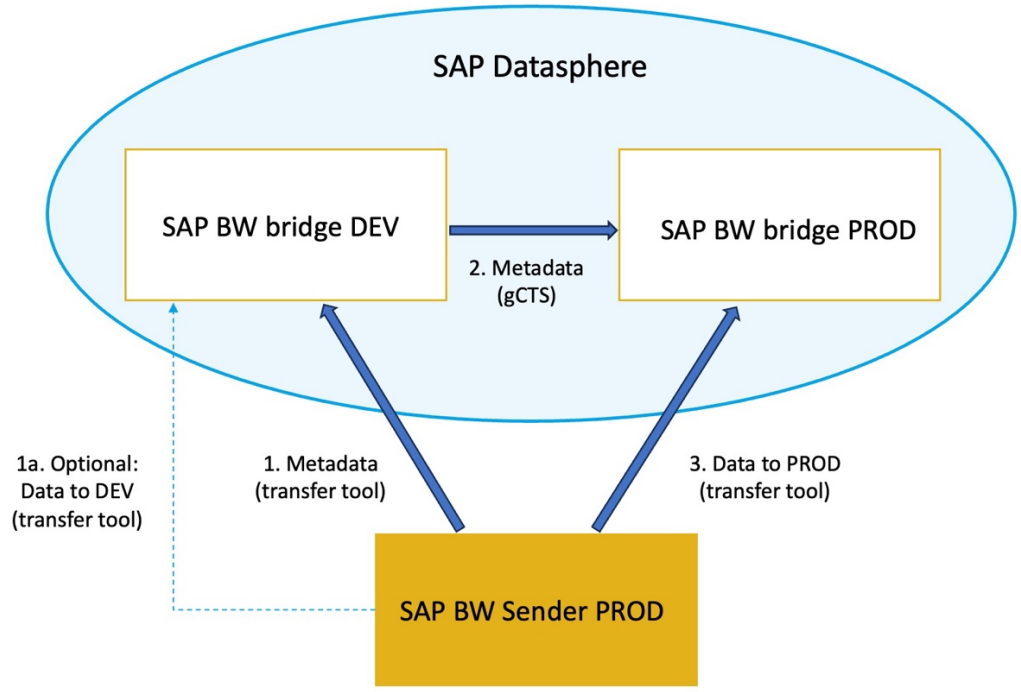


Figure 210: Data transfer to an SAP Datasphere, BW bridge 2-tier tenant landscape

For further details about the preparation activities and the data transfer steps, please check the blog *SAP Datasphere, BW bridge: How to transfer data from SAP BW or BW/4HANA to a 2-tier tenant landscape using the Remote Conversion*

© 2025 SAP SE or an SAP affiliate company. All rights reserved.
See Legal Notice on www.sap.com/legal-notice for use terms,
disclaimers, disclosures, or restrictions related to SAP Materials
for general audiences

