Dual Maintenance - WorkBook

SSST Internal - DM Working Procedure

Agenda

01 Introduction - Retrofit

02 Scoping

03 Service Scope Objects

04 Tool Scope Objects

05 Open Topics for Review / Discussions

Introduction - Retrofit

- SmartShift Retrofit solution is designed to merge the new developments and changes from one development environment to another parallel development environment.
- There are 2 types of retrofit supported by Smartshift,

1. Offline Retrofit
2. Online Retrofit

Scoping

Service Scope Objects

- All Object types OTHER THAN programs, includes, function groups, FMs, class methods,(public/protected/private) sections of class can be considered as a Service scope category.
  Meaning, the object types which have No/less code and less scope of smart (tool) rule to be applied.

- Some of the common object types under Service scope are:
  - ➢ DDIC objects like: TABL, TABD, VIEW, DTEL,TTYP, DOMA etc….
  - ➢ Text related stuff like: DOCU, TEXT, REPT, MSAG/MESS etc….
  - ➢ Generated objects like: IWPR, DYNP
  - ➢ Form objects like: SSFO, FORM, SFPI, SFPF etc….

- Most of the service scope object have versions, we can compare the versions to identify the Scope TR change.
  Few object types like Smartforms(SSFO), Scripts(FORM), Texts(TEXT), Documentations(DOCU), Smartstyles(SSST) doesn’t have versions - However we can still make use of SE03 TCode and consider them for ‘Safe to Import’ analysis.

- Using RFC connection from target to source, We can either make use of standard program RSSYSCOMP to do a mass comparison OR Open the object on Individual TCodes and do a Remote comparison(use ‘Fetch Remote versions’ button) to source to identify the difference between source and target.

  Few object types are not supported in RSSYSCOMP. Example - DOCU, MESS, FORM, TEXT(you can execute your assigned object types and try if it provides any output).

  Note: RSSYSCOMP compares Source Active to Target Active version; whereas when you open the object in individual TCodes and do Remote comparison → you have the flexibility to compare ANY target version to ANY source version.

- Very important Point to remember: See that Standard objects are not changed as part of Retrofit.

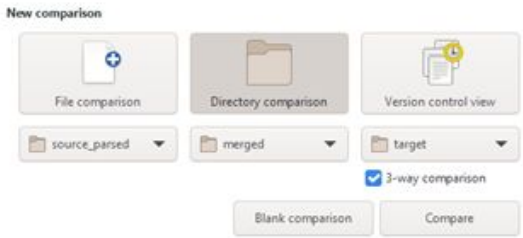
  Example - MATNR field length difference between ECC(length-18) and S/4(length-40) should be ignored when you are Retrofitting service scope objects.

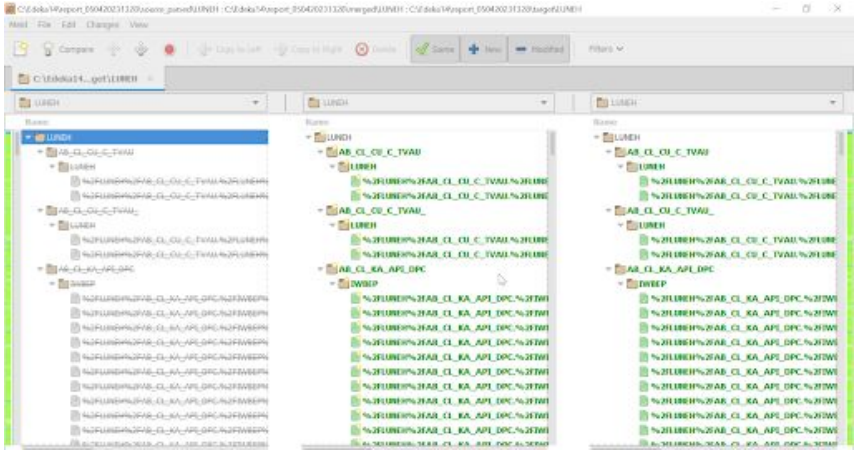
Tool Scope Object

- The objects which can be analyzed, parsed and upload back to the system by the smarttool is categorized into tool scope object. It can be said that all the repository objects are tool scope objects.
- Objects like programs, function groups and classes fall under this category.
- In case of offline retrofit, an xml extract of the system is downloaded using the extractor utility and handed over to the tool team for parsing.
- Once the xml extract has been parsed, we would have 3 folders such as source parsed, merged and target folder containing the different tool scope objects
- **source_parsed** – The code from source system will be parsed by the tool and respective tool rules has been applied to it
- **Merged** – The tool will analyze the source and target code of the objects and provides the merge code which will then be analyzed and suitable action will be taken by the developer during retrofit.
- **Target** – This folder contain the code extracted from the target system

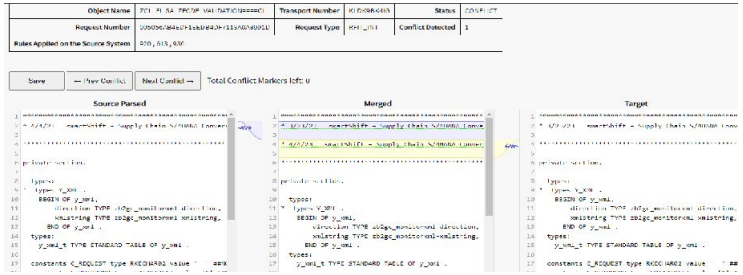
For an offline retrofit, we use an application called MELD for the retrofit.

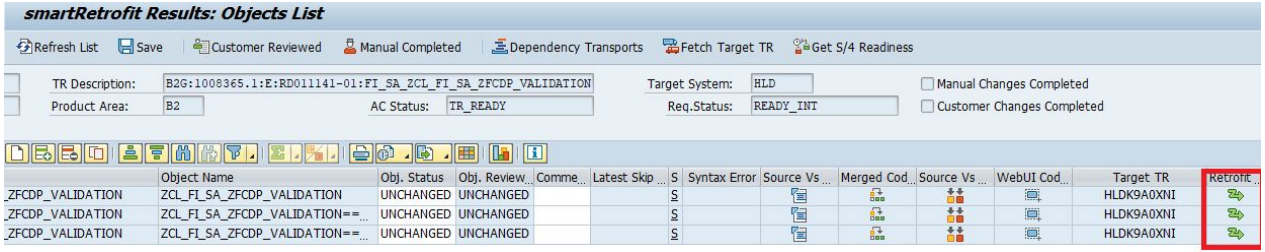
- Meld provides different ways to compare an object but in our case, we use 3 way comparison.
- All 3 folders need to be selected and submitted to the meld. Once the upload is completed, we will be able to open an object in 3 way comparison mode consisting of source parsed code, merged code and target code



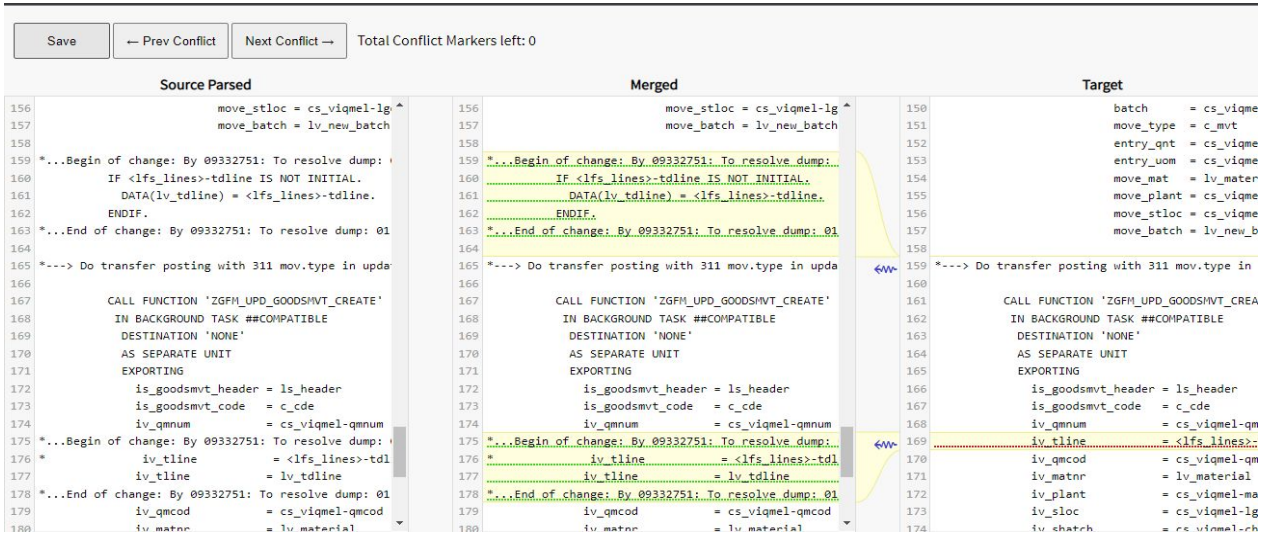


- The developer will next retrofit the code in the meld comparison's merged portion.
- After an object has been retrofitted, the merged code will be copied to the target system.
- In an online retrofit, we will make use of WebUI, which is available on the smartretrofit dashboard **/smash/rfit** which performs the same function as Meld.
- Once the retrofit for an object is completed, the code can be saved in WebUI. After that, it should be copied to the target system by clicking the retrofit done button.

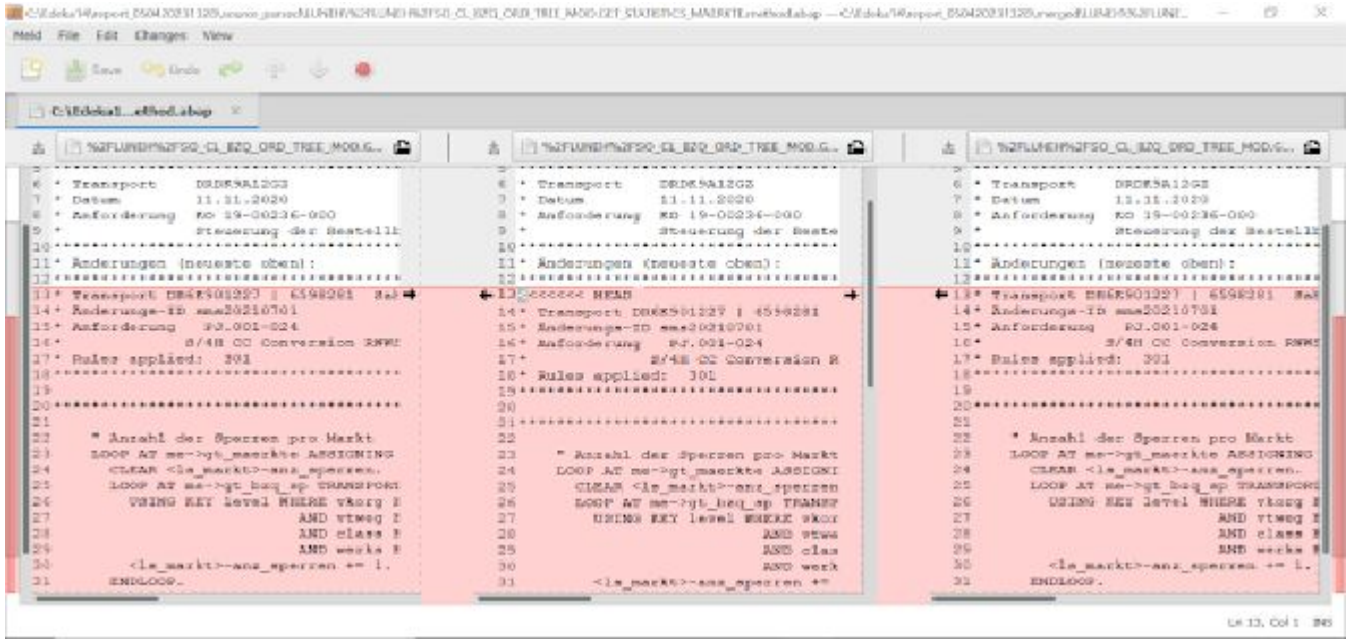




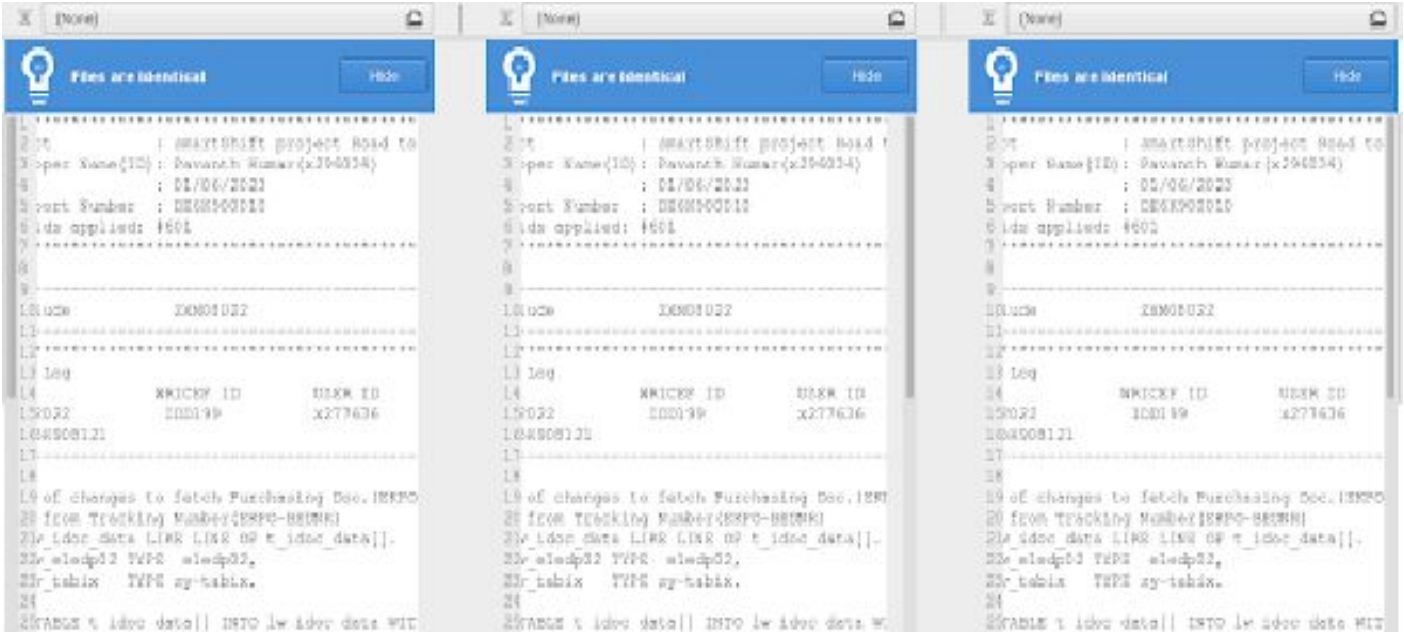
- There are 3 merge cases available both in offline and online retrofit.
  1. Automerge Case
  2. Conflict case
  3 Unchanged/Identical case
- **Automerge case**: Based on the git repository, the tool will analyse the code from source and target system and merges the relevant code automatically into the merged section without any manual intervention.



- **Conflict case**: The code section which has difference/changes both in target and source code and the tool needs some manual assistance to complete the merge is categorized under conflict cases. Here, the developer will analyse the code difference and selects the relevant code to be merged for the final version.



- **Unchanged/Identical case:** If there is no changes found between source parsed code and target code, then it would be a no action case.



- Once the retrofit is completed for all objects in scope, we will then run manual remediation checks on all the objects.
- Manual remediation would include UC , EPC , HCC and S4 checks.
- The objects can be said retrofitted only after both retrofit and remediation is completed.

Akshay

IDOC	IN	Service	Akshay
IEXT	IN	Service	Akshay
CHDO	IN	Service	Akshay
IWMO	IN	Service	Akshay
IWPR	IN	Service	Akshay
IWSV	IN	Service	Akshay
IWOM	IN	Service	Akshay
IWSG	IN	Service	Akshay
SFSW	IN	Service	Akshay

IWPR

Gateway Business Suite Enablement - Service Builder Project

- SAP OData is **a standard Web protocol used for querying and updating data present in SAP using ABAP**, applying and building on Web technologies such as HTTP to provide access to information from a variety of external applications, platforms and devices. In SAP, we use SEGW transaction code to create an OData Service.

- IWPR is the object type for the OData projects and services. The Tcode SEGW (SAP Gateway Service Builder) is used to view/change IWPR objects.

- There are different components available in IWPR objects some of the common components are Data Model, Service Implementation, Runtime Artifacts and Service maintenance.

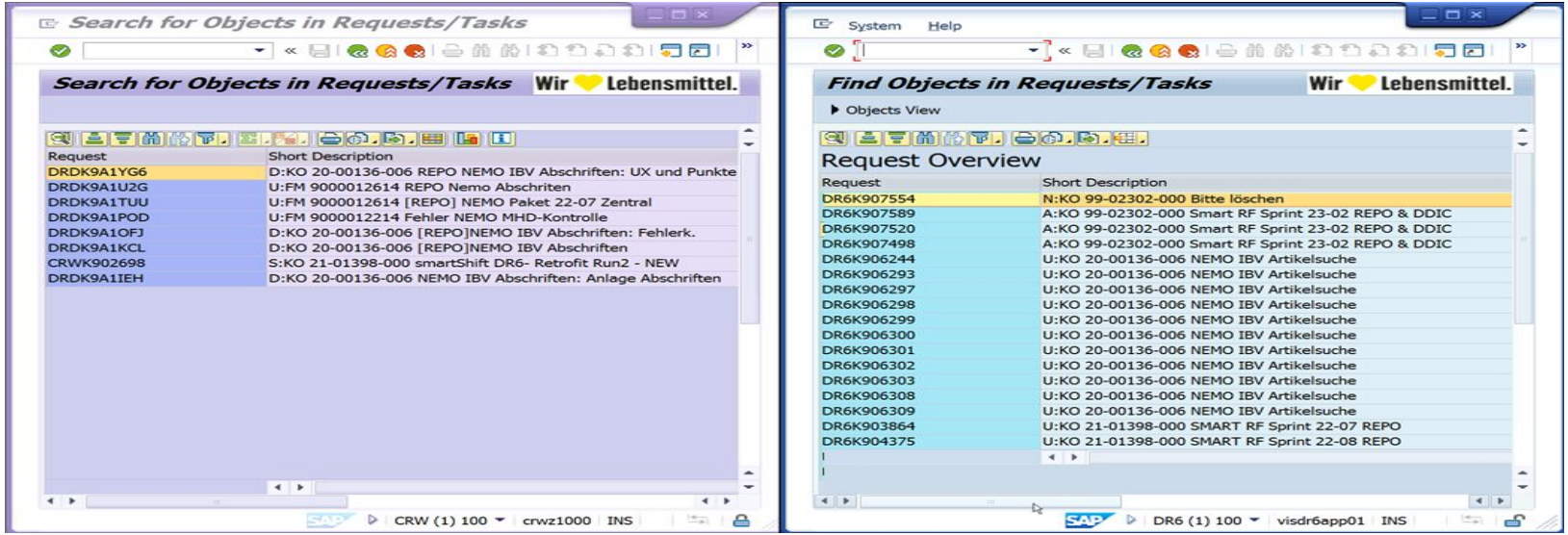
- We as a part of retrofit working on the Data model components except model referencing.

- Since we don't have a version management tool for this object type we need to do the retrofit carefully without overwriting any target changes by analysing the DPC and MPC classes of the IWPR object.

- IWPR objects are not part of the safe to import. Our scoping utility will not give existing/NEW information for IWPR objects, so first need to check the existence of the object in target system manually(go to SEGW Tcode and try to open IWPR).
- If NEW then mark it for ToC(so that we import the object via ToC to target); whereas for existing we need to check manually which is discussed in following slides

SE03 Analysis(for Already existing objects)

- The first step in the retrofit process of IWPR object type is checking the version information in both source and target system using Tcode SE03
- Here we will check whether any additional version present above our scope TR in the source system and also will check for the any customer changes in target system
- Sort the date field in the descending order. In the below example, CRW is the source system and TR DRDK9A1YG6 is the latest version available and also our scope version. DR6 is the target system and we can notice TR DR6K907554 which is not smartshift retrofit TRs by observing the description.



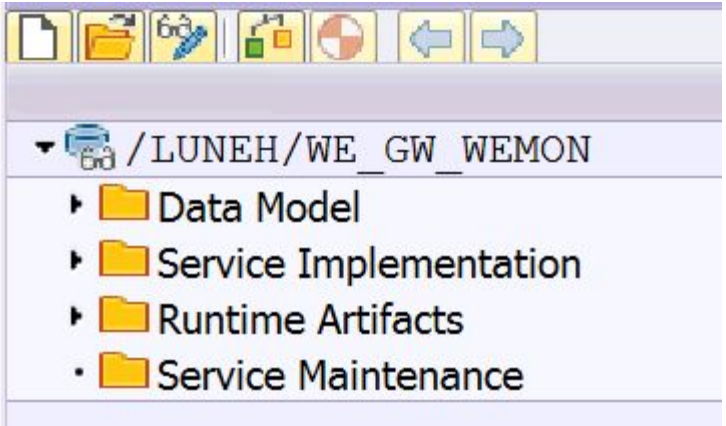
- As per the previous example there is a possibility for target level changes, so while doing the retrofit if there are additional changes present in the target system we need to cross verify it with respective DPC and MPC classes.

  Please note that we should not do any manual changes in the MPC and DPC classes.

Steps to retrofit IWPR

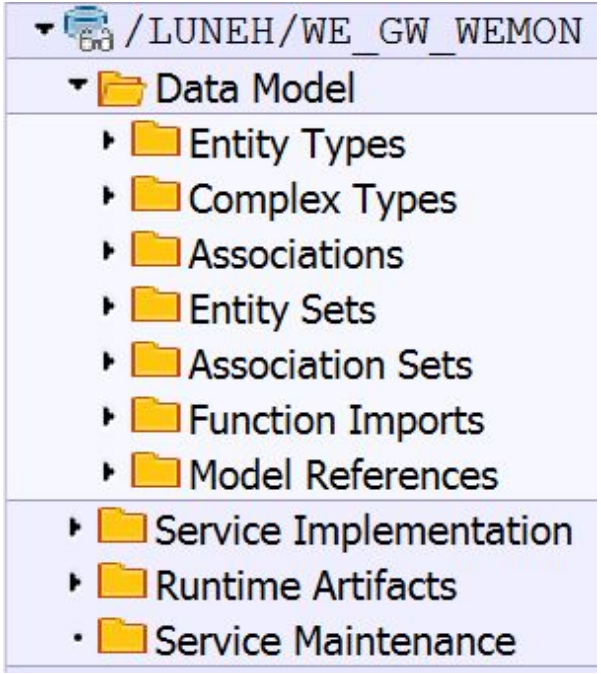
- Go to Tcode SEGW and click on open project icon. (second icon in the below screenshot). Please check the original language of project and login to original language before making changes.
- Use menu command Goto->Object Directory Entry to find original language

- Provide the object name in the pop-up and click on continue



- Kindly work on the display mode still you are finding some difference, then click on the display/change icon to make changes

- Open the project in both source and target system and then expand the data model



- We need to check all sub components of Data model except Model References

- Expand and double click on Entity Type in both source and target system
- Select all and copy the ALV of both source and target and compare it with any text comparing tools like Meld and WINmerge(check next slide for sample how the ALV screens look like)

[Assuming only source side changes exist for the object]

- In case of any changes in source then bring it to the target system by copying the additional entity type from source and paste it on the target ALV in the exact line (covered in next slides)
- Under entity type expand each sub tree
- Check each entity type - properties and navigation and repeat the same as explained in the previous slide
    - If there are any new additional changes in source, bring it to the target system
      (Kindly refer to next slide for the example)
- Check if anything needs to be changed in target system and analyse it with SE03 Tcode, DPC and MPC class of that particular project. At this stage, don't delete or change anything in target system
- Repeat the same steps for all other Entity Types

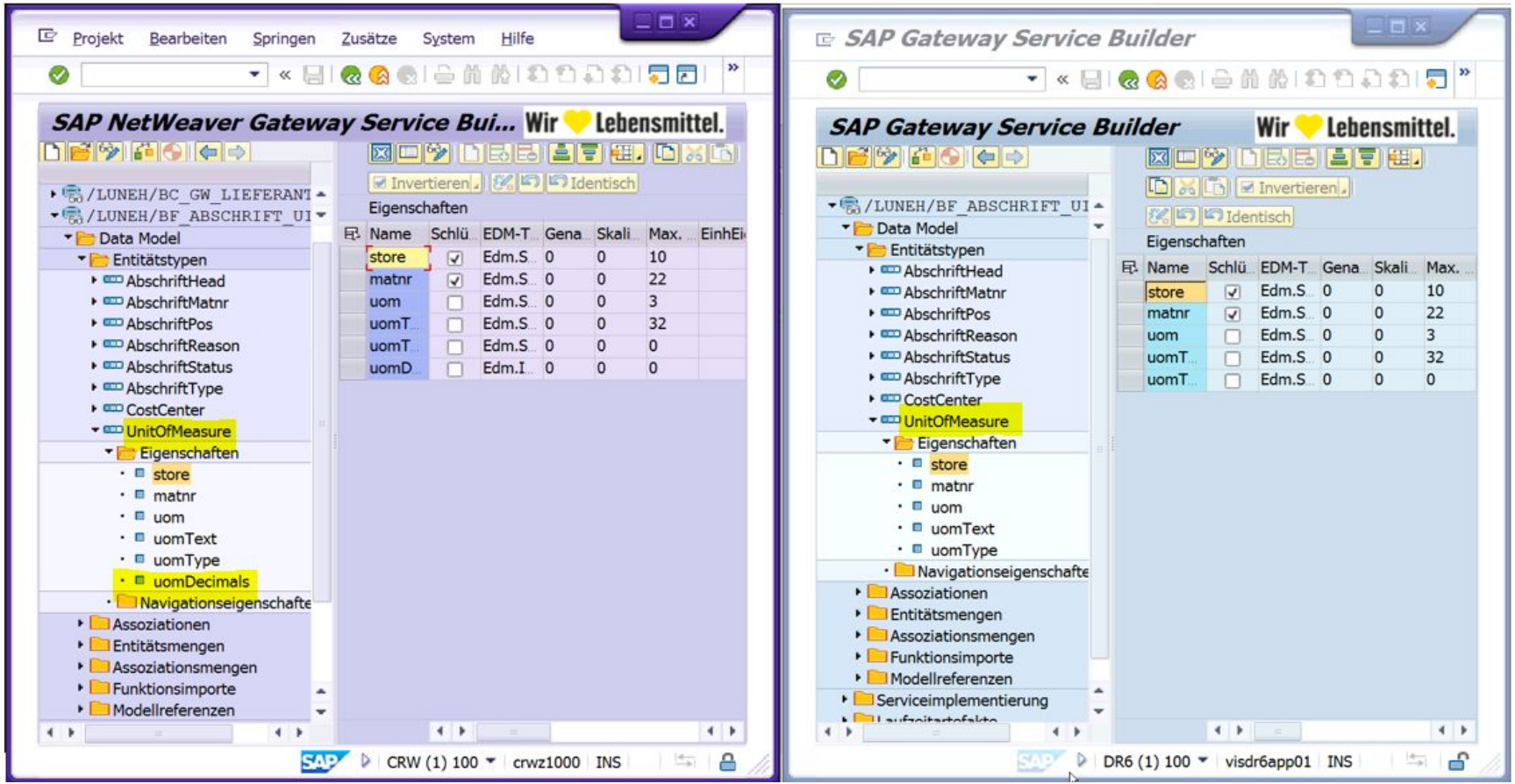


Image represent source and target system's ALV of Entity Type

- In the previous example, there is an additional property present for Unit of Measure Entity type which will be added into the target system.

Key 1 - To navigate between change and display

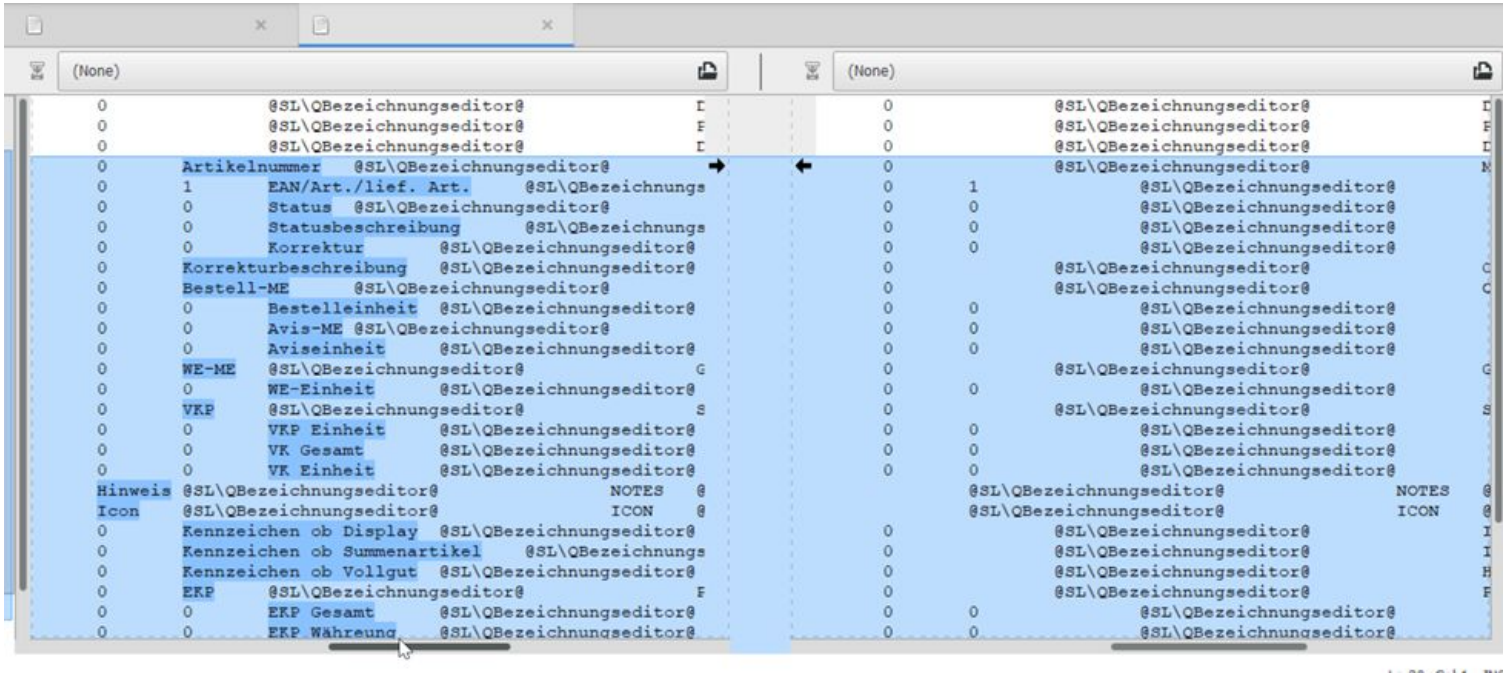
Key 2 - To append the row at the end of the list

Key 3 - To add the row at the particular line

- Follow the same process as explained in Entity Type for the below components(Complex type,Associations, Entity sets..in Data model)

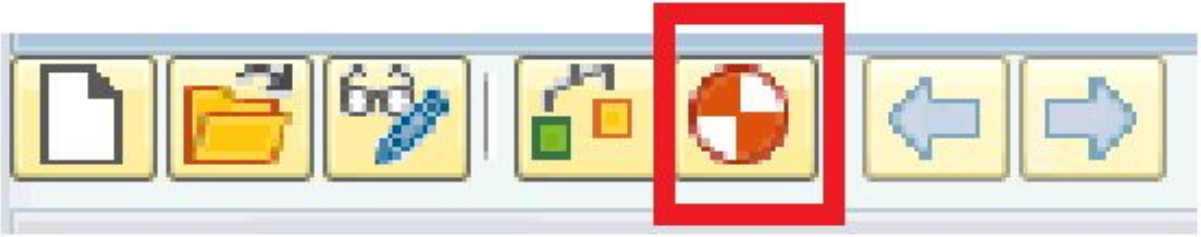
Reasons to check in Original Language

- The labels we are entering will be reflected in REPT only when we use Original language
- In below screenshot, the comparison of same property of source system in both original language(DE in this case) and EN is shown. We can see that labels are not maintained in EN.



Regenerate the project

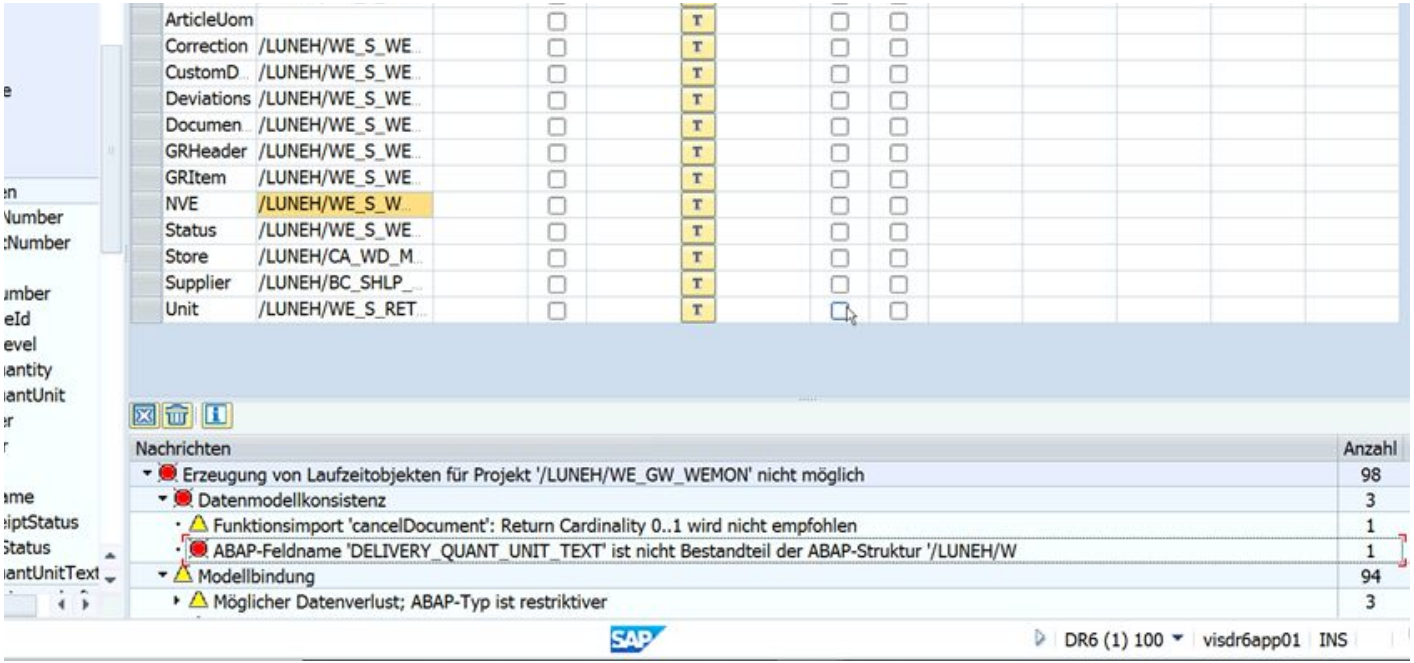
- Once the required changes have been made we need to regenerate the project.
- Click on the regenerate button and click on the continue button for all the runtime artifacts pop-up.



Regenerate button

- If the regeneration is successful, we will get success message else we will get the error message along with the failed generation message.

Common Error



- Some entity types are created by importing structure that means properties of entity type and fields of structure will be interrelated. So, let say you have added some additional property and those fields are missing in structure. Then we will get this ABAP-Field name error which will be resolved once we retrofit the corresponding structure.
- Please regenerate the project once the error is resolved.

If you find any conflict in screen to screen comparison, between source and target in SEGW Tcode, follow below steps:

How to verify using DPC and MPC class

- DPC and MPC class have methods for the every components in OData project
- For example take this METH UNITOFMEASURESET_UPDATE_ENTITY

  Here UNITOFMEASURESET is the entityType so, if we notice any target changes in this entityType then we need to check in methods where name starts with that entityType.
- Check version management of the respective method in both source and target system to confirm whether the extra change in target side is removed in source or added in the target system
- Same applies to all other components in OData project

IWSV IWMO IWOM IWSG

Supporting object types of IWPR

- The object types IWSV, IWMO, IWOM and IWSG are the backend support objects of the Odata Project.
- For this object types we will handle only the new entries in scoping sheet which should be imported via ToC
- This is applicable only if it is new in object status column and not applicable for the existing objects
- It is not part of safe to import

IDoc & IEXT

IDoc - Intermediate Document IEXT - IDoc Extension

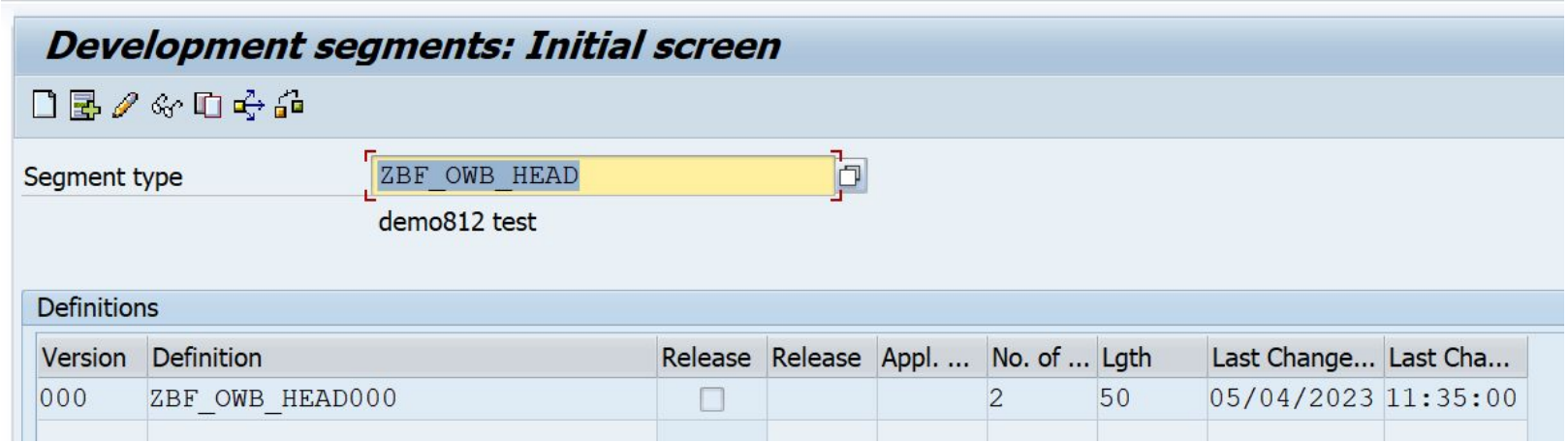
IDoc (intermediate document) is a standard data structure used in SAP applications to transfer data to and from SAP system applications and external systems.
For the retrofit process, if the IDoc or IEXT is NEW then we will create the object in target system manually instead of ToC
TCode: WE30
[We are not going with ToC import as we are not bringing the dependent TABU entries for Idoc related objects - segmenta/basic type/extension type]

Steps for retrofit process:

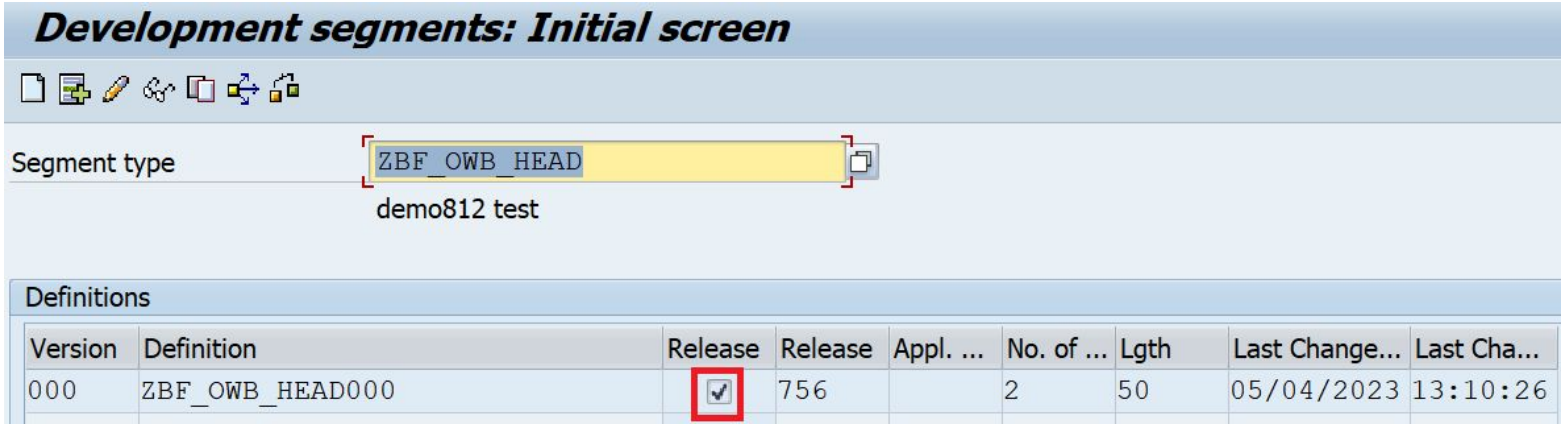
- Go to the TCode WE30 and give the object name
- For Object type IDOC, provide the object name and select the Basic type radio button
- For Object type IEXT, provide the object name and select the Extension radio button
- If you are creating an IDoc or IEXT then click on create button else click on display
- If required segment is already present in target system then add it else create manually

Creating NEW IDoc IEXT and segment in target system

- Once you create an IDoc or IEXT next is to create the segment. Please check in scoping sheet whether we have TABL entry for the segment in our scope. If so then proceed with creation else check with customer once.
- Go to TCode WE31 and enter the segment name, then click on create segment button
- Give all details as per source system and click on the save button
- Please be noted, if segment is not released in source keep it in that way in target as well



- If segment is in released status in source, then to set the release in target follow the menu command Edit->Set Release from menu bar
- [if direct set/cancel release option doesn't work, go for editing the dependent table via Debug option explained later.]



- Follow the same procedures to add remaining segments as per source system into target system
- There are two options to add segments based on hierarchy namely Add segment type as child and Add segment type at same level
- Select appropriate options to add the segment into IDoc or IEXT

For Existing IDOC or IEXT:

- For existing IDoc's and IEXT's follow the same procedure which we discussed in the previous slides(except creation) and add the source changes to the target system.
- To add new source changes, click on change button
- If you are unable to make the changes then check the release status in target system
- If status is released then change the status to Cancel release from menu bar option, If direct cancel release doesnt work, follow the below debug method.

Steps to set/release via Debug option :

- Open EDISDEF table in SE16 and give the segment name in SEGTYP field and display
- Enter debug mode by giving /h in command box
- Change the value of CODE from SHOW to EDIT and execute
- Delete the closed value X and save
- Now, the check in WE31 and flag X in release status will be removed
- Do the changes required and save then release the segment

Tcode and Table to set/cancel release Idoc segment, Basic type and Extension w.r.t Retrofit process

idoc related	Object Type	Tcode	Table to set/cancel release (if Direct option didn't work)
segment	TABL	we31	EDISDEF
basic type	IDOC	we30	EDBAS
extension type	IEXT	we30	EDCIM

Haritha

SHLP	IN	Service	Haritha
ENQU	IN	Service	Haritha
ENQD	IN	Service	Haritha
SHLD	IN	Service	Haritha
TOBJ	IN	Service	Haritha
NROB	IN	Service	Haritha
SUSK	IN	Service	Haritha
SUSO	IN	Service	Haritha

SHLP & SHLD

SHLP - Search help , SHLD - Search help Definition

- Format of Scoping Sheet
  - LIMU -SHLD -Object name , R3TR -SHLP- Object name
- About SHLP :
  - SHLP (Search Help) is a data Dictionary object type .
  - As part of Retrofit SHLP & SHLD will come Under Service scope .
  - Already existed DDIC objects are not Consider for ToC(Safe to Import), it will be fixed manually. As version management is available, we can compare versions and decide the changes to be made. Also, we can check them using prog **RSSYSCOMP**.
  - Same procedure will follow for SHLD & SHLP Object type .
  - SHLP & SHLD in TCode - Se11->Select Search help ->Enter object from Scoping Sheet -> click on display.

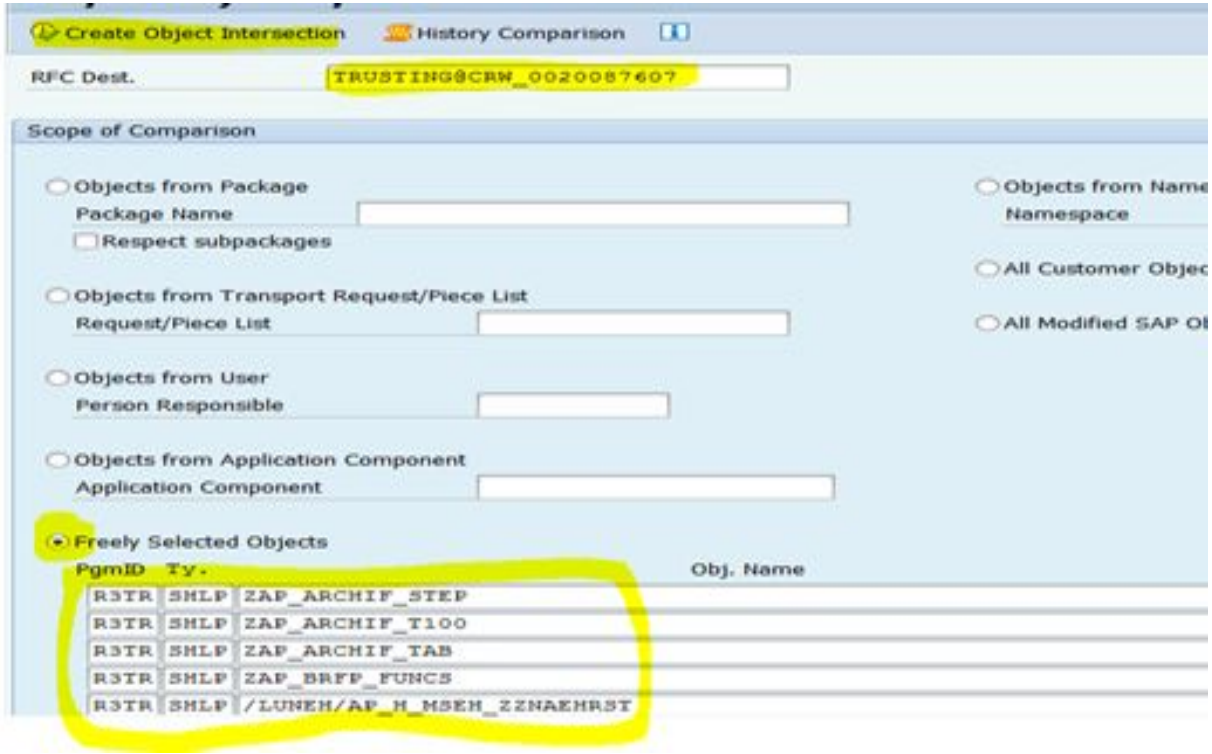
➔ When SHLP/SHLD is NEW to target:

we can add them to NEW/Safe to Import ToC and import to Target.

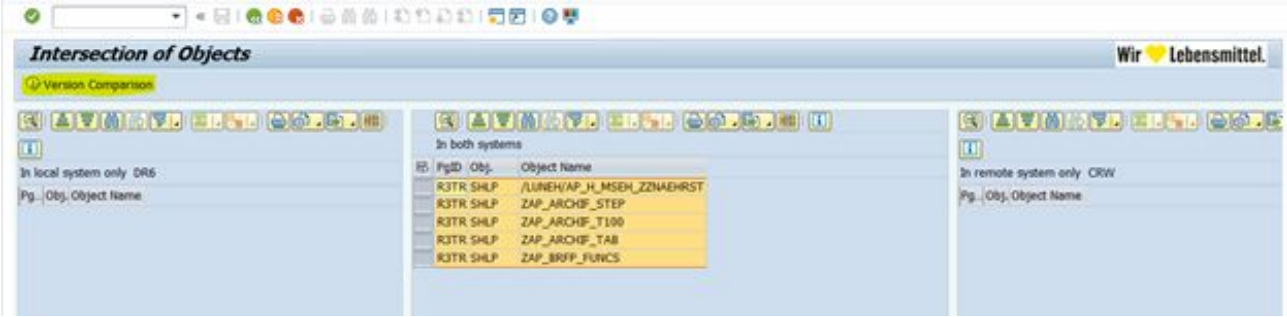
➔ When SHLP/SHLD is Already existed in target, then we can check in 2 steps.

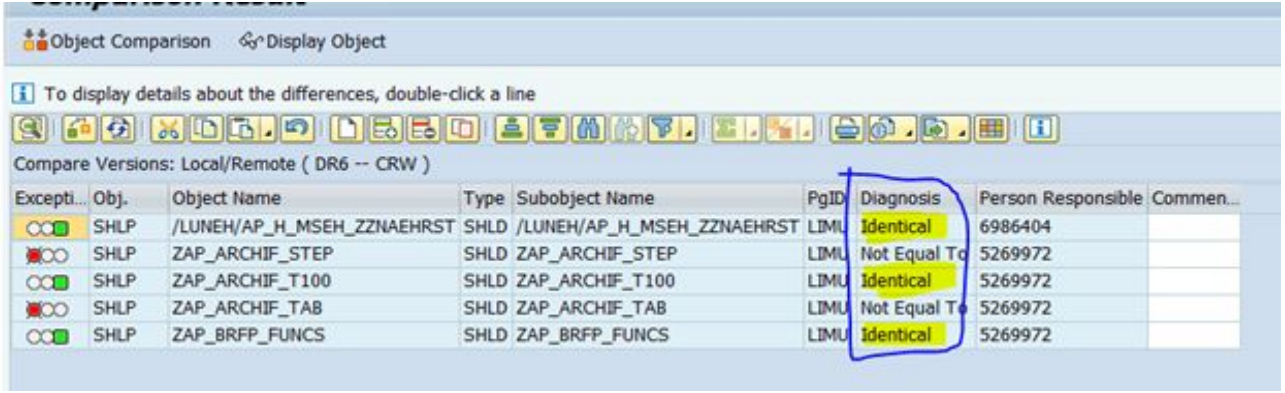
- **Procedure to check in RSSYSCOMP to identify and eliminate identical objects.**
  - With rssyscomp prog, we can compare 8 objects at a time using RFC (Target to Source and vice versa) instead of comparing individually from se11, which takes more time. Identical cases can be eliminated by using rssyscomp which will then be marked as No Action in the scope sheet. The remaining conflict cases need to be checked in SE11 manually by comparing versions..
  - Goto SE38 TCode ->enter program name - RSSYSCOMP -> execute
  - Note: rssyscomp always compares active version from source and target.

Enter RFC(Target to Source or vice versa) -> select freely selected objects -> Copy objects from Scoping sheet -> Execute. Create object Intersection like below screen shot .



Click on version Comparison, It displays the object comparison between source and target like identical or Not identical.

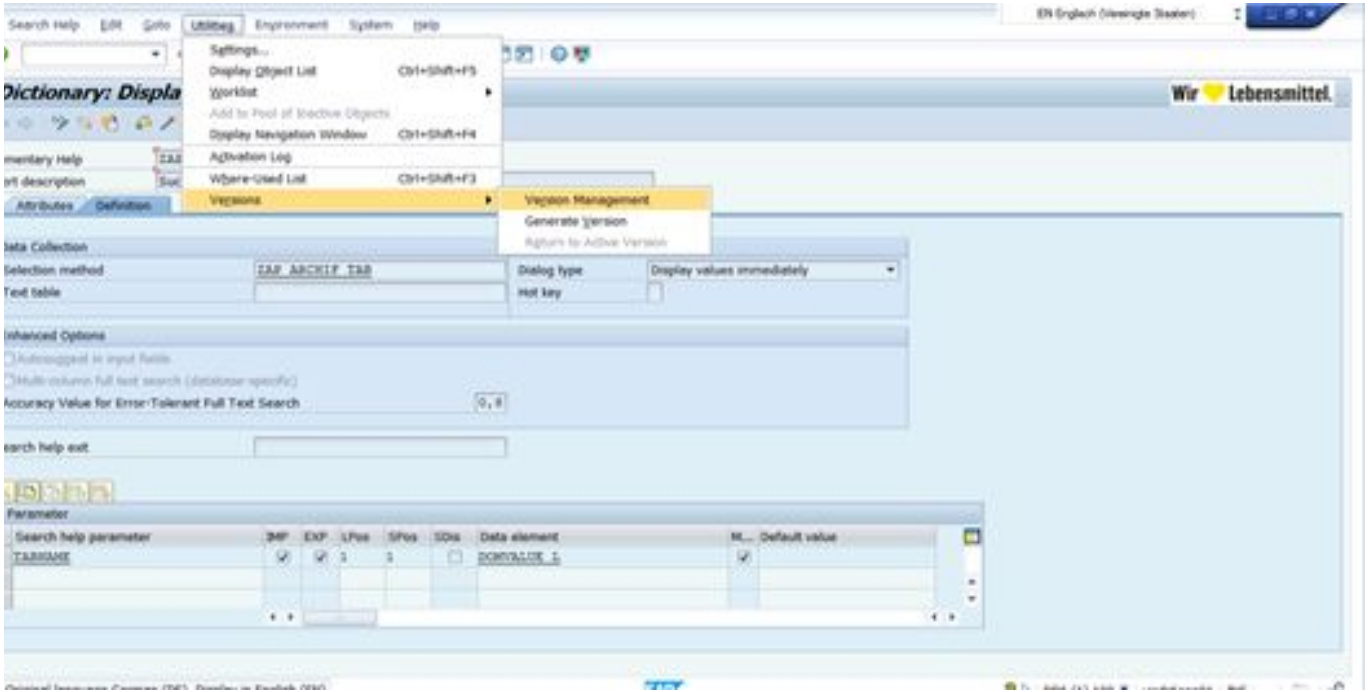




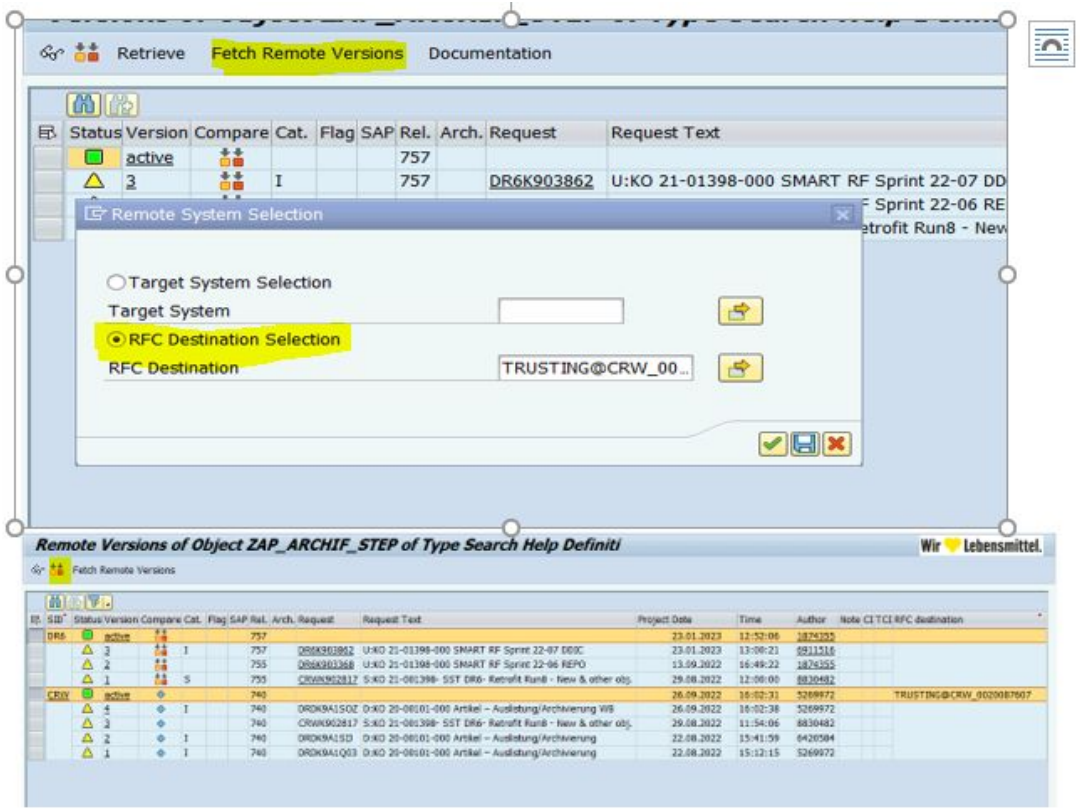
- **Procedure for fixing Conflict/Not-Identical(SHLD & SHLP) obj type:**

  - Goto Se11 Tcode, Select and enter object name in search help input field..
  - We need to find difference by comparing different versions in source and target. If RFC is available then, using ‘Fetch remote version’ option, scope TR from source system version can be compared with active version from target.
  - Go to Source version management and compare scope version TR with below versions and find if it is Actual source change or not. If the changes are newly made in the scope TR then, consider that as a source change and need to bring these changes into the target system.
  - If there is no difference found when comparing scope TR with below versions in source, then, target versions need to be compared to identify if it is target specific changes. Suppose object has only target changes, mark as no action and maintain “target has latest changes” in scoping sheet .
  - If both Source and target versions have changes, then we need to bring source changes to target by retaining target changes .
  - If there is a conflict, meaning, there is change in source and target simultaneously then, we need to take customer feedback.

Open object in Se11, go to version management. If RFC is available from target to source, then open versions in target or vice versa.



Click on “fetch remote versions” and enter RFC destination, this will display version from both target and source in single screen. Now, compare target active and scope version.



The difference will be highlighted like below. .

Version Display/Comparison of Search Help

Wir ♥ Lebensmittel.

Full Display Single/Double Column Field catalog

Object Name	Version	Syst. Name	Request	Date	Author	Object Name	Version	Syst. Name	Request
ZAP_ARCHIF_TAB	00002	DR6	DR6K903368	13.09.2022	1874355	ZAP_ARCHIF_TAB	Actv.	CRW	

Attr.

Attr.
Unchanged

Parameter

Src.hlp.pr	IMP	EXP	LPos	SPos	SDis	Data element	Mod	Default		Src.hlp.pr	IMP	EXP	LPos	SPos	SDis	Data element	Mod	Default
TABNAME	☑	☑	1	1	☐	ZAP_ARCHIF_TAB_TYPE	☐			TABNAME	☑	☑	1	1	☐	DOMVALUE_L	☑	

Parameter
Changed: 1

The conflict needs to be analysed to find out whether it is actual source change or target change by comparing scope version with other versions in source system. Below is an example for source change which was identified by comparing versions in source system. Therefore, we need to bring source changes into target .

Delta display Single/Double Column Field catalog

Object Name	Version	Request	Date	Author	Object Name	Version	Request
ZAP_ARCHIF_TAB	Activ		26.09.2022	5269972	ZAP_ARCHIF_TAB	00003	CRWK902817

Attribute

Elem. src. help	ZAP_ARCHIF_TAB	ZAP_ARCHIF_TAB
Short text	Suchhilfe für Tabellen in der Auslistung	Suchhilfe für Tabellen in der Auslistung
Selection method	ZAP_ARCHIF_TAB	ZAP_ARCHIF_TAB
Text table		
Search help exit		
Hot key		
Dialog type	Display values immediately	Display values immediately

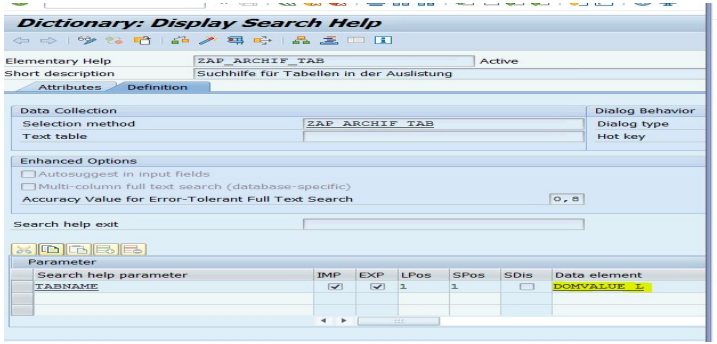
Attribute
Unchanged

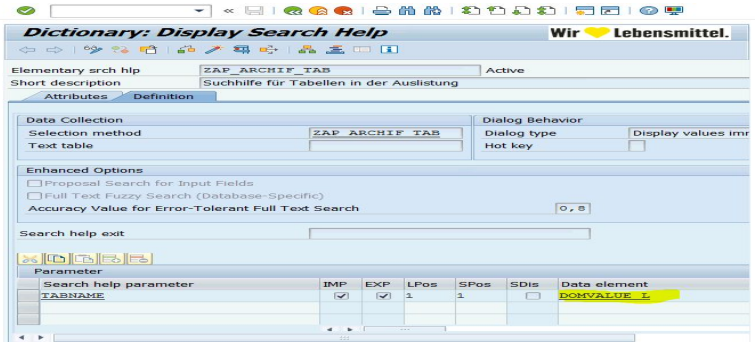
Parameter

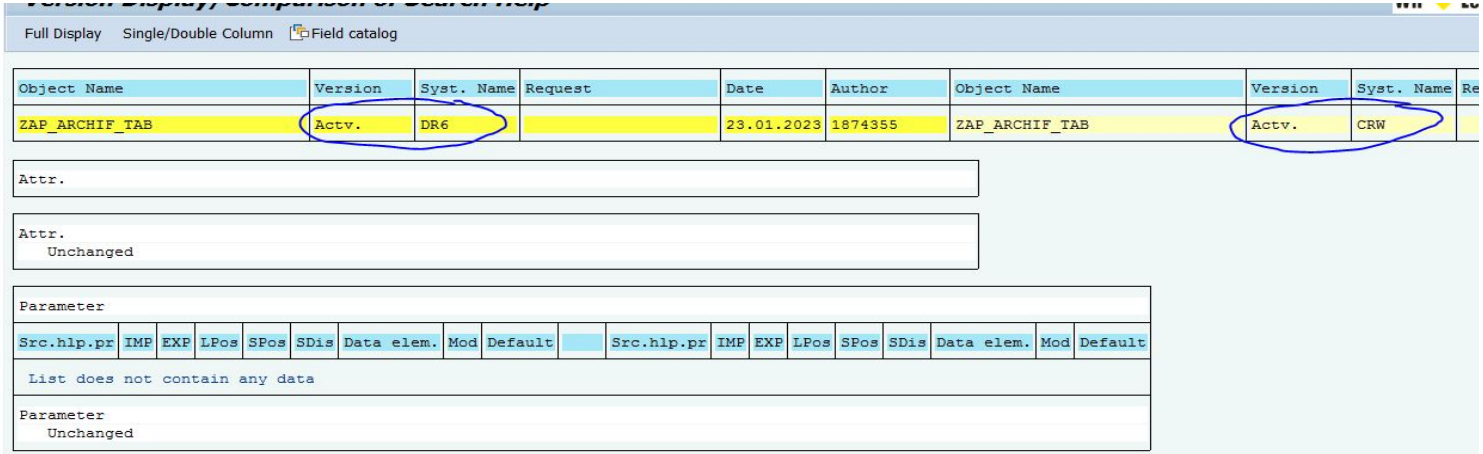
Src.hlp.pr	IMP	EXP	LPos	SPos	SDis	Data element	Mod	Default		Src.hlp.pr	IMP	EXP	LPos	SPos	SDis	Data element	Mod	Default
TABNAME	☑	☑	1	1	☐	DOMVALUE_L	☑			TABNAME	☑	☑	1	1	☐	ZAP_ARCHIF_TAB_TYPE	☐	

Parameter
Changed: 1

Once retrofitted, compare the source scope tr version with target active version, there should be no change.







ENQU & ENQD

ENQU- Lock Object ENQD - Lock Object Definition

- Format of Scoping Sheet
  - LIMU -ENQD -Object name , R3TR -ENQU- Object name
- About ENQU:
  - ENQU (Lock object) is a data Dictionary object type .
  - As part of Retrofit, ENQU & ENQD will come Under Service scope .
  - Already existed DDIC objects are not Consider for ToC(Safe to Import), it will be fixed manually. As version management is available, we can compare versions and decide the changes to be made. Also, we can check them using prog **RSSYSCOMP**.
  - Same procedure will follow for ENQU & ENQD Object type .
  - TCode - Se11->Select Lock object ->Enter object from Scoping Sheet -> click on display

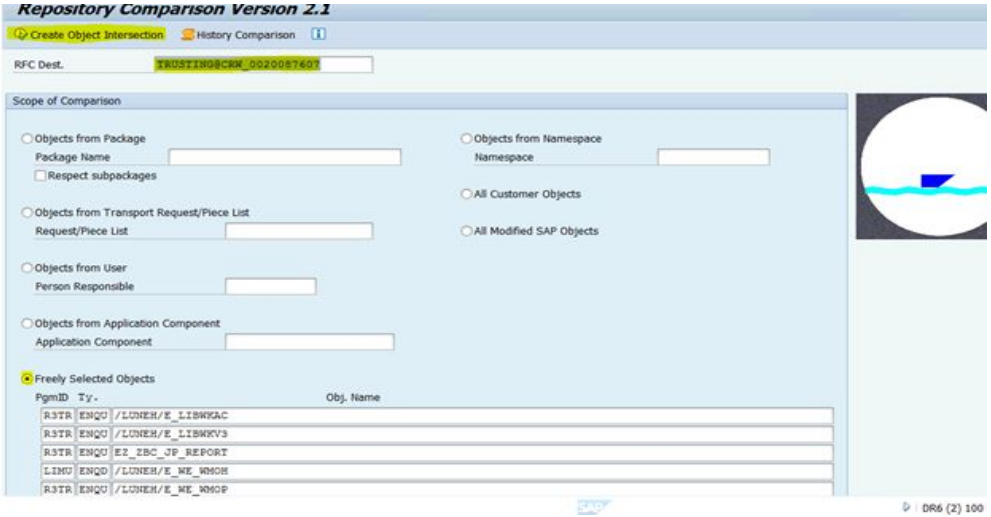
➔ When ENQU/ENQD is NEW to target:

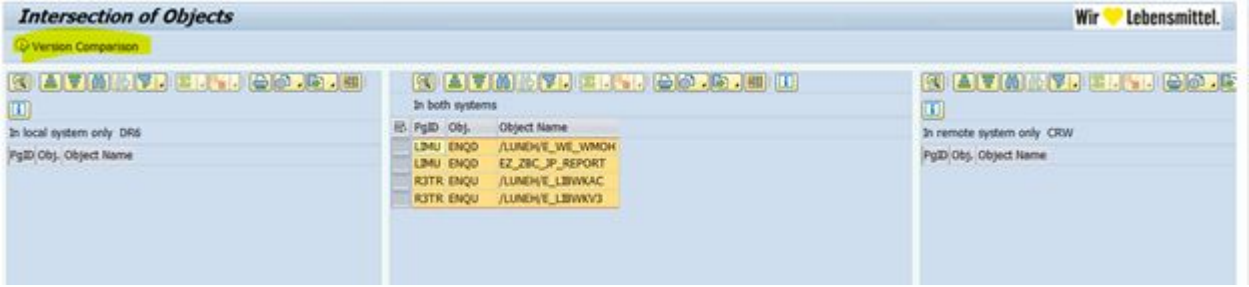
we can add them to NEW/Safe to Import ToC and import to Target.

➔ When ENQU/ENQD is Already existed in target, then we can check in 2 steps.

- **Procedure to check in RSSYSCOMP to identify and eliminate identical objects.**
  - With rssyscomp prog, we can compare 8 objects at a time using RFC (Target to Source and vice versa) instead of comparing individually from se11, which takes more time. Identical cases can be eliminated by using rssyscomp which will then be marked as No Action in the scope sheet. The remaining conflict cases need to be checked in SE11 manually by comparing versions..
  - Goto SE38 TCode ->enter program name - RSSYSCOMP -> execute
  - Note: rssyscomp always compares active version from source and target.

Enter RFC(Target to Source or vice versa) -> select freely selected objects -> Copy objects from Scoping sheet -> Execute. Create object Intersection like below screen shot .





Click on version Comparison, It displays the object comparison between source and target like identical or Not identical.

Comparison Result

Object Comparison Display Object

To display details about the differences, double-click a line 4 0 Total

Compare Versions: Local/Remote (DR6 -- CRW)

Exception	Obj.	Object Name	Type	Subobject Name	PgID	Diagnosis	Person Responsible	Comments
	ENQU	/LUNEH/E_WE_WMOH	ENQD	/LUNEH/E_WE_WMOH	LIMU	Not Equal To	5256266	
	ENQU	EZ_ZBC_JP_REPORT	ENQD	EZ_ZBC_JP_REPORT	LIMU	Not Equal To	N3COLOV3US	
	ENQU	/LUNEH/E_LIBWKAC	ENQD	/LUNEH/E_LIBWKAC	LIMU	Not Equal To	N3COLOV3US	
	ENQU	/LUNEH/E_LIBWKV3	ENQD	/LUNEH/E_LIBWKV3	LIMU	Not Equal To	N3COLOV3US	

Version Display/Comparison for Lock Objects

Full Display Single/Double Column Field catalog

Object Name	Version	Syst. Name	Request	Date	Author
/LUNEH/EAK_WALE	Actv.	DR6		20221115	6805275
/LUNEH/EAK_WALE	Actv.	TRUSTING@C		20220822	6805275

Lock object	/LUNEH/EAK_WALE

General Attributes:
Unchanged

Base Tables:

	Basis Table	LM
No Data		

Base Tables:
Unchanged

Lock Parameters:

	Lock par.	Table Name	Field
List does not contain any data			

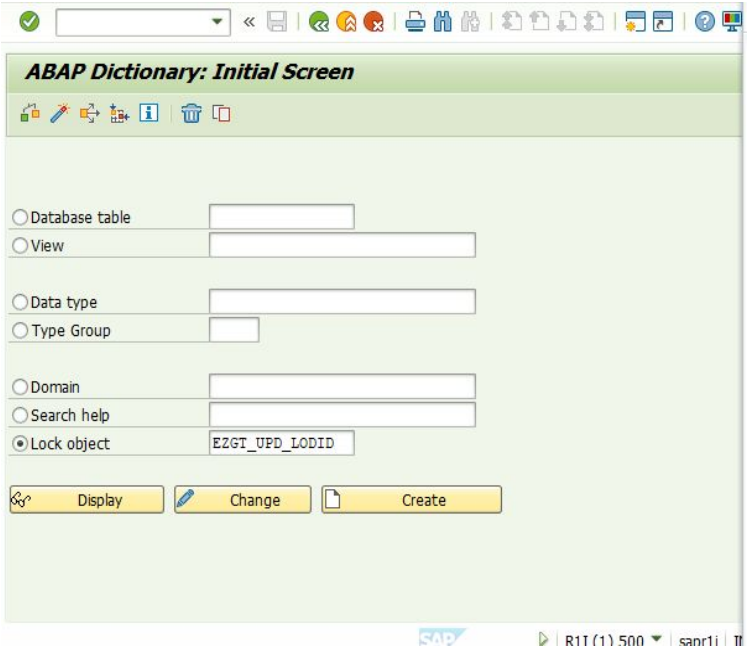
Lock Parameters:
Unchanged

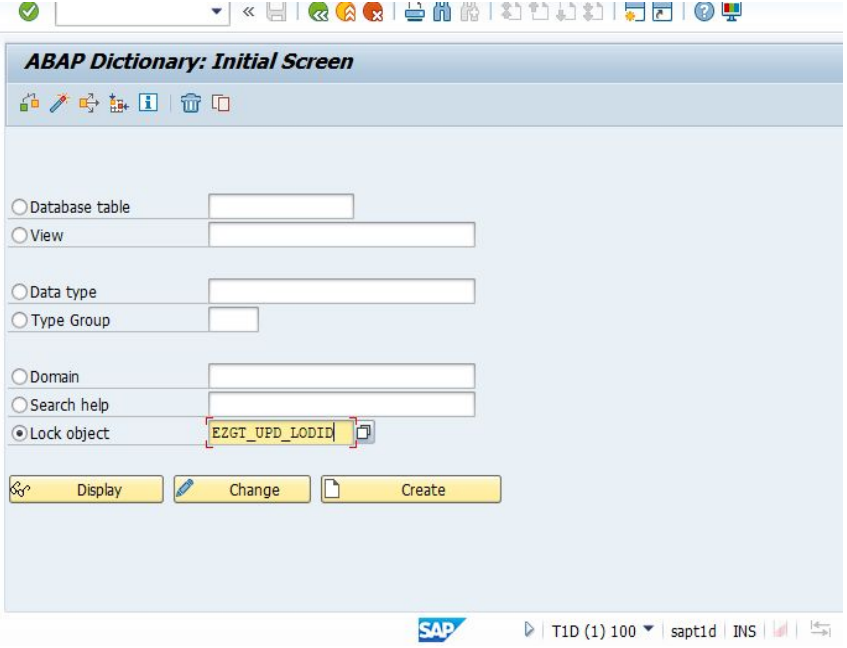
If we double click on “Not Identical” and select -> Delta display option, only differences are displayed. If an object is identical, then it displays, all attributes as unchanged .

- **Procedure for fixing Conflict/Not-Identical(ENQU & ENQD) obj type:**

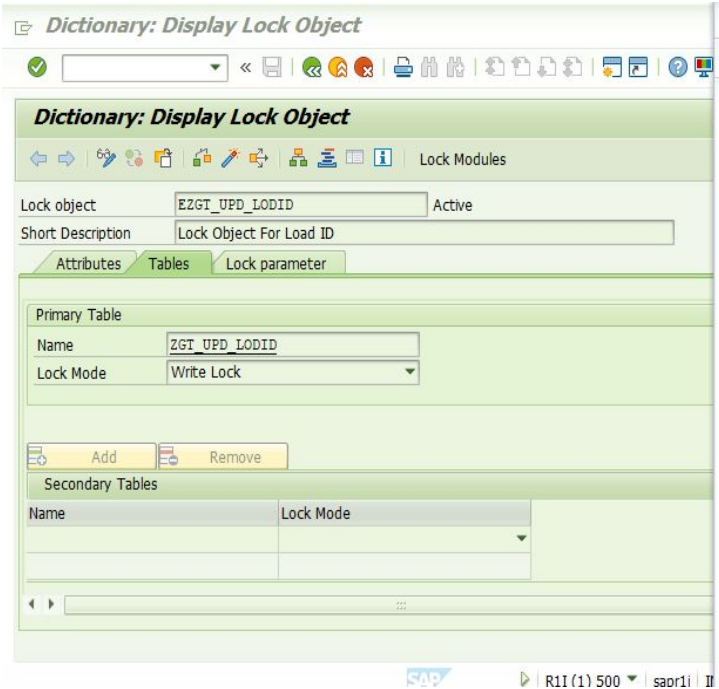
  - Goto Se11 Tcode Select and enter object name in lock object input field.
  - We need to find differences by comparing different versions in source and target. If RFC is available then, using ‘Fetch remote version’ option, scope TR from source system version can be compared with active version from target.
  - Go to Source version management and compare scope TR version with below versions and find if it is Actual source change or not. If the changes are newly made in the scope TR then, consider that as a source change and need to bring these changes into the target system.
  - If there is no difference found when comparing scope TR with below versions in source, then, target versions need to be compared to identify if it is target specific changes. Suppose object has only target changes, mark it as no action and maintain “target has latest changes” in scoping sheet .
  - If both Source and target versions have changes, then we need to bring source changes to target by retaining target changes.
  - If there is a conflict, meaning, there is change in source and target simultaneously then, we need to take customer feedback.

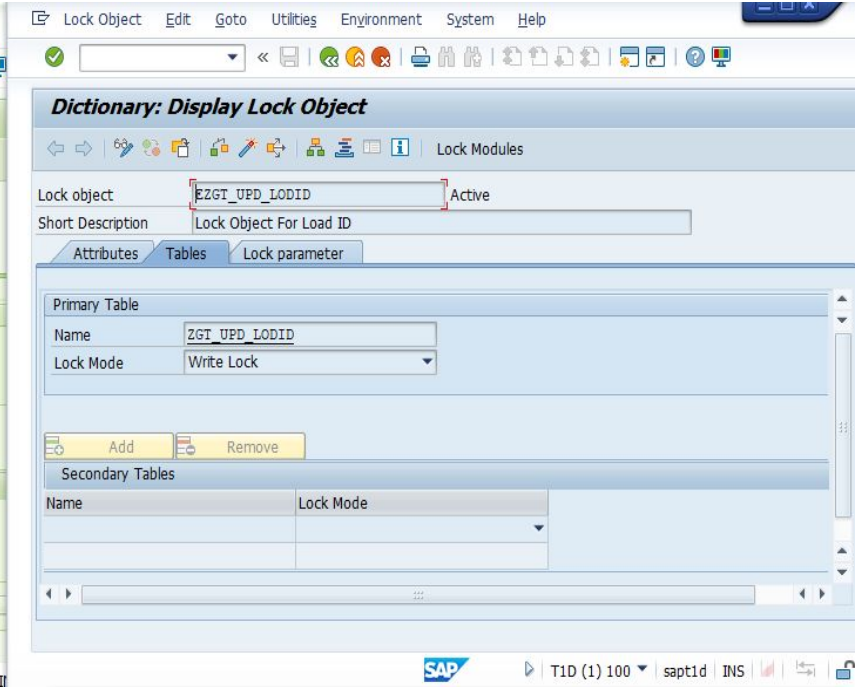
Open object in Se11 in both source and target systems



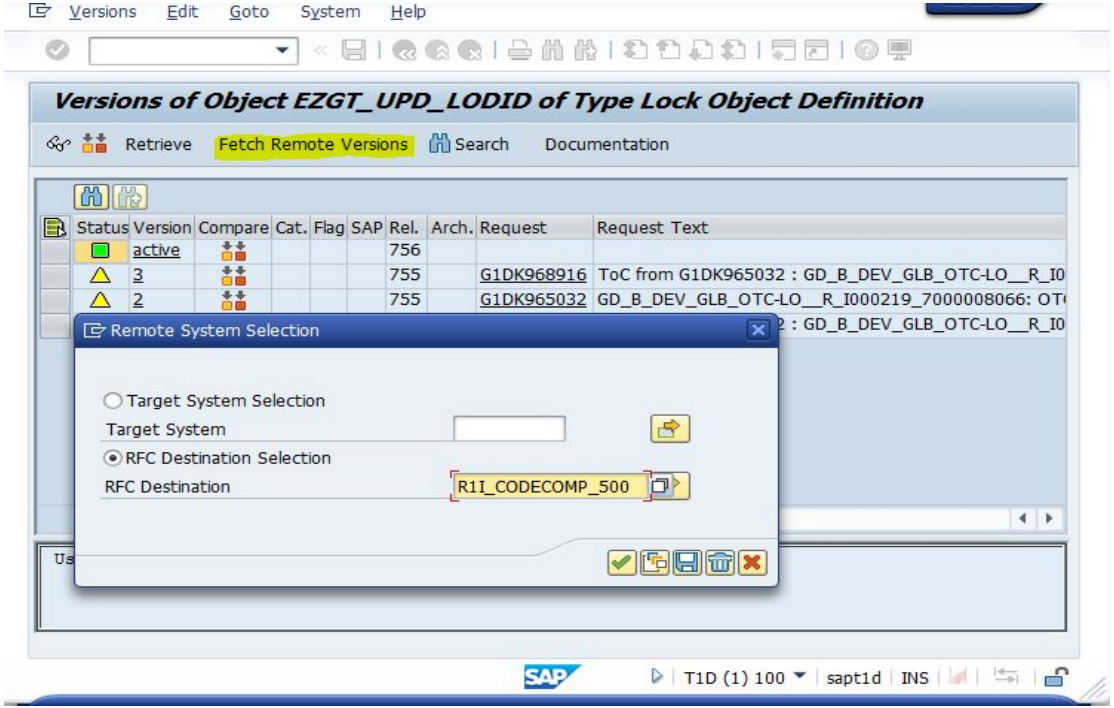


Compare all the tabs and attributes of lock object in both source and target systems. Go to version management.





Click on “fetch remote versions” and enter RFC destination, this will display version from both target and source in single screen. Now, compare target active and scope version.



In the below example, there is no difference between active version of target and scope tr in source. Therefore, the object is identical and no action is required

S...	Status	Version	Compare	Cat.	Flag	SAP	Rel.	Arch.	Request	Request Text
T1D		active					756			
		3					755		G1DK968916	ToC from G1DK965032 : GD_B_DEV_GLB_OTC-LO_
		2					755		G1DK965032	GD_B_DEV_GLB_OTC-LO__R_I000219_7000008066
		1					755		G1DK967120	ToC from G1DK965032 : GD_B_DEV_GLB_OTC-LO_
R1I		active					756		B1DK908993	GD_P_DEV_GLB_OTC-LO__CRCR12844_700001976

Object Name	Version	Syst. Name	Request	Date	Author
EZGT_UPD_LODID	Actv.	T1D		26.01.2023	09289281
EZGT_UPD_LODID	Actv.	R1I	B1DK908993	20.08.2022	09251706

General Attributes:

Lock object	EZGT_UPD_LODID

General Attributes:
Unchanged

Base Tables:

	Basis Table	LM

No Data

Base Tables:
Unchanged

Lock Parameters:

	Lock par.	Table Name	Field

List does not contain any data

Lock Parameters:
Unchanged

TOBJ -

TOBJ

- Format in scoping sheet: R3TR-TOBJ-ObjectName
- As part of Retrofit, we consider TOBJ under Service scope
- TCode : SOBJ
- We have to analyze tab by tab to understand the difference
- There is no version management option available here. Therefore, this object needs to be Check manually .

TOBJ can be considered as **part of Safe To Import** analysis/ToC

- Yes - For NEW, IN scope Objects-TOBJ can Consider for ToC .
- For Existing objects also can Consider for ToC based on Se03 Analysis .

Procedure to check TOBJ:

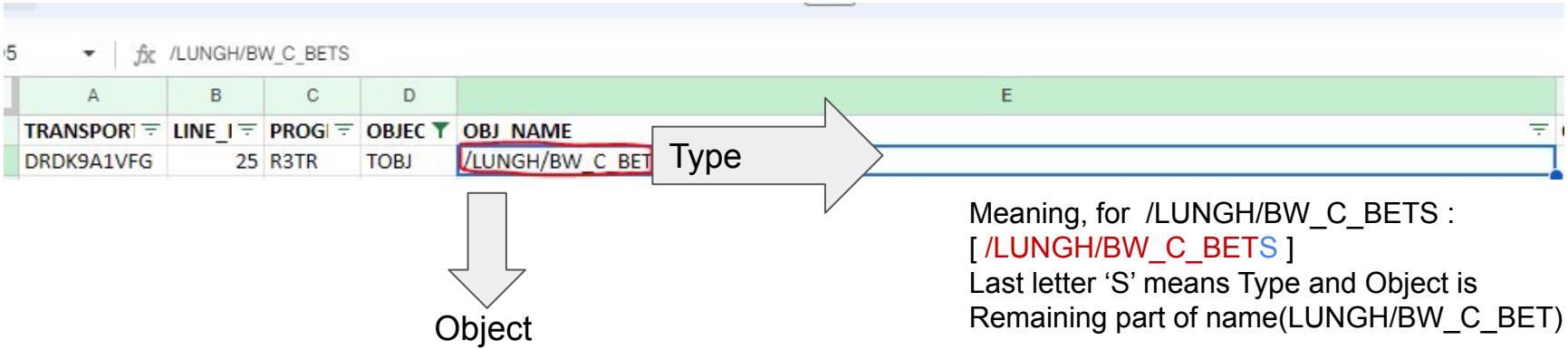
- Go to SOBJ - Tcode -> click on display ->Position->Enter object name and Type -> compare screen by screen in both source and target , check Consistency .

Enter SOBJ TCode and Display the object in source and target:

After entering the display mode in SOBJ, we have to find our object from the list.

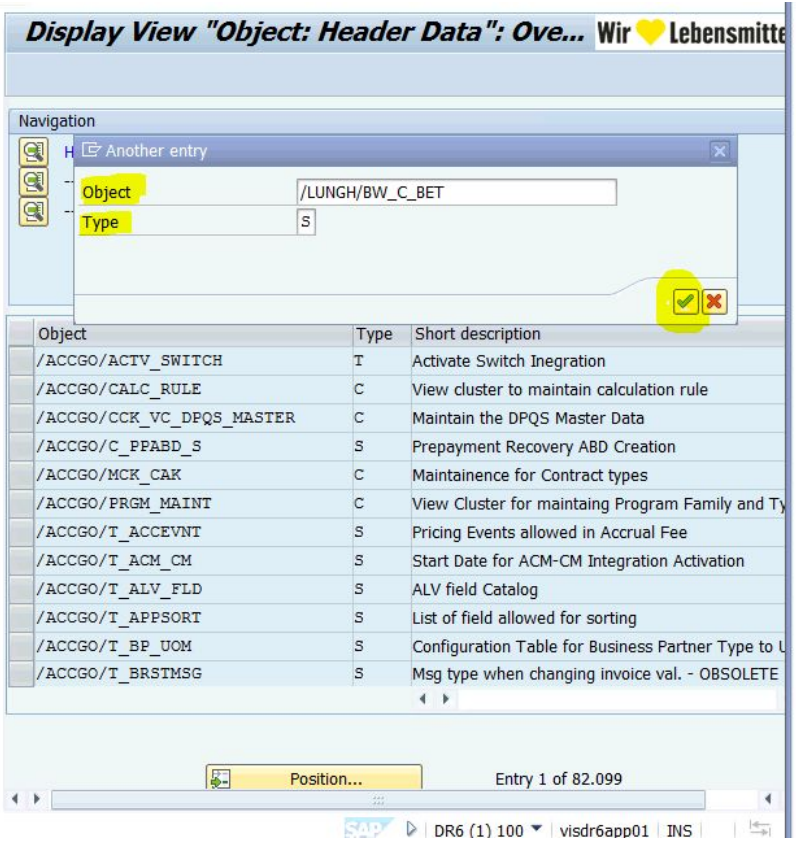
Note: For any TOBJ object, Last letter denotes TYPE and remaining part is the OBJECT

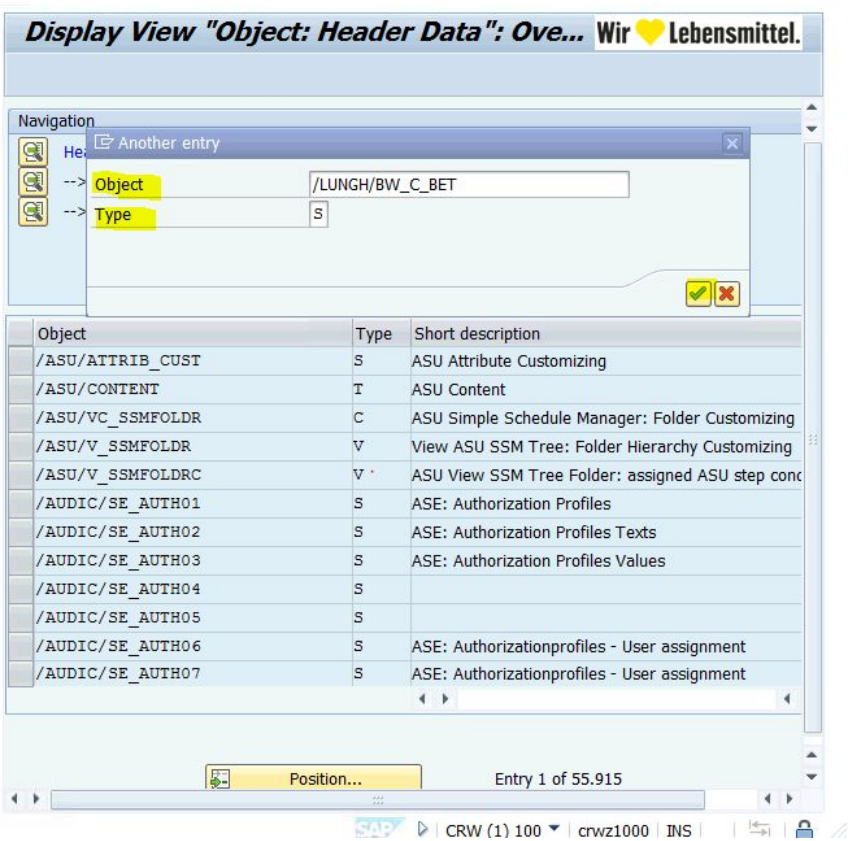
Example - /LUNGH/BW_C_BETS - In scoping sheet snippet below, red color part denotes object name, last letter 'S' denotes type .



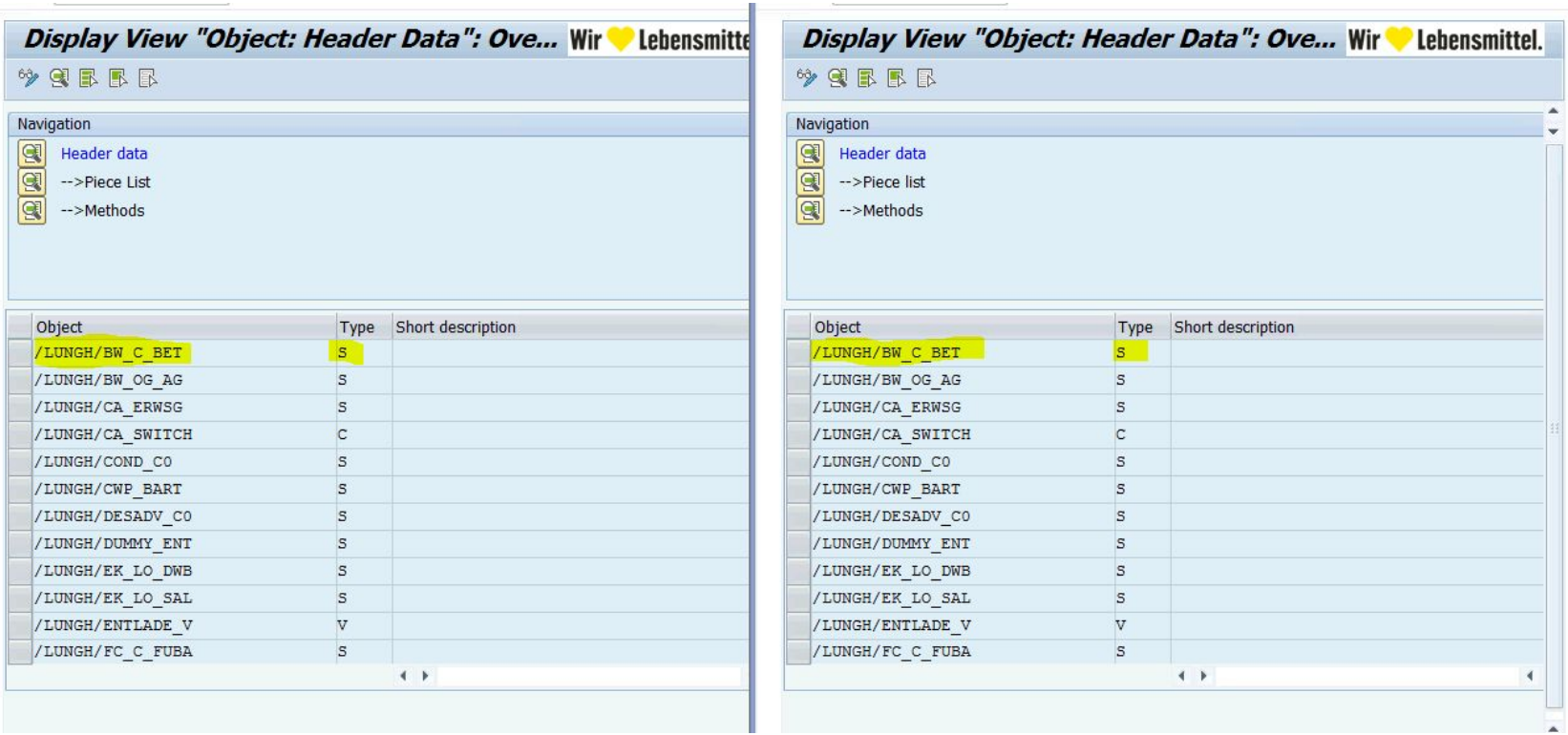
Meaning, for /LUNGH/BW_C_BETS :
[/LUNGH/BW_C_BETS]
Last letter 'S' means Type and Object is
Remaining part of name(LUNGH/BW_C_BET)

Click on position , Enter object name and Type in both source and target system to find object entry in SOBJ list of objects

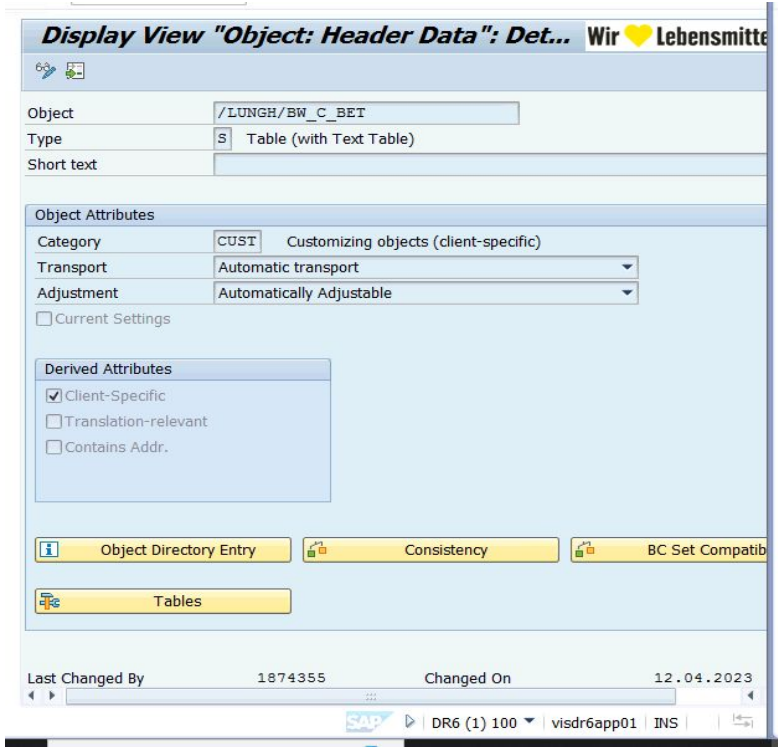


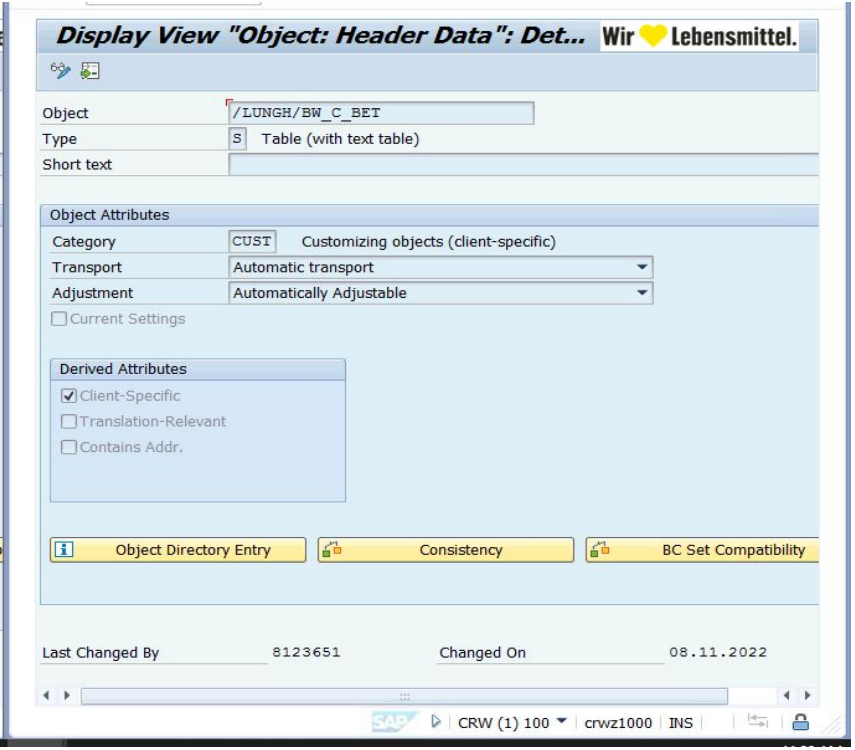


Here found entered object name with same Type . Double click will navigate into the object . And analyse in detail(screen comparison & Consistency check) after double clicking the object.



Compare TOBJ object type in Both Source and Target . In this example not found any Difference so that it's an Identical(No action) case .

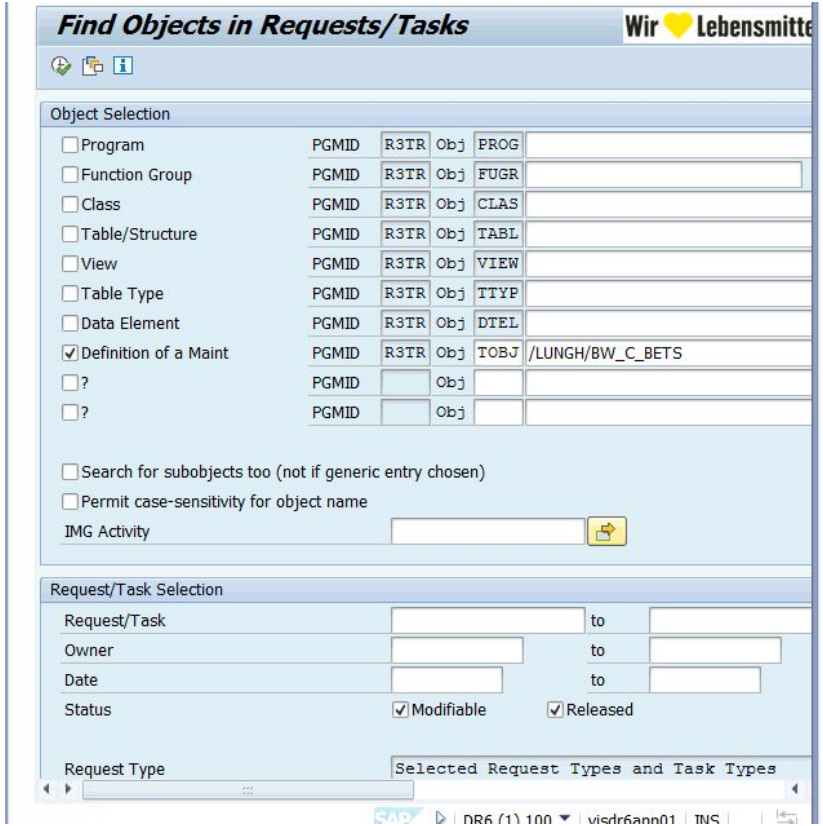


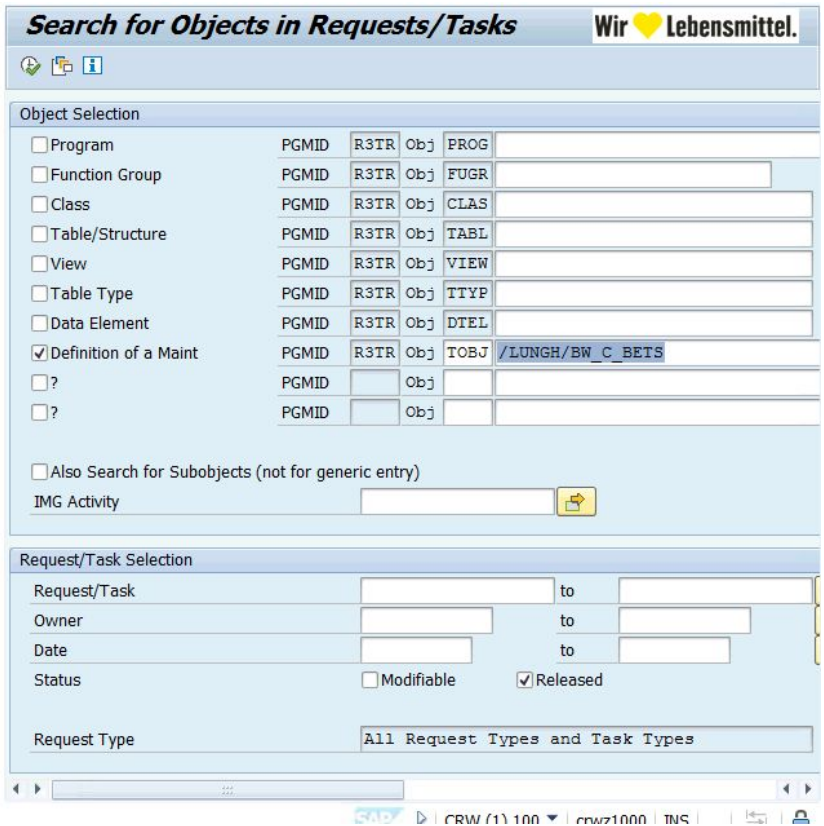


Scenario- 2

- ➔ Suppose the TOBJ object already exists in the target if find any Difference in manual Comparison , then we need to evaluate and decide whether importing the object is safe, meaning that it should not override any changes made specifically for the target.
- ➔ Using the SE03 transaction in both the source and the target,system, we can decide whether or not to import that object.
- ➔ **SE03 Screen:** Input object type, object name and execute, and sort by descending order of date identify if any common TR exists in both Source and Target systems. If there is a common TR between Source and Target, and in source if additional TRs exists above versions of common TR, then the object can be considered as Safe to Import.
  If there is a common TR between Source and Target for an object, but the Target has latest TR then we need add only source changes manually retaining target changes .

➔ **SE03 Screen:** Input object type, object name and execute in both source and Target Systems.





Find Objects in Requests/Tasks

Wir ♥ Lebensmitte

Objects View

Request Overview

Request	Short Description
DR6K907200	D:KO 99-02302-000 Smart RF Sprint 23-02 REPO & DDIC
SAPKCCD757	Commandfile for customer export
SAPKCCC752	Commandfile for customer export
CRWK902605	N:ZKTP Übernahme ECC-DEV nach S/4 (DR6)
CRWK902604	N:ZKTP Übernahme
DRDK9A17RA	D:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)
DRDK9A0VVD	N:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)
DRDK9A17K8	X:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)
DRDK9A17KF	X:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)
DRDK9A17KH	X:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)
DRDK9A17KT	X:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)
DRDK9A17L7	X:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)
DRDK9A17LJ	X:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)
DRDK9A17LO	T:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)
DRDK9A17JA	T:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)
DRDK9A177U	X:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)
DRDK9A176R	X:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)
DRDK9A16U3	X:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)
DRDK9A16RT	A:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)

Task Overview

DR6 (1) 100 | visdr6app01 | INS

Search for Objects in Requests/Tasks

Wir ♥ Lebensmittel.

Request	Short Description
DRDK9A1VFG	U:KO 22-00045-000 O&G digitaler Ordersatz Scheibe 2
CRWK902605	N:ZKTP Übernahme ECC-DEV nach S/4 (DR6)
CRWK902597	Übernahme "AltEntwicklungen" aus DRD(CRW) nach DR6 vor CCC
CRWK902598	N:ZKTP Übernahme "AltEntwicklungen" aus CRW nach DR6 vor CCC
CRWK902604	N:ZKTP Übernahme
DRDK9A1B8H	D:KO 19-00400-000 O&G Harmonisierung - Korrektur
DRDK9A1A81	D:KO 19-00400-000 O&G Harmonisierung
DRDK9A1B46	H:FM 6000460036 - Bugfixing PO3 für Prototyping - Paket 3
DRDK9A1ATO	H:FM 6000457845 O&G Harmonisierung
DRDK9A1AQR	H:KO 19-00400-000 O&G Harmonisierung
DRDK9A1ACS	H:FM 6000458171 Bugfixing RT3 für Prototyping
DRDK9A19TQ	H:FM 6000456963 Bugfixing RT3 fuer Prototyping
DRDK9A18GV	D:KO 19-00400-000 O&G Harmonisierung
DRDK9A19JF	H:KO 19-00400-000 O&G Harmonisierung
DRDK9A17RA	D:KO 19-00400-000 O&G Harmonisierung (DDIC/REPO)

Task	Short Description

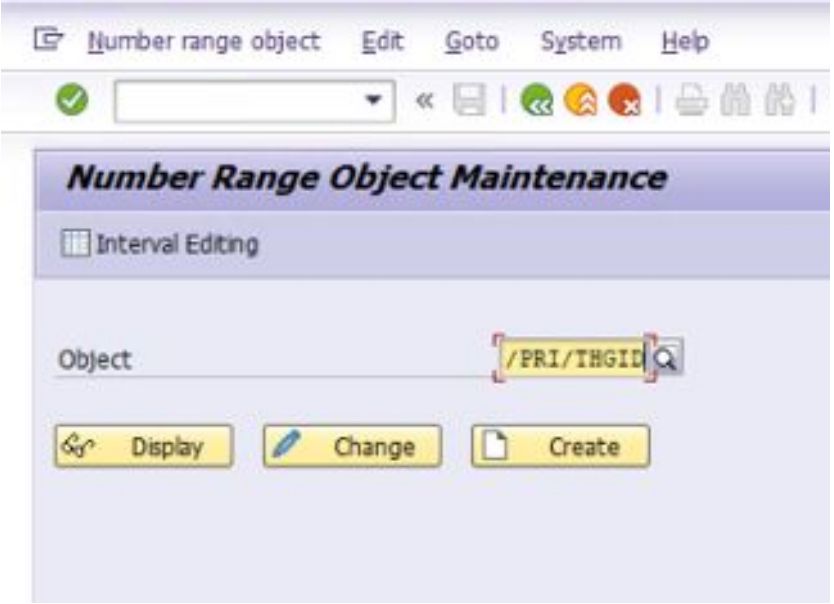
CRW (1) 100 | crwz1000 | INS

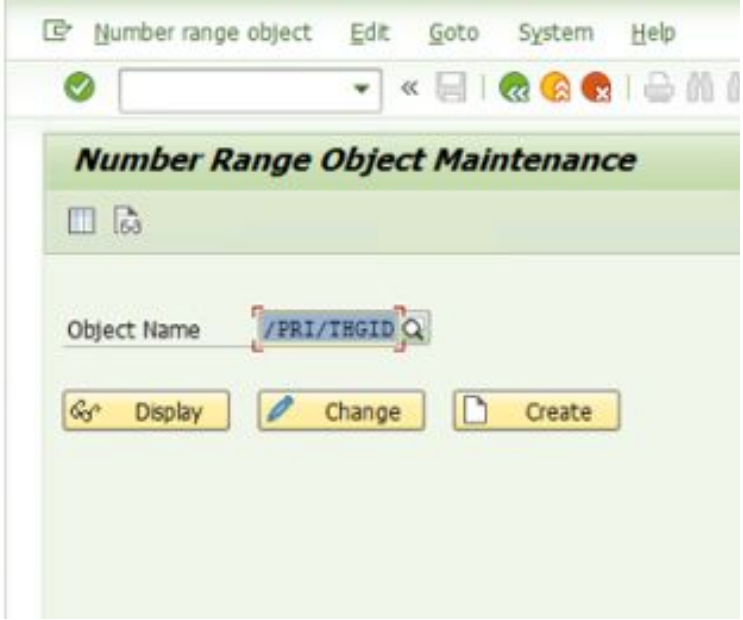
NROB

- Format of Scoping Sheet
  - R3TR-NROB-Object name
- About NROB:
  - NROB is a number range object type .
  - As part of Retrofit NROB will come Under Service scope .
  - Associated Tcode for NROB - SNRO

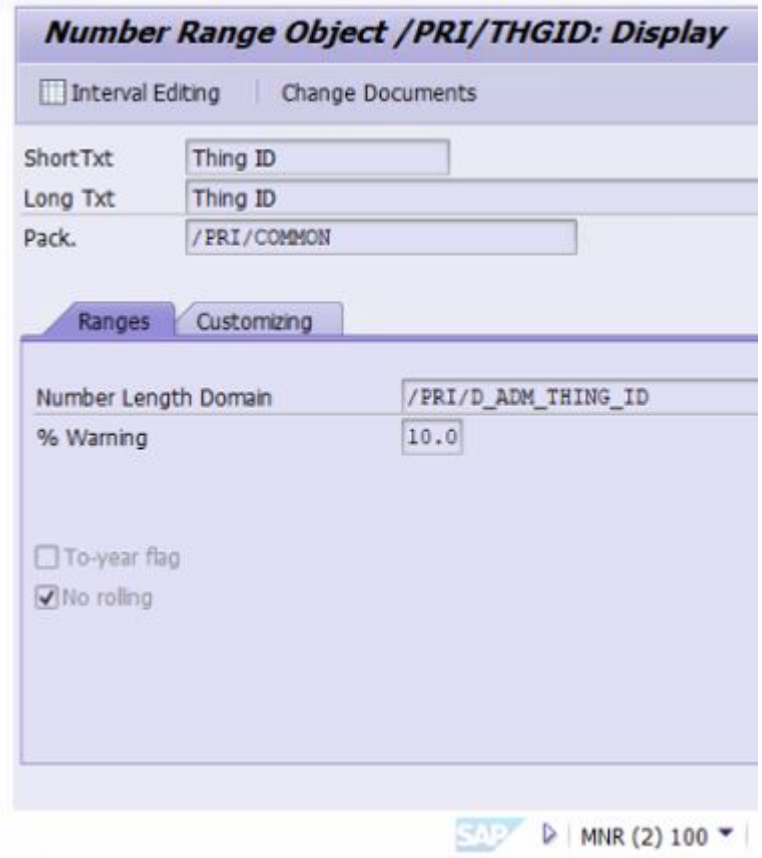
- Procedure to check NROB :
  - Goto TCode - SNRO and check if the object is already existing in target or not.
  - If NROB is New then we can Import as part of ToC.
  - If NROB existing one then we have to do Tab by Tab comparison in source and target and recheck in Se03 to confirm if it is a source change then we can edit under target working TR .

Open the NROB object in TCode SNRO in both source and Target

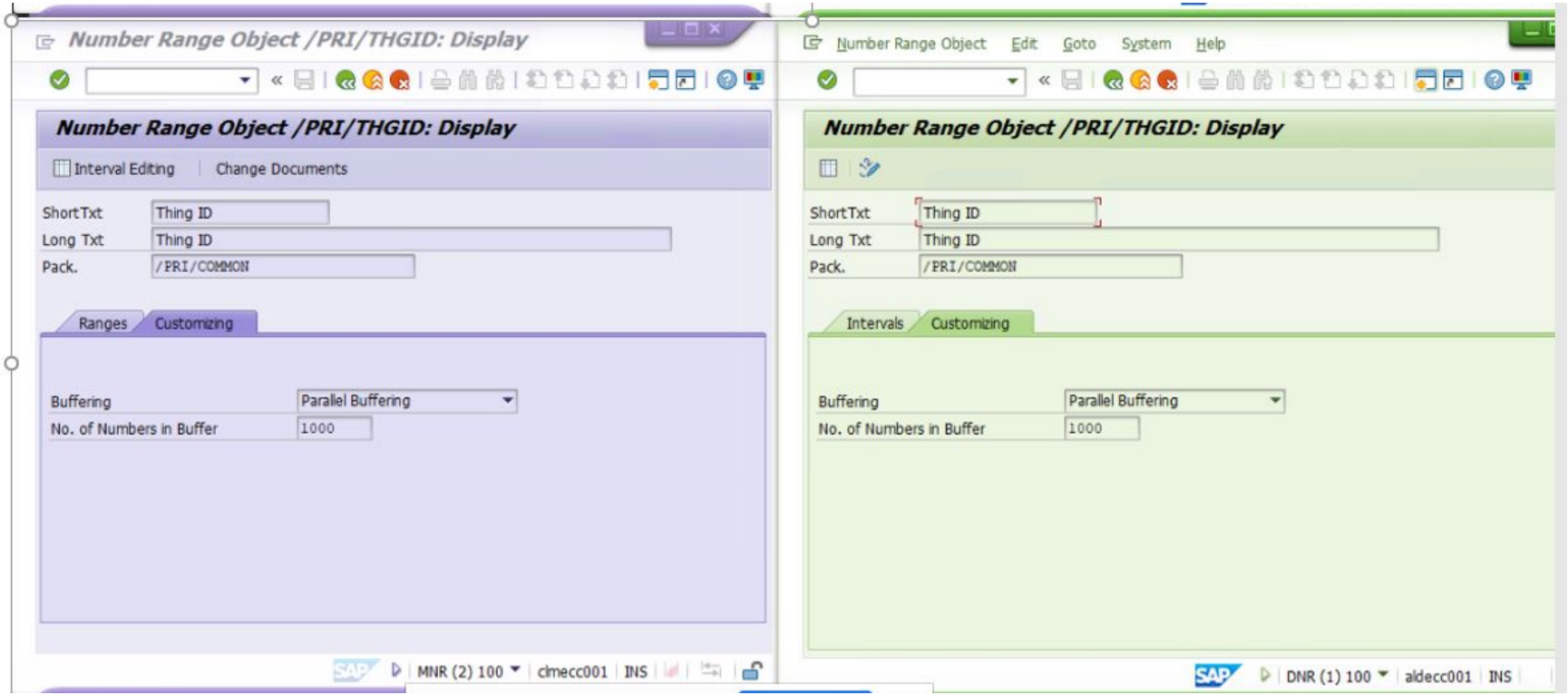


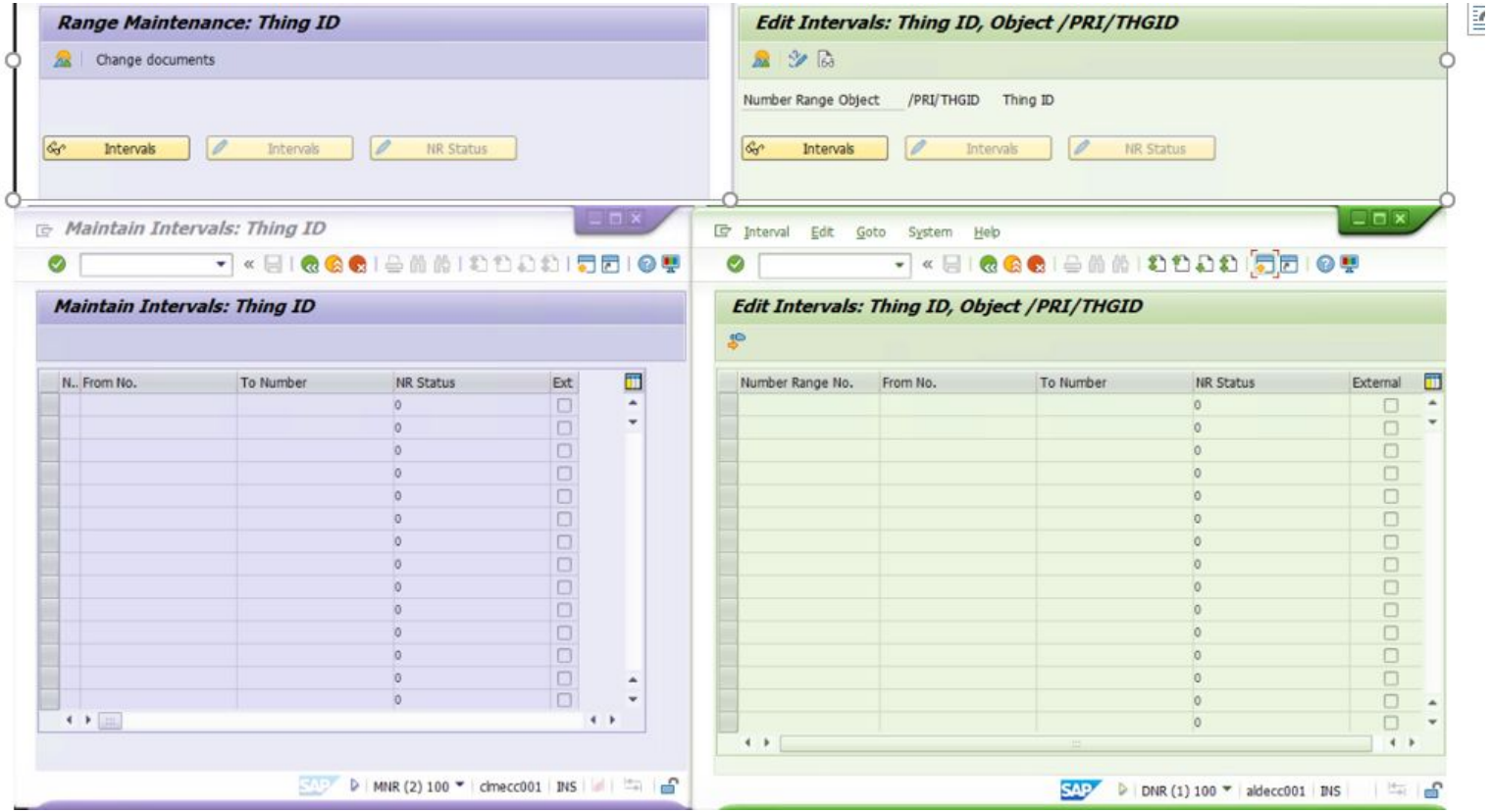


Compare 'Ranges' tab in Both source and target system. Here in current example, for Ranges tab there is no difference .



Compare Customizing tab screen in Both source and target system.Here in current example, for Customizing tab there is no difference .





Compare Interval Tab screen in both source and target system,here From number, To number changes if any-based on se03 analysis we can Retrofit(don't change NR Status). In the current example, there is no Difference found between source and target .

SUSK

Retrofit Process of SUSK :

Scenario 1: New object(ToC)

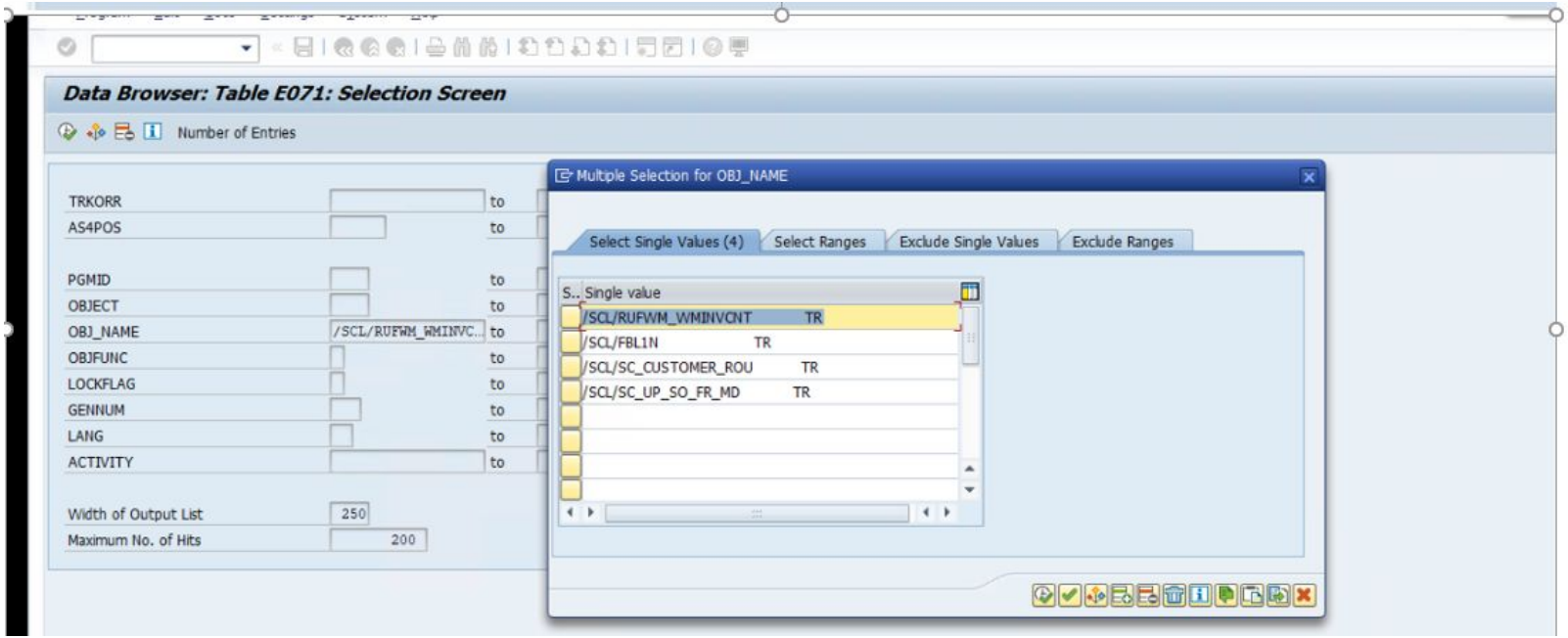
- SUSK is considered as Service scope object in retrofit .
- In Scoping sheet, ‘NEW /already existed’ column will be blank/empty for SUSK object type.;
- So, we need to find whether is it New or Already existed by using E071 table in Se16.
    - If object is New , i.e., No entry in E071 for checked object [we may get a message below like - ‘No table entries found in E071’], Then we can add them to NEW/Safe to Import ToC and import to Target.
    - Suppose Entries found in E071 Follow Scenario 2.

Scenario 2: ToC

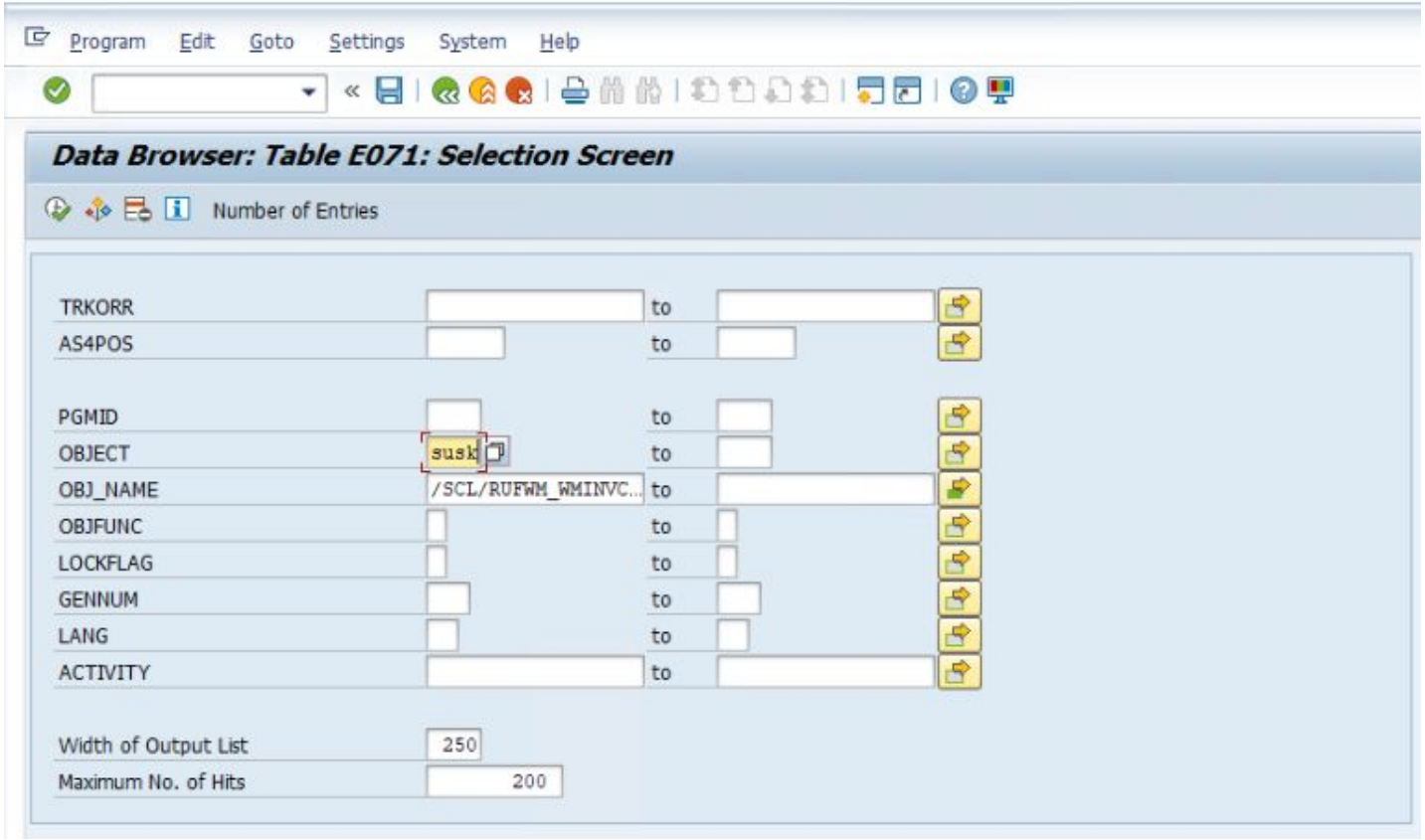
- In SE03, for an object identify if there is any common TR exists in both Source and Target systems by Sorting Descending with date. If there is a common TR between Source and Target and in source there are additional TRs exists on above versions of common TR then the object can be considered as Safe to Import.
- Suppose we got any target side Customer TR we need to change manually .
- Mostly existing objects are No Action cases, Customer rarely make changes to it.

Here in Scoping sheet not find NEW /already existed for SUSK object type, So we are checking in E071 whether object is new in Target/ already existed .

	A	B	C	D	E	F	M	N	S	T	U	V	W	X	Y	Z	AA
1	TRANSPORT	I	PRC	OBJE	OBJ_NAME	OBJ	OBJTYI	MAIN_OBJ	Tool	Object status(New/Existi	Duplic	De	Scop	Sc	A	C	Obj
.010	MNRK978139	2	R3TR	SUSK	/SCL/RUFWM_WMINVCNT TR								IN	In Sn	ManualN		Main_N
462	MNRK979015	2	R3TR	SUSK	/SCL/FBL1N TR								IN	In Sn	ManualN		Main_T
482	MNRK979109	2	R3TR	SUSK	/SCL/SC_CUSTOMER_ROU TR								IN	In Sn	ManualN		Main_T



Enter object type and object name in Se16 and execute .



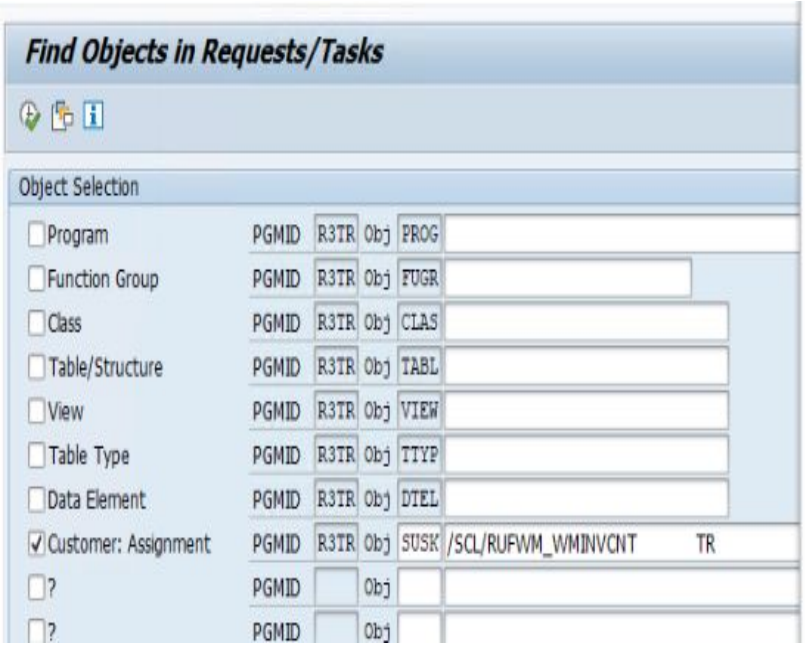
Here in this Example we have found all objects exist in E071 list . So those objects already exist .Suppose if got no table entries found consider it as new object and import (ToC).

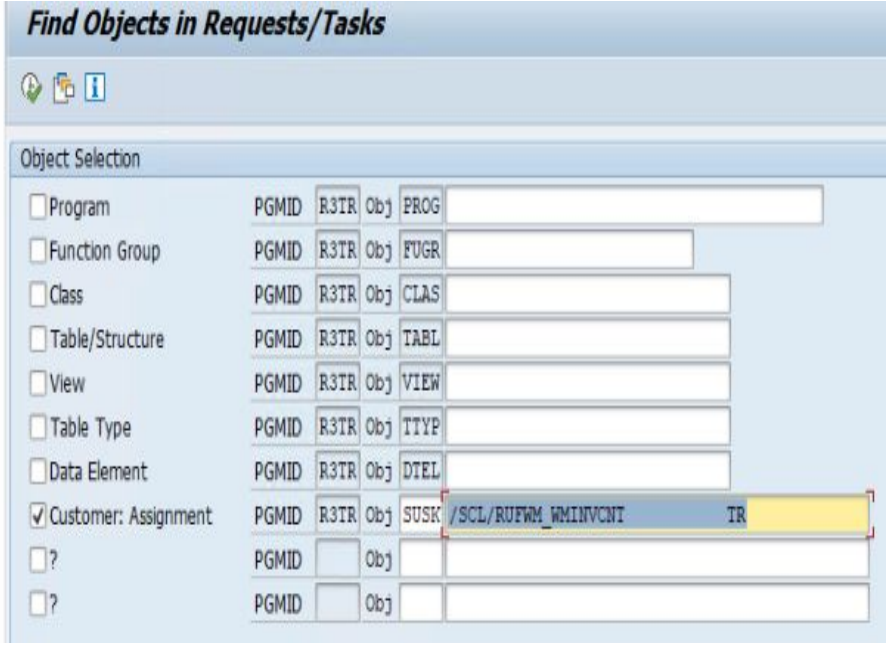
Data Browser: Table E071 Select Entries 8

Check Table...

TRKORR	AS4POS	PGMID	OBJECT	OBJ_NAME		OBJFUNC	LOCKFLAG	GENNUM	LANG	ACTIVITY
DNRK901168	000058	R3TR	SUSK	/SCL/RUFWM_WMINVCNT	TR		3			
DNRK901290	000057	R3TR	SUSK	/SCL/RUFWM_WMINVCNT	TR					
DNRK976195	000371	R3TR	SUSK	/SCL/RUFWM_WMINVCNT	TR					
DNRK976195	001099	R3TR	SUSK	/SCL/FBL1N	TR					
DNRK976195	001102	R3TR	SUSK	/SCL/SC_CUSTOMER_ROU	TR					
DNRK976195	001103	R3TR	SUSK	/SCL/SC_UP_SO_FR_MD	TR					
MNRK978139	000002	R3TR	SUSK	/SCL/RUFWM_WMINVCNT	TR		3			
MNRK978140	000001	R3TR	SUSK	/SCL/RUFWM_WMINVCNT	TR					

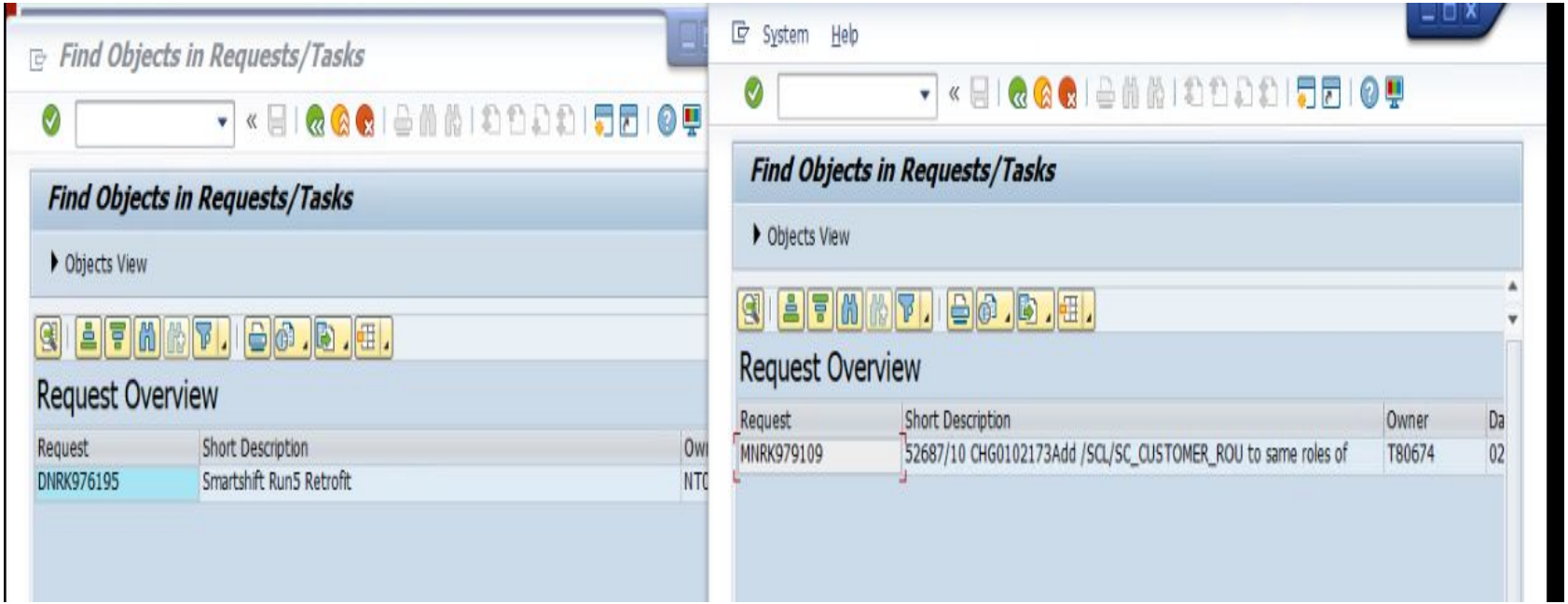
If object already exist check in E071 , We have to check in se03 'Find objects in request' option in both source and target .





(Note: in above screenshot, left-Target system, right-Source system)

Here in this example : left side of screenshot is target system , object locked in current retrofit TR .Right side source of screenshot showing only scope TR.so we can consider this as safe to import .



SUSO

Retrofit Process of SUSO

Scenario 1: New object(ToC)

- SUSO is considered as Service scope object in retrofit .
- In Scoping sheet, ‘NEW /already existed’ column will be blank/empty for SUSO object type.;
- So, we need to find whether is it New or Already existed by using E071 table in Se16.
  - If object is New , i.e., No entry in E071 for checked object [we may get a message below like - ‘No table entries found in E071’], Then we can add them to NEW/Safe to Import ToC and import to Target.
  - Suppose Entries found in E071 Follow Scenario 2.

Scenario 2: ToC

- In SE03, for an object identify if there is any common TR exists in both Source and Target systems by Sorting Descending with date. If there is a common TR between Source and Target and in source there are additional TRs exists on above versions of common TR then the object can be considered as Safe to Import.
- Suppose we got any target side Customer TR we need to change manually in SU21 Tcode .

➔ Mostly existing objects are No Action cases, Customer rarely make changes to it.

B	C	D	E	F	I	L	M	N	O	P	Q	R	S	T	U	V	W
LI	PR(	OBJE	OBJ_NAME	OB			OBJTY	MAIN_OBJECT	Seg	E	T	R	Tool /	Objec	Duplic	D	Scope
759	R3TR	SUSO	/PRI/AUTH														IN

Program Edit Goto Settings System Help

Data Browser: Table E071: Selection Screen

Number of Entries

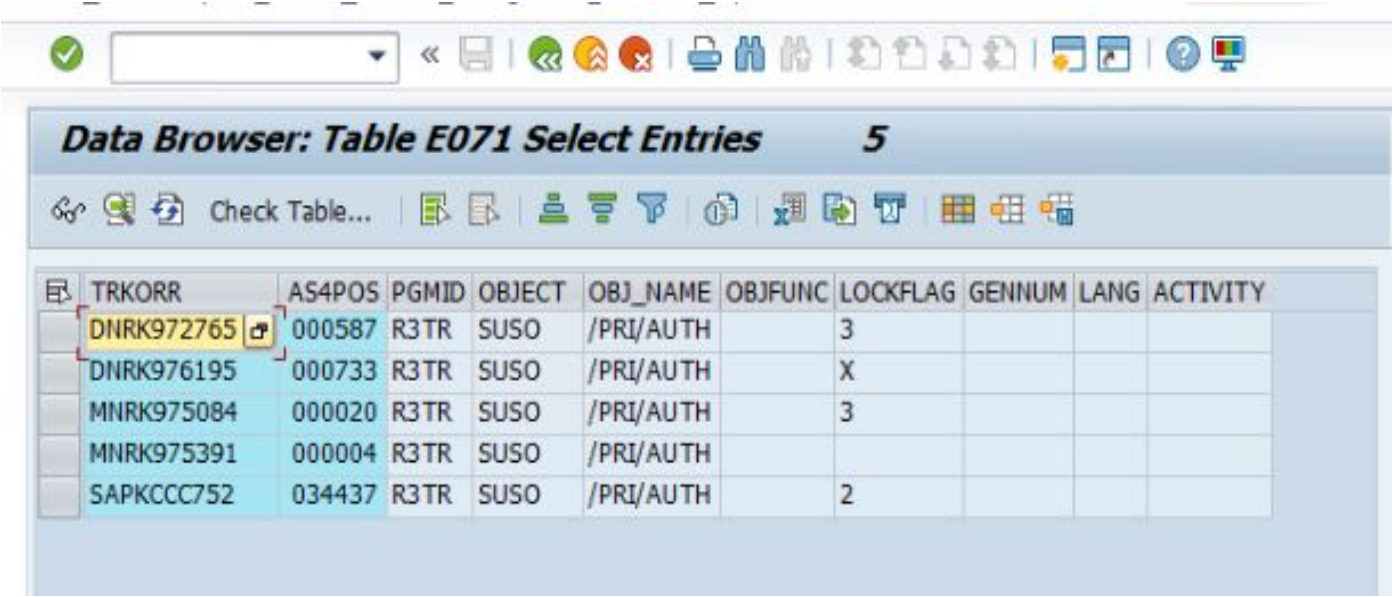
TRKORR		to	
AS4POS		to	
PGMID		to	
OBJECT	SUSO	to	
OBJ_NAME	/PRI/AUTH	to	
OBJFUNC		to	
LOCKFLAG		to	
GENNUM		to	
LANG		to	
ACTIVITY		to	

Width of Output List 250

Maximum No. of Hits 200

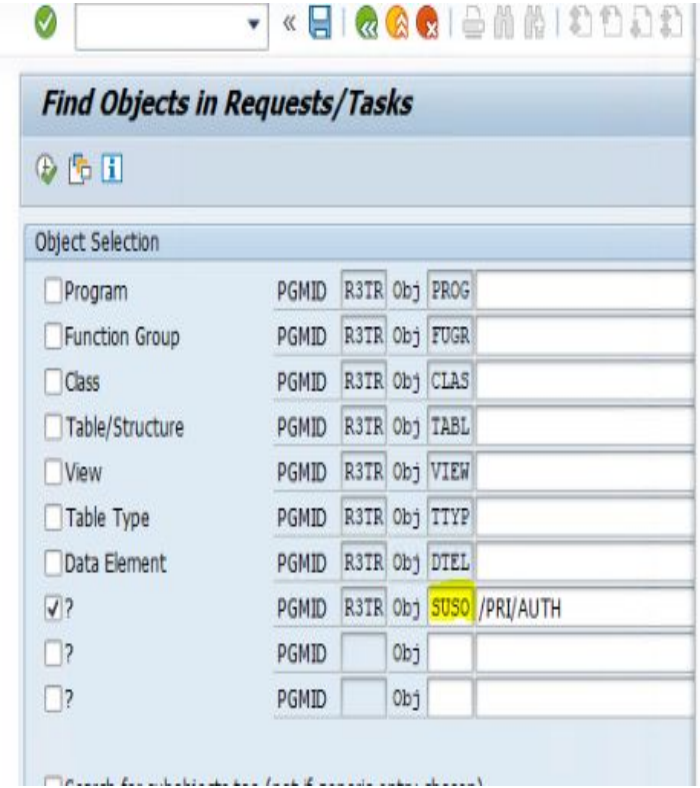
Here in Scoping sheet not find NEW /already existed for SUSO object type, So we are checking in E071 whether object is new in Target/ already existed .

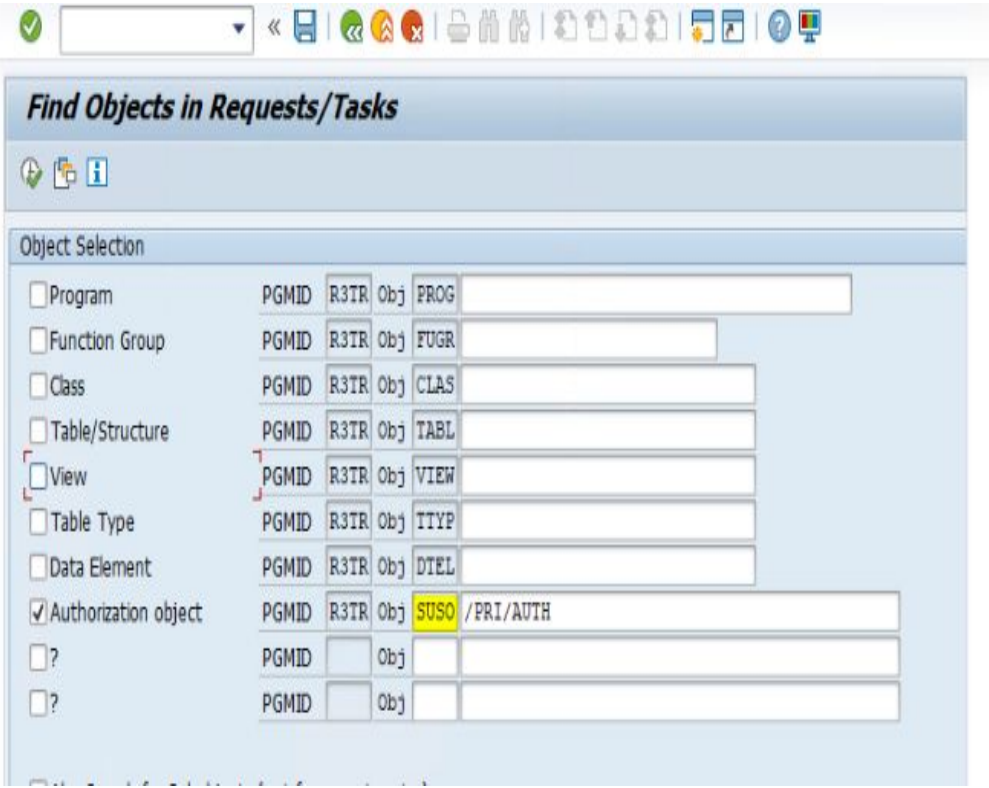
Here in this Example we have found object exist in E071 list . So those objects already exist .Suppose if got ‘No table entries found’ consider it as new object and import (ToC).



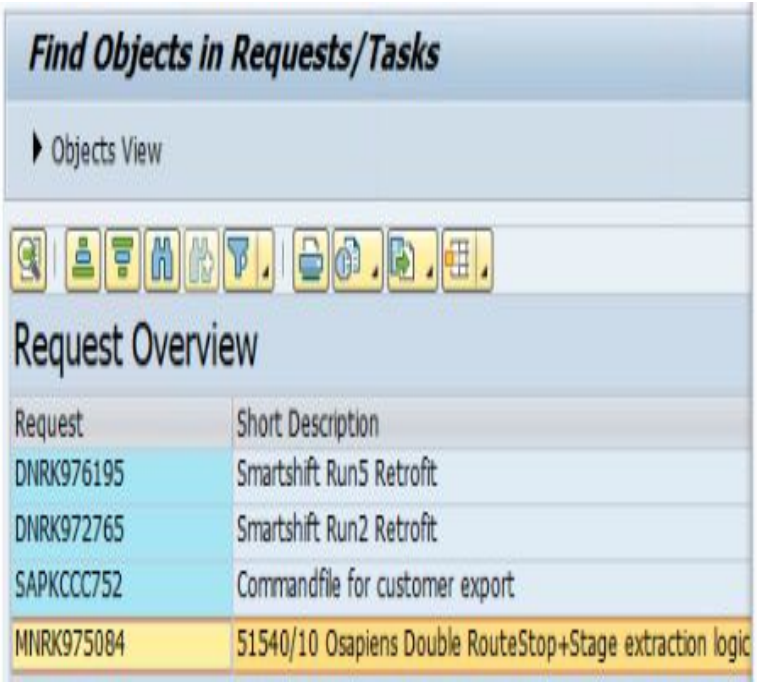
TRKORR	AS4POS	PGMID	OBJECT	OBJ_NAME	OBJFUNC	LOCKFLAG	GENNUM	LANG	ACTIVITY
DNRK972765	000587	R3TR	SUSO	/PRI/AUTH		3			
DNRK976195	000733	R3TR	SUSO	/PRI/AUTH		X			
MNRK975084	000020	R3TR	SUSO	/PRI/AUTH		3			
MNRK975391	000004	R3TR	SUSO	/PRI/AUTH					
SAPKCCC752	034437	R3TR	SUSO	/PRI/AUTH		2			

If object already exist check in E071 , We have to check in se03 'Find objects in request' option in both source and target .





Here in this example : left side of screenshot is target system , we have baseline(common)TR in both systems . top of baseline TR we have our Retrofit TRs and object locked in current retrofit TR .Right side source of screenshot showing scope TR above baseline TR ,there is no target side customer TR to overwrite so we can consider this as safe to import .



SSST(Smart Styles)

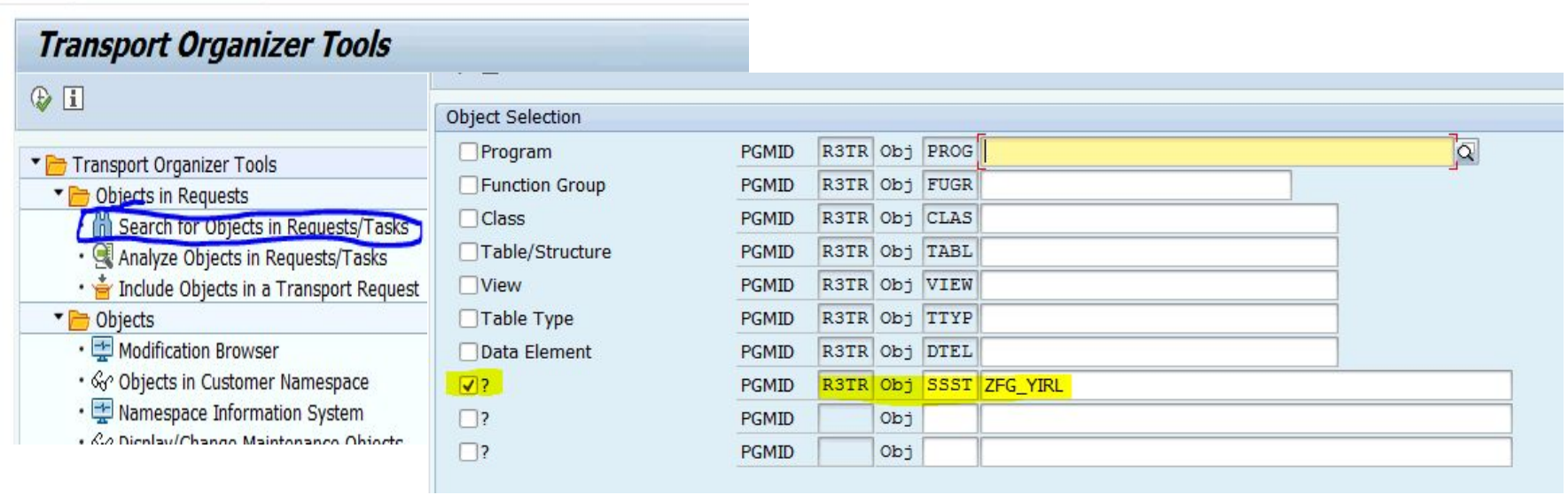
- ➔ Object type 'SSST' refers to SmartStyles objects.
- ➔ TCODE - Smartstyles.
- ➔ This object type is categorized under service scope objects.
- ➔ There is no version management option available here.

Case 1: New Object

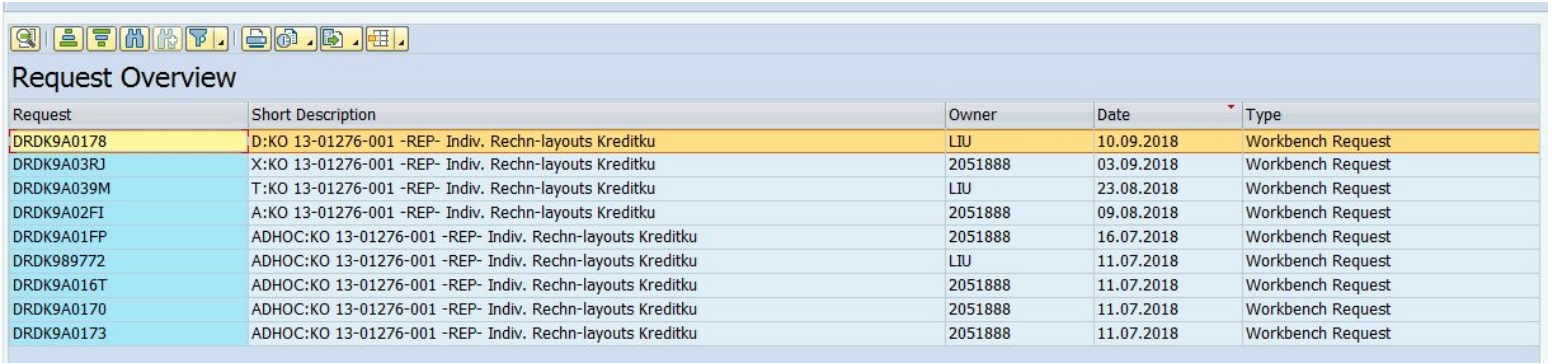
- If the SSST is new to the target system consider as part of ToC objects .

Case 2: Safe to Import

- If the SSST object already exists in the target, then we need to evaluate and decide whether importing the object is safe, meaning that it should not override any changes made specifically for the target.
- We can examine an object's version history using the SE03 transaction in both the source and the target,system and then decide whether or not to import that object.
- **SE03 Screen:** Input object type, object name and execute



- Sort the **date** field in descending order. Below screenshot is from a target system. The highlighted version is the latest version in target and it is from source system DRD.



Request Overview

Request	Short Description	Owner	Date	Type
DRDK9A0178	D:KO 13-01276-001 -REP- Indiv. Rechn-layouts Kreditku	LIU	10.09.2018	Workbench Request
DRDK9A03RJ	X:KO 13-01276-001 -REP- Indiv. Rechn-layouts Kreditku	2051888	03.09.2018	Workbench Request
DRDK9A039M	T:KO 13-01276-001 -REP- Indiv. Rechn-layouts Kreditku	LIU	23.08.2018	Workbench Request
DRDK9A02FI	A:KO 13-01276-001 -REP- Indiv. Rechn-layouts Kreditku	2051888	09.08.2018	Workbench Request
DRDK9A01FP	ADHOC:KO 13-01276-001 -REP- Indiv. Rechn-layouts Kreditku	2051888	16.07.2018	Workbench Request
DRDK989772	ADHOC:KO 13-01276-001 -REP- Indiv. Rechn-layouts Kreditku	LIU	11.07.2018	Workbench Request
DRDK9A016T	ADHOC:KO 13-01276-001 -REP- Indiv. Rechn-layouts Kreditku	2051888	11.07.2018	Workbench Request
DRDK9A0170	ADHOC:KO 13-01276-001 -REP- Indiv. Rechn-layouts Kreditku	2051888	11.07.2018	Workbench Request
DRDK9A0173	ADHOC:KO 13-01276-001 -REP- Indiv. Rechn-layouts Kreditku	2051888	11.07.2018	Workbench Request

- Perform the same steps in the source system. After sorting the date field, check the sequence of the latest version from target with the TR in scope for retrofit.
- As there is no target specific changes exists in target system, the latest version of the object can be safely imported via ToC or dashboard into the target.

Case 3: Manual Retrofit

- When a SSST object is already existed in target and contains target specific changes(identified in SE03), it would be difficult to retrofit since there is no version comparison accessible to identify which changes to be retained or added in the target.
- In such circumstances, we need to inform the customer about the issue/scenario and based on their feedback, further action can be made.

Smartforms

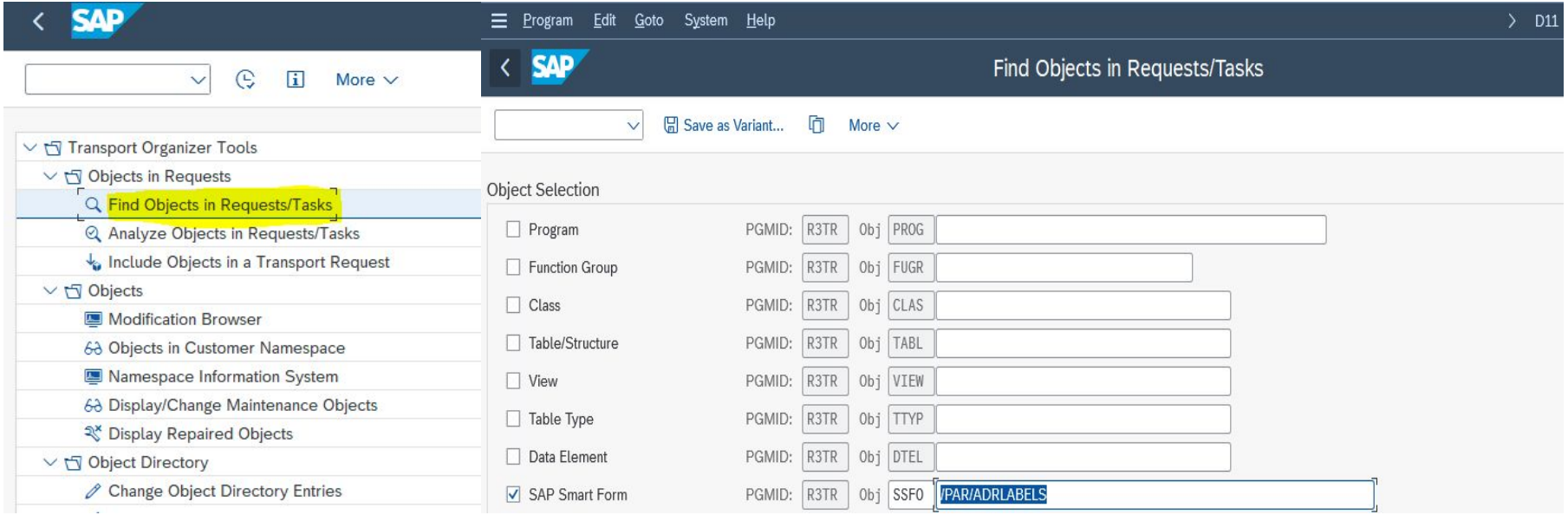
- ➔ Object type 'SSFO' refers to Smartforms objects.
- ➔ TCODE - Smartforms.
- ➔ This object type is categorized under service scope objects.
- ➔ There is no version management option available here.

Case 1: New Object

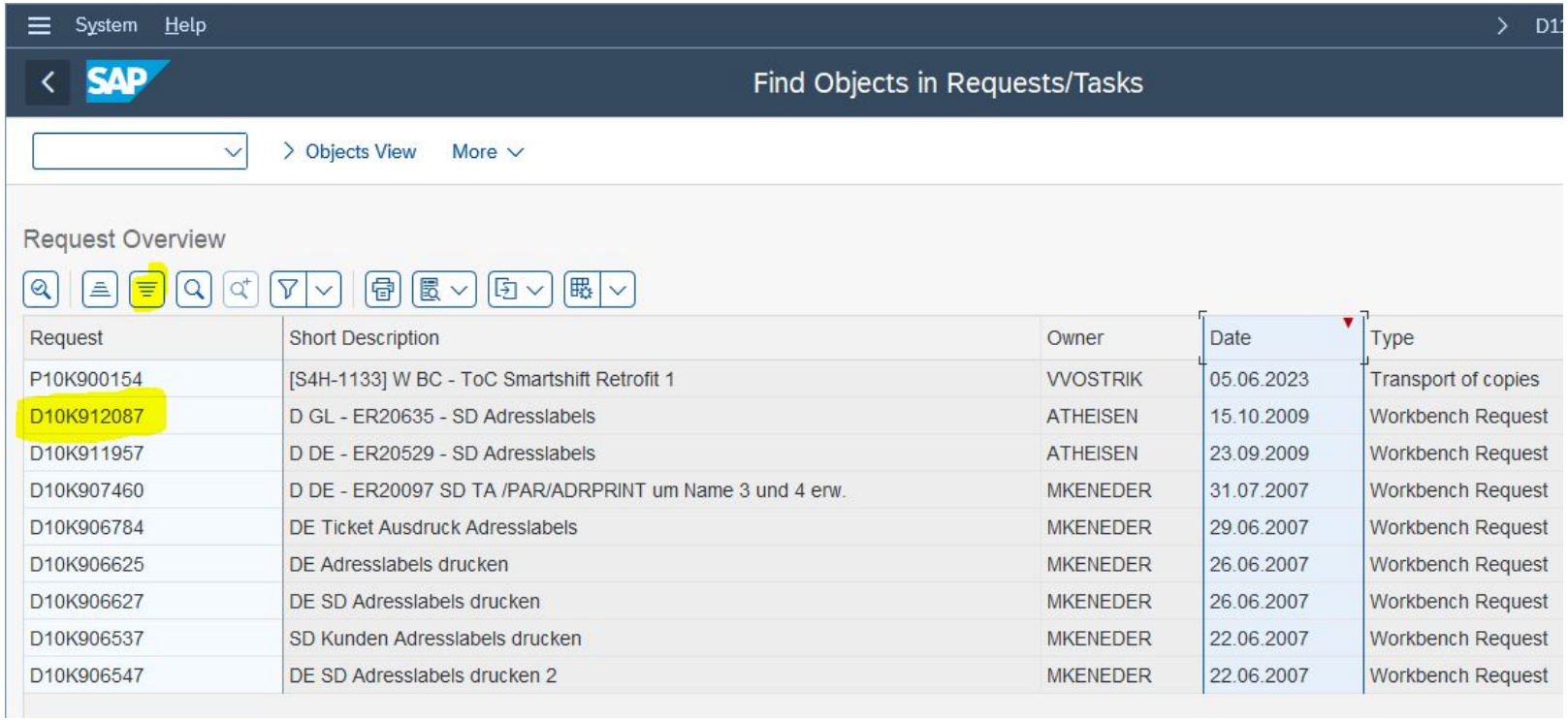
- If the SSFO is new to the target system consider as part of ToC objects .

Case 2: Safe to Import

- If the SSFO object already exists in the target, then we need to evaluate and decide whether importing the object is safe, meaning that it should not override any changes made specifically for the target.
- We can examine an object's version history using the SE03 transaction in both the source and the target,system and then decide whether or not to import that object.
- **SE03 Screen:** Input object type, object name and execute



- Sort the **date** field in descending order. Below screenshot is from a target system(D11). The highlighted version is the latest version in target and it is from source system D10 .



Request	Short Description	Owner	Date	Type
P10K900154	[S4H-1133] W BC - ToC Smartshift Retrofit 1	VVOSTRIK	05.06.2023	Transport of copies
D10K912087	D GL - ER20635 - SD Adresslabels	ATHEISEN	15.10.2009	Workbench Request
D10K911957	D DE - ER20529 - SD Adresslabels	ATHEISEN	23.09.2009	Workbench Request
D10K907460	D DE - ER20097 SD TA /PAR/ADRPRINT um Name 3 und 4 erw.	MKENEDER	31.07.2007	Workbench Request
D10K906784	DE Ticket Ausdruck Adresslabels	MKENEDER	29.06.2007	Workbench Request
D10K906625	DE Adresslabels drucken	MKENEDER	26.06.2007	Workbench Request
D10K906627	DE SD Adresslabels drucken	MKENEDER	26.06.2007	Workbench Request
D10K906537	SD Kunden Adresslabels drucken	MKENEDER	22.06.2007	Workbench Request
D10K906547	DE SD Adresslabels drucken 2	MKENEDER	22.06.2007	Workbench Request

- Perform the same steps in the source system. After sorting the date field, check the sequence of the latest version from target with the TR in scope for retrofit.

Request Overview

Request	Short Description	Owner	Date	Type
P10K900154	[S4H-1133] W BC - ToC Smartshift Retrofit 1	VVOSTRIK	05.06.2023	Transport of copies
D10K932926	[Case#37337] Adresslabel	VVOSTRIK	09.03.2023	Workbench Request
D10K934211	[Case#37337] Adresslabel	VVOSTRIK	09.03.2023	Workbench Request
D10K934226	[Case#37337] Adresslabel	VVOSTRIK	09.03.2023	Workbench Request
D10K934228	[Case#37337] Adresslabel	VVOSTRIK	09.03.2023	Workbench Request
D10K934230	[Case#37337] Adresslabel	VVOSTRIK	09.03.2023	Workbench Request
D10K934232	[Case#37337] Adresslabel	VVOSTRIK	09.03.2023	Workbench Request
D10K912087	D GL - ER20635 - SD Adresslabels	ATHEISEN	15.10.2009	Workbench Request
D10K911957	D DE - ER20529 - SD Adresslabels	ATHEISEN	23.09.2009	Workbench Request
D10K907460	D DE - ER20097 SD TA /PAR/ADRPRINT um Name 3 und 4 erw.	MKENEDER	31.07.2007	Workbench Request
D10K906784	DE Ticket Ausdruck Adresslabels	MKENEDER	29.06.2007	Workbench Request
D10K906625	DE Adresslabels drucken	MKENEDER	26.06.2007	Workbench Request
D10K906627	DE SD Adresslabels drucken	MKENEDER	26.06.2007	Workbench Request
D10K906537	SD Kunden Adresslabels drucken	MKENEDER	22.06.2007	Workbench Request
D10K906547	DE SD Adresslabels drucken 2	MKENEDER	22.06.2007	Workbench Request

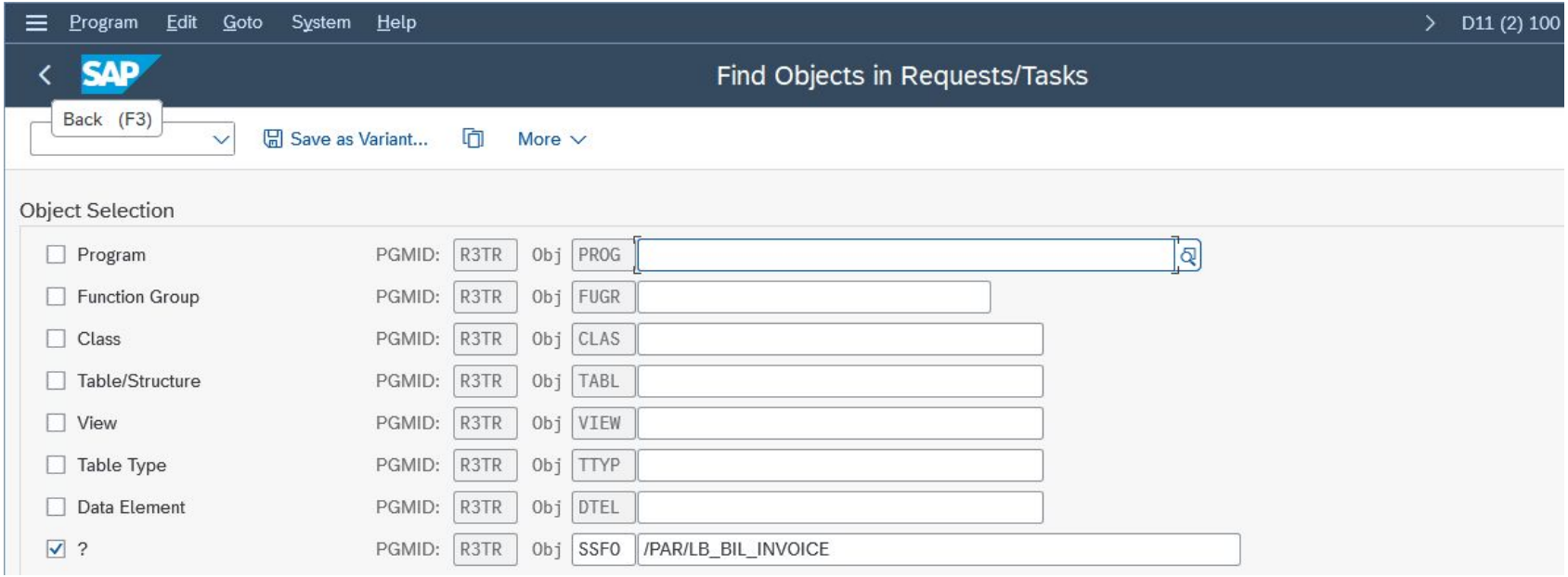
- As there is no target specific changes exists in target system, the latest version of the object can be safely imported via ToC (here P10 you see in source and target screen shot is the T0C TR for Current Run - After Safe to Import Analysis We have added object to ToC and imported from Source(Production p10) to Target (D11)

Case 3: Manual Retrofit

- When a SSFO object is already existed in target and contains target specific changes(identified in SE03), it would be difficult to retrofit since there is no version comparison accessible to identify which changes to be retained or added in the target.

EXAMPLE :

Analysis for safe to import in Se03 for Smartform .



In this Example - Highlighted ‘D11’ TR’s are Target Customer side TR’s and it contains target specific changes (Identified in Se03 Screen below)

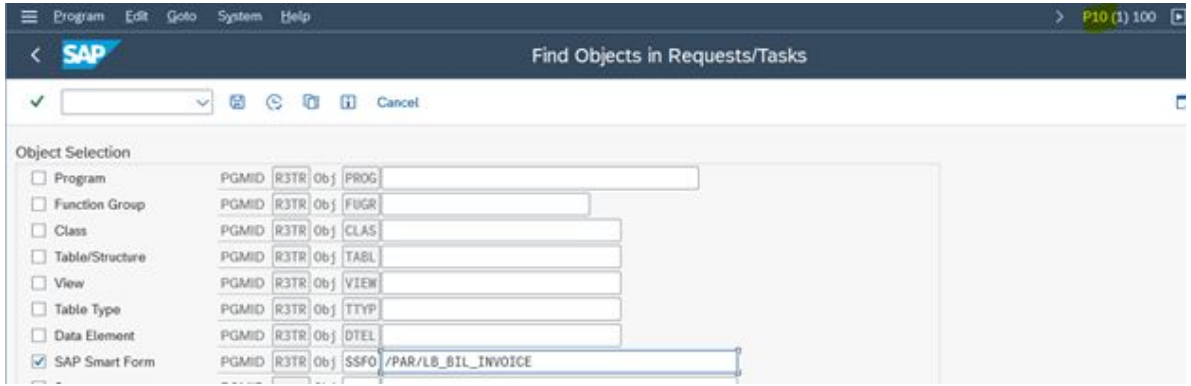
System Help D11 (2) 100

SAP Find Objects in Requests/Tasks

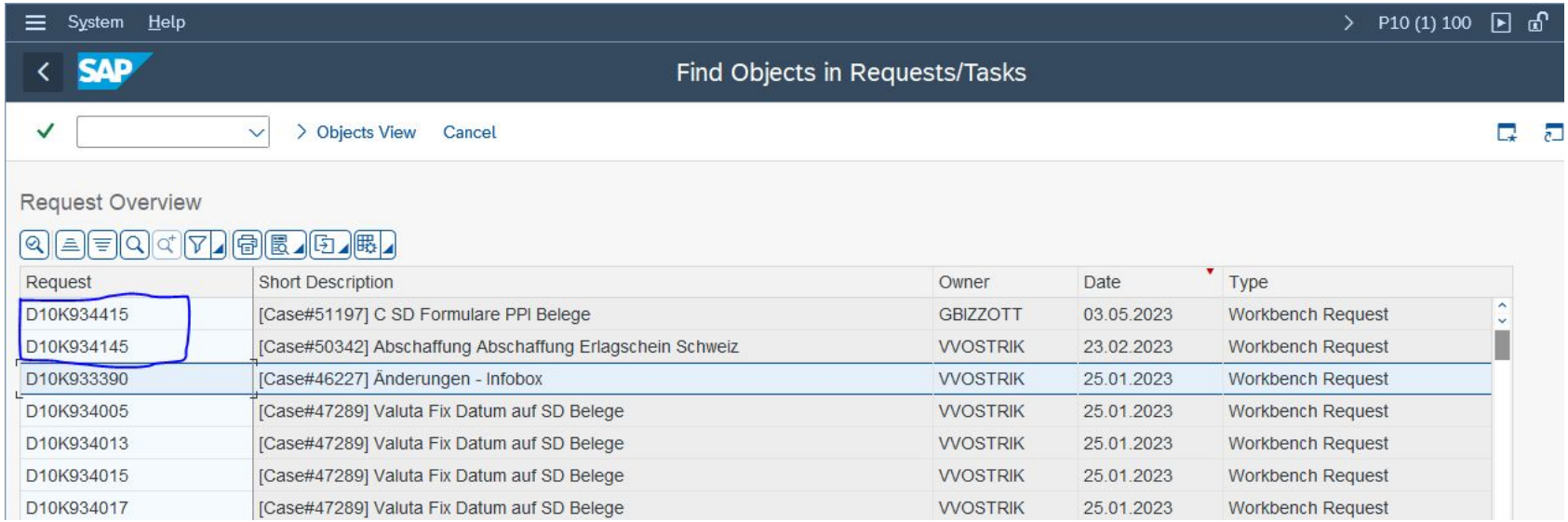
Objects View More Exit

Request Overview

Request	Short Description	Owner	Date	Type	Status
D11K901309	[S4H-1167] System Q10/P10, Partner PI_Q und PI_P festkodiert	VVOSTRIK	24.05.2023	Workbench Request	Released
D11K901293	[S4H-1130] Formular Rechnung FuBa fehlt	VVOSTRIK	15.05.2023	Workbench Request	Released
D11K901054	S4H-887 - W - Smartshift CCC	SMARTSHIFT2	17.04.2023	Workbench Request	Released
D10K933390	[Case#46227] Änderungen - Infobox	VVOSTRIK	23.02.2023	Workbench Request	Released
D10K934005	[Case#47289] Valuta Fix Datum auf SD Belege	VVOSTRIK	23.02.2023	Workbench Request	Released
D10K934013	[Case#47289] Valuta Fix Datum auf SD Belege	VVOSTRIK	23.02.2023	Workbench Request	Released
D10K934015	[Case#47289] Valuta Fix Datum auf SD Belege	VVOSTRIK	23.02.2023	Workbench Request	Released
D10K934017	[Case#47289] Valuta Fix Datum auf SD Belege	VVOSTRIK	23.02.2023	Workbench Request	Released
D10K934019	[Case#47289] Valuta Fix Datum auf SD Belege	VVOSTRIK	23.02.2023	Workbench Request	Released
D10K934145	[Case#50342] Abschaffung Abschaffung Erlagschein Schweiz	VVOSTRIK	23.02.2023	Workbench Request	Released



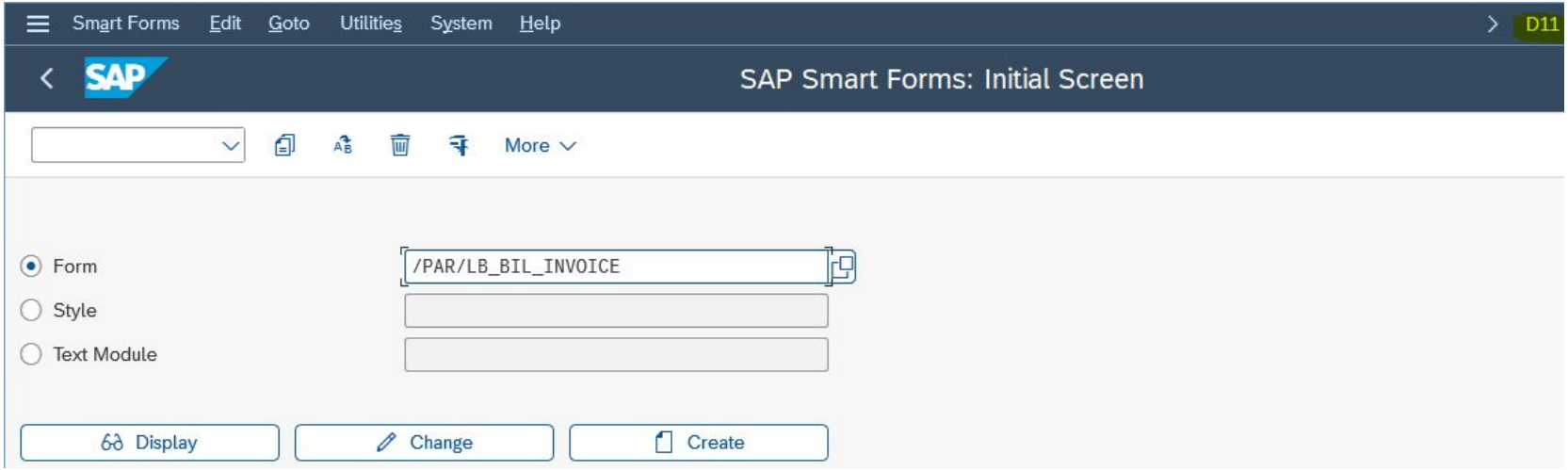
Highlighted D10 TR's are Scope TR's . we need to Add those Scope TR changes to Target retaining Target changes .



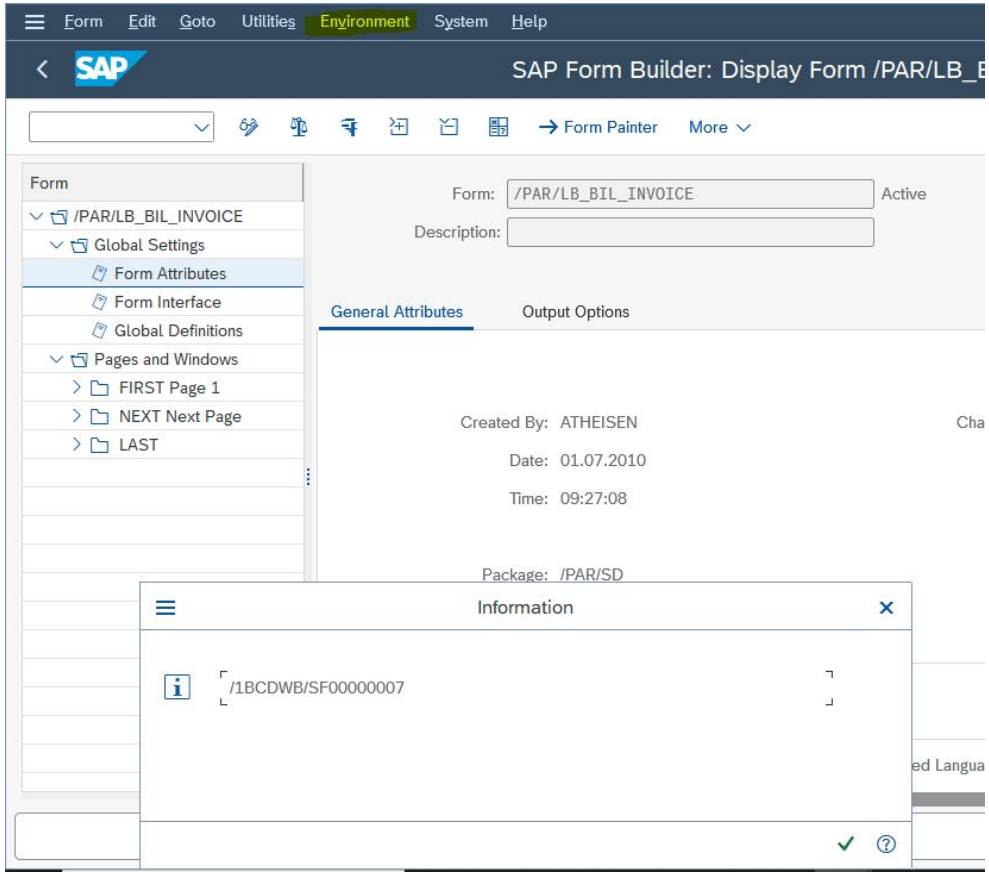
Request	Short Description	Owner	Date	Type
D10K934415	[Case#51197] C SD Formulare PPI Belege	GBIZZOTT	03.05.2023	Workbench Request
D10K934145	[Case#50342] Abschaffung Abschaffung Erlagschein Schweiz	VVOSTRIK	23.02.2023	Workbench Request
D10K933390	[Case#46227] Änderungen - Infobox	VVOSTRIK	25.01.2023	Workbench Request
D10K934005	[Case#47289] Valuta Fix Datum auf SD Belege	VVOSTRIK	25.01.2023	Workbench Request
D10K934013	[Case#47289] Valuta Fix Datum auf SD Belege	VVOSTRIK	25.01.2023	Workbench Request
D10K934015	[Case#47289] Valuta Fix Datum auf SD Belege	VVOSTRIK	25.01.2023	Workbench Request
D10K934017	[Case#47289] Valuta Fix Datum auf SD Belege	VVOSTRIK	25.01.2023	Workbench Request

Manual Comparison Check for Smartform

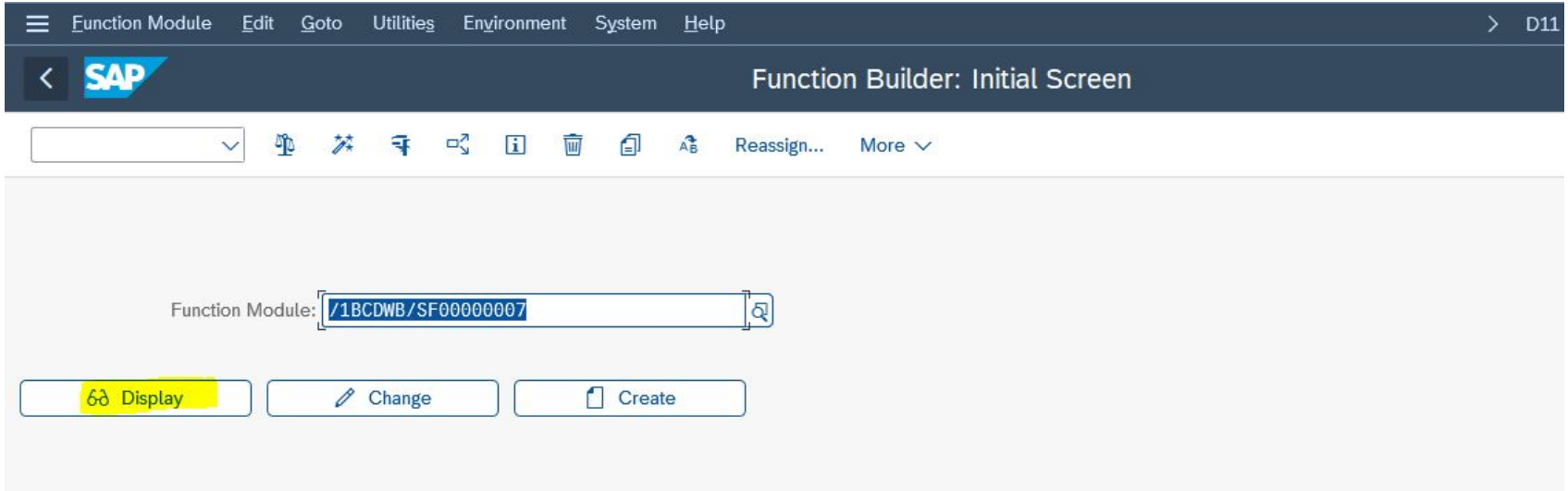
Go To Tcode - Smartform and enter object name and Display in Both Source and Target System Systems.



Go to ‘Environment ‘-> select Function module .



TCODE - SE37, Enter Function Module name and display .



Goto Attributes -> double click Program

Function Module Edit Goto Utilities Environment System Help D11

Function Builder: Display /1BCDWB/SF00000007

More

Function module: /1BCDWB/SF00000007 Active

Attributes Import Export Changing Tables Exceptions Source code

Classification

Function Group: /1BCDWB/SF00000007 Smart Form /PAR/LB_BIL_INVOICE

Short Text: Smart Form /PAR/LB_BIL_INVOICE

Processing Type

- (●) Regular Function Module
- ( ) Remote-Enabled Module

Scope: Not classified (unrestricted scope)

Interface Contract: Any [ ] BasXML supported

- ( ) Update Module
  - (●) Start immed.

General Data

Person Responsible:

Last Changed By: E-PBAIER

Changed on: 06.06.2023

Package:

Program Name: /1BCDWB/SAPLSF00000007

Include Name: /1BCDWB/LSF00000007U01

Double click on “F01” include .
Same steps follow in source system and compare both source and target system ‘F01’ code in Meld.

Program Edit Goto Utilities Environment System Help D11

SAP ABAP Editor: Display Function Group /1BCDWB/SAPLSF00000007

Pretty Printer Text Elements More

Function Group: /1BCDWB/SAPLSF00000007 Active

```
1  ************************************************************************
2  *   System-defined Include-files.                                      *
3  ************************************************************************
4    INCLUDE /1BCDWB/LSF00000007TOP.            " Global Declarations
5    INCLUDE /1BCDWB/LSF00000007UXX.            " Function Modules
6
7  ************************************************************************
8  *   User-defined Include-files (if necessary).                         *
9  ************************************************************************
10 * INCLUDE /1BCDWB/LSF00000007F...             " Subroutines
11 * INCLUDE /1BCDWB/LSF00000007O...             " PBO-Modules
12 * INCLUDE /1BCDWB/LSF00000007I...             " PAI-Modules
13 * INCLUDE /1BCDWB/LSF00000007E...             " Events
14 * INCLUDE /1BCDWB/LSF00000007P...             " Local class implement.
15 * INCLUDE /1BCDWB/LSF00000007T99.             " ABAP Unit tests
16 INCLUDE /1BCDWB/LSF00000007F01.
```

Here comparing source and target code in Meld. Above differences are Smartrules Applied during Transformation in target.

- Meld

Meld File Edit Changes View Tabs

Save Undo

(None) Source

```
777   ENDIF.
778 ELSE.
779 *Bei Service soll auch Easylog Barcode drauf.
780 *Auslieferung ist kein direkter Vorgängerbeleg, aber Nachfolger
781 *Auftrags
782   IF is_bil_invoice-hd_gen-bil_type = 'FR' OR
783      is_bil_invoice-hd_gen-bil_type = 'ZL3' OR
784      is_bil_invoice-hd_gen-bil_type = 'ZTS1'.
785     SELECT SINGLE vbelv INTO lv_reford FROM vbfa WHERE
786              vbeln = is_bil_invoice-hd_gen-bil_number AND
787              vbtyp_v = 'C'.
788     IF sy-subrc = 0.
789       SELECT SINGLE vbeln FROM vbfa INTO lv_refdlv WHERE
790                     vbelv = lv_reford AND
791                     vbtyp_n = 'J' AND
792                     bwart = '601'.
793       IF sy-subrc = 0.
794         SELECT SINGLE * FROM likp INTO ls_likp WHERE
795                                 vbeln = lv_refdlv.
796         SELECT SINGLE * FROM vbpa INTO ls_vbpa WHERE
797                                 vbeln = ls_likp-vbeln AND
798                                 parvw = 'SP'.
799         IF sy-subrc = 0.
800           SELECT SINGLE * FROM /par/easylog_abs INTO ls_abs
801                                 WHERE vkorg = ls_likp-vkorg AND
802                                       vstel = ls_likp-vstel AND
803                                       spedi = ls_vbpa-lifnr AND
804                                       kdgrp = ls_likp-kdgrp.
```

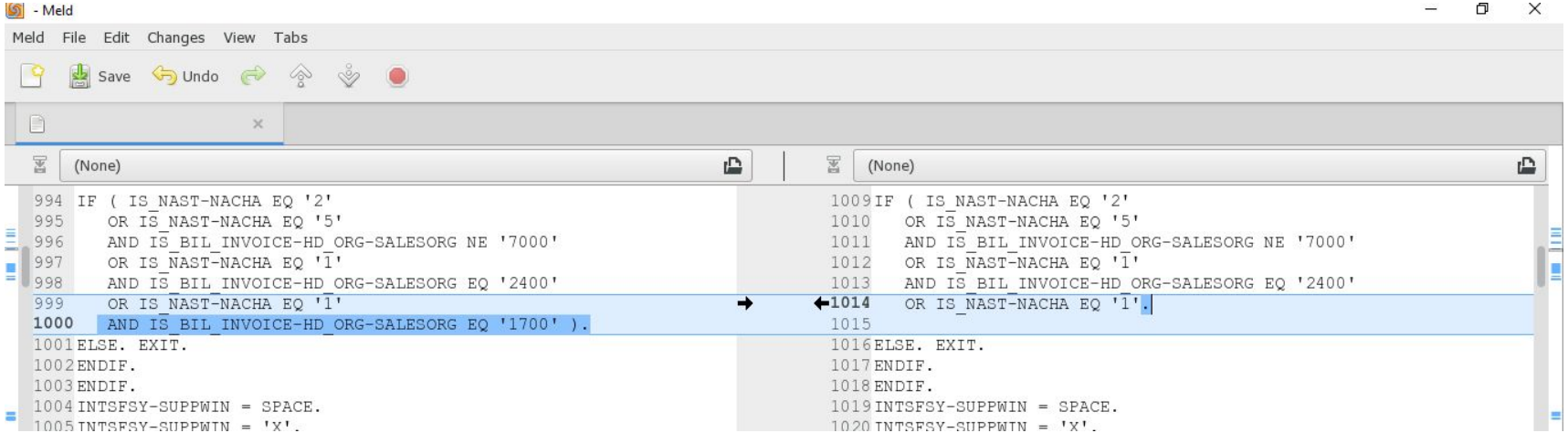
(None) Target

```
785   IF is_bil_invoice-hd_gen-bil_type = 'FR' OR
786      is_bil_invoice-hd_gen-bil_type = 'ZL3' OR
787      is_bil_invoice-hd_gen-bil_type = 'ZTS1'.
788     SELECT vbelv INTO lv_reford FROM vbfa UP TO 1 ROWS      "$sma
789       WHERE                                                 "$sma
790              vbeln = is_bil_invoice-hd_gen-bil_number AND
791              vbtyp_v = if_sd_doc_category=>order      "$smart: #
792                ORDER BY vbelv posnv vbeln posnn vbtyp_n.
793                                                   "$smart: #601 #
794     ENDSELECT.                                              "$sma
795     IF sy-subrc = 0.
796       SELECT vbeln FROM vbfa UP TO 1 ROWS INTO lv_refdlv    "$sma
797         WHERE                                               "$sma
798                     vbelv = lv_reford AND
799                     vbtyp_n = if_sd_doc_category=>delivery  "$sma
800                       AND                                   "$sma
801                     bwart = '601' ORDER BY vbelv posnv "$smart: #
802                       vbeln posnn vbtyp_n.             "$smart: #
803       ENDSELECT.                                            "$sma
804       IF sy-subrc = 0.
805         SELECT SINGLE * FROM likp INTO ls_likp WHERE
806                                 vbeln = lv_refdlv.
807         SELECT * FROM vbpa UP TO 1 ROWS INTO ls_vbpa WHERE "$sma
808                                 vbeln = ls_likp-vbeln AND
809                                 parvw = 'SP' ORDER BY PRIMARY KEY.
810                                                             "$sma
811         ENDSELECT.                                          "$sma
812         IF sy-subrc = 0.
813           SELECT SINGLE * FROM /par/easylog_abs INTO ls_abs
```

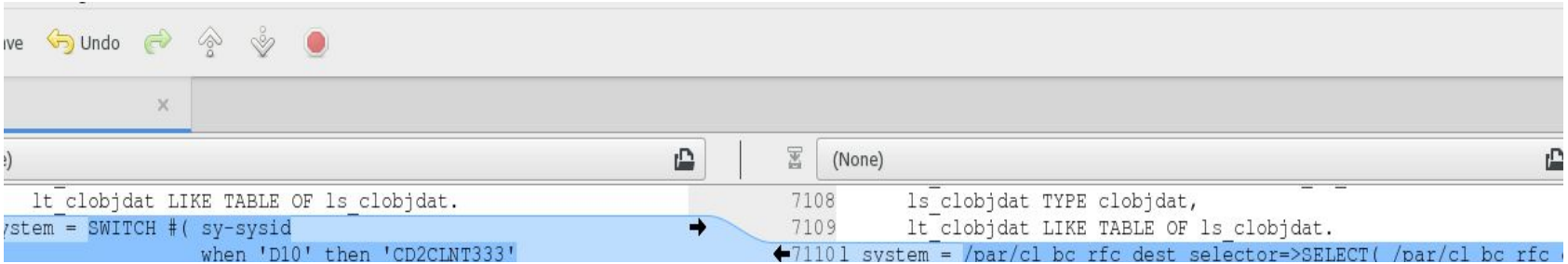
Ln 785, Col 1 INS

Type here to search 38°C Sunny 1:01 PM 6/8/2023

In the below screenshot, source side extra changes are found. We can consider this as Source change and need to bring above change into target system .



Here in beloe screenshot, it’s a conflict .
In such circumstances, if we found any Conflict changes we need to inform the customer about the issue/scenario and based on their feedback, further action can be made.



Note :
We can Follow Another way to find code for Smartform .
Goto Smartform Tcode , enter object name and Display , Goto ‘Utilities ‘- > select Form Information and compare code in meld from both systems .
Some Customers restrict Form information . Suppose If we have Authorization Follow above steps .

Divya Mary Terry

Object T	Sco	Scope	SPOC
PROG	IN	TOOL	Divya
REPS	IN	TOOL	Divya
REPT	IN	Service	Divya
DEVC	IN	Service	Divya
OA2S	IN	Service	Divya
TABU	IN	Service	Divya

REPT

➔ REPT - Report Text/Text Elements

- Format in scoping sheet: LIMU-REPT-ObjectName
- Is associated with Text Symbols, Selection Text, List Heading
- As part of Retrofit, we consider REPT under Service scope
- REPT can be maintained for programs,function groups, classes
- To display REPT, Open the object and use Menu → go to → Text Elements/Text

  Also we can check directly in se38 →Text elements radiobutton

  [when you try to open REPT from se38(for main obj CLAS/FUGR), see that Object name is taken correct from scoping sheet; Otherwise open the object in its actual TCode se24 for CLAS and se80 for FUGR and use GoTo → Text elements option]

- Can REPT be considered as **part of Safe To Import** analysis/ToC?
  - Yes - For NEW, IN scope Objects
  - Not Yet- For EXISTING, IN scope Objects; Discussions are going on to see if we can consider REPT as part of Safe to import analysis…As of now we have to Fix them manually.

- Points to note:
  - REPT should be first maintained in Original language(say primary language)
  - Then in secondary language(Translated language: most commonly Translated language may be English or German…As most of our clients are from this area) - For now we are not focusing on Translations in other languages.
  - Never fix REPT for **MPC** and **DPC** class manually! This should be checked at IWPR or corresponding structure level.

➔ Procedure to check REPT:

REPT - When we Retrofit REPT, we have to maintain in Original language as well as Translations; We won't be considering all language translations as it's not practical. Instead we choose something like a primary language and secondary language.

Primary language: will be the original language of the object

Secondary language: will be one of the translated language(EN or DE for most of our clients) { Check with your SPOC to understand the secondary language you have to use for REPT translation for Current client }

Note:

>>Some case REPT will only be maintained in Original language

>>Some cases Translation will be saved as a LANG entry by customer/source developer

If so you can find a LANG entry in scoping sheet as LANG-REPT-ObjectName ; LANG entry can be considered as part of Safe to Import ToC after safe to import analysis.

Whether we have a LANG entry or not, we have to follow below steps to Retrofit REPT.

Example:

Assume you have logged in to system (source and target) in EN (English language)

Scope TR: DEVxxx99

Object LIMU REPT ZSAMPLEPROGRAM

<1>First identify the Original language for current REPT object

Easy way to identify the original language is by just entering the Display mode of Text element(either via se38 Text elements radio button or open the object and use Menu Go to Text elements)

When you enter display mode,

- if you get a pop-up message or message towards the bottom of the screen like:
  ‘Untranslated text in XX will be displayed’ here XX will be the original language
  Or a message with similar meaning ‘Original language XX, Displayed in YY’
- Otherwise, (if you don’t get any message like above)
  Then it implies you are already in the original language for current object

Say, for the current object REPT ZSAMPLEPROGRAM you got message:
‘Untranslated text in DE will be displayed’ → Then it means original language is DE (German) for current object, NOT English (EN) the current language you are logged in.
In this scenario we can consider, primary language as DE and secondary language as EN

<2> Login in Original language in target--compare with scope TR--do Retrofit

[Steps to change the language you have logged in: (This is explained later with screenshots)]

Login in Original language in target- DE in our case, do a remote comparison to source system scope TR version, analyse the comparison difference and perform Retrofit.

When you compare, you have to **check all 3 tabs- Text Symbols, Selection Text, List Heading**

If there is No difference between Scope TR and Target Active version:Then it's a,

- No Action-Identical case in DE language.

If there is difference between Scope TR and Target Active version:

Analyse if it's a scope TR change or Target customer change(as we do for other object types, comparing with below versions)
And perform the Retrofit if necessary by entering the EDIT mode for the object.
[Note: Sometimes if EDIT is disabled, you may have to switch off the Modification assistant to enter change mode and save]
Here the scenarios can be:

- No Action – Target has the latest change for REPT in Original lang.
- Retrofit Done – Added scope TR changes to target in Original lang.
- Retrofit Done – Added scope TR changes to target in Original lang DE, retaining customer target changes.

<3> Login in Secondary language in target--compare with scope TR--do Retrofit

Login in Secondary language in target- EN in our case, do a remote comparison to source system scope TR version, analyse the comparison difference and perform Retrofit.

Above mentioned scenarios of Original language will be applicable in Translation as well.

- No Action-Identical case in EN language translation.
- No Action – Target has the latest change for REPT in Translated lang EN.
- Retrofit Done – Added scope TR changes to target in Translated lang EN.
- Retrofit Done – Added scope TR changes to target in Translated lang EN, retaining customer target changes.

To Retrofit the Translation,
First try, if u can directly edit in secondary lang(here EN) for REPT
i.e., directly enter edit mode in EN and see if all other text are also in English and you can make your changes in EN ; If you can, then save the changes from this direct textelement screen.

Most of the time, there will be a restriction to directly edit REPT in other (other than original) languages; If so, When we are in EDIT mode in Original language, we have to enter the translation mode from Menu:Go To→Translation(In the pop-up screen mention translation from which to which language, here DE to EN). Now perform the Retrofit, SAVE your changes, Come back to REPT/Textelements screen and Activate the Object again.

[Note: Some case REPT will only be maintained in Original language]

<4> Update Status and Comments in scoping sheet
Status and Comments in scoping sheet should be a combination of Action you have taken for primary and secondary language.
If you have Retrofitted in at least one language, then status will be ‘Retrofit Done’
If you made No changes in any of the language, then status will be ‘No Action’

Example for status comments:
Retrofit Done – Added scope TR changes to target in Original lang DE and translation maintained in EN.
Retrofit Done – Added translations in EN as per latest scope TR; REPT looks identical in Original lang DE.
Retrofit Done – Added scope TR changes to target in Original lang DE and translation maintained in EN, retaining customer target changes.
No Action – Identical in source and target in both Original lang DE and EN translation.
No Action – Target has the latest change for REPT in both Original lang DE and EN translation.

Capsule steps for REPT Retrofit:

1> Identify the Original lang & Secondary lang for current REPT object.
2> Login in Original lang(say DE)—compare with scope TR--do Retrofit
3> Login in Secondary (EN/DE) lang--compare with scope TR--do Retrofit
    3.1>if u can directly edit in Secondary lang REPT-do that
    3.2>Otherwise, Login in Original lang-->GoTo-->Translation--> Secondary lang----do Retrofit and SAVE-->come back to REPT and Activate again.
4> Update Status and Comments in scoping sheet

Above mentioned steps were for Individual comparison, Object by object.

You can perform a bulk comparison (8 REPT objects at a time) using RSSYSCOMP program, Provided you know the Original language and translation maintained secondary language for all 8 objects.
So first you have to login in original language and use RSSYSCOMP, Later login in secondary(Translated) language and use RSSYSCOMP; And based on the analysis you can make changes.

Note: When we compare and find that a selection text is newly added as part of Scope TR change, and we have to add the Selection text to target as part of Retrofit,- Then we may have to keep the REPT on Hold and wait till tool scope for the object is completed, As there is NO direct add/+/Insert icon for Selection Text; However deletion is possible.

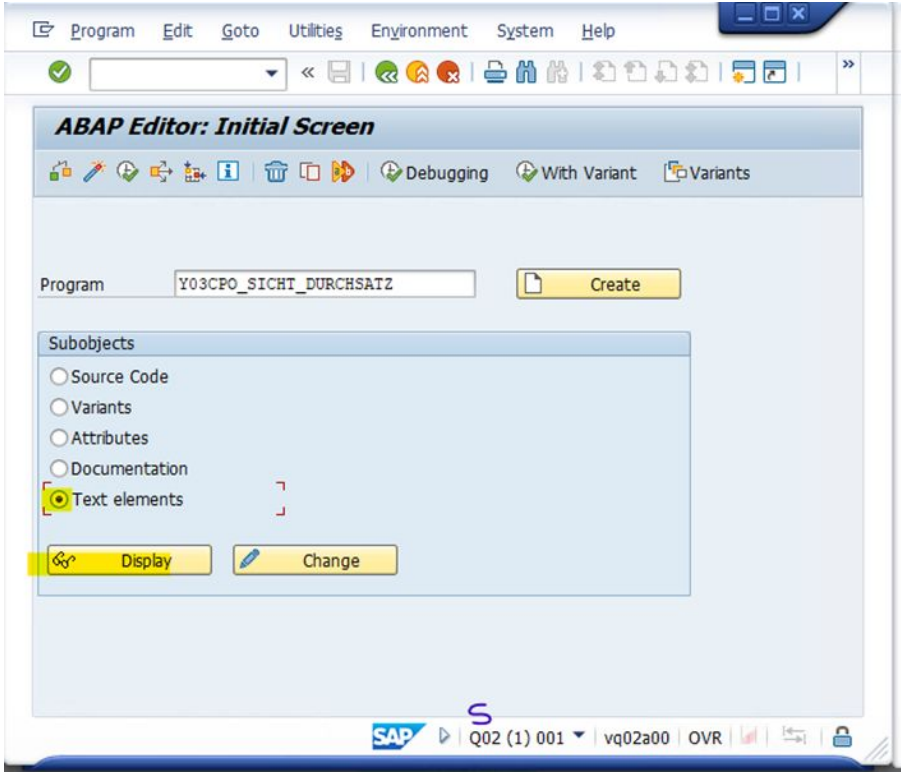
REPT Example:

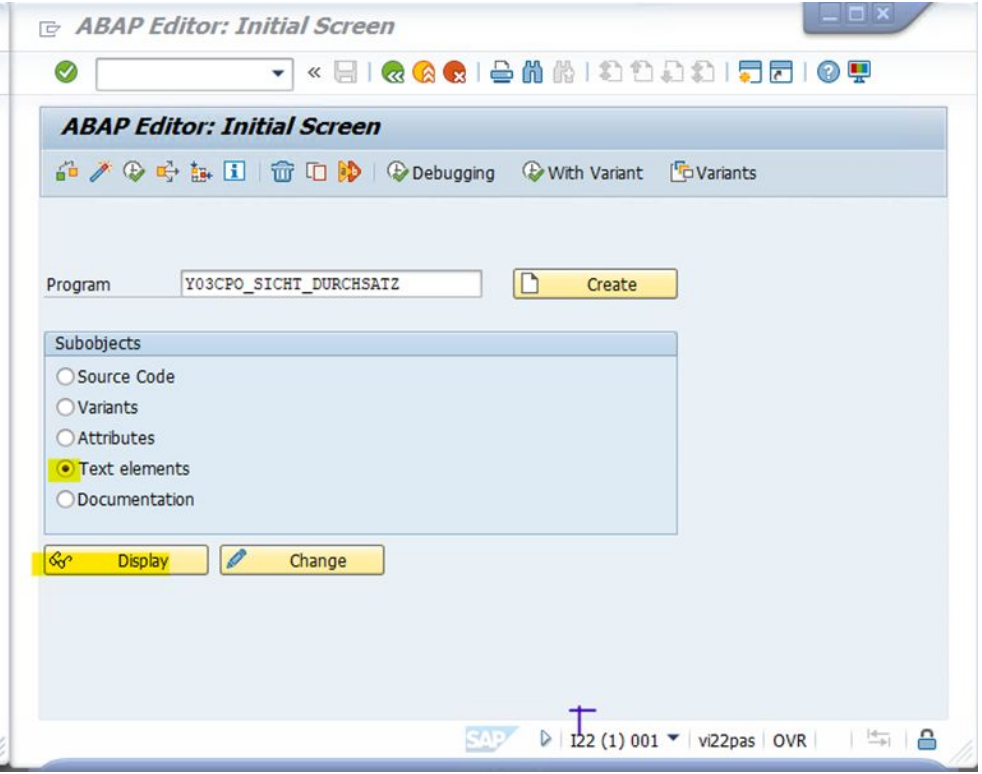
Scope TR	I02K9P7987
Object	LIMU - REPT - Y03CPO_SICHT_DURCHSATZ
Source system	Q02
Target system	I22
Working(Retrofit) target TR	I22K907482
Client/Customer	FESTO is the client; Therefore, if original lang is EN-we'll check if Translations available in DE if original lang is DE-we'll check if Translations available in EN

Steps to cover:

1) Identifying original language and secondary language
2) Compare in Original language(primary lang) with scope TR and Retrofit accordingly
3) Compare in Secondary language(for maintaining translation lang) with scope TR and Retrofit accordingly
4) Update combined status and comments in Scoping sheet

Step (1) Trying to Identifying original language and secondary language





Display the REPT from se38 by choosing text elements radio button

ABAP Text Elements: Display Text Symbols Language English

Program: Y03CPO_SICHT_DURCHSATZ — Active

Text Symbols | Selection Texts | List Headings

Next Free Number

S...	Text	Lngth	Max.
001	Time Int	8	8
002	Selection	9	12
003	Posting Date	12	25
004	Start Date greater End Date	27	27
005	to	2	3
006	Confirmation Report	19	40
007	Booking Period or Date missing!	31	80
008	Booking Period	14	15
009	No Confirmations found	22	28

Original language German (DE), ... | Q02 (1) 001 | vq02a00 | OVR

Text Elements | Edit | Goto | Utilities | Environment | System | Help

ABAP Text Elements: Display Text Symbols Language English

Program: Y03CPO_SICHT_DURCHSATZ — Active

Text Symbols | Selection Texts | List Headings

Next Free Number

S...	Text	Lngth	Max.
001	Time Int	8	8
002	Selection	9	12
003	Posting Date	12	25
004	Start Date greater End Date	27	27
005	to	2	3
006	Confirmation Report	19	40
007	Booking Period or Date missing!	31	80
008	Booking Period	14	15
009	No Confirmations found	22	28

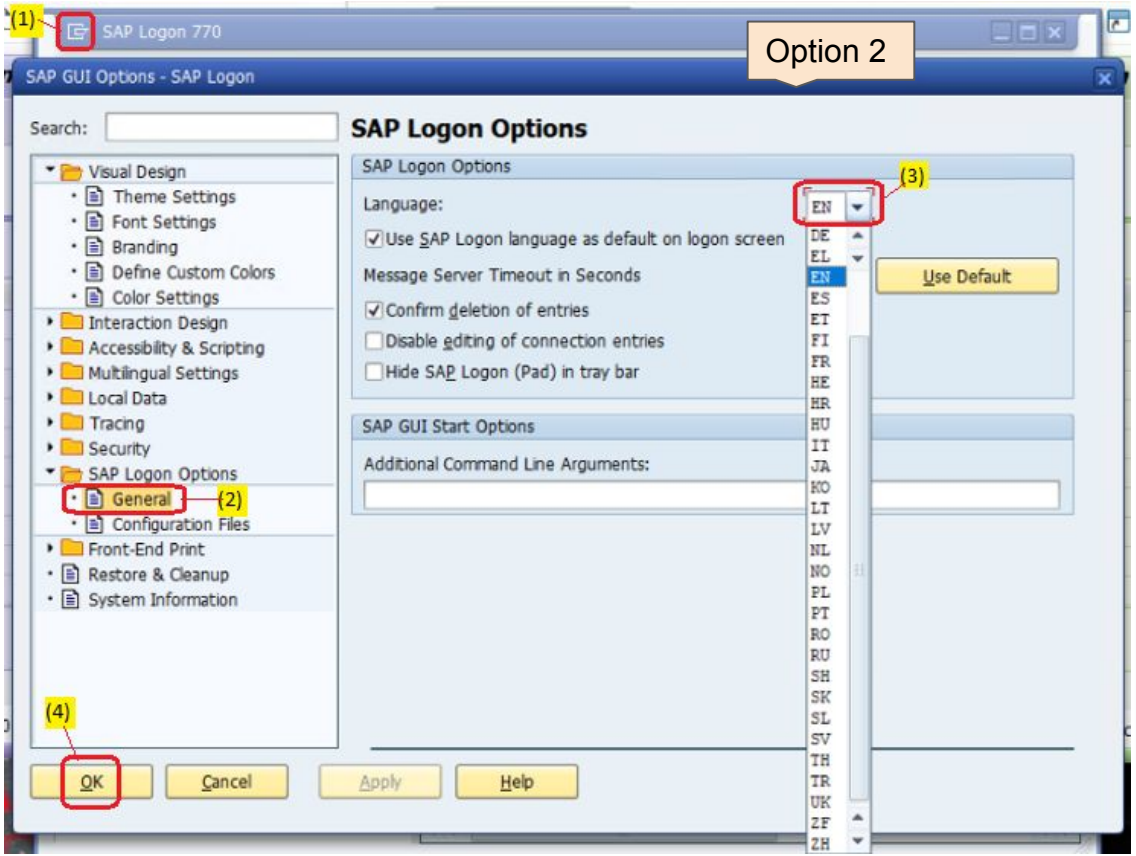
Original language German (DE), Display in English (EN) | I22 (1) 001 | vi22p

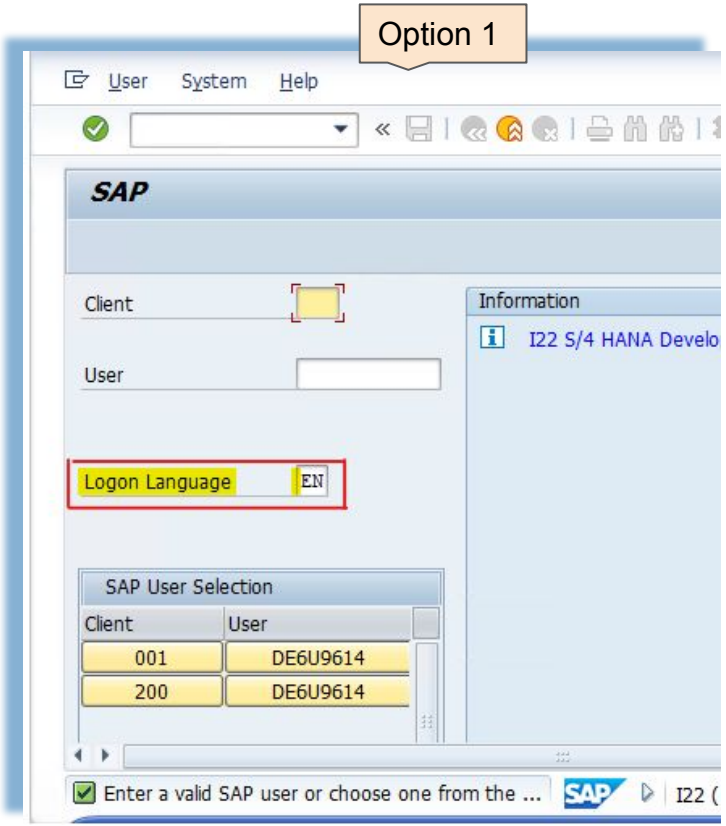
From above screenshot it's clear that Original language is DE for the current object, so we can choose EN as secondary language.

Now we have to login in original language, Here we have to change from EN(English) to DE(German) login

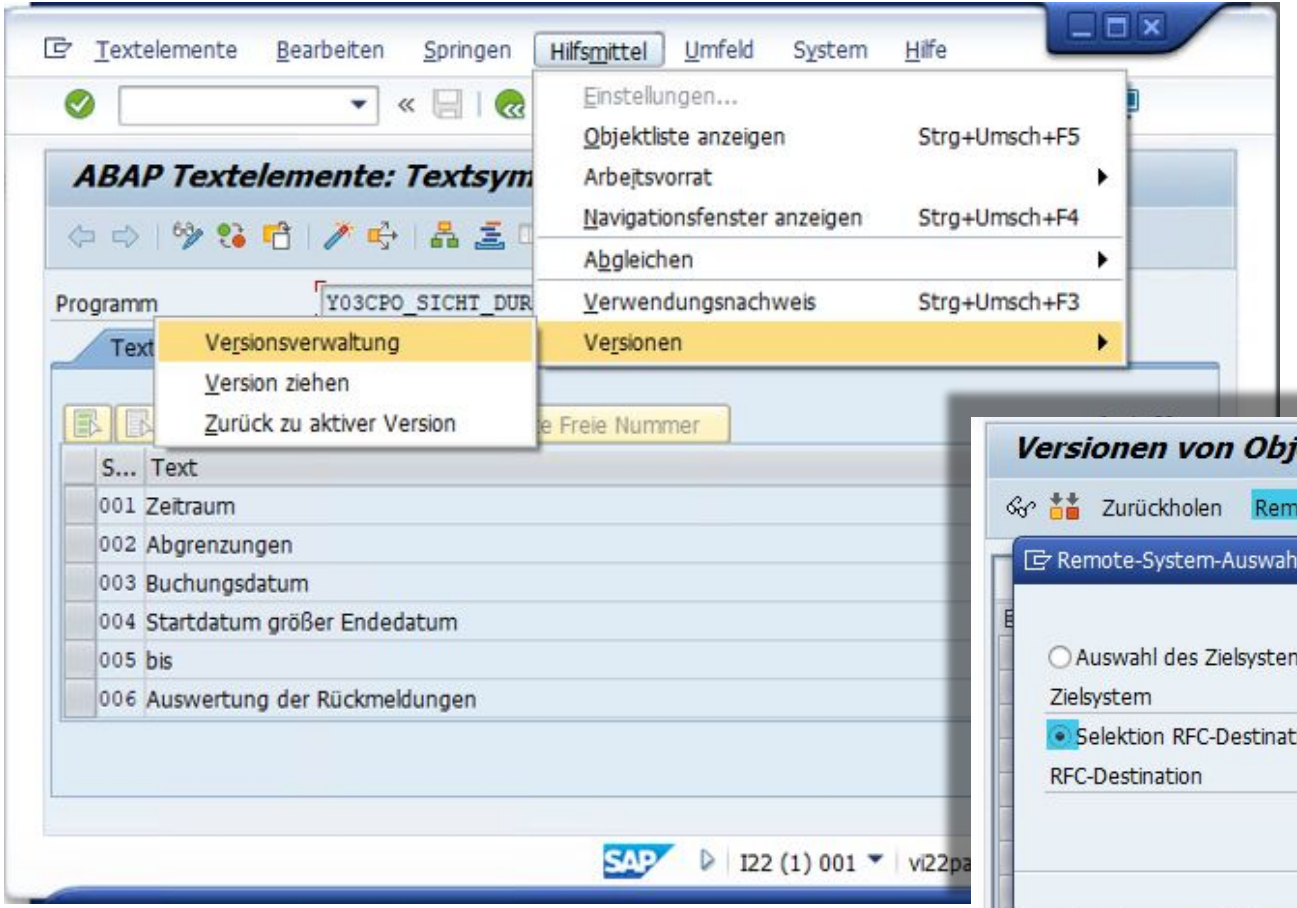
How to change the SAP login language??

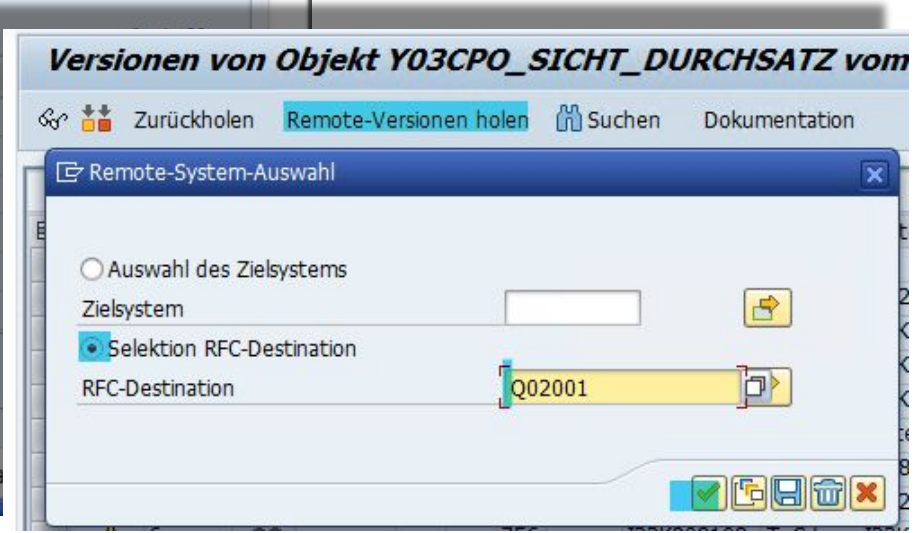
If option to change Logon language is given in login screen change there(refer option 1), Otherwise follow the steps in option 2.





Step (2) Compare in Original language(primary lang) with scope TR and Retrofit accordingly





→Login in original language DE
→Open the Text element
→Using Fetch Remote version, giving RFC connection id to source, get the source versions to compare

Versionen Bearbeiten Springen System Hilfe

Remote-Versionen von Objekt Y03CPO_SICHT_DURCHSATZ vom Typ Reporttexte

Remote-Versionen holen Suchen

	SID	Status	Version	Vergl.	Kat...	Knz.	SAP	Rel.	Arc...	Anforderung	Anforderungstext	Projekt
S	Q02		active					750				
			5		I			750		I02K9P7BRF	S 8200026485: 0430 : Y Transaction Migration to NGE (S	
			4		I	Scope TR--->		750		I02K9P7987	S 8200025665: YCFR time	
			3		I			750		I02K9P73NF	S 8200023815: YCFR add personnel number to ALV grid	
			2		I			750		I02K9P41BU	CB_3446002 ; YCFR Translation EN	
			1		S			750		I03K940266	Programme WAS aus Paket Y03CPO	
T	I22		active					756		I22K907482	S 8200027312: CM Retrofit#4 handover 1	I22_P00004
			4					756		I22K903455	S 8200025092: CM Code Retrofit #3 all	I22_P00004
			3					756		I22K900755	S 8200021738: change origin system I22 after carveout 2.2	
			2					756		I22K901151	ToC NGE: consolidated transport for C22 preparation (0 Test)	
			1		S			756				

Suchen Sie mit der Such-Drucktaste in der Symbolleiste nach Änderungen innerhalb von Versionen.

I22 (1) 00

Compare target with Scope TR

Textelemente Bearbeiten Springen System Hilfe

Versionsvergleich: Textelemente

Note: There is No 'Delta display' button when you compare REPTs; What you see itself is the delta view

Vergleich von ABAP-Textelementen Programm Y03CPO_SICHT_DURCHSATZ

Textelemente: Version 00004 Sprache D
Lokales System: I22
Auftrag: I22K903455 06.01.2023 DE6U3189
Text:

Textelemente: Version 00004 Sprache D
Remote System: Q02
Auftrag: I02K9P7987 29.12.2022 US0WAS
Text:

Id	Schlüssel	Länge	DDIC	Text
S	P_RUECK	36		Einzelrückm. (User Zeitzone)
S	P_RUECK	38		Einzelrückm. (User Zeitzone)

SAP I22 (1) 001 vi22pas OVR

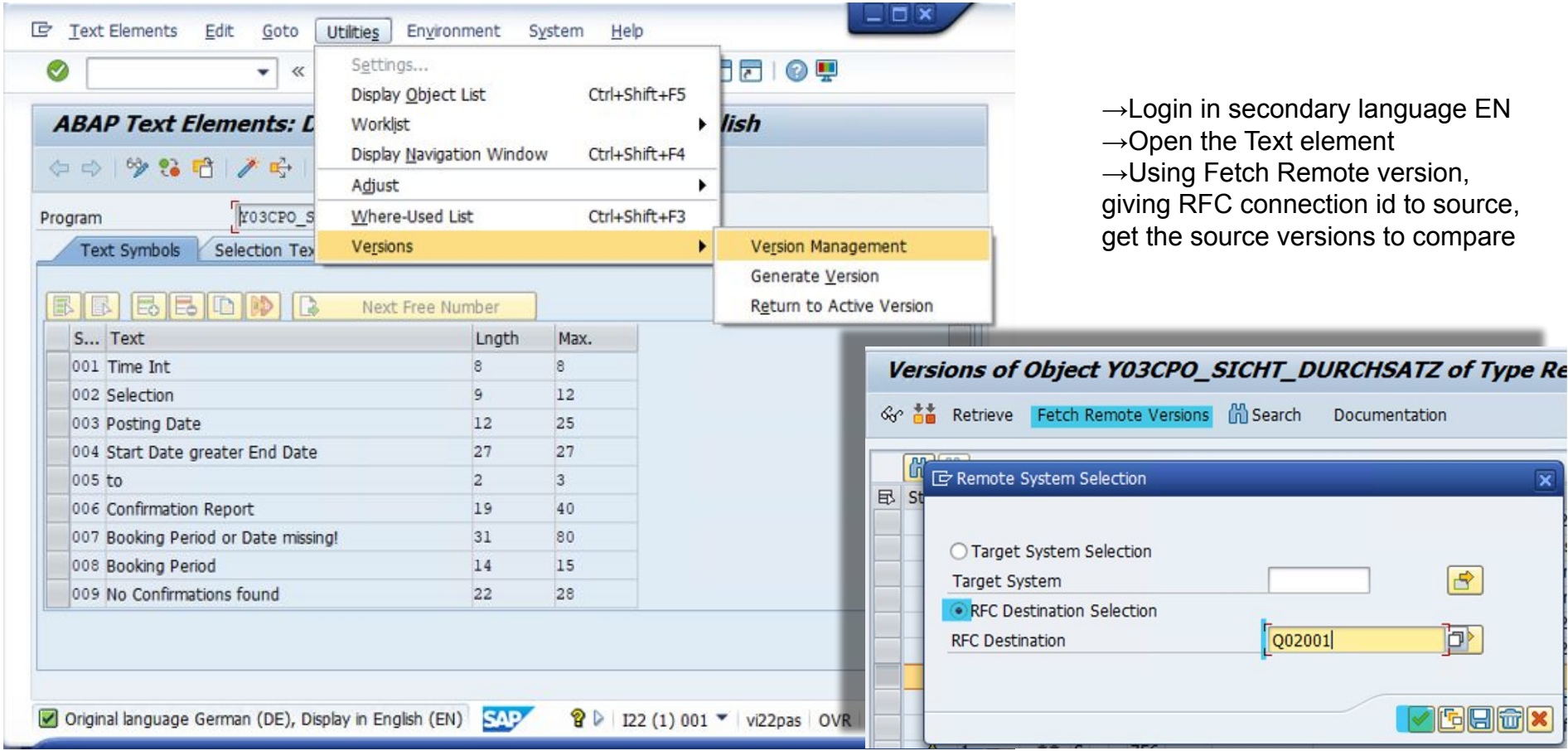
→ When we compare target with Scope TR, we get only difference of Länge(Length in english) in Selection Text; and this length difference can be ignored- As we are not giving any length explicitly for selection text

status and comments for Original lang DE will be:

No Action - REPT looks identical in original language DE.

→ Now we have to check secondary language EN (english) to compare and identify if any difference exist.

Step (3) Compare in Secondary language(for maintaining translation lang) with scope TR and Retrofit accordingly



→Login in secondary language EN
→Open the Text element
→Using Fetch Remote version, giving RFC connection id to source, get the source versions to compare

Remote Versions of Object Y03CPO_SICHT_DURCHSATZ of Type Report Texts

Remote Versions of Object Y03CPO_SICHT_DURCHSATZ of Type Report Texts

Fetch Remote Versions | Search

	SID	Status	Version	Compare	Cat.	Flag	SAP	Rel.	Arch.	Request	Request Text	Project
S	Q02		active					750				
			5		I			750		I02K9P7BRF	S 8200026485: 0430 : Y Transaction Migration to NGE (S	
			4		I	Scope TR--->		750		I02K9P7987	S 8200025665: YCFR time	
			3		I			750		I02K9P73NF	S 8200023815: YCFR add personnel number to ALV grid	
			2		I			750		I02K9P41BU	CB_3446002 ; YCFR Translation EN	
			1		S			750		I03K940266	Programme WAS aus Paket Y03CPO	
T	I22		active					756		I22K907482	S 8200027312: CM Retrofit#4 handover 1	I22_P00004
			4					756		I22K903455	S 8200025092: CM Code Retrofit #3 all	I22_P00004
			3					756		I22K900755	S 8200021738: change origin system I22 after carveout 2.2	
			2					756		I22K901151	ToC NGE: consolidated transport for C22 preparation (0 Test)	
			1		S			756				

Use toolbar Search button to search for changes within versions

T

I22 (1) 001 | vi22pas

Compare target with Scope TR

Compare Versions: Text Elements

Compare ABAP text elements in program Y03CPO_SICHT_DURCHSATZ

Text elements: Version 00004 Language E
Local system: I22 T
Request: I22K903455 06.01.2023 DE6U3189
Text:

Text elements: Version 00004 Language E
Remote system: Q02 S
Request: I02K9P7987 29.12.2022 US0WAS
Text:

Compare text symbols
Only exist in the remote system version:

ID	Key	Length	Text
I	ZZ1	76	You don't have permission for Plant:

Compare ABAP text elements in program Y03CPO_SICHT_DURCHSATZ

Text elements: Version 00004 Language E
Local system: I22
Request: I22K903455 06.01.2023 DE6U3189
Text:

T

SAP I22 (1) 001

→ When we compare target with Scope TR in EN lang, we get difference in Text Symbol and Selection Text;

→ Now we have to compare Scope TR with below versions to reconfirm if its a Source change(if we don't find proof for scope TR change, We have to check Target versions- As This can be a target change where customer have removed/changed the translations for them in Target)

Compare ABAP text elements in program Y03CPO_SICHT_DURCHSATZ

Text elements: Version 00004 Language E
Local system: I22 T
Request: I22K903455 06.01.2023 DE6U3189
Text:

Text elements: Version 00004 Language E
Remote system: Q02 S
Request: I02K9P7987 29.12.2022 US0WAS
Text:

Compare selection texts
Different from the remote system version are:

ID	Key	Length	DDIC	Text
S	P_RUECK	38		Single Confirmations
S	P_RUECK	38		Single Conf. (user time zone)

T

SAP I22 (1) 001 vi22pas OVR

scroll down to find other differences for the object

Comparing scope TR with below versions, to re-validate if the difference we've identified is a scope TR/Source side change.

Versions Edit Goto System Help

Versions of Object Y03CPO_SICHT_DURCHSATZ of Type Report Texts

Retrieve Fetch Remote Versions Search Documentation

Status	Version	Compare	Cat.	Fl	SA	Rel.	Arc	Request	Request Text	Project	Date
	active					750					
	5		I			750		I02K9P7BRF	S 8200026485: 0430 : Y Transaction Migration to NGE (S	I02_P...	28.02.2023
	4		I			750		I02K9P7987	S 8200025665: YCFR time	I02_P...	29.12.2022
	3		I			750		I02K9P73NF	S 8200023815: YCFR add personnel number to ALV grid	I02_P...	23.09.2022
	2		I			750		I02K9P41BU	CB_3446002 ; YCFR Translation EN		24.09.2020
	1		S			750		I03K940266	Programme WAS aus Paket Y03CPO		25.02.2008

Use toolbar Search button to search for changes within versions

S

Q02 (1) 001 vq02a00 OVR

Compare Versions: Text Elements

Compare ABAP text elements in program Y03CPO_SICHT_DURCHSATZ

Text elements: Version 00004 Language E
Request: I02K9P7987 29.12.2022 US0WAS
Text: S 8200025665: YCFR time

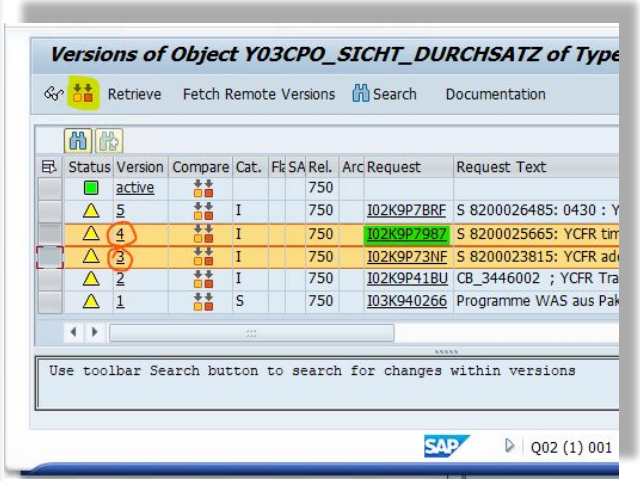
Text elements: Version 00003 Language E
Request: I02K9P73NF 23.09.2022 US0WAS
Text: S 8200023815: YCFR add personnel number to ALV grid

Compare selection texts
were changed:

ID	Key	Length	DDIC	Text
S	P_RUECK	38		Single Conf. (user time zone) V4
S	P_RUECK	38		Single Confirmations

S

Q02 (1) 00



On comparing source version 4(scope TR) to version 3, it's clear that Selection Text is a source side change.

Now lets go one more version below and compare version 4 and version 2, to check if we'll get Text symbol difference als as a source side change.

Text elements Edit Goto System Help

Compare Versions: Text Elements

Compare ABAP text elements in program Y03CPO_SICHT_DURCHSATZ

Text elements: Version 00004 Language E
Request: I02K9P7987 29.12.2022 US0WAS
Text: S 8200025665: YCFR time

Text elements: Version 00002 Language E
Request: I02K9P41BU 24.09.2020 WAS
Text: CB_3446002 ; YCFR Translation EN

Compare text symbols
Inserted:

ID	Key	Length	Text
I	ZZ1	76	You don't have permission for Plant:

Compare ABAP text elements in program Y03CPO_SICHT_DURCHSATZ

SAP Q02 (1) 001 vq02a00 OVR

Compare Versions: Text Elements

Compare ABAP text elements in program Y03CPO_SICHT_DURCHSATZ

Text elements: Version 00004 Language E
Request: I02K9P7987 29.12.2022 US0WAS
Text: S 8200025665: YCFR time

Text elements: Version 00002 Language E
Request: I02K9P41BU 24.09.2020 WAS
Text: CB_3446002 ; YCFR Translation EN

Compare selection texts
were changed:

ID	Key	Length	DDIC	Text
S	P_RUECK	38		Single Conf. (user time zone)
S	P_RUECK	38		Single Confirmations

SAP Q02 (1) 001 vq02a00 OVR

On comparing source version 4(scope TR) to version 2, it's clear that both Selection Text and Text symbol are source side changes. → So we have to bring them to target.

I22		active				756	I22K907482	S 8200027312: CM Retrofit#4 handover 1
		4				756	I22K903455	S 8200025092: CM Code Retrofit #3 all
		3				756	I22K900755	S 8200021738: change origin system I22 after carveout 2.2
		2				756	I22K901151	ToC NGE: consolidated transport for C22 preparation (0 Test)
		1		S		756		

```
Use toolbar Search button to search for changes with
```

Text elements Edit Goto System Help

Compare Versions: Text Elements

```
Compare ABAP text elements in program Y03CPO_SICHT_DURCHSATZ

Text elements:  Version 00004 Language E
Request:        I22K903455            06.01.2023 DE6U3189
Text:           S 8200025092: CM Code Retrofit #3 all

Text elements:  Version 00001 Language E
Request:                              19.07.2022 DE6U3189
Text:

            No differences in text elements found
                    between these versions.
```

SAP I22 (1) 001 vi22pas OVR

Just to reconfirm, we are not gonna overwrite any Target changes, We have to do a version comparison with Target below versions.

- ➢ There is NO difference between Active target version and Target version 4; Active version just has the object locked in our working Retrofit TR.
- ➢ Target versions 4 to 1 on comparison shows No difference/Identical in EN language.

⇒ Meaning, there are customer target changes as such, so we can go ahead and bring the scope TR changes for EN language.

Text Elements Edit Goto Utilities Environment System Help

ABAP Text Elements: Display Selection Texts Language English

Program Y03CPO_SICHT_DURCHSATZ Active

Text Symbols | Selection Texts | List Headings

1 / 18

Name	Text	DDIC Refe...
P_DAY	Day View	☐
P_MONTH	Month View	☐
P_RUECK	Single Confirmations	☐
P_WEEK	Week View	☐
P_WEMNG	Consider delivered	
S_ARBPL	Work Center	
S_AUART	Order Type	
S_AUFNR	Order	
S_BUPER	Month	
S_DISPO	MRP Controller	

Different original and logon languages

The logon language does not match the original language.

Logon language EN English

Original language DE German

Maint. in orig. lang. | Change orig.language | Cancel

I22 (4) 001 vi22p

{1} When we try to enter edit mode for REPT in Non-original language, we may get the highlighted message.

Textelemente Bearbeiten Springen Hilfsmittel Umfeld Syste

Objektkatalogeintrag

Dokumentation

Übersetzung

Online-Handbuch Strg+F8

Zurück F3

ABAP Textelemente:

Programm Y03CPO_SICHT_DURCHSATZ aktiv

Textsymbole | Selektionstexte | Listüberschriften

{2} In such case, we have to enter Translation mode from DE to EN, and maintain the changes there.

To enter translation mode, select: Menu → Goto → Translations and

{3} give DE to EN.

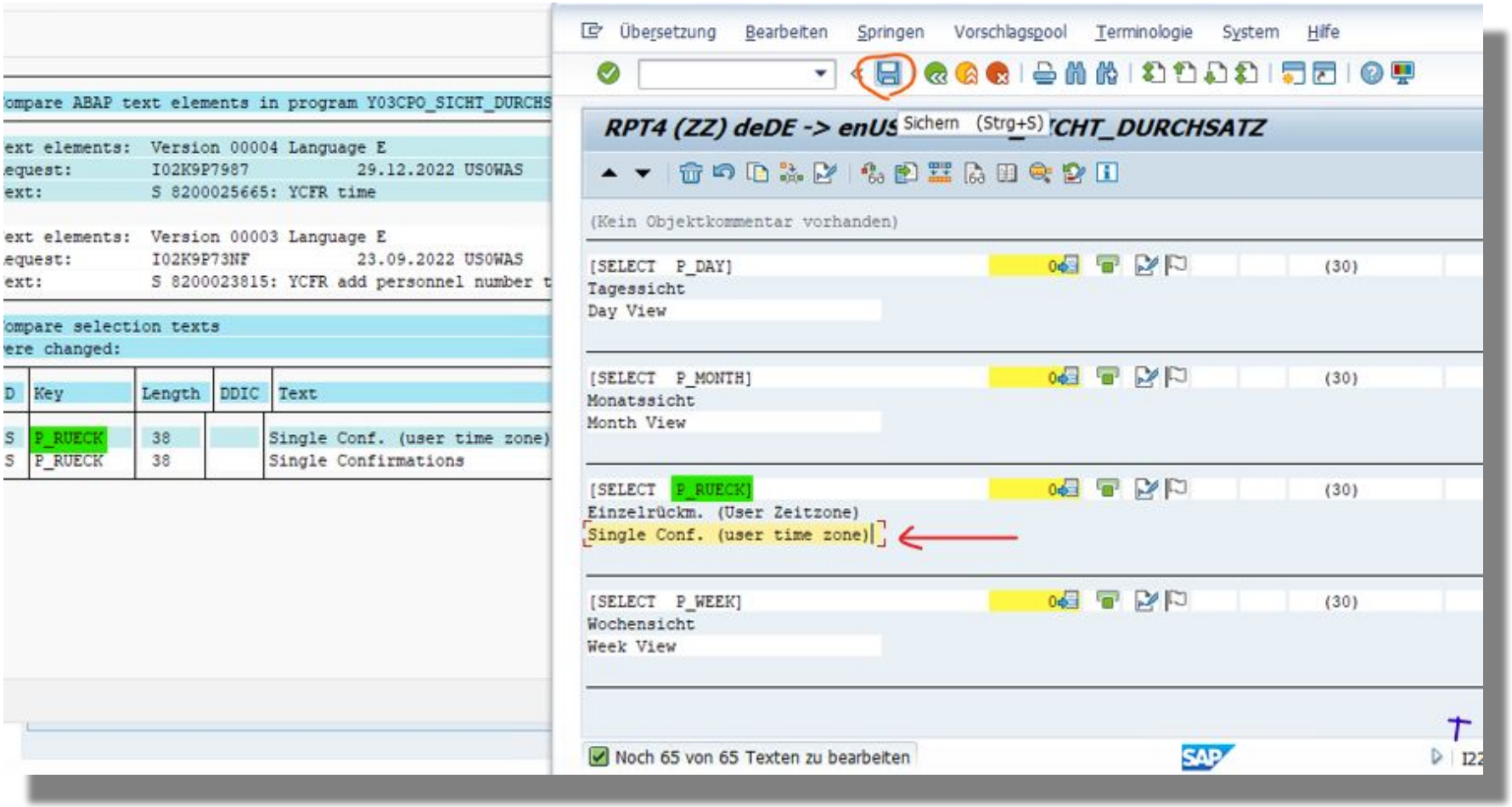
Zielsprache für die Übersetzung

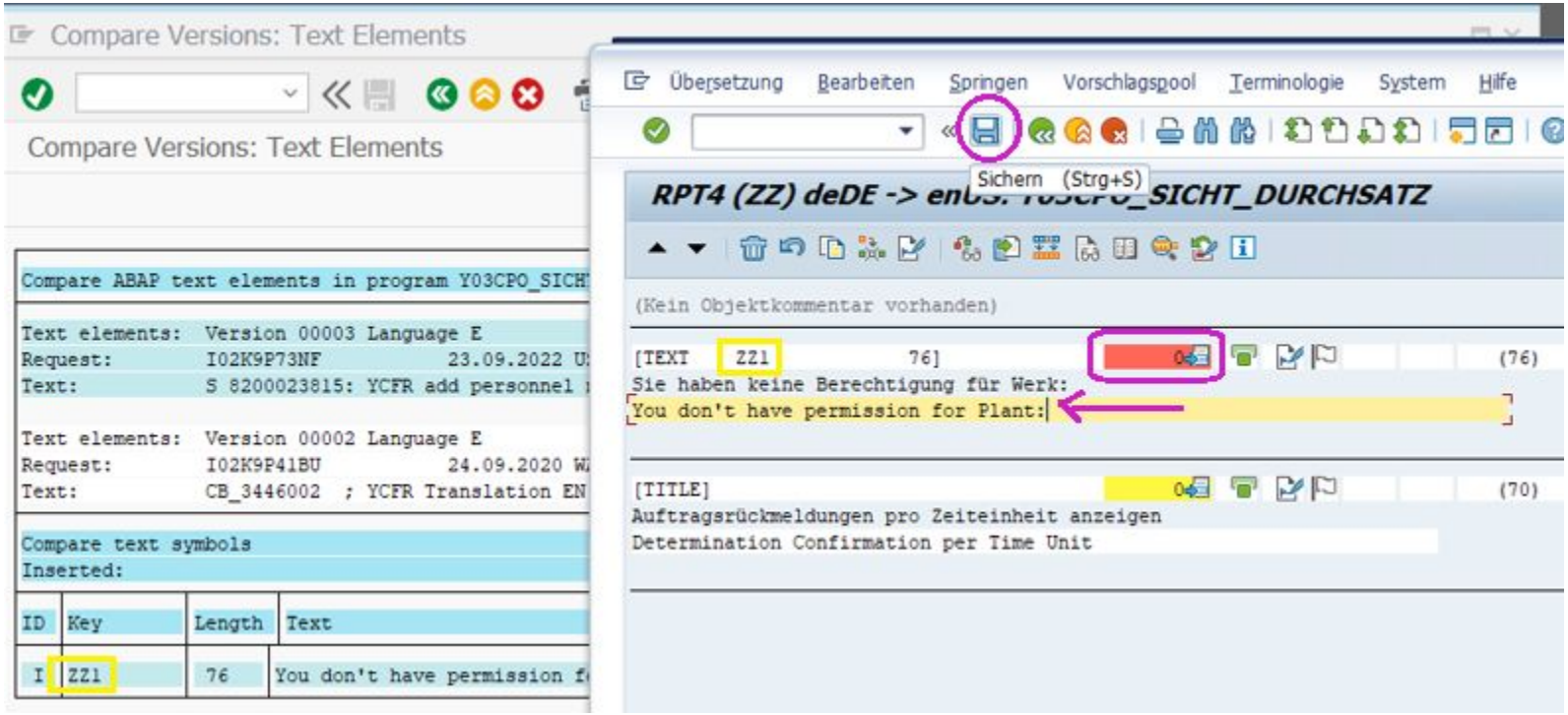
Originalsprache DE Deutsch

Zielsprache EN

en

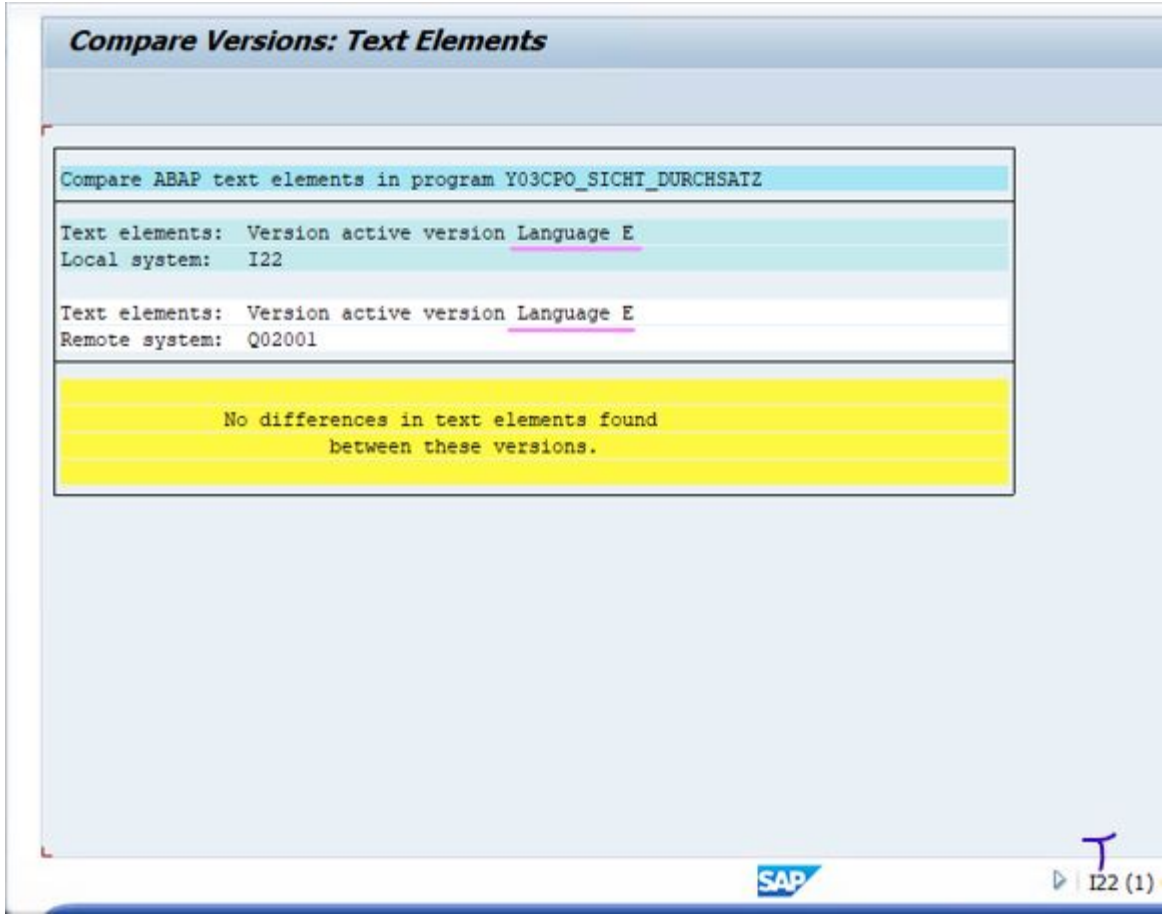
{4} maintain the Translations and Save:





Only once we enter the text and save, the processing color will change from red to yellow

Maintain the translation texts, save and then go back at the REPT level and Activate the Object.



After Activating at the REPT level, let's do a remote comparison to scope TR to re-validate the changes are made correctly.

For the current object, it had only source changes and,
⇒ After remote comparison, there are No more differences, implying we did the retrofit right.

status and comments for secondary lang EN will be:

Retrofit Done - Maintained translations in EN language as per scope TR.

Step (4) Update combined status and comments in Scoping sheet

When you update the final status and comments in scoping sheet, see that you are not using terms like primary and secondary language there, as those were just meant for easy explanation purpose.

status and comments for Original lang DE:

No Action - REPT looks identical in original language DE.

+

status and comments for secondary lang EN:

Retrofit Done - Maintained translations in EN language as per scope TR.

→

Combined status and comments for the object:

Retrofit Done - As per scope TR, Maintained translations for the REPT in EN language in target and REPT looks identical in original language DE.

DEVC

- DEVC -Packages
  - Format in scoping sheet: R3TR-DEVC-PackageName
  - As part of Retrofit, we consider DEVC under Service scope
  - TCode : se80, se21
  - We have to analyze tab by tab to understand the difference

- Can DEVC be considered as **part of Safe To Import** analysis/ToC?
  - Yes - For NEW, IN scope Objects.
  - Not Required for Existing , IN scope Objects.- we just have to compare tab by tab (usually we observe changes in Properties tab : short desc, super package, transportation layer, check box || Changes in sub-packages, Package hierarchy, and Transportation layer can be ignored if those are target specific – to confirm you can either check in Version history or SE03)

- Common error/issue related to package:
  - We may see message like '*Only edit objects from package PackageName in local requests*' and when we get this kind of error, we may not be able to Retrofit(Modify and Save) or lock the package or objects belonging to this package under our Retrofit working TR.
    ⇒ And the solution for this case would be, we have to inform Customer and they have to update the correct Transportation layer for the package in target.
    [Note: For the same package, Transportation layer maynot be same in source and Target.]
    This issue can occur when Transportation Layer is BLANK or INCORRECTLY MAINTAINED.

Package Edit Goto Utilities Environment System Help

Display Package

Package Check | Package Check Including Subpackages

Repository Browser

Package

/PRI/OBSOLETE

Object Name	D
/PRI/OBSOLETE	obs
Dictionary Objects	
Data Elements	
/PRI/E_BO_NODE_SCHEME_ITEM_ID	Bus

Package /PRI/OBSOLETE

Properties | Use Accesses | Package Interfaces

Basic Data

Short Description	obsolete objects
Application Component	CA
Person Responsible	OSAPIENS
Created by	PRIMINGCLOUD
Created On	01/02/2020
Last Changed By	PRIMINGCLOUD
Changed on	02/19/2021

Package Properties

Superpackage /PRI/CONNECTOR

☐ Main Package

☐ Adding further objects not possible

Transport Attributes

Transport Layer

Software Component HOME Release DEV

☑ Record Object Changes in Transport Requests

SAP DNR (2) 100 aldecc001 INS

Change request/task DNRK976195

TRANSLATION FOR: VIEW DE...	LANG	VIED	/PRI/V_DIMMAPFLD
TRANSLATION FOR: VIEW DE...	LANG	VIED	/PRI/V_DIMMAPHD
TRANSLATION FOR: VIEW DE...	LANG	VIED	/PRI/V_INTFIMPL
TRANSLATION FOR: VIEW DE...	LANG	VIED	/PRI/V_PARAM
TRANSLATION FOR: VIEW DE...	LANG	VIED	/PRI/V_QU
TRANSLATION FOR: VIEW DE...	LANG	VIED	/PRI/V_QU_DT
TRANSLATION FOR: VIEW DE...	LANG	VIED	/PRI/V_QU_SY
Data Element	R3TR	DTEL	/PRI/E_BO_NODE_SCHEME_ITEM_ID

Only edit objects from package /PRI/OBSOLETE in local requests SAP

Common error/issue related to packages

⇒Only once the Transportation layer is fixed, we can change or save the dependent object in working TR.

Until then the dependent object will be on Hold

[Note: For the same package, Transportation layer maynot be same in source and Target.]

Request/Task Edit Goto Utilities Environment System Help

Change request/task DNRK976195

Request/Task DNRK976195 Workbench Request

Properties | Objects | Documentation | Status

Short Description	Pr...	Ob...	Object Name
TRANSLATION FOR: VIEW DE...	LANG	VIED	/PRI/V_QU
TRANSLATION FOR: VIEW DE...	LANG	VIED	/PRI/V_QU_DT
TRANSLATION FOR: VIEW DE...	LANG	VIED	/PRI/V_QU_SY
Data Element	R3TR	DTEL	/PRI/E_BO_NODE_SCHEME_ITEM_ID

Only edit objects from package /PRI/OBSOLETE in local requests

Only edit objects from package /PRI/OBSOLETE in local requests

Message No. TK332

Diagnosis

Objects in package /PRI/OBSOLETE cannot be edited in transportable Workbench requests in the current system DNR. The transport routes are configured such that objects from package /PRI/OBSOLETE can only be edited in local Workbench requests.

System Response

The function terminates.

Procedure

You have the following options:

- If you want to transport objects in package /PRI/OBSOLETE into the target QVR once only, without changing the configuration of the transport routes, then create a transport of copies in the extended view of the Transport Organizer (Transaction SE01). Go to the request overview and choose *Request/task -> Object list -> Include objects* to include your objects in this transport of copies.
- If you would like objects in package /PRI/OBSOLETE to be transported generally into the target system QVR, contact your transport administrator. In this case, the configuration of the transport routes must be corrected in the Transport Management System (TMS). For details, see the documentation on the Transport Management System.

Now we have reported this to Customer asking them to Fix the transportation layer

Update - After few days Customer has replied back saying they have fixed the Transportation layer

Package Edit Goto Utilities Environment System Help

Display Package

Package Check | Package Check Including Subpackages

Package: /PRI/OBSOLETE — Saved

Properties | Use Accesses | Package Interfaces | Subpackages | Package Hierarchy

Basic Data

Short Description	obsolete objects
Application Component	CA — Cross-Application Components
Person Responsible	OSAPIENS
Created by	PRIMINGCLOUD
Created On	01/02/2020
Last Changed By	CAPBASIS
Changed on	04/20/2023

Package Properties

Superpackage	/PRI/CONNECTOR — Change

☐ Main Package
☐ Adding further objects not possible

Transport Attributes

Transport Layer	SAP — DNR to QVR
Software Component	HOME — Release DEV

SAP | DNR (1) 100 | aldecc001 | INS

Request/Task Edit Goto Utilities Environment System Help

Display request/task DNRK976195

Request/Task: DNRK976195 — Workbench Request

Properties | Objects | Documentation | Status

2,683 / 2,688

Short Description	Pr...	Ob...	Object Name	Fu...	Lock/Import
Translation for: View De...	LANG	VIED	/PRI/V_QU		
Translation for: View De...	LANG	VIED	/PRI/V_QU_DT		
Translation for: View De...	LANG	VIED	/PRI/V_QU_SY		
Data Element	R3TR	DTEL	/PRI/E_BO_NODE_SCHEME_ITEM_ID		Object l
Comment Entry: Released	CORR	RELE	DNRK976365 20230420 200529 A41746		
Report Source Code	LIMU	REPS	ZRV60AFZZ04_GLOBAL		Object l

SAP | DNR (1) 100 | aldecc001 | INS

⇒As the Transportation layer is fixed, we were able to save/lock the dependent IN scope object in working TR.

Now the team can Retrofit the dependent object and update status&comments accordingly.

➔ **Retrofit Checks for Existing DEVC objects**:

Open the package in Source and Target (Tcode: se80 or se21) , Then we have to do a tab by tab comparison to identify the difference between source and target; If you find any difference, you have to further analyze if its a source change or a target change by validating in se03.

When you compare, please ignore the difference in ‘Transportation Layer’ and ‘Software Component’ between source and target; As those will be target specific changes.
All tabs are important, but mainly we have to focus on: Properties, Use Access and Package Interface.
Sub-Package and Package Hierarchy tab will be automatically updated based on where all current package is used and superpackage details.

TABU

➔ About TABU

- Table Contents
- Format in scoping sheet: R3TR-TABU-ObjectName
- For Retrofit TABU is taken care as part of Service scope
- As per new SCOPE categorization, Not all TABUs are IN our(smartShift) scope.
- IN scope TABUs include TDDAT,TVDIR,TVIMF and Custom table entries.

➔ Can TABU be considered as part of NEW/Safe To Import analysis/ToC?

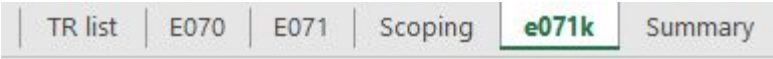
- Yes - but do the analysis and TABU addition to ToC carefully as its a sensitive topic.
  Note : We have to get the ToC reviewed from critical team, when we add TABU as part of ToC.
- We have to compare the Table contents at system level, For the IN scope TR, We have to open the TABLE in source and Target and Compare the Key-Value combination.
  - **NEW keys** → NEW entries can be brought to target
    If the Key combination is new, then the current entry we can consider as new; So we can add this to safe to import ToC.
  - **Existing keys → with same values**
    For the existing entries where both Key-Value combination is same in source and Target, We don't have to take any action(As these are already available, there is no reason to bring the same again)
  - **Existing keys → with different values** → This is a Conflict case
    For the existing entries where, for same Key combination, Values are different in source and Target, We can't blindly take a decision that the difference in value is a source/target change - These conflicting TABU contents should be reported to customer; and based on Customer feedback we can keep them as a 'No Action case - Target has the latest change' OR 'Add to ToC-if confirmed as a source change'.

➔ How to perform Safe to import analysis for TABU:

steps:

Note: As per new SCOPE categorization, Not all TABUs are IN our(smartShift) scope.
IN scope TABUs include TDDAT,TVDIR,TVIMF and Custom table entries.
So our first step will to find these are the entries in our IN scope TABU TR.

1. From the scoping sheet, take all the scope TRs which belong to scope=IN.
2. Give the above taken TR list to E071K table in source system and export(copy) the contents to a new tab in scoping sheet. You can name the new tab as E071K.

   (note that you are clearing the limit for maximum hit in table before executing,
   otherwise we may not have complete TABU details)
3. Now add 2 columns ‘Status’ and ‘Comments’ towards the end in E071K tab.
4. We have to analyze each row in e071k tab and update the status and comments.
   Possible status can be:
   - OUT – In case TABU objects are Other than TDDAT,TVDIR,TVIMF and Custom table entries.
   - ToC – In case of NEW or Safe to import TABU content
   - No Action – In case same key-value combination already exists in both source and target
   - Customer Feedback/Customer Review – In case of conflicting key-value combination

steps cont:

5. Understand the important E071K fields
   - TRKORR - IN Scope TABU TR
   - OBJNAME - TABU object name where we can check, if the Name space is IN our scope or not
   - TABKEY - Table keys
   - Status & Comments

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	TRKORR	PGMID	OBJECT	OBJNAME	AS	MASTE	MASTERNAME	V	OI	TABKEY	SO	FL	LAI	AC	Status	Comments
2	MNRK965506	R3TR	TABU	/SCL/TVARV	1	TABU	/SCL/TVARV			/SCL/OTC_MANUAL_COND_CH	2					
3	MNRK965506	R3TR	TABU	/SCL/TVARV	2	TABU	/SCL/TVARV			/SCL/OTC_MANUAL_COND_CH	2					
4	MNRK965506	R3TR	TABU	/SCL/TVARV	3	TABU	/SCL/TVARV			/SCL/OTC_MANUAL_COND_CH	2					
5	MNRK965796	R3TR	TABU	TDDAT	1	TABU	TDDAT			/SCL/TOTCM_PMA	2					
6	MNRK965796	R3TR	TABU	TVDIR	1	TABU	TVDIR			/SCL/TOTCM_PMA	2					
7	MNRK967249	R3TR	TABU	T682	1	CDAT	V_T682	V_T		AV Z3PL	2					
8	MNRK967249	R3TR	TABU	T682T	1	CDAT	V_T682	V_T		EAV Z3PL	2					
9	MNRK973464	R3TR	TABU	/SCL/TVARV	1	TABU	/SCL/TVARV			ZXVVCU01 PRO	2					
10	MNRK973464	R3TR	TABU	/SCL/TVARV	2	TABU	/SCL/TVARV			ZXVVCU01 TCO	2					
11	MNRK973464	R3TR	TABU	/SCL/TVARV	3	TABU	/SCL/TVARV			ZXVVCU01 TCO	2					

6. Analyze the content and Update Status and Comments

   6.1. Mark the OUT scope objects, we don't have to analyze them
   Check column OBJNAME for TABU objects
   >>Other than TDDAT,TVDIR,TVIMF and Custom table entries → Mark these as status OUT.
   [for custom table entries check IN scope namespaces, and go to step 6.4]
   >>For TDDAT,TVDIR,TVIMF and Custom table entries → we have to analyze further(next step)

   6.2. For IN scope TDDAT & TVDIR -check if Table exist in target or not
   You can get the table name from TABKEY column OR open the corresponding table in se09, check the key maintained for TDADAT, TVDIR.

Now check if the table is available in target system.
If the table is available in target system → Mark the status as No Action.
And if table is NOT available in target system → Mark the status as ToC.

1.3. For IN scope TVIMF - [TVIMF gets created when a TMG Event for a table is changed(created/deleted/updated)]
Open the table in both source and target and check the TMG events screen
If the Event screen is matching in source & target → Mark the status as No Action.
Otherwise → Mark the status as ToC.

1.4. For Custom table entries(OBJNAME will contain a custom Name space),
We should only work on Namespaces which are IN our scope,
Other custom Namespaces you can mark as OUT.

Note for step 6.4-For new projects..for Custom table entries-we directly mark them as 'Customer Review' with comments 'As its business related entries, require customer confirmation to import via ToC to target.'

For the IN scope namespace one,
Open the scope TR in source system[TCode-se09] → Filter on the current TABU entry and double click on the key.
You will see a 'Key combination' for a list of newly inserted/changed/deleted entry here.
Now open the custom table in both source & target and provide the 'Key combination' you see in se09 key sub-screen one by one and compare the values you see in source with target.

- If the 'Key combination' only exist in source and doesn't exist in Target
  ⇒ means its a NEW entry→ Mark the status as ToC.
- If the 'Key combination' doesn't exist in both Source and Target
  ⇒ means its a Deleted entry in source side, but as target doesn't have this entry yet we don't have to take any action → Mark the status as No Action.
- If the 'Key combination' doesn't exist in Source, but Exist in Target
  ⇒ means its a Deleted entry in source side as part of scope TR, scope TR change so we have to bring to target therefore → Mark the status as ToC.

- If the ‘Key combination’ exist in both Source and Target - We have to analyze further
  Check/Compare the values for the ‘Key combination’.
  - ➢ If the values are matching/same in both Source & Target for the current key
    ⇒ we don't have to take any action, as the same entry is already present in target side
    → thus we can Mark the status as No Action.
  - ➢ If the values are **NOT** matching/same in Source & Target for the current key
    ⇒ It's clearly a CONFLICT case : we have to take customer Feedback for this scenario,
    as we cant blindly take decision in this case
    → thus we can Mark the status as Customer Review.

Example - Our TR is DEVxxx98 and custom Table zemployee and table keys are (EmpId,DOJ)
Once you double click on the key in se09 after opening TR DEVxxx98 and filtering on zemployee object,
You will see a list of newly inserted/changed/deleted employee id and Date of Joining Key here.
Now for each key [EmpId,DOJ] compare the values in Table zemployee in both source and target system.
>For new employees → we can add to ToC and bring to target
>For existing employees, if value is changed(say for example salary field got changed) → customer has to review this case and tell us which side has latest changes. If customer updates us that,
>>Target has latest changes – then we don't have to bring source changes as it may overwrite target changes.
→ thus we can update the status from ‘Customer Review’ to ‘No Action’.
>>Source has latest changes – then we have to bring source changes to target changes.
→ thus we can update the status from ‘Customer Review’ to ‘ToC’.

1.5. For Customer Review cases, get the feedback from Customer and accordingly mark the status as No Action or ToC.
1.6. Follow the next step ‘How to add TABU contents/objects to ToC’ to add the TABUs marked as ToC status to ToC TR.

➔ How to add TABU contents/objects to ToC:

steps:

1. Open the ToC TR in se09. double click and enter edit mode of TR
2. Use include TR option and add the 1st 'IN scope-Need to add to ToC' TR based on our analysis in the previous step (How to perform Safe to import analysis for TABU)
3. Once the IN scope TR is included to ToC, this will have all the objects of current TR added to ToC - but we just want the TABU entry. In this case you can select and use remove icon emove unwanted entries.
4. Now click on the key symbol for TABU and validate the KEYS.
   You can double click on each row and check. If any key combination is not required, then you can select and delete unwanted entries alone using remove icon.
5. Once the 1st TR TABUs are included, Save the ToC TR so that our changes are not lost.
6. Now go for the next IN scope TABU TR for second step and repeat steps 3 to 5 until you add all required TABUs.

Note : Never add TABUs manually just by entering edit mode for ToC, Please make you of Include TR option and follow above steps as there is a chance of below mistake.

Common error/issue related to TABU addition to ToC TR : Missing keys/ Having SPACE or '*' in kep position - This may lead to complete overwrite or wipe out of Target Table contents and Impact customer data/testing.

PROG

- About PROG
  - Programs ; TCode - SE38
  - Format in scoping sheet: R3TR-PROG-ProgramName
  - As PROG is an R3TR entry, it means all sub-objects of the PROG should also be considered for Retrofit.
    Sub-Objects of PROG includes:
    REPS,REPT, DYNP,CUAD,DOCU
  - For Retrofit, PROG is taken care as part of Tool scope- where we have to focus on REPS subobjects of PROG, Other sub-objects REPT,DYNP,CUAD,DOCU will be taken care as part of Service scope.

- Can PROG be considered as **part of NEW/Safe To Import** analysis/ToC?
  - Yes - for NEW,IN scope Objects; After ToC import-Tool rules have to be applied to all its sub-objects.
  - No - for For EXISTING, IN scope Objects.

➔ Procedure to check PROG

Case 1: New Object

- If the PROG is new to the target system, it can be directly imported from the dashboard in online retrofits or included in the ToC import in offline retrofits. Depending on how each project is conducted.
- If the scope TR contains NEW R3TR PROG, the entire object—including Textelements(REPT), sub-Includes(REPS) like TOP/F01/I01/O01 include, DYNP, CUAD—will be imported to the target during import.
- If the scope TR has an entry of sub objects of a program like LIMU REPS and this is NEW, then only that specific Include will be imported into the target.
- Post import, source_parsed version of the code(with relevant tool rules) needs to be added into the target if the object has got any tool changes.

Case 2: Existing Object

- If the PROG object is already existed in target, then we need to retrofit it using the tool report/result containing source_parsed, merged and target code either in MELD application(Offline retrofit) or WEBUI(Online retrofit).
- If scope TR contains LIMU objects, then we need to retrofit only that specific object.
  If scope TR contains R3TR PROG object, then we need to retrofit all the subobjects of that Program.

Example (On how to apply Filter in scoping sheet): R3TR-PROG-/SCL/RDLETRAR_DRIVER_SW

scope= ‘IN’, MAIN_OBJECT= ‘/SCL/RDLETRAR_DRIVER_SW’

	A	B	C	D	E	F	I	L	M	N	O	P	Q	R	S	T	U	V	W
1	TRANSPORT	LI	PRO	OB	OBJ_NAME	C	L	B	OBJTYI	MAIN_OBJECT				I	Tool /	Object	Duplic		Scope
56	MNRK973711	3	R3TR	PROG	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SC	ALREADY	Final		IN
60	MNRK974467	4	LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SC	ALREADY	Duplicate		IN
61	MNRK974467	5	LIMU	REPT	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					SERVICE_	ALREADY	Final		IN
72	MNRK974913	3	LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SC	ALREADY	Duplicate		IN
187	MNRK976550	4	LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SC	ALREADY	Duplicate		IN
929	MNRK977935	3	LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SC	ALREADY	Final		IN
4358	MNRK973711		LIMU	DYNP	/SCL/RDLETRAR_DRIVER_SW	1000			PROG	/SCL/RDLETRAR_DRIVER_SW					SERVICE_	ALREADY			IN
4359	MNRK973711		LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SC	ALREADY			IN
4360	MNRK973711		LIMU	REPT	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					SERVICE_	ALREADY			IN

Now remove Duplicate,

	A	B	C	D	E	F	I	L	M	N	O	P	Q	R	S	T	U	V	W
1	TRANSPORT	LI	PRO	OB	OBJ_NAME	C	L	B	OBJTYI	MAIN_OBJECT				I	Tool /	Object	Duplic		Scope
56	MNRK973711	3	R3TR	PROG	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SC	ALREADY	Final		IN
60	MNRK974467	4	LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SC	ALREADY	Duplicate		IN
61	MNRK974467	5	LIMU	REPT	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					SERVICE_	ALREADY	Final		IN
72	MNRK974913	3	LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SC	ALREADY	Duplicate		IN
187	MNRK976550	4	LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SC	ALREADY	Duplicate		IN
929	MNRK977935	3	LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SC	ALREADY	Final		IN
4358	MNRK973711		LIMU	DYNP	/SCL/RDLETRAR_DRIVER_SW	1000			PROG	/SCL/RDLETRAR_DRIVER_SW					SERVICE_	ALREADY			IN
4359	MNRK973711		LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SC	ALREADY			IN
4360	MNRK973711		LIMU	REPT	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					SERVICE_	ALREADY			IN

	A	B	C	D	E	F	I	L	M	N	O	P	Q	R	S	T	U	V	W
1	TRANSPORT	L	PRC	OB	OBJ_NAME	C	L	B	OBJ	MAIN_OBJECT					Tool / Service	Object status(N	Dup		Sco
56	MNRK973711		R3TR	PROG	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SCOPE	ALREADY_EXISTED	Final		IN
61	MNRK974467		LIMU	REPT	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					SERVICE_SCOPE	ALREADY_EXISTED	Final		IN
929	MNRK977935		LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SCOPE	ALREADY_EXISTED	Final		IN
4358	MNRK973711		LIMU	DYNP	/SCL/RDLETRAR_DRIVER_SW 1000				PROG	/SCL/RDLETRAR_DRIVER_SW					SERVICE_SCOPE	ALREADY_EXISTED			IN
4359	MNRK973711		LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SCOPE	ALREADY_EXISTED			IN
4360	MNRK973711		LIMU	REPT	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					SERVICE_SCOPE	ALREADY_EXISTED			IN

Service scope sub-objects might have already been taken care under ‘Service_scope’, If not-we have to perform the retrofit for Service scope sub-objects as well - As we’re working on R3TR entry.
Now focus on Tool scope sub-object and perform the Retrofit.

Note: If you have both ‘FINAL’/’BLANK’ entry for same object-we have to focus on the latest scope TR and perform Retrofit; and update the status and comment for the other one as Retrofit Done/No Action-As per the latest scope TR.
You can refer version history/se03/E071-E070 to get the latest version.

	A	B	C	D	E	F	I	L	M	N	O	P	Q	R	S	T	U	V	W
1	TRANSPORT	L	PRC	OB	OBJ_NAME	C	L	B	OBJ	MAIN_OBJECT					Tool / Service	Object status(N	Dup		Sco
56	MNRK973711	3	R3TR	PROG	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SCOPE	ALREADY_EXISTED	Final		IN
929	MNRK977935	3	LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SCOPE	ALREADY_EXISTED	Final		IN
4359	MNRK973711		LIMU	REPS	/SCL/RDLETRAR_DRIVER_SW				PROG	/SCL/RDLETRAR_DRIVER_SW					TOOL_SCOPE	ALREADY_EXISTED			IN

To perform Retrofit for any tool scope object,
make use of Tool results(source_parsed,merged and target file) and Source/Target version history,
and decide on how Retrofitted code should look like.
We should bring the latest scope TR changes, retaining target changes.

- To get all subobjects of an R3TR-PROGentry:

  - Execute RSSYSCOMP program with R3TR-PROG-ObjectName, We can get all the sub-objects of the current main object PROG.
  - Also in scoping sheet, you can filter as below, and update status and comments for Retrofit action taken.
    scope=IN,Duplicate_Transport=Remove duplicates, MAIN_OBJECT=currnentPROGobjectName
    ⇒ Now in PROGRAM_ID,OBJECT_TYPE,OBJ_NAME we can see the sub-objects(LIMUs) of the R3TR-PROG entry.

    Note: Scope TR(column TRANSPORT) might have different scope TRs for few objects; As we are removing Duplicates - Final/Latest scope TR will be there in the Filtered list in scope sheet ⇒ See that in such scenarios Retrofit is done for the latest scope TR.
    Therefore, It's always better to first complete Retrofit for R3TR entries when we start Tool scope, so that dependent sub-objects with different scope Tr is already Taken care, and later we can go for independent LIMUs in our scope.

- All the objects displayed in the output screen of rssyscomp report should be fixed as part of R3TR PROG retrofit.
  Making use of Tool results(source_parsed,merged and target file) and Source/Target version history, we can take a decision on how Retrofitted code should look like.
- Difference in how we handle Online & Offline PROG objects—-pending—-

REPS

- **About REPS**
  - REPS denotes Includes
  - REPS Can be a sub-object of R3TR entry PROG, FUG or CLAS
  - Format in scoping sheet: LIMU-REPS-IncludeName
  - For Retrofit, REPS is taken care as part of Tool scope.

- **Can PROG be considered as part of NEW/Safe To Import analysis/ToC?**
  - Yes - for NEW,IN scope Objects; After import-Tool rules have to be applied them(if any).
  - No - for For EXISTING, IN scope Objects.

- **Points to note:**
  - Never fix(Retrofit) Generated objects manually.
    In first few lines of code you can find 'Generated' wording.
    Example - UXX includes, TMG related includes/FMs like VIEWPROC, TABLEPROC (covered under FUGR)

- **How to Retrofit REPS objects**
  - Make use of Tool results(source_parsed,merged and target file) and MELD app in case of Manual/Offline Retrofit and WEBUI in case of Online Retrofit.
  - Now build/prepare merge file, Analyzing source latest scope TR change and Target changes comparing with below versions.
    Once the merge file is ready with the Retrofitted code(having scope TR change, retaining target changes if any)
    We can update the Target side with Retrofitted code and maintain Status and comments in scoping sheet based on the Action taken.

➔ Possible Status/Comments for REPS:

Status	Comments {Senario}
No Action	Identical in source and target {No Source/Target changes and No Rule changes}
	Target has the latest changes {Target has the latest changes}
	TMG related table was not changed as part of current Retrofit.
Retrofit Done	Imported via ToC and updated tool rules. {for NEW objects}
	Added latest changes to target. {only Source changes}
	Added scope TR changes to target, retaining target changes. {both Source & Target has changes}
	Regenerated TMG for table tableName {for TMG related includes like VIEWPROC, TABLEPROC etc..}
	Repaired FUGR FunctionGroupName {for UXX include}
Customer Action	{when Retrofit is performed, However the object may be in Error due to dependent object which is NOT IN our(smartShift) scope }

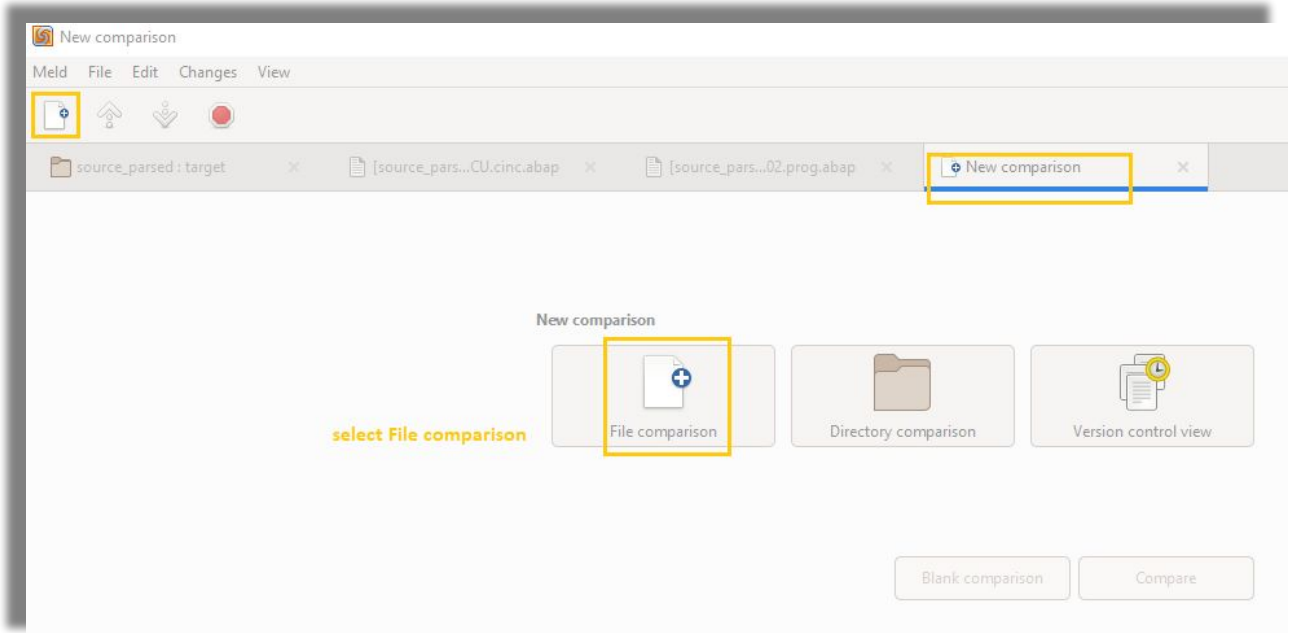
Example (On how to Retrofit REPS/Offline Retrofit): **LIMU-REPS-ZXVKPU02**

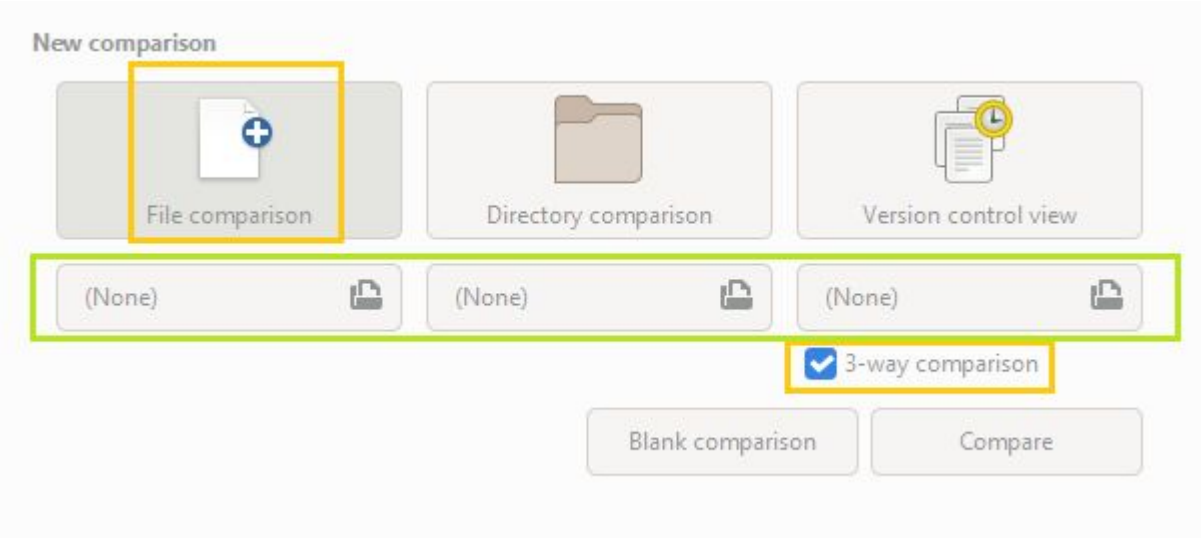
Source- CRW
Target- DR6
Working TR- DR6K907200

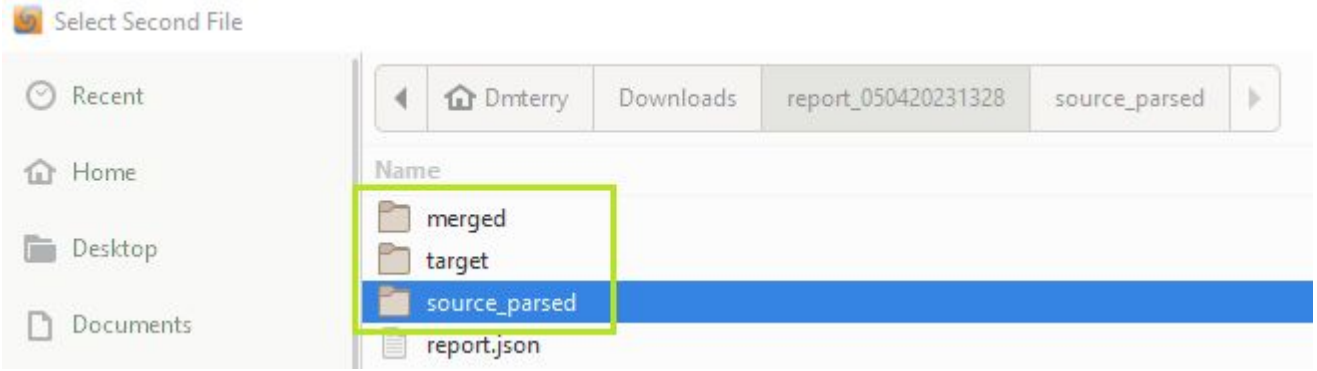
Scoping sheet snippet :

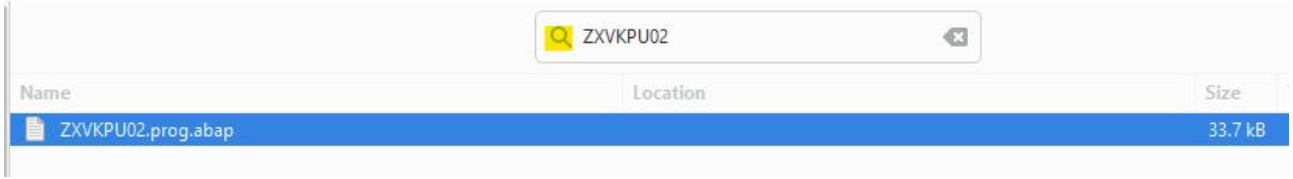
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	TRANSPORT	LI	PROGR	OBJEC	OBJ_NAME	O	L	G	L	A	PAC	B	OB	MAIN_OBJECT	Segme	ENHAN
2478	DRDK9A1ZLF	33	LIMU	REPS	ZXVKPU02		3				ZENTW		FUGR	XVKP		X

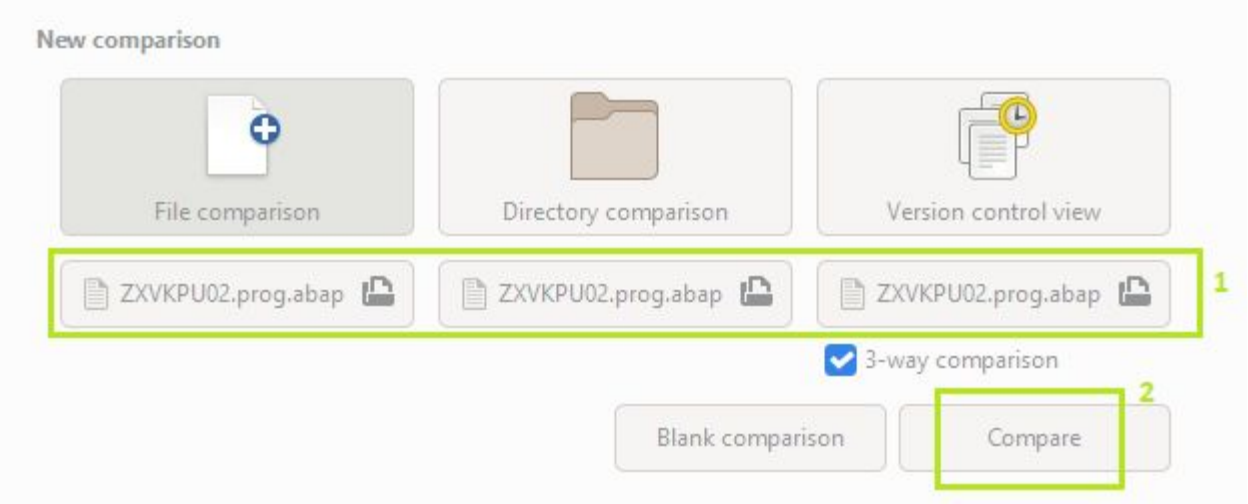
Open MELD app:
And load source_parsed, merger and target file for the object(from the extracted folder shared by tool team).



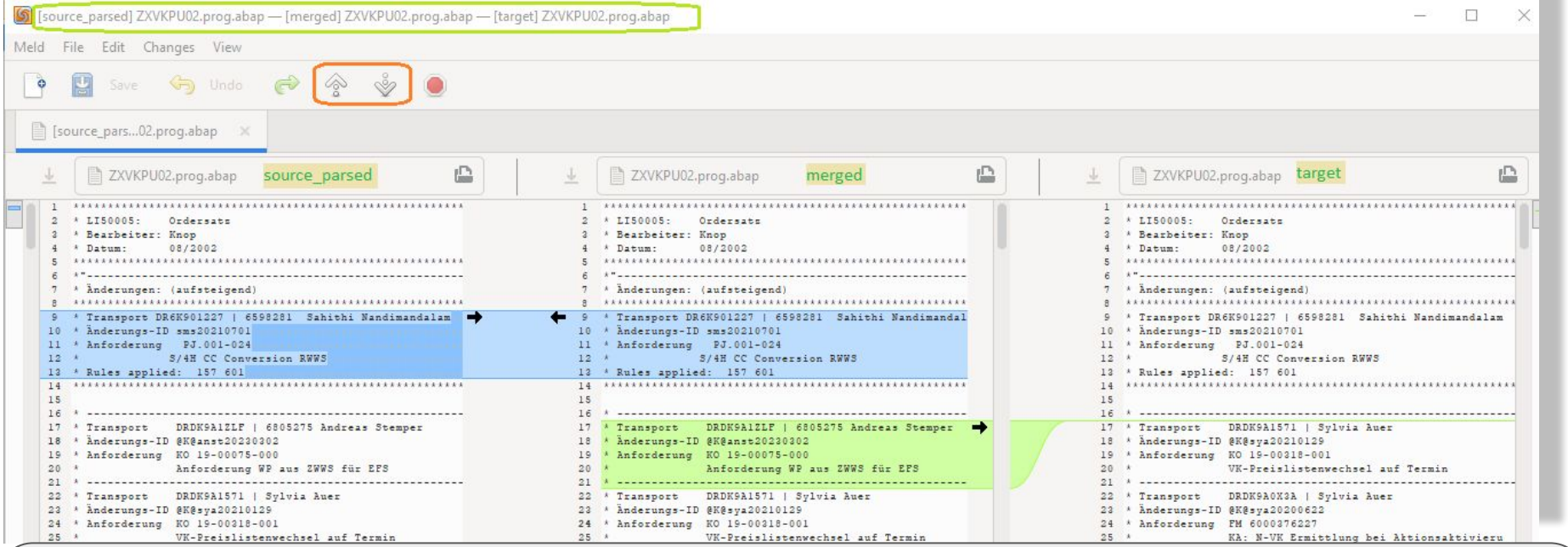

New comparison
File comparison
Directory comparison
Version control view
(None)
(None)
(None)
3-way comparison
Blank comparison
Compare


Select Second File
Recent
Home
Desktop
Documents
Dmterry
Downloads
report_050420231328
source_parsed
Name
merged
target
source_parsed
report.json





Once the files are loaded, click on ‘Compare’.



- source_parsed file→ contains the source active code + tool rules applied.
- target file→ contains the target active code.
- merged file→ contains the source_parsed & target code merged, which is most likely to be the code we have to update in target as part of current retrofit; but the merged file is not 100% accurate, thus we have to analyze each difference at SAP system level by comparing with below versions-and update the merge file accordingly.

Use the Arrows to navigate to next difference in meld.

Note: As part of Retrofit we have to bring the changes up to latest scope TR(and this latest scope TR may not be the active version, instead a below version in source side -compare the difference clearly), retaining the target changes(there can be a chance that the source side difference was changed/removed as part of target change - so compare each difference clearly before finalizing the merged code)

- Note: if any Smart(M) fixes are there in target, then retain them- if the fix was made for same logic; if there is a difference/scope TR change for the same line- analyze clearly and take the decision whether to retain or remove Smart(M) fix.
  Even if the Smart(M) fix is missed during Retrofit, this will be handled/taken care as part of Remediation fix for the current Retrofit run.
- Once the merge file is built correctly, we can update the target side with same code and Activate/save under Retrofit working TR.
- Later, Update the status and comments in scoping sheet.
- Incase, the Retrofitted object contains errors, analyze the errors and update the status and comments clearly in scoping sheet. [Most of the time, the error may be due to dependent objects in current retrofit for which Retrofit is not yet performed or object is marked as NEW, for which we have to wait for New objects ToC import for the error to be resolved.

OA2S

- About OA2S
  - OAuth2 Scope
  - For Retrofit, OA2S is taken care as part of Service scope

- Can OA2S be considered as **part of NEW/Safe To Import** analysis/ToC?
  - Yes - after Safe to import analysis.
  - As of now, in ,current scoping utility OA2S is NOT categorized as existing or new - so first we have to identify that.
    - In target, check the OA2S object in E071 or you can directly check the physical existence of the object from se80/open Other development object option.
    - If there are No hits for the current OA2S entry - we can consider them as NEW and mark the status as ToC and import it along with other safe to import objects.
    - If entry already exists, we have to analyze further on Safe to import analysis in se03/ tab-by-tab comparison and mark as ToC or No Action case accordingly.

KALPANA

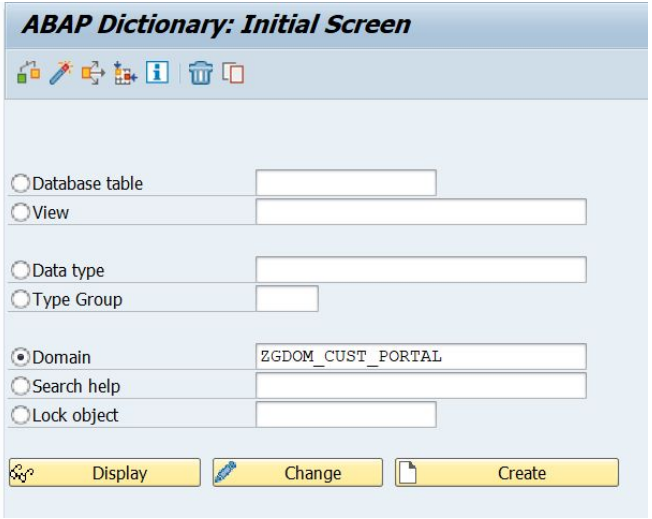
Object T	Sco	Scope	SPOC	Status
DOMA	IN	Service	kalpana	
DOMD	IN	Service	kalpana	
DTEL	IN	Service	kalpana	
DTED	IN	Service	kalpana	
MESS	IN	Service	Kalpana	
MSAD	IN	Service	Kalpana	
MSAG	IN	Service	Kalpana	
WDYC	IN	Service	kalpana	
WDYV	IN	Service	kalpana	

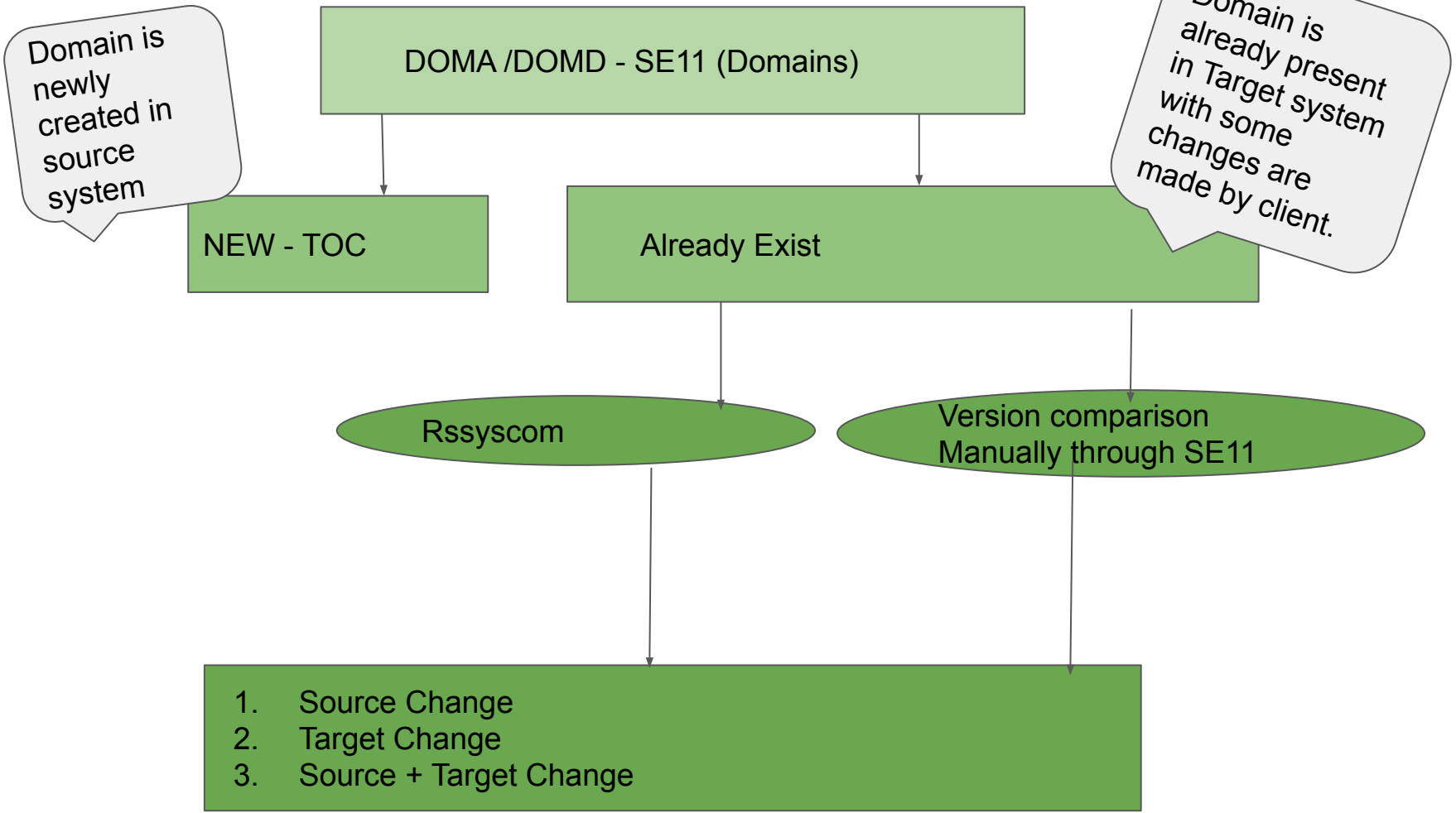
DOMA / DOMD

Object Types : Domains

- Object type **DOMA and DOMD** refers to **Domains.**
- **TCODE - SE11**
- This object type is categorized under service scope objects as there is no support from the smarttool.

SE11 Screen




Domain is newly created in source system
DOMA /DOMD - SE11 (Domains)
Domain is already present in Target system with some changes are made by client.
NEW - TOC
Already Exist
Rssyscom
Version comparison
Manually through SE11
1. Source Change
2. Target Change
3. Source + Target Change

NEW Objects are handled by Transport of Copies.

Case 1 - Source Change:

- Step1: Compare Scope TR version with below version in Source system.
- Step2: Add required changes in Target system.

Case 2 - Target change:

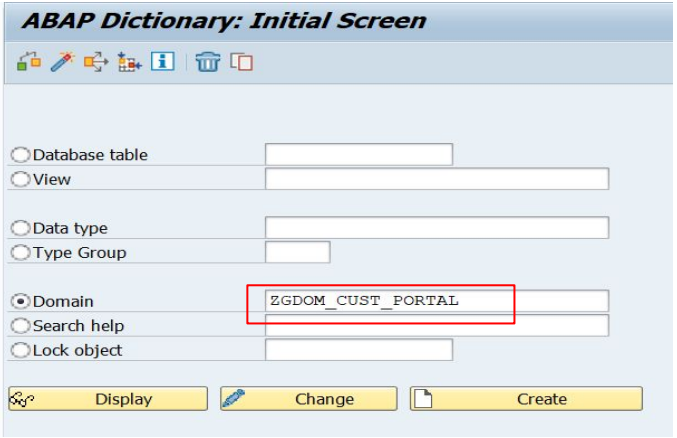
- Step 1: Compare Target Active TR version with below version in Target system.
- Step 2: Changes are completely relevant to target, mostly client has added changes in Target.

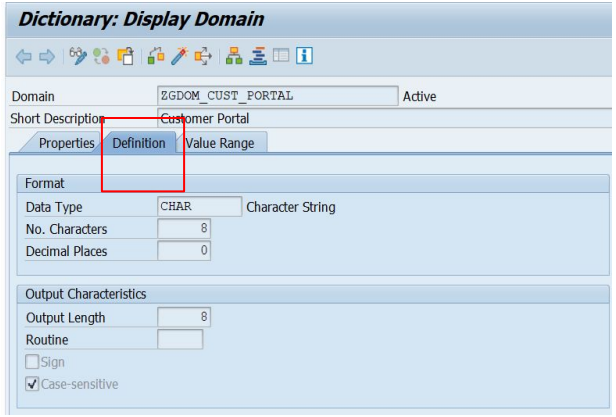
  In that case,
- Step 3: Retained Target changes

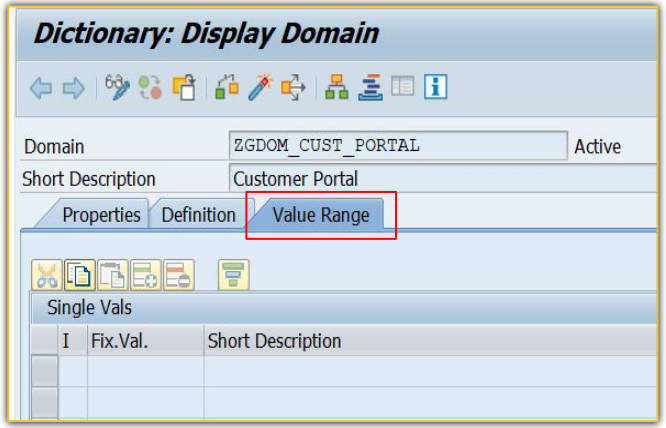
Case 3 -Target + Source change:

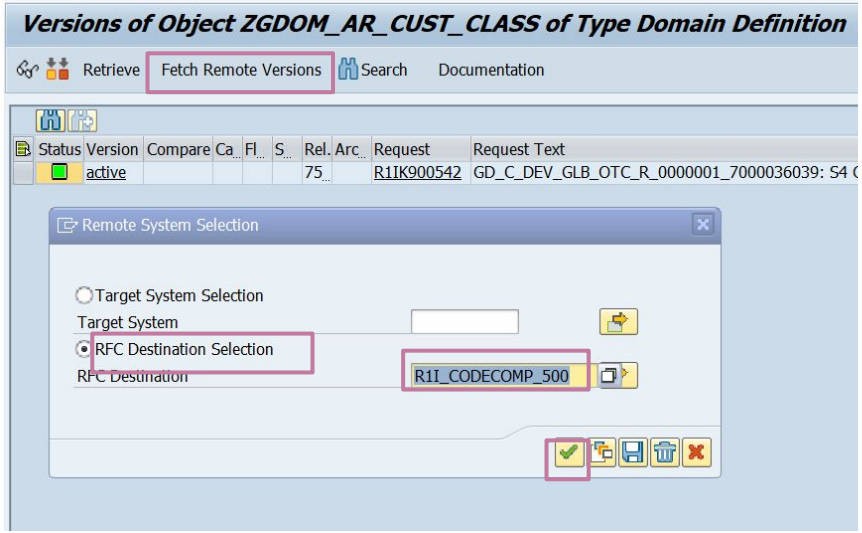
- Step 1: Compare Scope TR version with below version in Source system.
- Step 2: Compare Target Active TR version with below version in Target system.
- Step 3: Make required changes in target system based on scenarios.

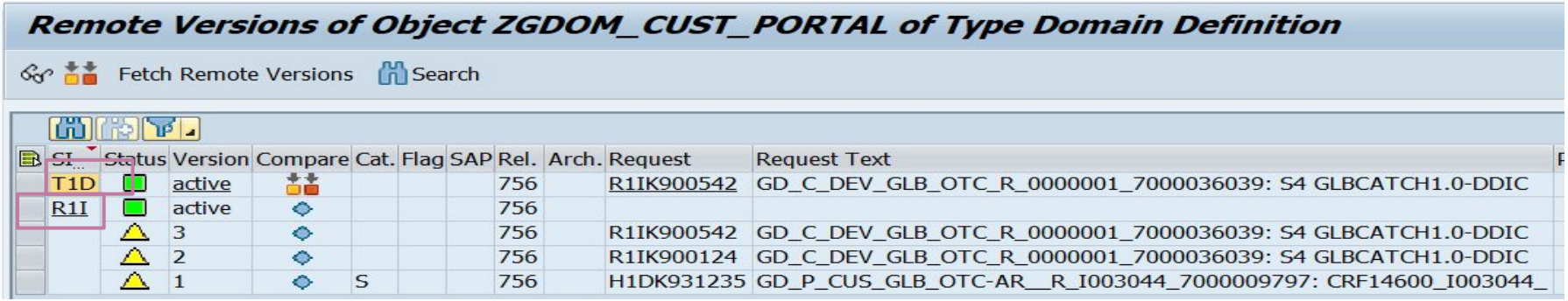
For all those cases, below screens are the steps to follow…








Versions of Object ZGDOM_AR_CUST_CLASS of Type Domain Definition
Retrieve
Fetch Remote Versions
Search
Documentation
Status Version Compare Ca... Fl... S... Rel. Arc... Request Request Text
active 75 R1IK900542 GD_C_DEV_GLB_OTC_R_0000001_7000036039: S4 (
Remote System Selection
Target System Selection
Target System
RFC Destination Selection
RFC Destination
R1I_CODECOMP_500


Remote Versions of Object ZGDOM_CUST_PORTAL of Type Domain Definition
Fetch Remote Versions
Search
SI... Status Version Compare Cat. Flag SAP Rel. Arch. Request Request Text
T1D active 756 R1IK900542 GD_C_DEV_GLB_OTC_R_0000001_7000036039: S4 GLBCATCH1.0-DDIC
R1I active 756
3 756 R1IK900542 GD_C_DEV_GLB_OTC_R_0000001_7000036039: S4 GLBCATCH1.0-DDIC
2 756 R1IK900124 GD_C_DEV_GLB_OTC_R_0000001_7000036039: S4 GLBCATCH1.0-DDIC
1 S 756 H1DK931235 GD_P_CUS_GLB_OTC-AR__R_I003044_7000009797: CRF14600_I003044_

For example: Based on Version comparison there is no changes in Domains

Version Display/Comparison of Domains

Delta display | Single/Double Column

Object Name	Version	Syst. Name	Request	Date	Author
ZGDOM_CUST_PORTAL	Actv.	T1D	R1IK900542	30.03.2023	09324660
ZGDOM_CUST_PORTAL	00003	R1I	R1IK900542	30.03.2023	09336505

General Attributes

Domain name	ZGDOM_CUST_PORTAL
Short Description	Customer Portal
Data Type	CHAR
No. of Characters	000008
Decimal Places	000000
Output Length	000008
Conversion Routine	
Sign	☐
Lowercase Letters	☑
Value table	
Domain name	
Proxy Usage	☐
Output Style	
AM/PM-Enabled	☐

General Attributes
Unchanged

Version Display/Comparison of Domains

Full display | Single/Double Column | Field catalog

Object Name	Version	Syst. Name	Request	Date	Author
ZGDOM_CUST_PORTAL	Actv.	T1D	R1IK900542	30.03.2023	09324660
ZGDOM_CUST_PORTAL	00003	R1I	R1IK900542	30.03.2023	09336505

General Attributes

Domain name	ZGDOM_CUST_PORTAL

General Attributes
Unchanged

DTEL & DTED

Object Type: Data Element

- Object type **DTEL & DTED** refers to **Data Element** objects**.**
- **TCODE - SE11**
- There is version management option here. Therefore, objects can check through rssyscomp or manual version comparison.

Case 1 - Source Change:
- Step1: Compare Scope TR version with below version in Source system.
- Step2: Add required changes in Target system.

Case 2 - Target change:
- Step 1: Compare Target Active TR version with below version in Target system.
- Step 2: Changes are completely relevant to target, mostly client has added changes in Target. In that case,
- Step 3: Retained Target changes

Case 3 -Target + Source change:
- Step 1: Compare Scope TR version with below version in Source system.
- Step 2: Compare Target Active TR version with below version in Target system.
- Step 3: Make required changes in target system.

ABAP Dictionary: Initial Screen

Database table
View
Data type: ZGDEL_CUST_PORTAL
Type Group
Domain
Search help
Lock object

Display | Change | Create

Dictionary: Display Data Element

Display Original | Documentation | Supplementary Documentation

Data element: ZGDEL_CUST_PORTAL — Active
Short Description: Customer Portal

Attributes | Data Type | Further Characteristics | Field Label

Elementary Type
- Domain: ZGDOM_CUST_PORTAL — Customer Portal
  - Data Type: CHAR — Character String
  - Length: 8
- Built-in type
  - Data Type:
  - Length: 0

Reference Type
- Referenced Type:
- Reference to built-in type
  - Data Type:
  - Length: 0

Dictionary: Display Data Element

Display Original | Documentation | Supplementary Documentation

Data element: ZGDEL_CUST_PORTAL — Active
Short Description: Customer Portal

Attributes | Data Type | Further Characteristics | Field Label

Search Help
- Name
- Parameters

Parameter ID

Default Component Name

Change Document
No Input History

Bidirectional Options
- Basic direction is set to LTR
- No BIDI Filtering

Dictionary: Display Data Element

Display Original | Documentation | Supplementary Documentation

Data element: ZGDEL_CUST_PORTAL — Active
Short Description: Customer Portal

Attributes | Data Type | Further Characteristics | Field Label

	Length	Field Label
Short	10	Portal
Medium	20	Customer Portal
Long	40	Customer Portal
Heading	55	Customer Portal

In below Example : DTEL has source change

Versions of Object /SCL/BAY_DIAGRAM of Type Data Element Defin

Retrieve | Fetch Remote Versions | Documentation

Status	Version	Compare	Cat.	Flag	SAP	Rel.	Arch.	Request	Request Text
	modified					757		DNRK976206	Smartshift Run5 Retrofit
	active					757		DNRK976206	Smartshift Run5 Retrofit
	3		I			757		DNRK972813	Smartshift Run3 Retrofit

Remote System Selection

- Target System Selection
- Target System
- RFC Destination Selection
- RFC Destination: PNRCLNT100_RETRO

Remote Versions of Object /SCL/BAY_DIAGRAM of Type Data Element Defini

Fetch Remote Versions

SID	Status	Version	Compare	Cat.	Flag	SAP	Rel.	Arch.	Request	Request Text	Project	Date
PNR		active					750		MNRK977656	52122/10 Reduce Bay diagram printing in RFIM		03/05/2023
		2			I		750		PNRK900203	CONA Retrofit Run#3 ToC request		12/22/2022
		1		S			750		MNRK975621	51648/10 Reduce print documents in the Shipment process		12/11/2022
DNR		modified					757		DNRK976206	Smartshift Run5 Retrofit		04/13/2023
		active					757		DNRK976206	Smartshift Run5 Retrofit		04/01/2023
		3		I			757		DNRK972813	Smartshift Run3 Retrofit		04/01/2023
		2					750		MNRK977656	52122/10 Reduce Bay diagram printing in RFIM		01/09/2023
		1					750		MNRK975621	51648/10 Reduce print documents in the Shipment process		10/28/2022

Version Display/Comparison of Data Elements

Full Display | Single/Double Column | Field catalog

Object Name	Version	Request	Date	Author
/SCL/BAY_DIAGRAM	modif	DNRK976206	04/13/2023	NT003120
/SCL/BAY_DIAGRAM	Actv.	DNRK976206	04/01/2023	NT002133

General Attributes

Data element	/SCL/BAY_DIAGRAM
Short Field Label	Bay Dgm SM
Short Field Label	Bay Diagm
Long field label	Print Bay Diagram Shipment Monitor
Long field label	Print Bay Diagram

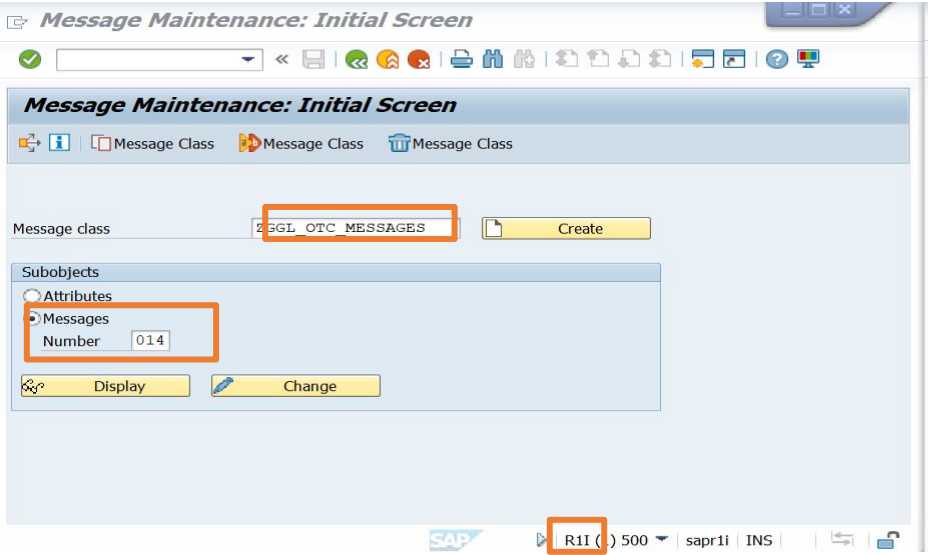
General Attributes
Changed: 2

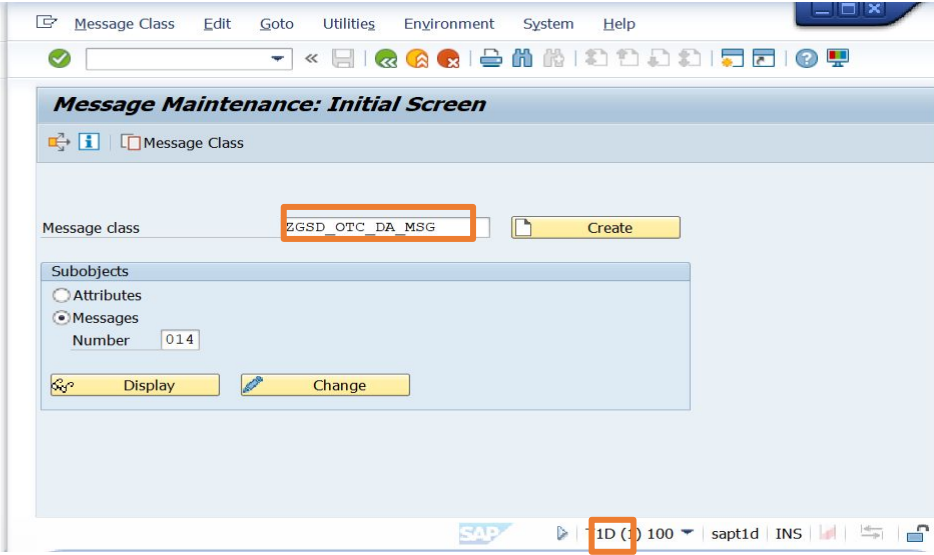
MESS / MSAD / MSAG

Object Type: Message Maintenance

- Object type **MESS/ MSAD/ MSAG** refers to **Message Objects.**
- **TCODE - SE91**
- This object type is categorized under service scope objects as there is less/no support from the smarttool.
- There is no version management option available here. Hence,for this object type, we need to compare source screen and Target screen by manually..

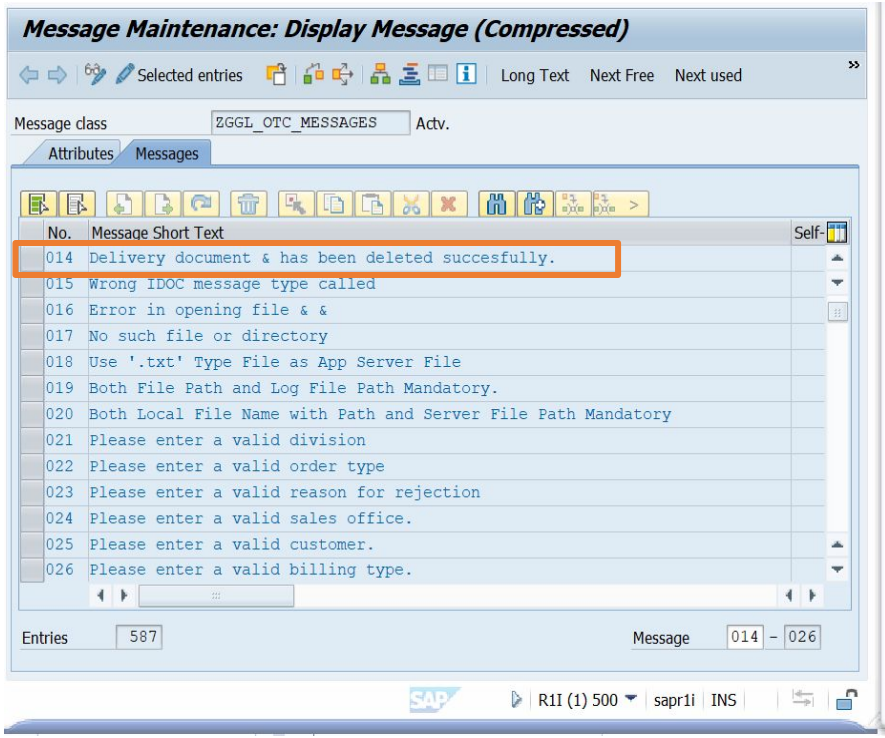
Below is an example of screen comparison…

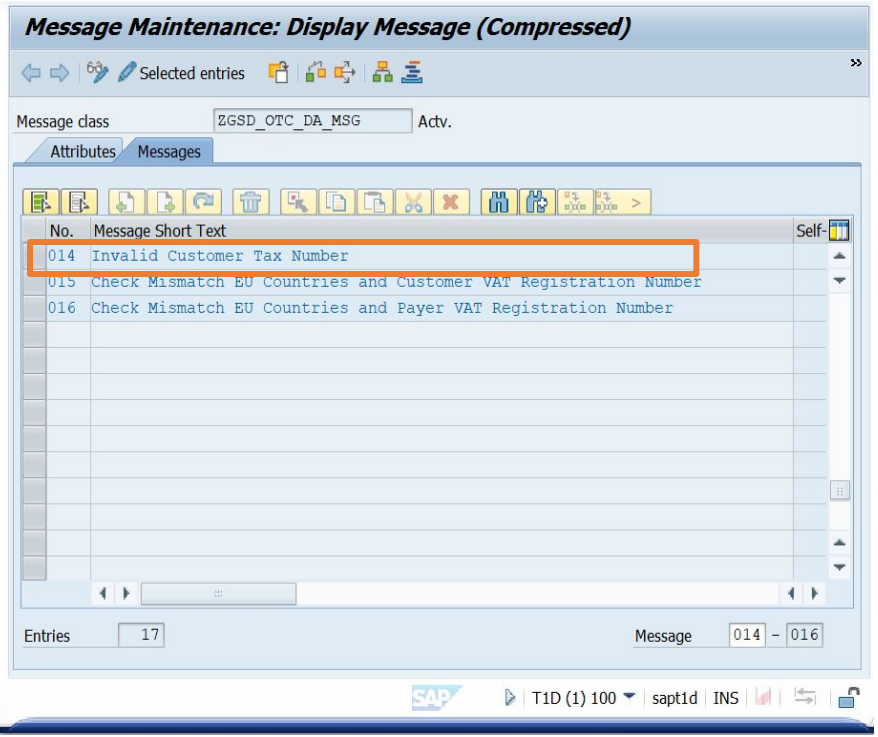




R1D - Source system
T1D - Target system

Below case, difference in message no **014.** Therefore, modify target message as like source message





Self- Explanatory checkbox should be considered in both source and target

Message Maintenance: Display Message (Compressed)

Selected entries | Long Text | Next Free | Next used

Message class: ZGGL_OTC_MESSAGES Actv.

Attributes | Messages

No.	Self-Explanatory	Last Changed By	Changed On
014	☑	09240252	03.01.2019
015	☑	09240252	14.01.2019
016	☑	09240252	13.02.2019
017	☑	09240252	19.02.2019
018	☑	09240252	19.02.2019
019	☑	09240252	19.02.2019
020	☑	09240252	19.02.2019
021	☑	09249814	22.02.2019
022	☑	09249814	22.02.2019
023	☑	09249814	22.02.2019
024	☑	09251632	07.03.2019
025	☑	09251632	07.03.2019
026	☑	09251632	07.03.2019

Entries 587 Message 014 – 026

R1I (1) 500 | sapr1i | INS

Message Maintenance: Display Message (Compressed)

Selected entries | Long Text | Next Free | Next used

Message class: ZGSD_OTC_DA_MSG Actv.

Attributes | Messages

No.	Self-Explanatory	Last Changed By	Changed On
014	☑	09283392	11.05.2022
015	☑	09283392	11.05.2022
016	☑	09283392	11.05.2022

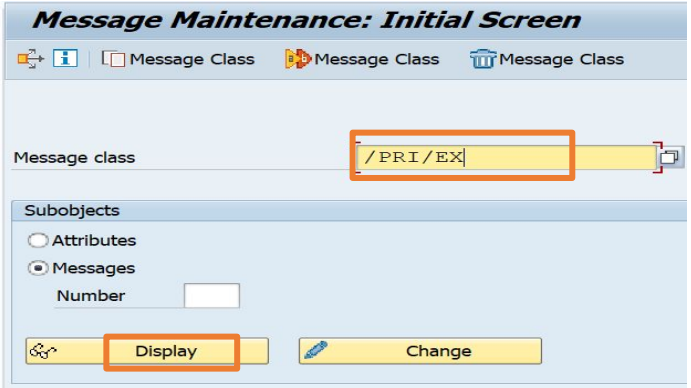
Entries 17 Message 014 – 016

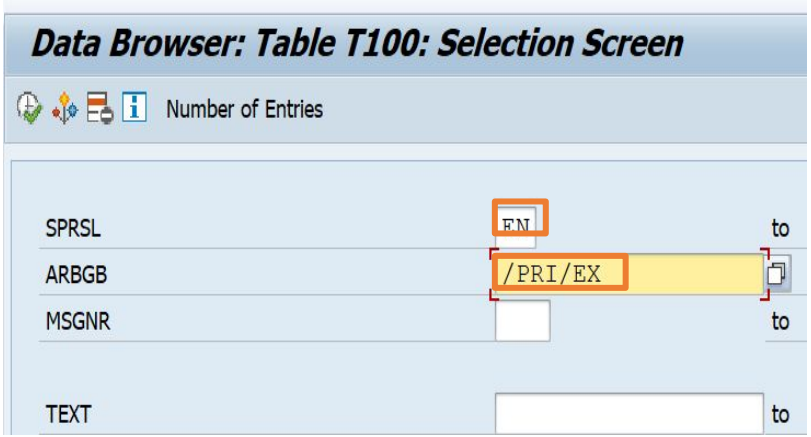
T1D (1) 100 | sapt1d | INS

MSAD / MSAG

Object Type: Message Class

- There is no Message number is present in MSAG type, so Need to check entire message in SE91.
- Step 1: Source system - SE11 - T100 . Pass Message Original Lang and Name. Execute.
- Local file - Copy to clipboard and compare in Meld.
- Step 2: Source system - SE11 - T100 . Pass Message Original Lang and Name. Execute.Local file - Copy to clipboard and compare in Meld.
- Local file - Copy to clipboard and compare in Meld.
- Compare both files and do required changes.





Copy both source and target details to clipboard and check it in meld.

Data Browser: Table T100 Select Entries 87

Check Table...

SPRSL	ARBGB	MSGNR	TEXT
E	/PRI/EX	000	&1 &2 &3 &4
E	/PRI/EX	001	Table &1 already exists in DDIC. Manually adjust if needed.
E	/PRI/EX	002	Table &1 does not exist in DDIC.
E	/PRI/EX	003	No (working) RFC destination to service available.
E	/PRI/EX	004	Could not create itab. Type &1 is unknown.
E	/PRI/EX	005	Error during batch generation for object &1.
E	/PRI/EX	006	Structure of &1 insonsistent. &2 missing.
E	/PRI/EX	007	Cloud not obtain lock on app &1 queue &2 partition &3.
E	/PRI/EX	008	Customer &1 is not setup in this system.
E	/PRI/EX	009	Queue &1 application &2 is not setup in this system.
E	/PRI/EX	010	Application &1 for customer &2 is not setup in this system.
E	/PRI/EX	011	Could not successfully submit request.
E	/PRI/EX	012	Incompatible/unexpected data type received from data mapping.
E	/PRI/EX	013	Could not persist polling data record with data uid &1 in table &2.
E	/PRI/EX	014	Could not compress data.
E	/PRI/EX	015	Could not persist new queue ack ID &1 in database.
E	/PRI/EX	016	No data received from data mapping.
E	/PRI/EX	017	Could not instantiate class &1 for interface &2.
E	/PRI/EX	018	Class configured for interface &1 does not implement the same.
E	/PRI/EX	019	Data object id &1 (&2) is not setup in this system.
E	/PRI/EX	020	Could not update proc uid &1 in DB table &2 for &3 records.
E	/PRI/EX	021	Invalid RFC queue name &1 (subrc &2) using &3 instead.
E	/PRI/EX	022	Could not obtain lock on app &4 queue id &1 partition &2 proc uid &3.
E	/PRI/EX	023	Conversion routine not maintained for dim type &1 item &2 field &3.
E	/PRI/EX	024	Customer ID &1 is not setup.
E	/PRI/EX	025	Application ID &1 is not setup.

Data Browser: Table T100 Select Entries 87

Check Table...

SPRSL	ARBGB	MSGNR	TEXT
E	/PRI/EX	000	&1 &2 &3 &4
E	/PRI/EX	001	Table &1 already exists in DDIC. Manually adjust if needed.
E	/PRI/EX	002	Table &1 does not exist in DDIC.
E	/PRI/EX	003	No (working) RFC destination to service available.
E	/PRI/EX	004	Could not create itab. Type &1 is unknown.
E	/PRI/EX	005	Error during batch generation for object &1.
E	/PRI/EX	006	Structure of &1 insonsistent. &2 missing.
E	/PRI/EX	007	Cloud not obtain lock on app &1 queue &2 partition &3.
E	/PRI/EX	008	Customer &1 is not setup in this system.
E	/PRI/EX	009	Queue &1 application &2 is not setup in this system.
E	/PRI/EX	010	Application &1 for customer &2 is not setup in this system.
E	/PRI/EX	011	Could not successfully submit request.
E	/PRI/EX	012	Incompatible/unexpected data type received from data mapping.
E	/PRI/EX	013	Could not persist polling data record with data uid &1 in table &2.
E	/PRI/EX	014	Could not compress data.
E	/PRI/EX	015	Could not persist new queue ack ID &1 in database.
E	/PRI/EX	016	No data received from data mapping.
E	/PRI/EX	017	Could not instantiate class &1 for interface &2.
E	/PRI/EX	018	Class configured for interface &1 does not implement the same.
E	/PRI/EX	019	Data object id &1 (&2) is not setup in this system.
E	/PRI/EX	020	Could not update proc uid &1 in DB table &2 for &3 records.
E	/PRI/EX	021	Invalid RFC queue name &1 (subrc &2) using &3 instead.
E	/PRI/EX	022	Could not obtain lock on app &4 queue id &1 partition &2 proc uid &3.
E	/PRI/EX	023	Conversion routine not maintained for dim type &1 item &2 field &3.
E	/PRI/EX	024	Customer ID &1 is not setup.
E	/PRI/EX	025	Application ID &1 is not setup.

Below is an example for IDENTICAL case.

Meld File Edit Changes View

Save Undo

(None) | (None)

```
1 :          T100                                                  1 /2023    Dynamic List Display
2 ayed Fields:   4 of   4        Fixed Columns:                    2 ----------------------------------------------------------
3 ----------------------------------------------------------------- 3 ----------------------------------------------------------
4 RSL|ARBGB                |MSGNR|TEXT                             4 RSL|ARBGB   |MSGNR|TEXT
5 ----------------------------------------------------------------- 5 ----------------------------------------------------------
6     |/PRI/EX             |000  |&1 &2 &3 &4                      6     |/PRI/EX|000  |&1 &2 &3 &4
7     |/PRI/EX             |001  |Table &1 already exists in I      7     |/PRI/EX|001  |Table &1 already exists in DDIC. Manually a
8     |/PRI/EX             |002  |Table &1 does not exist in I      8     |/PRI/EX|002  |Table &1 does not exist in DDIC.
9     |/PRI/EX             |003  |No (working) RFC destinatior      9     |/PRI/EX|003  |No (working) RFC destination to service ava
10    |/PRI/EX             |004  |Could not create itab. Type       10    |/PRI/EX|004  |Could not create itab. Type &1 is unknown.
11    |/PRI/EX             |005  |Error during batch generatic      11    |/PRI/EX|005  |Error during batch generation for object &1
12    |/PRI/EX             |006  |Structure of &1 insonsistent      12    |/PRI/EX|006  |Structure of &1 insonsistent. &2 missing.
13    |/PRI/EX             |007  |Cloud not obtain lock on app      13    |/PRI/EX|007  |Cloud not obtain lock on app &1 queue &2 pa
14    |/PRI/EX             |008  |Customer &1 is not setup in       14    |/PRI/EX|008  |Customer &1 is not setup in this system.
15    |/PRI/EX             |009  |Queue &1 application &2 is r      15    |/PRI/EX|009  |Queue &1 application &2 is not setup in thi
16    |/PRI/EX             |010  |Application &1 for customer       16    |/PRI/EX|010  |Application &1 for customer &2 is not setup
17    |/PRI/EX             |011  |Could not successfully submi      17    |/PRI/EX|011  |Could not successfully submit request.
18    |/PRI/EX             |012  |Incompatible/unexpected data      18    |/PRI/EX|012  |Incompatible/unexpected data type received
19    |/PRI/EX             |013  |Could not persist polling da      19    |/PRI/EX|013  |Could not persist polling data record with
20    |/PRI/EX             |014  |Could not compress data.          20    |/PRI/EX|014  |Could not compress data.
21    |/PRI/EX             |015  |Could not persist new queue       21    |/PRI/EX|015  |Could not persist new queue ack ID &1 in da
22    |/PRI/EX             |016  |No data received from data m      22    |/PRI/EX|016  |No data received from data mapping.
23    |/PRI/EX             |017  |Could not instantiate class       23    |/PRI/EX|017  |Could not instantiate class &1 for interfac
24    |/PRI/EX             |018  |Class configured for interfa      24    |/PRI/EX|018  |Class configured for interface &1 does not
25    |/PRI/EX             |019  |Data object id &1 (&2) is nc      25    |/PRI/EX|019  |Data object id &1 (&2) is not setup in this
26    |/PRI/EX             |020  |Could not update proc uid &1      26    |/PRI/EX|020  |Could not update proc uid &1 in DB table &2
27    |/PRI/EX             |021  |Invalid RFC queue name &1 (s      27    |/PRI/EX|021  |Invalid RFC queue name &1 (subrc &2) using
```

There is difference between Source and Target, Here we need to find which changes is belongs to Either SOURCE or TAREGT. For that, Check versions in SE03.

```
81 |  E    |/PRI/MSG|075   |Started task UUID &1 on app server &
82 |  E    |/PRI/MSG|076   |Processing status &1 overwritten wit
83 |  E    |/PRI/MSG|077   |BO and BO Node Validation - Tables /
84 |  E    |/PRI/MSG|078   |Table &1 linked to BO with Short ID
85 |  E    |/PRI/MSG|079   |BO Node Association Validation - Tab
86 |  E    |/PRI/MSG|080   |BO Node Data Provider Validation - T
87 |  E    |/PRI/MSG|081   |BO Node Schema Validation - Table /P
88 |  E    |/PRI/MSG|082   |BO Node Role Assignment - Table /PRI
89 |  E    |/PRI/MSG|083   |Trigger &1 created for table &2
90 |  E    |/PRI/MSG|084   |Trigger &1 dropped for table &2
91 |  E    |/PRI/MSG|085   |Trigger &1 for table &2 not found at
92 |  E    |/PRI/MSG|086   |End parallel execution due to functi
93 |  E    |/PRI/MSG|087   |End parallel execution due to timewc
94 |  E    |/PRI/MSG|088   |Queue lifetime not found or lower th
95 |  E    |/PRI/MSG|089   |Data entry with UUID &2 (&1) modifie
96 |  E    |/PRI/MSG|090   |No Row has beeen selected
97 |  E    |/PRI/MSG|091   |Selected WL tables will be truncated
```

```
80 |  E    |/PRI/MSG|074   |Database Trigger definition: Table &
81 |  E    |/PRI/MSG|075   |Started task UUID &1 on app server &
82 |  E    |/PRI/MSG|076   |Processing status &1 overwritten wit
83 |  E    |/PRI/MSG|077   |BO and BO Node Validation - Tables /
84 |  E    |/PRI/MSG|078   |Table &1 linked to BO with Short ID
85 |  E    |/PRI/MSG|079   |BO Node Association Validation - Tab
86 |  E    |/PRI/MSG|080   |BO Node Data Provider Validation - T
87 |  E    |/PRI/MSG|081   |BO Node Schema Validation - Table /P
88 |  E    |/PRI/MSG|082   |BO Node Role Assignment - Table /PRI
89 |  E    |/PRI/MSG|083   |Trigger &1 created for table &2
90 |  E    |/PRI/MSG|084   |Trigger &1 dropped for table &2
91 |  E    |/PRI/MSG|085   |Trigger &1 for table &2 not found at
92 |  E    |/PRI/MSG|086   |End parallel execution due to functi
93 |  E    |/PRI/MSG|087   |End parallel execution due to timewc
94 |  E    |/PRI/MSG|090   |No Row has beeen selected
95 |  E    |/PRI/MSG|091   |Selected WL tables will be truncated
96 |  E    |/PRI/MSG|092   |Selected RS tables will be truncated
```

```
105|099  |Data object &1 - Data Key: &3 UUID: &2
106|100  |Task &1 triggered by user &2.
107|101  |RS tables below truncated in BO Manager by user &1.
108|102  |WL tables below truncated in BO Manager by user &1.
109|103  |DB Trigger for tables below was locked by user &1.
110|104  |DB Trigger for tables below was unlocked by user &1.
111|105  |SPBT initialize failed with server group &1
112|106  |Failover to non parallelized processing.
113|107  |SPBT resource info failed with server group &1
114|108  |Update replicaton: no node data found for BO &1 with
115|109  |Update replicaton: no replic state found for BO &1 wi
116|110  |Pulled &1 entries from app &4 queue &2 partition &3
117|111  |Cloud queue has more: &1; ack key: &2
118|112  |Application Id &1 does not exist
119|113  |BO &1 does not exist in Application Id &2
120|-----------------------------------------------------------
```

```
100|096  |BO configuraton errors were found.
101|097  |Database Trigger definition: Pooled Table &1 not supp
102|098  |&3 pending entries for queue &1 and partition &2 were
103|099  |Data object &1 - Data Key: &3 UUID: &2
104|100  |Task &1 triggered by user &2.
105|101  |RS tables below truncated in BO Manager by user &1.
106|102  |WL tables below truncated in BO Manager by user &1.
107|103  |DB Trigger for tables below was locked by user &1.
108|104  |DB Trigger for tables below was unlocked by user &1.
109|105  |SPBT initialize failed with server group &1
110|106  |Failover to non parallelized processing.
111|107  |SPBT resource info failed with server group &1
112|108  |Update replicaton: no node data found for BO &1 with
113|109  |Update replicaton: no replic state found for BO &1 wi
114|110  |Pulled &1 entries from app &4 queue &2 partition &3
115|111  |Cloud queue has more: &1; ack key: &2
```

In SE03 - Give MSAG and respective message class name. Then execute , Sort based on dates. Below case There is a baseline TR MNRK966972 which is present in Both Source and Target system. Whereas Scope TR is MNRK978369 which is recently added in source.

Target doesn't have any Client change TR (SAPKCCC752 belongs o Basis activity, not relevant to Message changes).

Find Objects in Requests/Tasks

Object Selection

Program	PGMID	R3TR	Obj	PROG	
Function Group	PGMID	R3TR	Obj	FUGR	
Class	PGMID	R3TR	Obj	CLAS	
Table/Structure	PGMID	R3TR	Obj	TABL	
View	PGMID	R3TR	Obj	VIEW	
Table Type	PGMID	R3TR	Obj	TTYP	
Data Element	PGMID	R3TR	Obj	DTEL	
Message Class	PGMID	R3TR	Obj	MSAG	/PRI/MSG

S

Request Overview

Request	Short Description	Owner	Date
MNRK978369	Osapiens Transports for Sascha/Peter	T76742	01/27/2023
OSDK900278	osapiens connector - rel20221207	OSAPIENS	01/17/2023
MNRK966972	RT 49676 - Priming Cloud TOC	T76742	12/23/2021
OSDK900028	osapiens connector - rel 20211222	OSAPIENS	12/22/2021
OSDK900022	osapiens Connector - rel20211217	OSAPIENS	12/17/2021
DMRK901975	osapiens connector - rel20211026	OSAPIENS3	10/26/2021

Find Objects in Requests/Tasks

Objects View

T

Request Overview

Request	Short Description	Owner	Date
SAPKCCD757	Commandfile for customer export	DDIC	03/27/2023
SAPKCCC752	Commandfile for customer export	DDIC	03/25/2023
MNRK966972	RT 49676 - Priming Cloud TOC	T76742	12/23/2021
OSDK900028	osapiens connector - rel 20211222	OSAPIENS	12/22/2021
OSDK900022	osapiens Connector - rel20211217	OSAPIENS	12/17/2021
DMRK901975	osapiens connector - rel20211026	OSAPIENS3	10/26/2021

SE91- Edit - Do required changes as like source based on Scope TR

Message Maintenance: Change Messages

Selected entries | Long Text | Next Free | Next used | Compact Display | Last Changed By

Message class: /PRI/MSG Actv.

Attributes | Messages

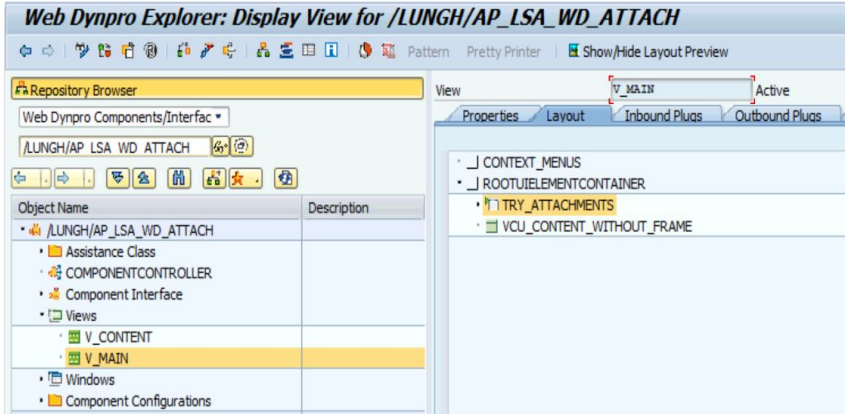
No.	Message Short Text	Self-Explanatory
087	End parallel execution due to timewout while waiting for async task.	✓
088		✓
089		✓
090	No Row has beeen selected	✓
091	Selected WL tables will be truncated. Are you sure?	✓
092	Selected RS tables will be truncated. Are you sure?	✓
093	Error when executing the Truncate SQL statement	✓

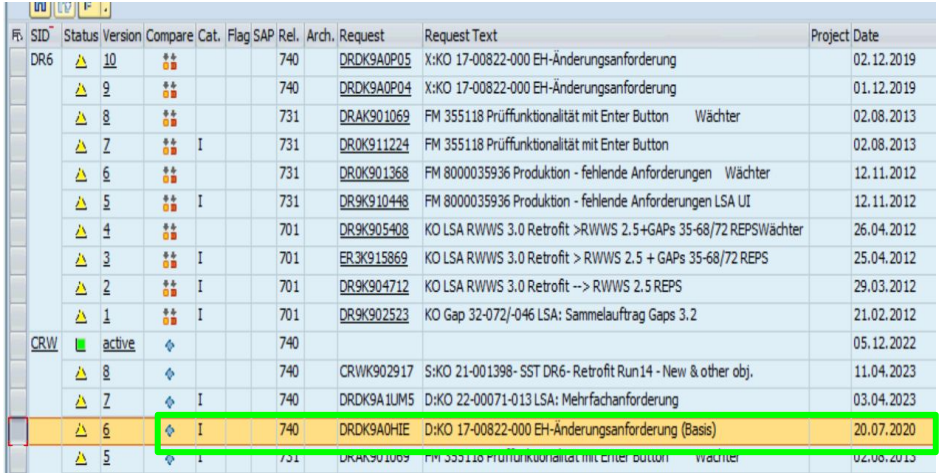
WDYC / WDYV

Object Type: Webdynpro

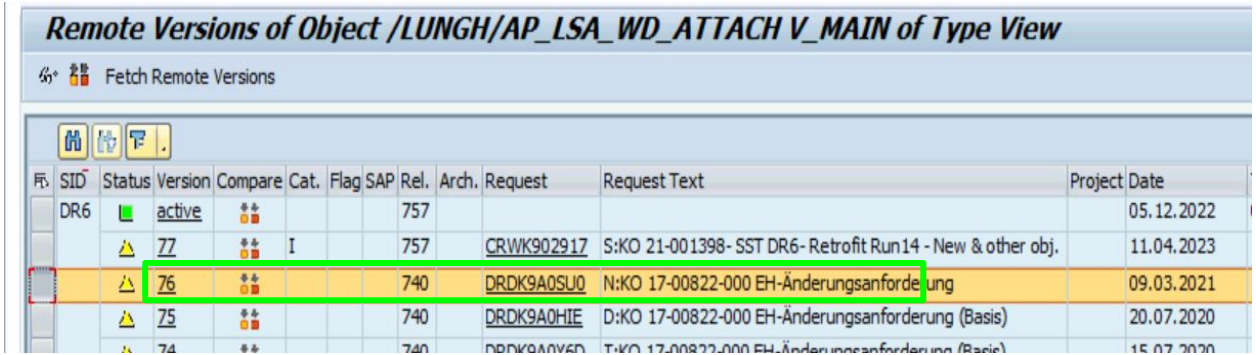
- Object type **WDYC/ WDYV** refers to **Webdynpro Component and Views.**
- **TCODE - SE80**
- This object type is categorized under service scope objects as there is less/no support from the smarttool.
- Mostly WDYC and WDYV objects are coming under TOC. But some cases we do retrofit based on certain conditions
- Case 1: Only source change consider as Safe to Import.
- Case 2: If Target has layout change and Source also have Layout changes then particular case will be considered as Customer action. Other than Layout like Methods, Actions, Events etc will be doing retrofit based on version comparisons between Source and Target with respective scope TR
- Case 3: Once target has latest change consider as NO action.

Example for TOC - WDYV TR version comparison between Source and Target. Among one of Source Pervious Version with Target active version should be same. So active source version can be directly move to target.





SID	Status	Version	Compare	Cat.	Flag	SAP	Rel.	Arch.	Request	Request Text	Project	Date
DR6		10					740		DRDK9A0P05	X:KO 17-00822-000 EH-Änderungsanforderung		02.12.2019
		9					740		DRDK9A0P04	X:KO 17-00822-000 EH-Änderungsanforderung		01.12.2019
		8					731		DRAK901069	FM 355118 Prüffunktionalität mit Enter Button Wächter		02.08.2013
		7		I			731		DR0K911224	FM 355118 Prüffunktionalität mit Enter Button		02.08.2013
		6					731		DR0K901368	FM 8000035936 Produktion - fehlende Anforderungen Wächter		12.11.2012
		5		I			731		DR9K910448	FM 8000035936 Produktion - fehlende Anforderungen LSA UI		12.11.2012
		4					701		DR9K905408	KO LSA RWWS 3.0 Retrofit >RWWS 2.5+GAPs 35-68/72 REPSWächter		26.04.2012
		3		I			701		ER3K915869	KO LSA RWWS 3.0 Retrofit > RWWS 2.5 + GAPs 35-68/72 REPS		25.04.2012
		2		I			701		DR9K904712	KO LSA RWWS 3.0 Retrofit --> RWWS 2.5 REPS		29.03.2012
		1		I			701		DR9K902523	KO Gap 32-072/-046 LSA: Sammelauftrag Gaps 3.2		21.02.2012
CRW		active					740					05.12.2022
		8					740		CRWK902917	S:KO 21-001398- SST DR6- Retrofit Run14 - New & other obj.		11.04.2023
		7		I			740		DRDK9A1UM5	D:KO 22-00071-013 LSA: Mehrfachanforderung		03.04.2023
		6		I			740		DRDK9A0HIE	D:KO 17-00822-000 EH-Änderungsanforderung (Basis)		20.07.2020
		5		I			731		DRAK901069	FM 355118 Prüffunktionalität mit Enter Button Wächter		02.08.2013

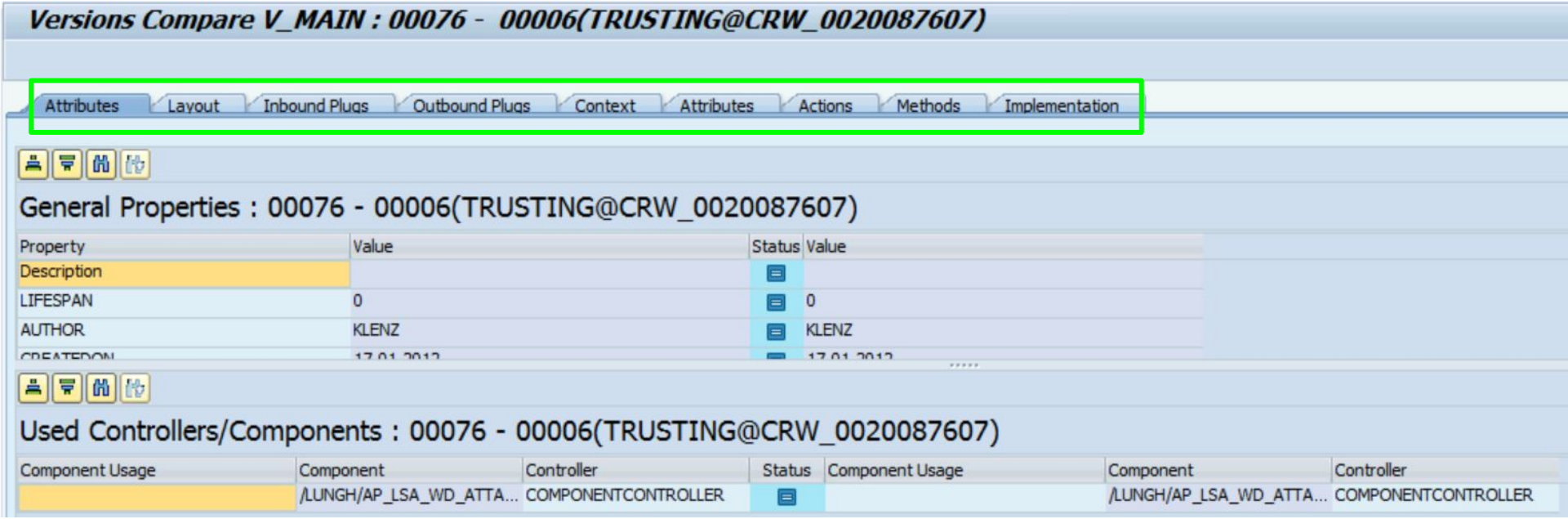


Remote Versions of Object /LUNGH/AP_LSA_WD_ATTACH V_MAIN of Type View

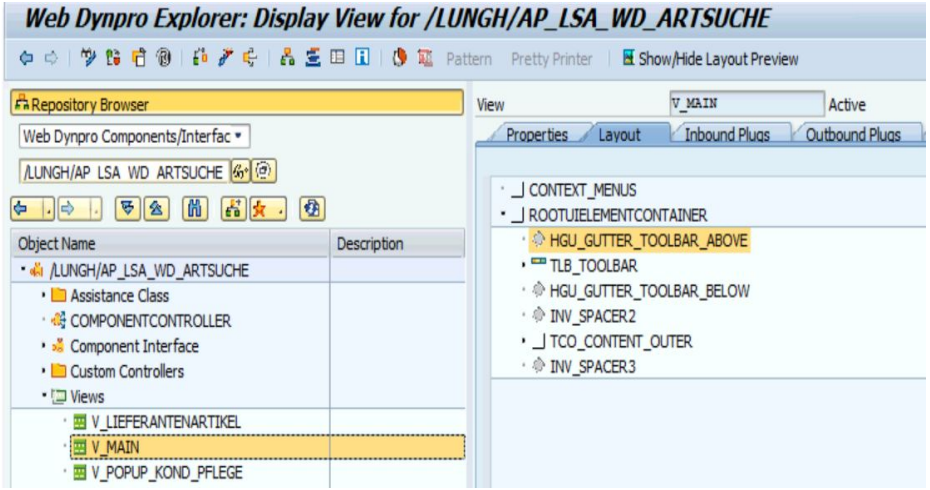
Fetch Remote Versions

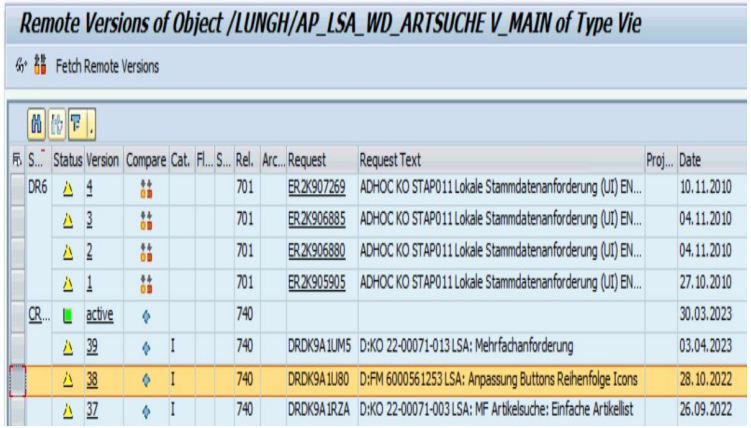
SID	Status	Version	Compare	Cat.	Flag	SAP	Rel.	Arch.	Request	Request Text	Project	Date
DR6		active					757					05.12.2022
		77		I			757		CRWK902917	S:KO 21-001398- SST DR6- Retrofit Run14 - New & other obj.		11.04.2023
		76					740		DRDK9A0SU0	N:KO 17-00822-000 EH-Änderungsanforde ung		09.03.2021
		75					740		DRDK9A0HIE	D:KO 17-00822-000 EH-Änderungsanforderung (Basis)		20.07.2020
		74					740		DRDK9A0Y6D	T:KO 17-00822-000 EH-Änderungsanforderung (Basis)		15.07.2020

Comparing Source one of the below version of scope TR with Target active version, we conclude there is no difference found. So, directly we can proceed with TOC for SOURCE ACTIVE CHANGES

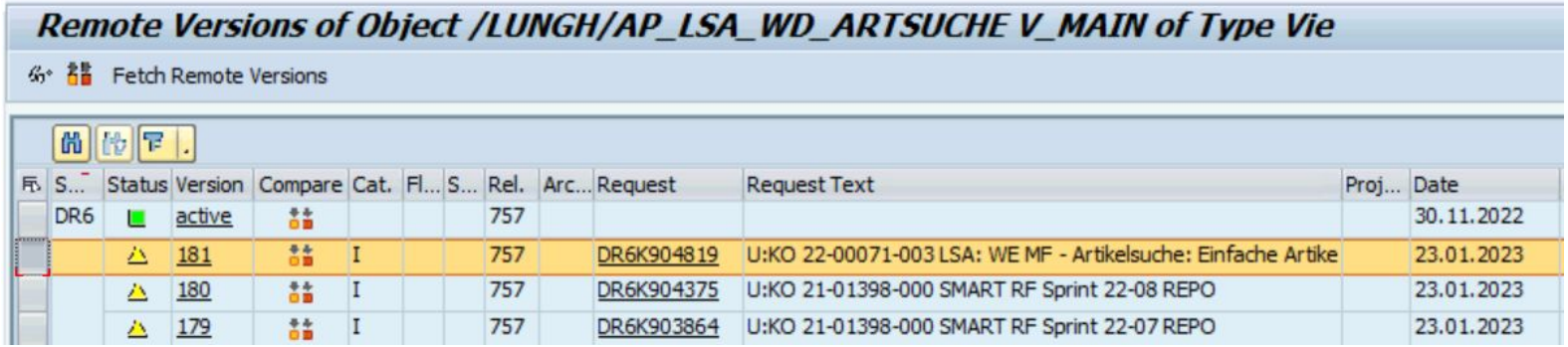


Example for Customer Action - Layout changes in Active Target version comparing with Scope TR in Source



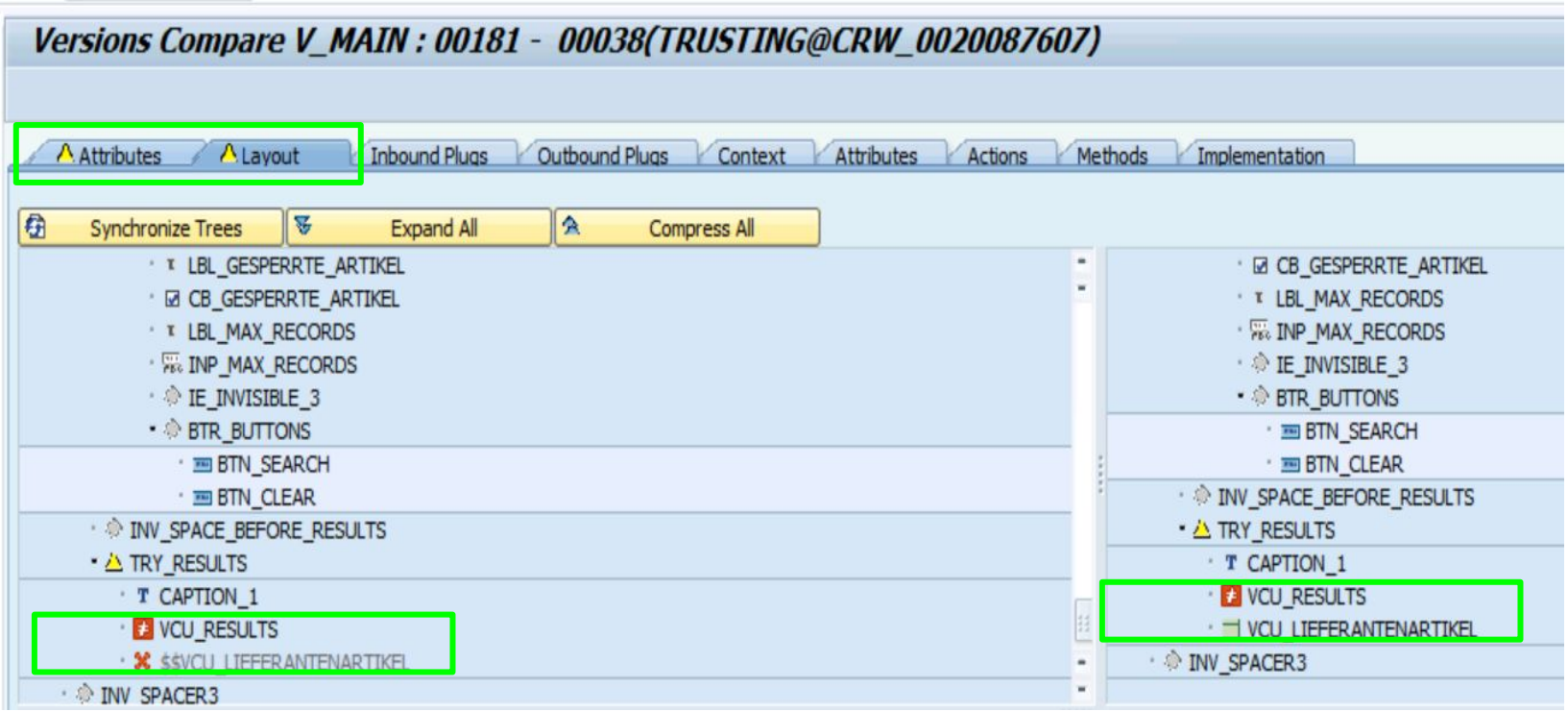


Fr.	S...	Status	Version	Compare	Cat.	Fl...	S...	Rel.	Arc...	Request	Request Text	Proj...	Date
	DR6		4					701		ER2K907269	ADHOC KO STAP011 Lokale Stammdatenanforderung (UI) EN...		10.11.2010
			3					701		ER2K906885	ADHOC KO STAP011 Lokale Stammdatenanforderung (UI) EN...		04.11.2010
			2					701		ER2K906880	ADHOC KO STAP011 Lokale Stammdatenanforderung (UI) EN...		04.11.2010
			1					701		ER2K905905	ADHOC KO STAP011 Lokale Stammdatenanforderung (UI) EN...		27.10.2010
	CR...		active					740					30.03.2023
			39		I			740		DRDK9A1UM5	D:KO 22-00071-013 LSA: Mehrfachanforderung		03.04.2023
			38		I			740		DRDK9A1U80	D:FM 6000561253 LSA: Anpassung Buttons Reihenfolge Icons		28.10.2022
			37		I			740		DRDK9A1RZA	D:KO 22-00071-003 LSA: MF Artikelsuche: Einfache Artikellist		26.09.2022

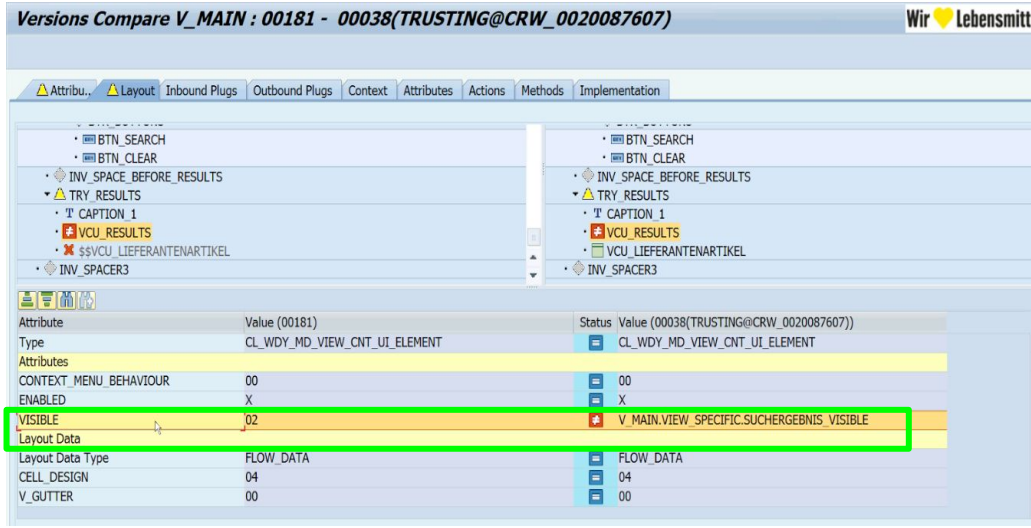


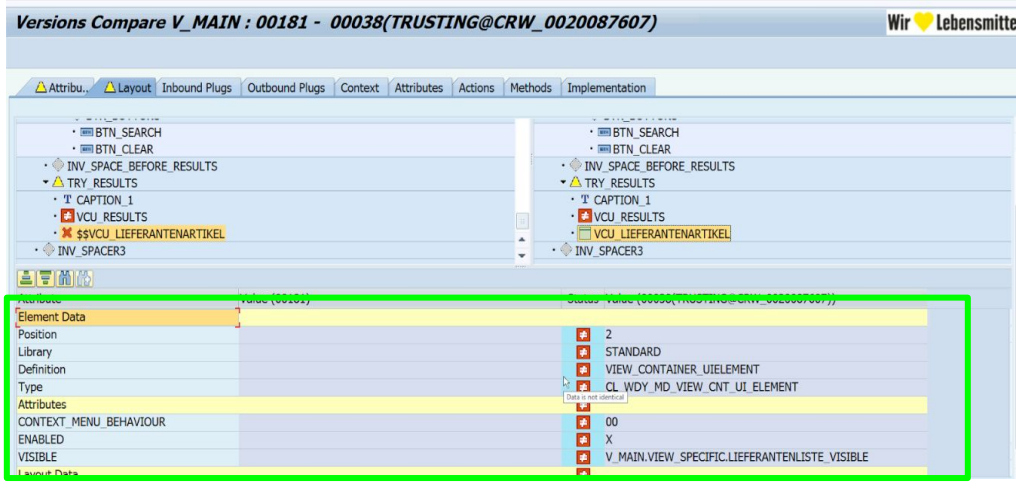
Fr.	S...	Status	Version	Compare	Cat.	Fl...	S...	Rel.	Arc...	Request	Request Text	Proj...	Date
	DR6		active					757					30.11.2022
			181		I			757		DR6K904819	U:KO 22-00071-003 LSA: WE MF - Artikelsuche: Einfache Artike		23.01.2023
			180		I			757		DR6K904375	U:KO 21-01398-000 SMART RF Sprint 22-08 REPO		23.01.2023
			179		I			757		DR6K903864	U:KO 21-01398-000 SMART RF Sprint 22-07 REPO		23.01.2023

Layout changes in Target system and source system will be consider as Customer action.
NOTE: Yellow mark will consider as Changes in two system



Example In Screenshot we have Two Differences one is Visibility check and Other one is Element which is deleted in Target





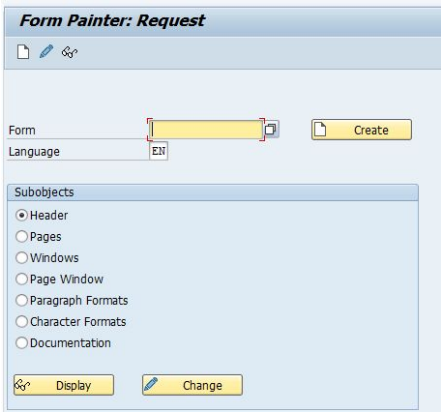
CLAS	IN	TOOL	Vedika
METH	IN	TOOL	Vedika
CINC	IN	TOOL	Vedika
CLSD	IN	TOOL	Vedika
CPRI	IN	TOOL	Vedika
CPRO	IN	TOOL	Vedika
CPUB	IN	TOOL	Vedika
CUAD	IN	Service	Vedika
FORM	IN	Service	Vedika
TEXT	IN	Service	Vedika
ADIR	IN	Service	Vedika
VARI	IN	Service	Vedika
VARX	IN	Service	Vedika

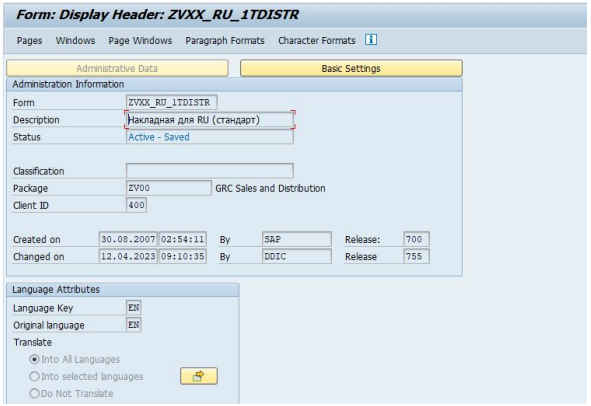
FORM

Object Type: Sapscripts

- Object type **FORM** refers to **SAPscript objects.**
- **TCODE - SE71**
- This object type is categorized under service scope objects as there is less/no support from the smarttool.
- There is no version management option available here. Therefore, this object needs to be dealt differently.

SE71 Screen



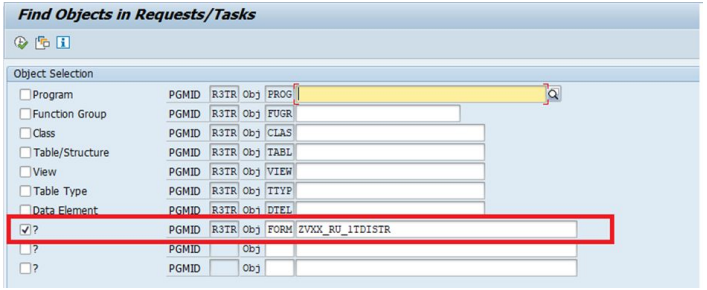


Case 1: New Object

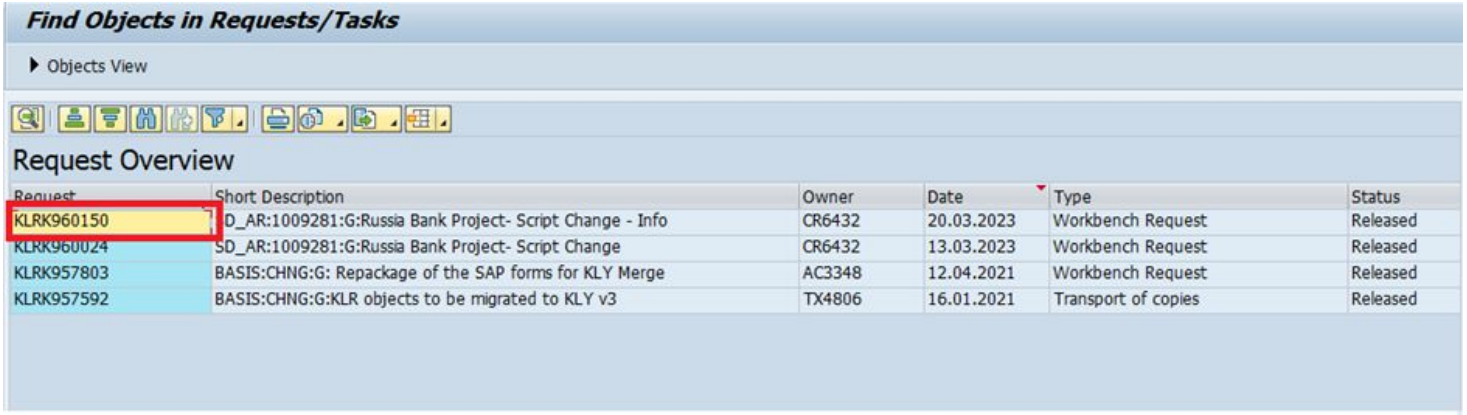
- If the FORM is new to the target system, it can be directly imported from the dashboard in online retrofits or included in the ToC import for offline retrofits.Depending on how each project is conducted.

Case 2: Safe to Import

- If the SAPscript object already exists in the target, then we need to evaluate and decide whether importing the object is safe, meaning that it should not override any changes made specifically for the target.
- We can examine an object's version history using the SE03 transaction in both the source and the target,system and then decide whether or not to import that object.
- **SE03 Screen:** Input object type, object name and execute



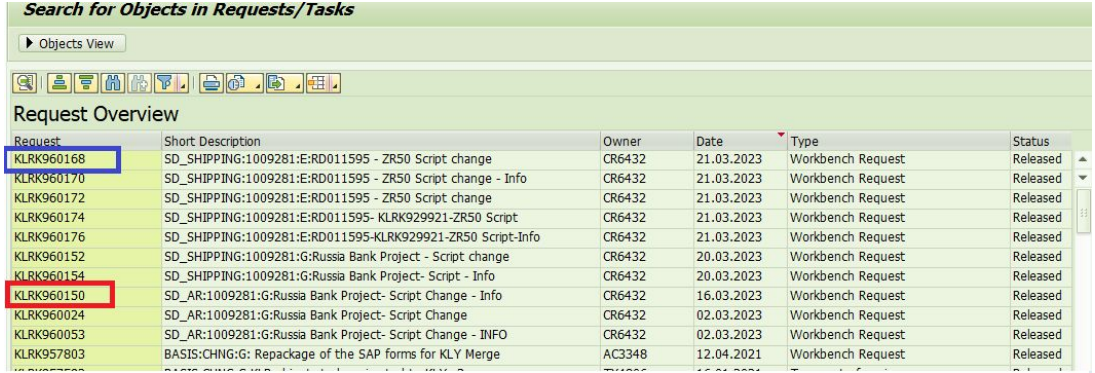
- Sort the **date** field in descending order. Below screenshot is from a target system. The highlighted version is the latest version in target and it is from source system KLR.



Request	Short Description	Owner	Date	Type	Status
KLRK960150	D_AR:1009281:G:Russia Bank Project- Script Change - Info	CR6432	20.03.2023	Workbench Request	Released
KLRK960024	SD_AR:1009281:G:Russia Bank Project- Script Change	CR6432	13.03.2023	Workbench Request	Released
KLRK957803	BASIS:CHNG:G: Repackage of the SAP forms for KLY Merge	AC3348	12.04.2021	Workbench Request	Released
KLRK957592	BASIS:CHNG:G:KLR objects to be migrated to KLY v3	TX4806	16.01.2021	Transport of copies	Released

- Perform the same steps in the source system. After sorting the date field, check the sequence of the latest version from target with the TR in scope for retrofit.

- In the below screenshot, TR KLR****150 is the latest version in target and KLR****174 is the scope TR.



Search for Objects in Requests/Tasks

Objects View

Request Overview

Request	Short Description	Owner	Date	Type	Status
KLRK960168	SD_SHIPPING:1009281:E:RD011595 - ZR50 Script change	CR6432	21.03.2023	Workbench Request	Released
KLRK960170	SD_SHIPPING:1009281:E:RD011595 - ZR50 Script change - Info	CR6432	21.03.2023	Workbench Request	Released
KLRK960172	SD_SHIPPING:1009281:E:RD011595 - ZR50 Script change	CR6432	21.03.2023	Workbench Request	Released
KLRK960174	SD_SHIPPING:1009281:E:RD011595- KLRK929921-ZR50 Script	CR6432	21.03.2023	Workbench Request	Released
KLRK960176	SD_SHIPPING:1009281:E:RD011595-KLRK929921-ZR50 Script-Info	CR6432	21.03.2023	Workbench Request	Released
KLRK960152	SD_SHIPPING:1009281:G:Russia Bank Project - Script change	CR6432	20.03.2023	Workbench Request	Released
KLRK960154	SD_SHIPPING:1009281:G:Russia Bank Project- Script - Info	CR6432	20.03.2023	Workbench Request	Released
KLRK960150	SD_AR:1009281:G:Russia Bank Project- Script Change - Info	CR6432	16.03.2023	Workbench Request	Released
KLRK960024	SD_AR:1009281:G:Russia Bank Project- Script Change	CR6432	02.03.2023	Workbench Request	Released
KLRK960053	SD_AR:1009281:G:Russia Bank Project- Script Change - INFO	CR6432	02.03.2023	Workbench Request	Released
KLRK957803	BASIS:CHNG:G: Repackage of the SAP forms for KLY Merge	AC3348	12.04.2021	Workbench Request	Released

- As there is no target specific changes exists in target system, the latest version of the object can be safely imported via ToC or dashboard into the target.

Case 3: Manual Retrofit

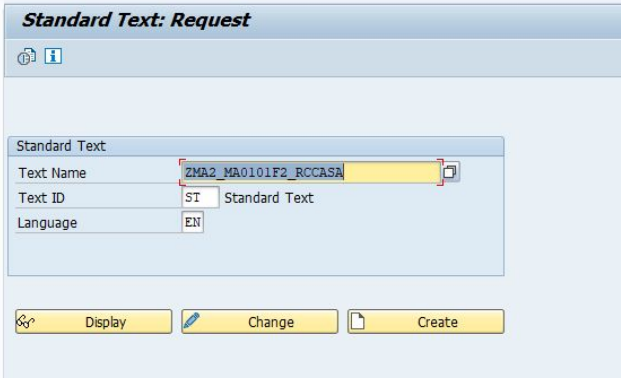
- When a SAPscript object is already existed in target and contains target specific changes(identified in SE03), it would be difficult to retrofit since there is no version comparison accessible to identify which changes to be retained or added in the target.
- In such circumstances, we need to inform the customer about the issue/scenario and based on their feedback, further action can be made.

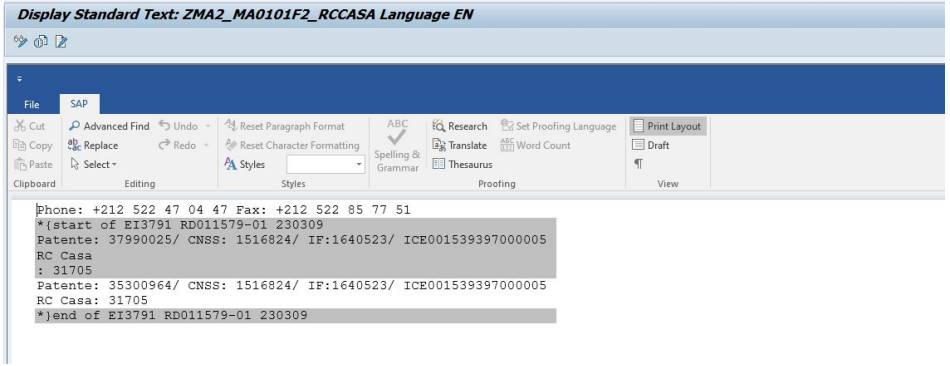
TEXT

Object Type: SAPscript Text

- Object type **TEXT** refers to **SAPscript text** objects**.**
- **TCODE - SO10**
- There is no version management option here. Therefore, this object will be dealt similar to the SAPscript object.

SO10 screen



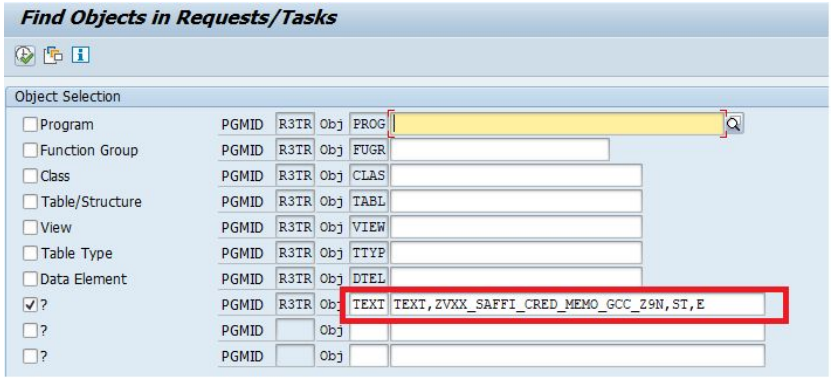


Case 1: New Object

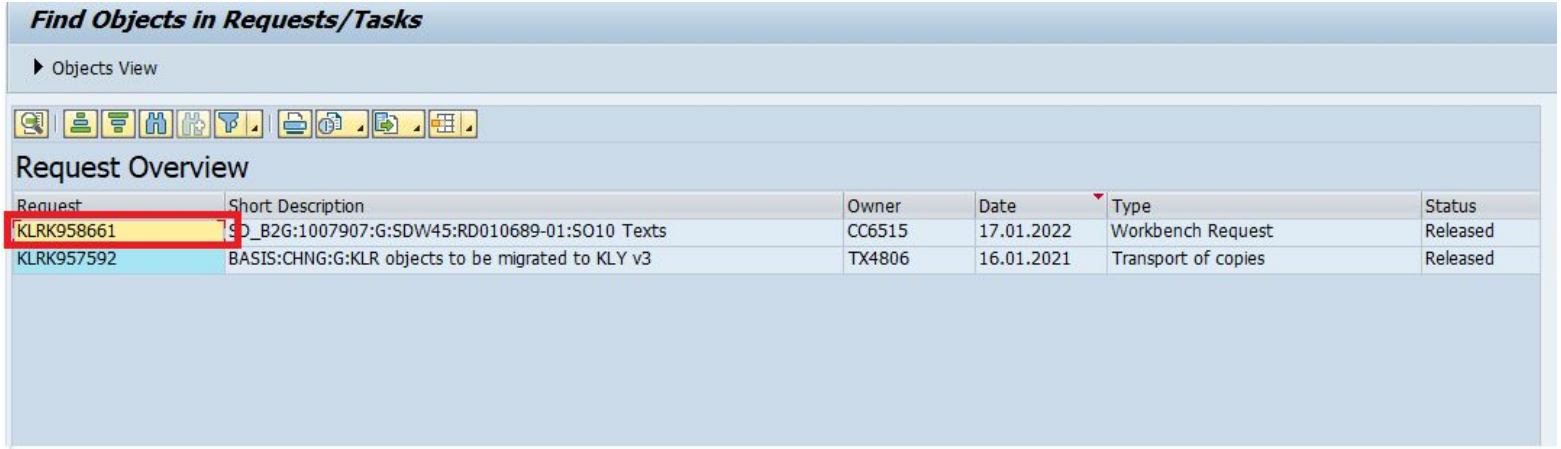
- If the TEXT is new to the target system, it can be directly imported from the dashboard in online retrofits or included in the ToC import for offline retrofits.Depending on how each project is conducted.

Case 2: Safe to Import

- If the sapscript text object already exists in the target, then we need to evaluate and decide whether importing the object is safe, meaning that it should not override any changes made specifically for the target.
- Using the SE03 transaction in both the source and the target,system, we can decide whether or not to import that object.
- **SE03 Screen:** Input object type, object name and execute



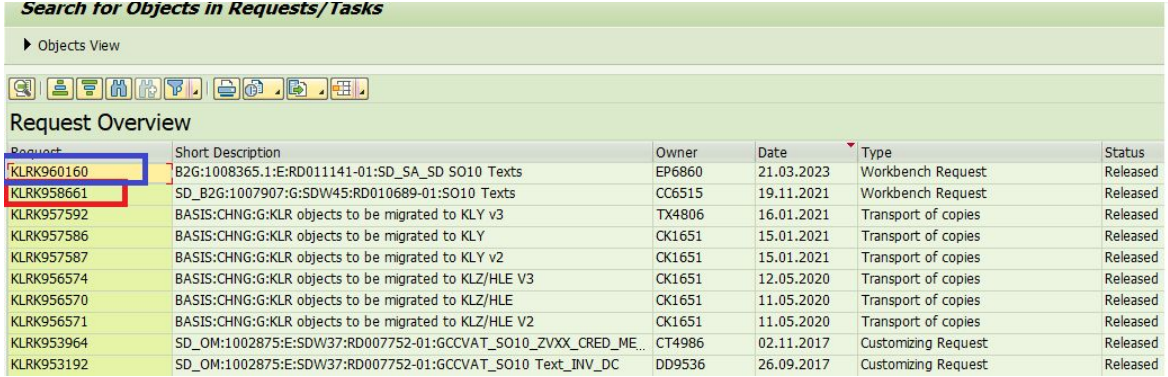
- Sort the date field in the descending order. In the below example, KLR is the source system and TR KLR****661 is the latest version available in target system. This indicates that there is no target specific changes for the object.



Request	Short Description	Owner	Date	Type	Status
KLRK958661	SD_B2G:1007907:G:SDW45:RD010689-01:SO10 Texts	CC6515	17.01.2022	Workbench Request	Released
KLRK957592	BASIS:CHNG:G:KLR objects to be migrated to KLY v3	TX4806	16.01.2021	Transport of copies	Released

- Having confirmed that the target does not have any target specific changes, let's perform the same steps in source system to check the sequence of latest TR in target with the TR in scope to ensure we are not downgrading the version by importing the scope TR.

- In the below screenshot, TR KLR****661 is the latest version in target and KLR****160 is the scope TR.



Search for Objects in Requests/Tasks

Objects View

Request Overview

Request	Short Description	Owner	Date	Type	Status
KLRK960160	B2G:1008365.1:E:RD011141-01:SD_SA_SD SO10 Texts	EP6860	21.03.2023	Workbench Request	Released
KLRK958661	SD_B2G:1007907:G:SDW45:RD010689-01:SO10 Texts	CC6515	19.11.2021	Workbench Request	Released
KLRK957592	BASIS:CHNG:G:KLR objects to be migrated to KLY v3	TX4806	16.01.2021	Transport of copies	Released
KLRK957586	BASIS:CHNG:G:KLR objects to be migrated to KLY	CK1651	15.01.2021	Transport of copies	Released
KLRK957587	BASIS:CHNG:G:KLR objects to be migrated to KLY v2	CK1651	15.01.2021	Transport of copies	Released
KLRK956574	BASIS:CHNG:G:KLR objects to be migrated to KLZ/HLE V3	CK1651	12.05.2020	Transport of copies	Released
KLRK956570	BASIS:CHNG:G:KLR objects to be migrated to KLZ/HLE	CK1651	11.05.2020	Transport of copies	Released
KLRK956571	BASIS:CHNG:G:KLR objects to be migrated to KLZ/HLE V2	CK1651	11.05.2020	Transport of copies	Released
KLRK953964	SD_OM:1002875:E:SDW37:RD007752-01:GCCVAT_SO10_ZVXX_CRED_ME...	CT4986	02.11.2017	Customizing Request	Released
KLRK953192	SD_OM:1002875:E:SDW37:RD007752-01:GCCVAT_SO10 Text_INV_DC	DD9536	26.09.2017	Customizing Request	Released

- As the scope TR is the latest version in source system, it can be safely imported via ToC into the target.

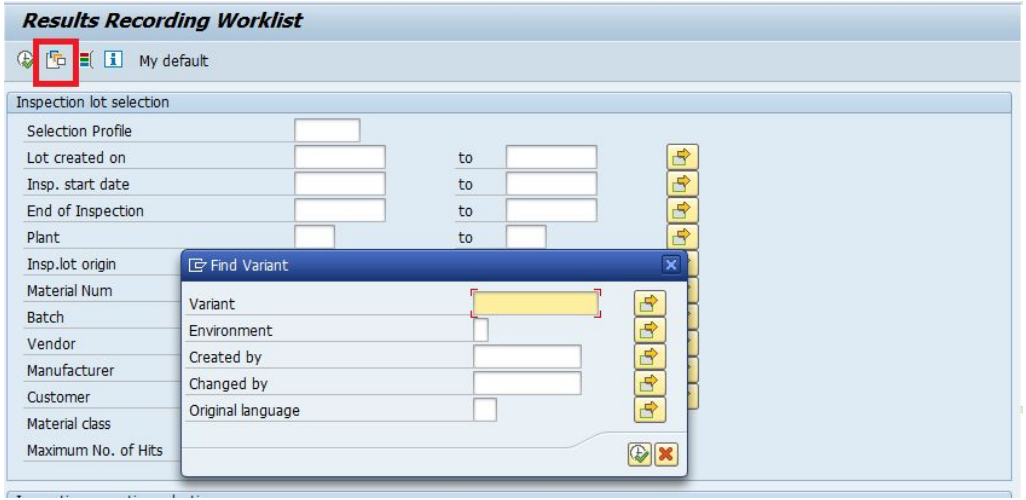
Case 3: Manual Retrofit

- When a SAPscript text object is already existed in target and contains target specific changes(identified in SE03), it would be difficult to retrofit since there is no version comparison accessible to identify which changes to be retained or added in the target.
- In such circumstances, we need to inform the customer about the issue/scenario and based on their feedback, further action can be made.

VARX/VARI

Object Types: VARX - Report Program Application Variant
VARI - Report Program System Variant

- Object type **VARX/VARI** refers to report program variants.
- **TCODE - SE38**
- Execute the report program in SE38. We can see the variants option highlighted in the below screenshot.



Case 1: New Object

- If a variant for a custom program is new to the target system, it can be directly imported from the dashboard in online retrofits or included in the ToC import for offline retrofits.Depending on how each project is conducted.
- If a variant is created for a standard program and it is new to target, we need to take confirmation from the customer for its import into target.

Case 2: Existing Object

- If the variants for an object already exists in target, we need to inform the customer about the issue/scenario and based on their feedback, further action can be made.

ADIR

Object Type: Object Directory Entry

- Object type **ADIR** refers to **Object directory entry.**
- **TCODE -** can be checked in **E071 table**

Case 1: New Object

- When the ADIR entry is new, the object can be imported by ToC.

Case 2: Existing

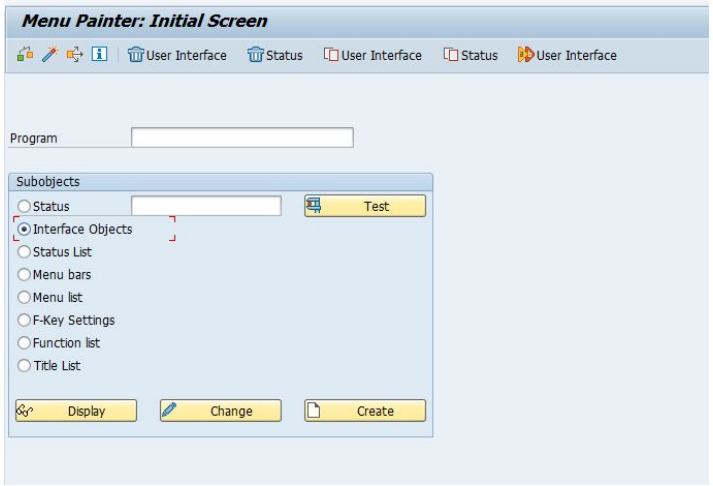
- If ADIR object exists in target, no action would be taken in the target side.It would be marked as **No Action**.

CUAD

Object Type: GUI definition

- Object type **CUAD** refers to **GUI definition/Menu painter** objects.
- **TCODE - SE41**
- This is a version comparison available for this object type.

SE41 Screenshot

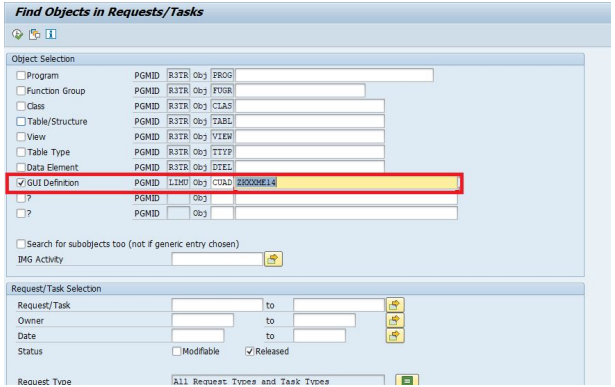


Case 1: New Object

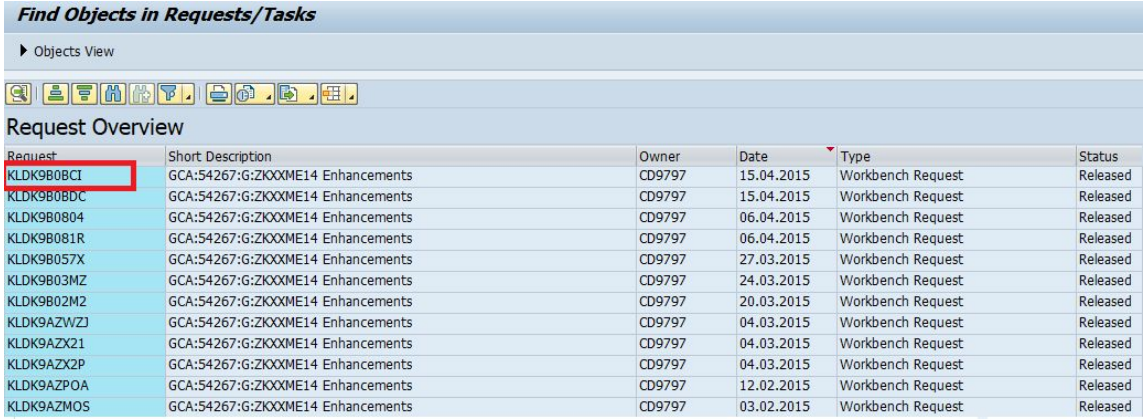
- If the CUAD is new to the target system, it can be directly imported from the dashboard in online retrofits or included in the ToC import in offline retrofits.Depending on how each project is conducted.

Case 2: Safe to Import

- If the CUAD object already exists in the target, then we need to evaluate and decide whether importing the object is safe, meaning that it should not override any changes made specifically for the target.
- Using the SE03 transaction in both the source and the target,system, we can decide whether or not to import that object.
- **SE03 Screen:** Input object type, object name and execute



- Sort the date field in the descending order. In the below example, KLD is the source system and TR KLD****BCI is the latest version available in target system. This indicates that there is no target specific changes for the object. All the changes are from source KLD.



Find Objects in Requests/Tasks

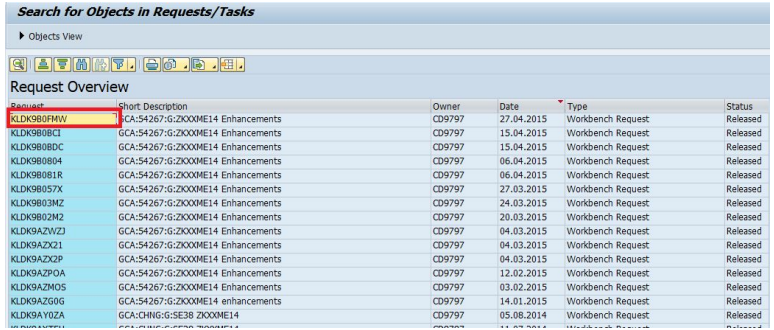
Objects View

Request Overview

Request	Short Description	Owner	Date	Type	Status
KLDK9B0BCI	GCA:54267:G:ZKXXME14 Enhancements	CD9797	15.04.2015	Workbench Request	Released
KLDK9B0BDC	GCA:54267:G:ZKXXME14 Enhancements	CD9797	15.04.2015	Workbench Request	Released
KLDK9B0804	GCA:54267:G:ZKXXME14 Enhancements	CD9797	06.04.2015	Workbench Request	Released
KLDK9B081R	GCA:54267:G:ZKXXME14 Enhancements	CD9797	06.04.2015	Workbench Request	Released
KLDK9B057X	GCA:54267:G:ZKXXME14 Enhancements	CD9797	27.03.2015	Workbench Request	Released
KLDK9B03MZ	GCA:54267:G:ZKXXME14 Enhancements	CD9797	24.03.2015	Workbench Request	Released
KLDK9B02M2	GCA:54267:G:ZKXXME14 Enhancements	CD9797	20.03.2015	Workbench Request	Released
KLDK9AZWZJ	GCA:54267:G:ZKXXME14 Enhancements	CD9797	04.03.2015	Workbench Request	Released
KLDK9AZX21	GCA:54267:G:ZKXXME14 Enhancements	CD9797	04.03.2015	Workbench Request	Released
KLDK9AZX2P	GCA:54267:G:ZKXXME14 Enhancements	CD9797	04.03.2015	Workbench Request	Released
KLDK9AZPOA	GCA:54267:G:ZKXXME14 Enhancements	CD9797	12.02.2015	Workbench Request	Released
KLDK9AZMOS	GCA:54267:G:ZKXXME14 Enhancements	CD9797	03.02.2015	Workbench Request	Released

- Having confirmed that there is no target specific changes, let’s perform the same steps in source system to check the sequence of latest TR in target with the TR in scope to ensure we are not downgrading the version by importing the scope TR.

- In the below screenshot, TR KLD****BCI is the latest version in target and KLD****FMW is the scope TR.



Search for Objects in Requests/Tasks

Objects View

Request Overview

Request	Short Description	Owner	Date	Type	Status
KLDK9B0FMW	GCA:54267:G:ZKXXME14 Enhancements	CD9797	27.04.2015	Workbench Request	Released
KLDK9B0BCI	GCA:54267:G:ZKXXME14 Enhancements	CD9797	15.04.2015	Workbench Request	Released
KLDK9B0BDC	GCA:54267:G:ZKXXME14 Enhancements	CD9797	15.04.2015	Workbench Request	Released
KLDK9B0804	GCA:54267:G:ZKXXME14 Enhancements	CD9797	06.04.2015	Workbench Request	Released
KLDK9B081R	GCA:54267:G:ZKXXME14 Enhancements	CD9797	06.04.2015	Workbench Request	Released
KLDK9B057X	GCA:54267:G:ZKXXME14 Enhancements	CD9797	27.03.2015	Workbench Request	Released
KLDK9B03MZ	GCA:54267:G:ZKXXME14 Enhancements	CD9797	24.03.2015	Workbench Request	Released
KLDK9B02M2	GCA:54267:G:ZKXXME14 Enhancements	CD9797	20.03.2015	Workbench Request	Released
KLDK9AZWZJ	GCA:54267:G:ZKXXME14 Enhancements	CD9797	04.03.2015	Workbench Request	Released
KLDK9AZX21	GCA:54267:G:ZKXXME14 Enhancements	CD9797	04.03.2015	Workbench Request	Released
KLDK9AZX2P	GCA:54267:G:ZKXXME14 Enhancements	CD9797	04.03.2015	Workbench Request	Released
KLDK9AZPOA	GCA:54267:G:ZKXXME14 Enhancements	CD9797	12.02.2015	Workbench Request	Released
KLDK9AZMOS	GCA:54267:G:ZKXXME14 Enhancements	CD9797	03.02.2015	Workbench Request	Released
KLDK9AZG0G	GCA:54267:G:ZKXXME14 enhancements	CD9797	14.01.2015	Workbench Request	Released
KLDK9AY0ZA	GCA:CHNG:G:SE38 ZKXXME14	CD9797	05.08.2014	Workbench Request	Released

- As the scope TR is the latest version in source system, it can be safely imported via ToC into the target.

Case 3: Manual Retrofit

- When a CUAD object is already existed in target and contains target specific changes, then the object needs to be handled/retrofitted manually using se41 tcode.

CLAS

Object Type: Class Objects

Sub categories: CINC, CLSD, CPRI, CPRO, CPUB, METH

- Object type **CLAS** refers to **Class objects.**
- **TCODE - SE24**
- This object type is categorized under tool scope objects and have tool reports available for the retrofit which contains source_parsed code, target code and merged code.
- Main object (R3TR) **CLAS** would contain below sub objects and all of them will follow the same process.

CINC	Class Include
CLSD	Class Definition
CPUB	Public Header
CPRI	Private Header
CPRO	Protected Header
METH	Methods

Case 1: New Object

- If the CLAS is new to the target system, it can be directly imported from the dashboard in online retrofits or included in the ToC import in offline retrofits. Depending on how each project is conducted.
- If the scope TR contains NEW R3TR CLAS, the entire object—including the header section, definitions, and methods—will be imported to the target during import.
- If the scope TR has an entry of sub objects of a class like LIMU METH and it is new, then, only that specific method will be imported into the target.
- Post import, source_parsed version of the code(with relevant tool rules) needs to be added into the target.

Case 2: Existing Object

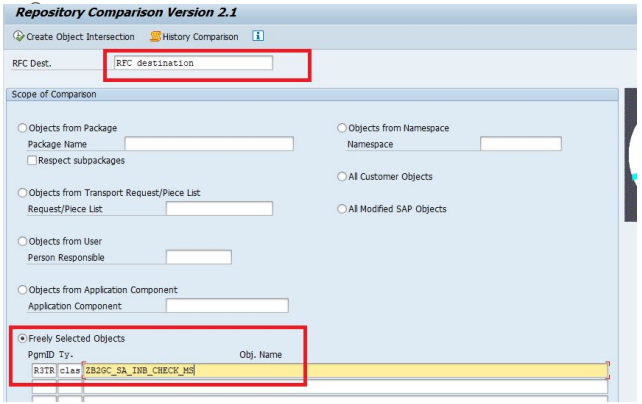
- If the CLAS object is already existed in target, then we need to retrofit it using the tool report containing source_parsed, merged and target code either in MELD application(Offline retrofit) or WEBUI(Online retrofit).
- If scope TR contains LIMU objects like below, then we need to retrofit only that specific object.

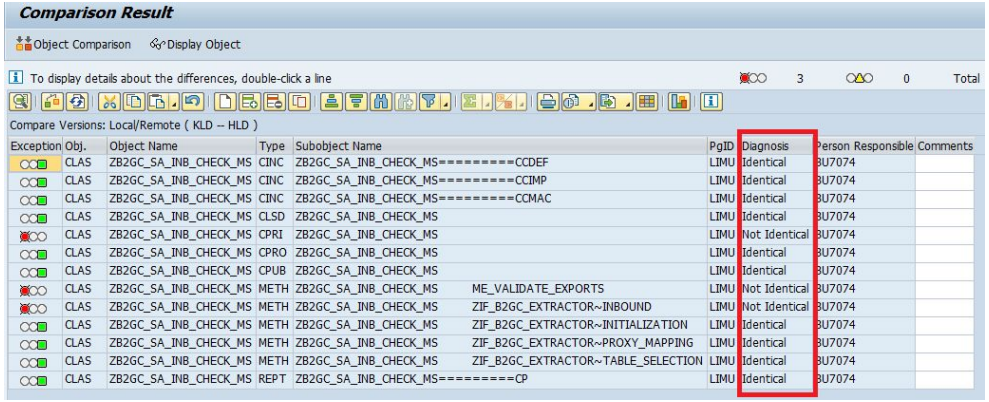
LIMU	CPRI	class_name
LIMU	METH	class_name method_name

- If scope TR contains R3TR CLAS object like below, then we need to retrofit all the subobjects of that class.

R3TR	CLAS	class_name

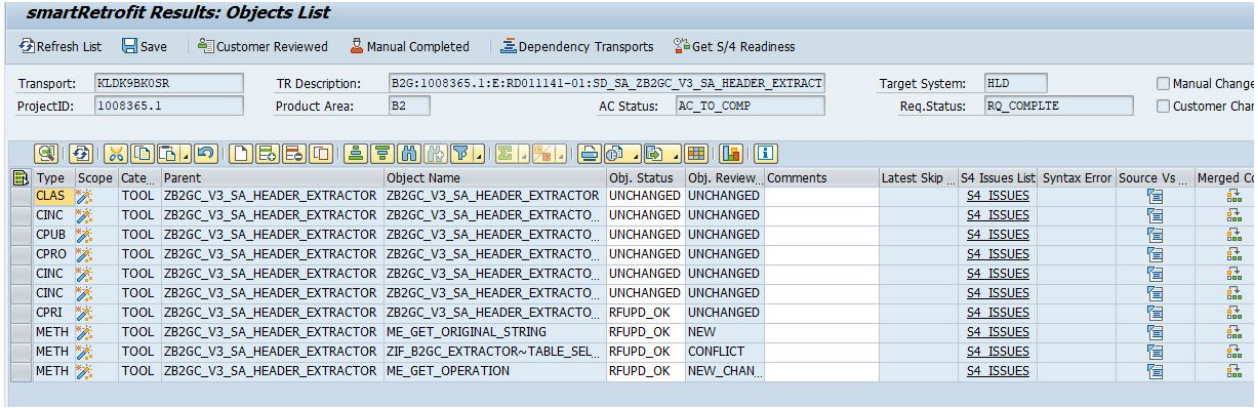
- In offline retrofit, whenever R3TR class is in scope, we can filter the scoping sheet on main object column to get all the subobjects of that class. Also, using **RSSYSCOMP** report, we can get all the sub objects of a Class.It also provides information about object status like identical, conflicts, not in source/target by comparing between source and target systems.





- All the objects displayed in the output screen of rssyscomp report should be fixed as part of R3TR Class retrofit.

- In online retrofit, the dashboard will display all the subobject to be retrofitted for an R3TR entry except REPT object which needs to be checked manually.



Type	Scope	Cate...	Parent	Object Name	Obj. Status	Obj. Review...	Comments	Latest Skip ...	S4 Issues List	Syntax Error	Source Vs ...	Merged Co
CLAS		TOOL	ZB2GC_V3_SA_HEADER_EXTRACTOR	ZB2GC_V3_SA_HEADER_EXTRACTOR	UNCHANGED	UNCHANGED			S4_ISSUES			
CINC		TOOL	ZB2GC_V3_SA_HEADER_EXTRACTOR	ZB2GC_V3_SA_HEADER_EXTRACTO...	UNCHANGED	UNCHANGED			S4_ISSUES			
CPUB		TOOL	ZB2GC_V3_SA_HEADER_EXTRACTOR	ZB2GC_V3_SA_HEADER_EXTRACTO...	UNCHANGED	UNCHANGED			S4_ISSUES			
CPRO		TOOL	ZB2GC_V3_SA_HEADER_EXTRACTOR	ZB2GC_V3_SA_HEADER_EXTRACTO...	UNCHANGED	UNCHANGED			S4_ISSUES			
CINC		TOOL	ZB2GC_V3_SA_HEADER_EXTRACTOR	ZB2GC_V3_SA_HEADER_EXTRACTO...	UNCHANGED	UNCHANGED			S4_ISSUES			
CINC		TOOL	ZB2GC_V3_SA_HEADER_EXTRACTOR	ZB2GC_V3_SA_HEADER_EXTRACTO...	UNCHANGED	UNCHANGED			S4_ISSUES			
CPRI		TOOL	ZB2GC_V3_SA_HEADER_EXTRACTOR	ZB2GC_V3_SA_HEADER_EXTRACTO...	RFUPD_OK	UNCHANGED			S4_ISSUES			
METH		TOOL	ZB2GC_V3_SA_HEADER_EXTRACTOR	ME_GET_ORIGINAL_STRING	RFUPD_OK	NEW			S4_ISSUES			
METH		TOOL	ZB2GC_V3_SA_HEADER_EXTRACTOR	ZIF_B2GC_EXTRACTOR~TABLE_SEL...	RFUPD_OK	CONFLICT			S4_ISSUES			
METH		TOOL	ZB2GC_V3_SA_HEADER_EXTRACTOR	ME_GET_OPERATION	RFUPD_OK	NEW_CHAN...			S4_ISSUES			

There are 3 merge cases available.

- No action would be necessary for all the identical/unchanged cases notified in meld or webui.
- In Automerge case, the code changes are already added into the merged file by the tool and the retrofit developer needs to validate the merge code before copying the code into the target

- In conflict cases, developer needs to analyse the code section having the conflict between source and target and decide upon the code to be merged.
- Version management of the object can also be used to decide the changes to be retained in the merged section.
- The developer can understand changes made in each TR and correct the object by comparing the active version with other versions in the target system or the scope TR version with various lower versions of an object in the source system.

Different code conflict cases are:

- If the source system's code changes are the cause of the code conflict and there are no target-specific changes for that object, the developer must keep the source changes in the merged section.
- If the code conflict is due to the target specific changes, then, the developer must retain the target code.
- If the code conflict is due to change in source as well as target specific code change - mark it for customer action. Further action will be taken based on the customer's feedback.

- In the below screenshot, there is a conflict wherein both source and target code have different conditions after matnr value. The code conflict needs to be analyzed to understand which version of the code to be retained.

```
    CLEAR e_event_track.
    UNASSIGN <fs_xvbap>.
*" Line item where there is change in line item rejection
    LOOP AT ct_xvbap
      ASSIGNING <fs_xvbap>
      WHERE abgru IS NOT INITIAL.
      UNASSIGN <fs_yvbap>.
      READ TABLE ct_yvbap
        ASSIGNING <fs_yvbap>
        WITH KEY matnr = <fs_xvbap>-matnr
          werks = <fs_xvbap>-werks.
      IF  sy-subrc EQ 0
      AND <fs_xvbap> IS ASSIGNED.
*" If no change in line item rejection continue
        IF  <fs_xvbap>-abgru EQ <fs_yvbap>-abgru.
          CONTINUE.
```

```
    ASSIGNING <fs_xvbap>
    WHERE abgru IS NOT INITIAL.
    UNASSIGN <fs_yvbap>.
    READ TABLE ct_yvbap
      ASSIGNING <fs_yvbap>
      WITH KEY matnr = <fs_xvbap>-matnr
<<<<<<< HEAD
          posnr = <fs_xvbap>-posnr.
=======
          werks = <fs_xvbap>-werks.
>>>>>>> 4f934b9392a547b20203e973b958bc6f7df9eb1f
    IF  sy-subrc EQ 0
    AND <fs_xvbap> IS ASSIGNED.
*" If no change in line item rejection continue
      IF  <fs_xvbap>-abgru EQ <fs_yvbap>-abgru.
        CONTINUE.
```

```
    CLEAR e_event_track.
    UNASSIGN <fs_xvbap>.
*" Line item where there is change in line item rejection
    LOOP AT ct_xvbap
      ASSIGNING <fs_xvbap>
      WHERE abgru IS NOT INITIAL.
      UNASSIGN <fs_yvbap>.
      READ TABLE ct_yvbap
        ASSIGNING <fs_yvbap>
        WITH KEY matnr = <fs_xvbap>-matnr
          posnr = <fs_xvbap>-posnr.
    IF  sy-subrc EQ 0
    AND <fs_xvbap> IS ASSIGNED.
*" If no change in line item rejection continue
      IF  <fs_xvbap>-abgru EQ <fs_yvbap>-abgru.
        CONTINUE.
```

- As there is a change in source where werks condition is newly added in source version and even in target, posnr was added exclusively in target by customer. We need to get customer confirmation on which code to be retained om the target. Based on customer inputs, we can retrofit the changes.

Case 3: Hybrid Object (Existing object with few new sub-objects)

- This a special category where the scope TR has a R3TR CLAS object having both already existed and new sub-objects. The new METHs can be imported via ToC in offline retrofit.
- But, it is recommended to start with the header sections(public, private, protected) of the class to avoid any accidental errors.
- By retrofitting and activating the header section would create or delete required class methods as per changes done as part of the retrofit. In case of deletion of METH, check whether methods are present in scoping sheet with deletion flag.
- Once the new methods are created in the class level, source_parsed code must be copied to the target.
- In an online retrofit, new methods in an R3TR CLAS should never be handled until the header actions have been fixed and the CLAS has been activated in the SAP GUI.
- Retrofitting new methods prior to the header section would result in empty methods having no implementation.

Remediation

- Once the retrofit is completed for all the objects in scope, manual remediation checks must be performed to identify any syntax errors, runtime errors, HANA code compliance issues or S4 HANA issues.

Vishnupriya

1	Object T	Sco	Scope	SPOC
85	TYPE	IN	Service	Vishnupriya
86	TTYD	IN	Service	Vishnupriya
87	TTYP	IN	Service	Vishnupriya
88	TRAN	IN	Service	Vishnupriya
89	SOBJ	IN	Service	Vishnupriya
90	INDX	IN	Service	Vishnupriya

TTYP/TTYD

Object Type: Table Type / Table Type Definition

- A table type is a SE11 Data dictionary, Data element object which is used to describe structure and functions of an internal table in an ABAP program.
- A table type definition describes the basic attributes of a table type.
- For PRE-EXISTING TTYP/TTYD objects in the system: A utility program RSSYSCOMP, can be used to compare the active version of target with the source TR version from the source using a RFC (remote function call triggered from target to source or vice-versa). Maximum 8 objects can be executed at once, using this utility.
- Steps to execute the utility program:

1. Open the utility program in SE38 and execute.
2. Give the RFC Dest. and select “Freely selected Objects” radio button

   PgmID: R3TR

   Ty. : TTYP

   Obj. Name: Object name

Repository Comparison Version 2.1

Create Object Intersection | History Comparison

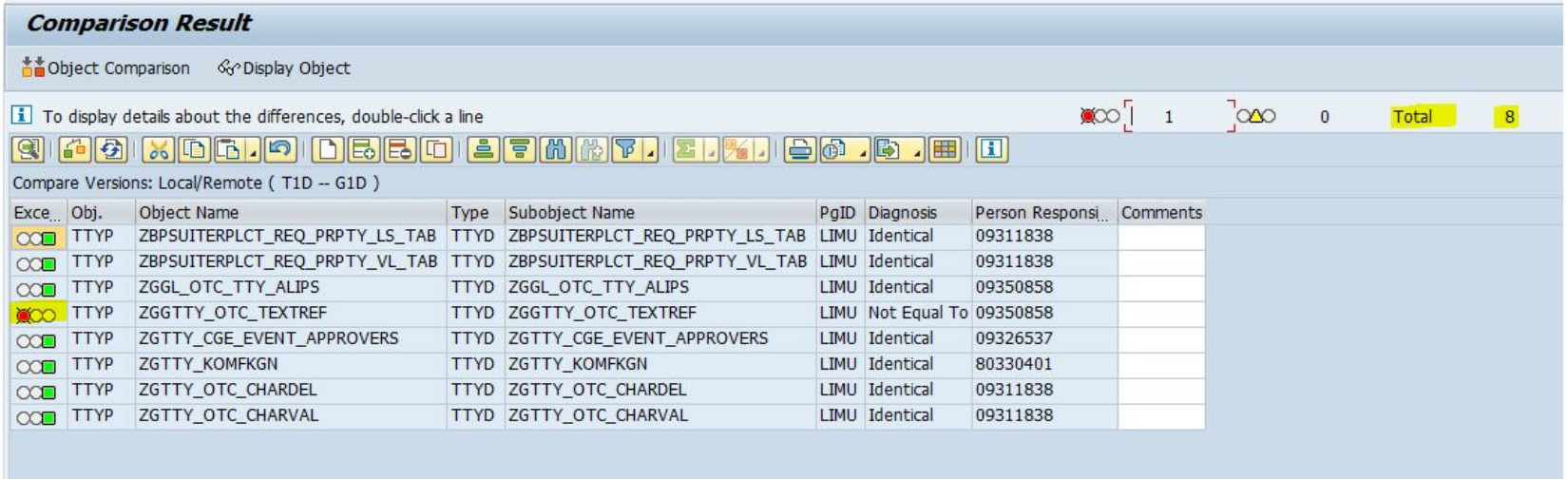
RFC Dest. Smartshift_G1D

Scope of Comparison

- Objects from Package
  - Package Name
  - Respect subpackages
- Objects from Transport Request/Piece List
  - Request/Piece List
- Objects from User
  - Person Responsible
- Objects from Application Component
  - Application Component
- Freely Selected Objects
- Objects from Namespace
  - Namespace
- All Customer Objects
- All Modified SAP Objects

PgmID	Ty.	Obj. Name
R3TR	TTYP	ZBPSUITERPLCT_REQ_PRPTY_LS_TAB
R3TR	TTYP	ZBPSUITERPLCT_REQ_PRPTY_VL_TAB
R3TR	TTYP	ZGTTY_OTC_CHARDEL
R3TR	TTYP	ZGTTY_OTC_CHARVAL
R3TR	TTYP	ZGGL_OTC_TTY_ALIPS
R3TR	TTYP	ZGGTTY_OTC_TEXTREF

3. Further, click on “Create Object Intersection” and on the NEXT screen click on “Version Comparison”.



Comparison Result

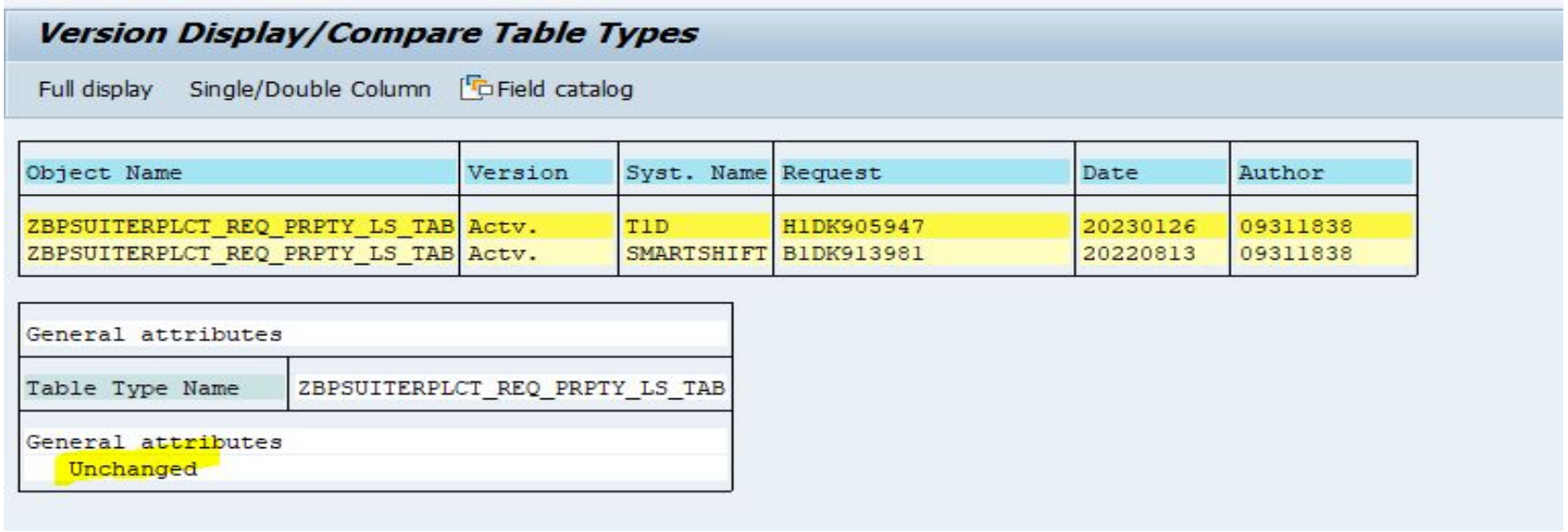
Object Comparison Display Object

To display details about the differences, double-click a line 1 0 Total 8

Compare Versions: Local/Remote (T1D -- G1D)

Exce...	Obj.	Object Name	Type	Subobject Name	PgID	Diagnosis	Person Responsi...	Comments
	TTYP	ZBPSUITERPLCT_REQ_PRPTY_LS_TAB	TTYD	ZBPSUITERPLCT_REQ_PRPTY_LS_TAB	LIMU	Identical	09311838	
	TTYP	ZBPSUITERPLCT_REQ_PRPTY_VL_TAB	TTYD	ZBPSUITERPLCT_REQ_PRPTY_VL_TAB	LIMU	Identical	09311838	
	TTYP	ZGGL_OTC_TTY_ALIPS	TTYD	ZGGL_OTC_TTY_ALIPS	LIMU	Identical	09350858	
	TTYP	ZGGTTY_OTC_TEXTREF	TTYD	ZGGTTY_OTC_TEXTREF	LIMU	Not Equal To	09350858	
	TTYP	ZGTTY_CGE_EVENT_APPROVERS	TTYD	ZGTTY_CGE_EVENT_APPROVERS	LIMU	Identical	09326537	
	TTYP	ZGTTY_KOMFKGN	TTYD	ZGTTY_KOMFKGN	LIMU	Identical	80330401	
	TTYP	ZGTTY_OTC_CHARDEL	TTYD	ZGTTY_OTC_CHARDEL	LIMU	Identical	09311838	
	TTYP	ZGTTY_OTC_CHARVAL	TTYD	ZGTTY_OTC_CHARVAL	LIMU	Identical	09311838	

4. Here, GREEN color represents that the objects are IDENTICAL between source and target. Double click the object name and further click on DELTA DISPLAY option at the top, to find whether the objects are Identical or NOT.



Version Display/Compare Table Types

Full display Single/Double Column Field catalog

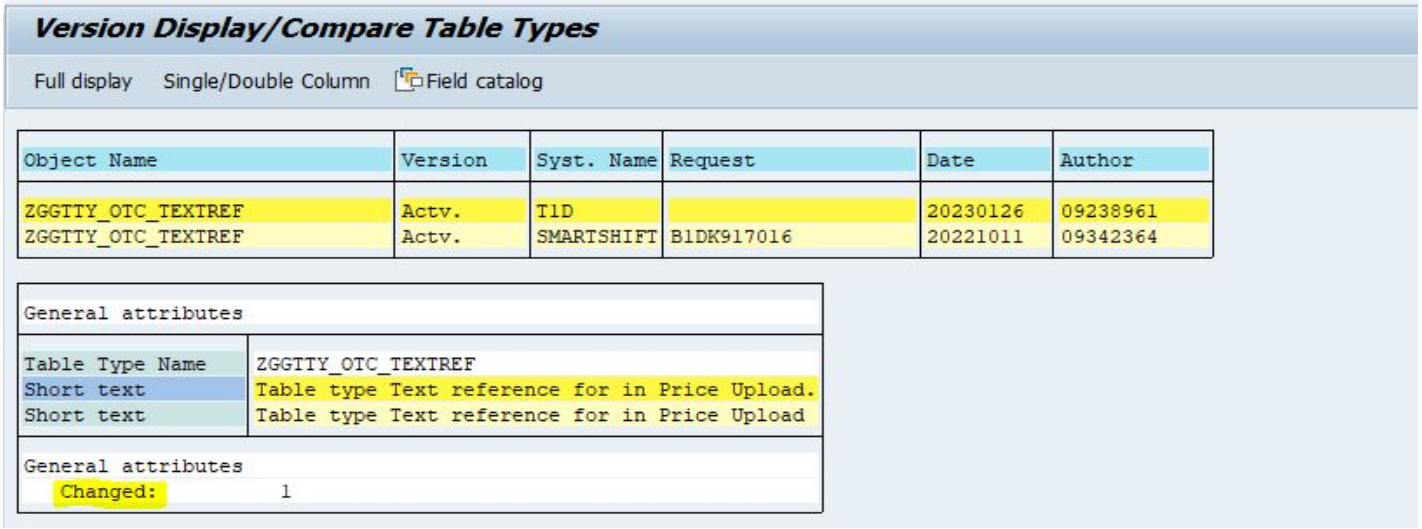
Object Name	Version	Syst. Name	Request	Date	Author
ZBPSUITERPLCT_REQ_PRPTY_LS_TAB	Actv.	T1D	H1DK905947	20230126	09311838
ZBPSUITERPLCT_REQ_PRPTY_LS_TAB	Actv.	SMARTSHIFT	B1DK913981	20220813	09311838

General attributes	
Table Type Name	ZBPSUITERPLCT_REQ_PRPTY_LS_TAB

General attributes
Unchanged

Here, LIGHT yellow color refers to Source and DARK yellow color refers to target.

5. RED color represents that there is difference between the objects in source and target. Further, double click on the object name and click on DELTA DISPLAY at the top to visualize the difference between source and target.



Version Display/Compare Table Types

Full display | Single/Double Column | Field catalog

Object Name	Version	Syst. Name	Request	Date	Author
ZGGTTY_OTC_TEXTREF	Actv.	T1D		20230126	09238961
ZGGTTY_OTC_TEXTREF	Actv.	SMARTSHIFT	B1DK917016	20221011	09342364

General attributes

Table Type Name	ZGGTTY_OTC_TEXTREF
Short text	Table type Text reference for in Price Upload.
Short text	Table type Text reference for in Price Upload

General attributes

Changed: 1

6. **LIMITATIONS** of using the utility program RSSYSCOMP:

This program always compares the active version of target to the active version of the source. So, **if the scope TR is not the active TR in source, then one should not use rssyscomp utility program for comparison.**

Example: There might be additional TR's on the top of scope TR in source, which might contain updated code changes. In that case, if one uses the utility program for source-target comparison, then they might visualize conflict in the code.

Points to cross check before proceeding with any changes in the target system: Whether the particular code/ change is a source change OR a target change

1. Source change: If the piece of code is newly introduced in the source code.Compare the latest TR in source with the below versions to understand if that is a source change or not.
2. Target change: If the piece of code is newly introduced in the target code.Compare the latest TR in target with the below versions to understand if that is a target change or not.
3. Always bring the SCOPE TR source changes to target, **without hampering the target changes**.

- If SCOPE TR is not the latest TR in source, then we shouldn't use the RSSYSCOMP utility program. Instead, one should try doing the version comparison on the system level.

  Steps:

  SE11 -> Data type (put the table type name) -> Utilities -> Versions -> Version Management -> Fetch Remote version -> RFC Destination and execute

  Further, select the versions from source and target to compare and understand the differences. Accordingly, proceed with the changes.

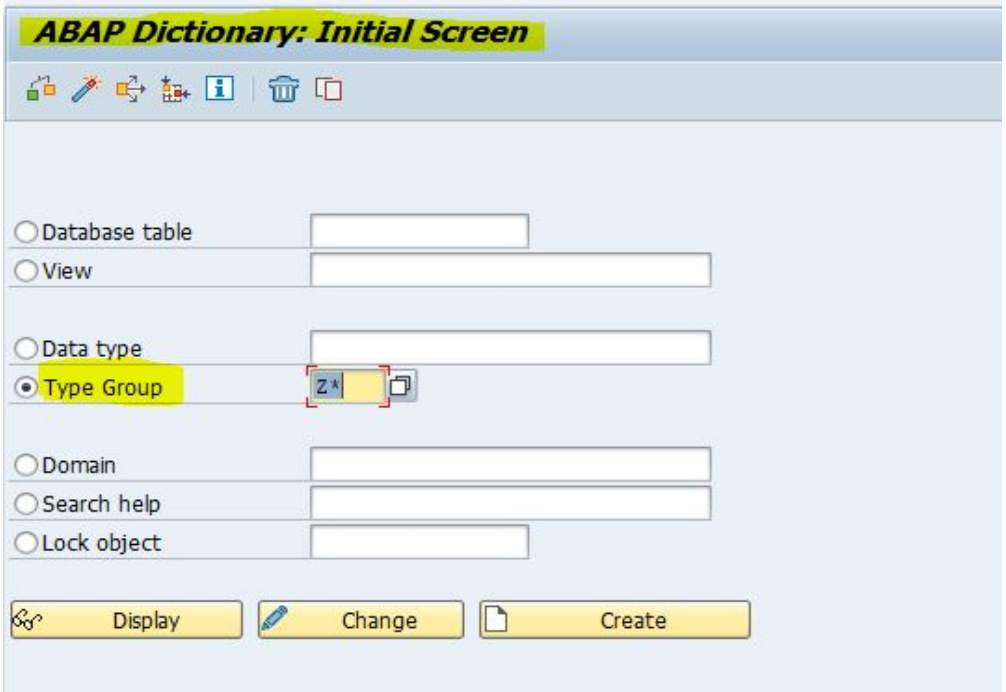
- ## NEW OBJECTS:

  Objects that are not existing in the system and are NEW, can be imported via ToC.

TYPE

Object type: Type Group

- A type group is a SE11 Data Dictionary object and an ABAP program, that is initiated by the statement **TYPE-POOL which contains ABAP statements for defining globally visible data types, constants, and macros**.



- For PRE-EXISTING TYPE objects in the system: A utility program RSSYSCOMP, can be used to compare the active version of target with the source TR version from the source using a RFC (remote function call triggered from target to source or vice-versa). Maximum 8 objects can be executed at once, using this utility.
- Steps to execute the utility program:

1. Open the utility program in SE38 and execute.
2. Give the RFC Dest. and select “Freely selected Objects” radio button

   PgmID: R3TR

   Ty. : TYPE

   Obj. Name: Object name

3. Further, click on “Create Object Intersection” and on the NEXT screen click on “Version Comparison”.

4. Here, GREEN color represents that the objects are IDENTICAL between source and target. Double click the object name and further click on DELTA DISPLAY option at the top, to find whether the objects are Identical or NOT.

Here, LIGHT yellow color refers to Source and DARK yellow color refers to target.

5. RED color represents that there is difference between the objects in source and target. Further, double click on the object name and click on DELTA DISPLAY at the top to visualize the difference between source and target.

6. **LIMITATIONS** of using the utility program RSSYSCOMP:

   This program always compares the active version of target to the active version of the source. So, **if the scope TR is not the active TR in source, then one should not use rssyscomp utility program for comparison.**

   Example: There might be additional TR's on the top of scope TR in source, which might contain updated code changes. In that case, if one uses the utility program for source-target comparison, then they might visualize conflict in the code.

- If SCOPE TR is not the latest TR in source, then we shouldn't use the RSSYSCOMP utility program. Instead, one should try doing the version comparison on the system level.

  Steps:

  SE11 -> Type Group radiobutton (put the type group name) -> Utilities -> Versions -> Version Management -> Fetch Remote version -> RFC Destination and execute

  Further, select the versions from source and target to compare and understand the differences. Accordingly, proceed with the changes.

Points to cross check before proceeding with any changes in the target system: Whether the particular code/ change is a source change OR a target change

1. Source change: If the piece of code is newly introduced in the source code.Compare the latest TR in source with the below versions to understand if that is a source change or not.
2. Target change: If the piece of code is newly introduced in the target code.Compare the latest TR in target with the below versions to understand if that is a target change or not.
3. Always bring the SCOPE TR source changes to target, **without hampering the target changes**.

- ## NEW OBJECTS:

  Objects that are not existing in the system and are NEW, can be imported via ToC.

INDX

Object type: Table Index

- Indexes are used to increase the efficiency of data fetch from database table. it is done on the basis of maping between index of original table to the corresponding column entry value.
- For PRE-EXISTING INDEX objects in the system: Utility program RSSYSCOMP, can be used to compare the active version of target with the source TR version from the source using a RFC (remote function call triggered from target to source or vice-versa). Maximum 8 objects can be executed at once, using this utility.
- Steps to execute the utility program:

1. Open the utility program in SE38 and execute.
2. Give the RFC Dest. and select “Freely selected Objects” radio button

    PgmID: LIMU

    Ty. : INDX

    Obj. Name: Object name

3. Further, click on “Create Object Intersection” and on the NEXT screen click on “Version Comparison”.

4. Here, GREEN color represents that the objects are IDENTICAL between source and target. Double click the object name and further click on DELTA DISPLAY option at the top, to find whether the objects are Identical or NOT.

Here, LIGHT yellow color refers to Source and DARK yellow color refers to target.

5. RED color represents that there is difference between the objects in source and target. Further, double click on the object name and click on DELTA DISPLAY at the top to visualize the difference between source and target.

6. **LIMITATIONS** of using the utility program RSSYSCOMP:

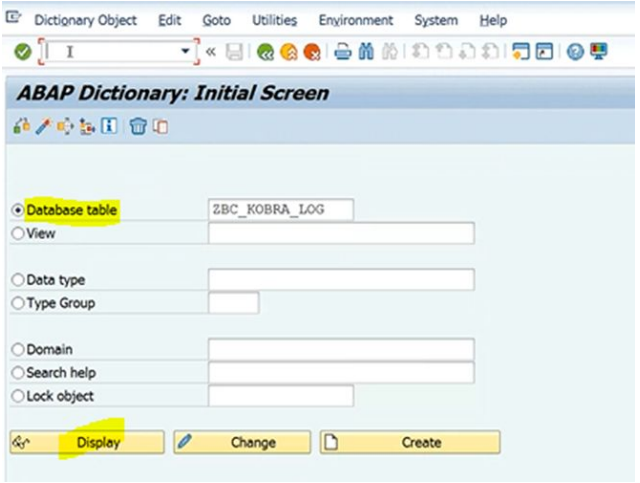
   This program always compares the active version of target to the active version of the source. So, **if the scope TR is not the active TR in source, then one should not use rssyscomp utility program for comparison.**

- If SCOPE TR is not the latest TR in source, then one should try doing the version comparison on the system level to understand the difference between the two systems:.

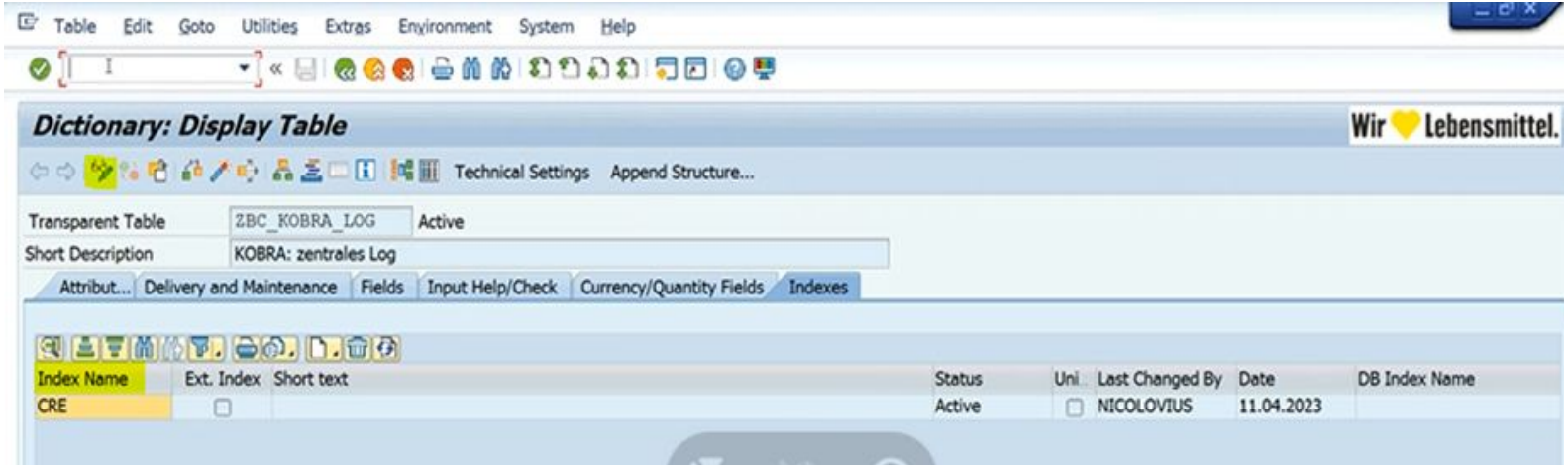
Steps:

A. SE11 -> put TABLE name in Database table -> Display -> Indexes in the toolbar (OR)

SE11 -> put the INDEX name in the Database table -> DISPLAY.

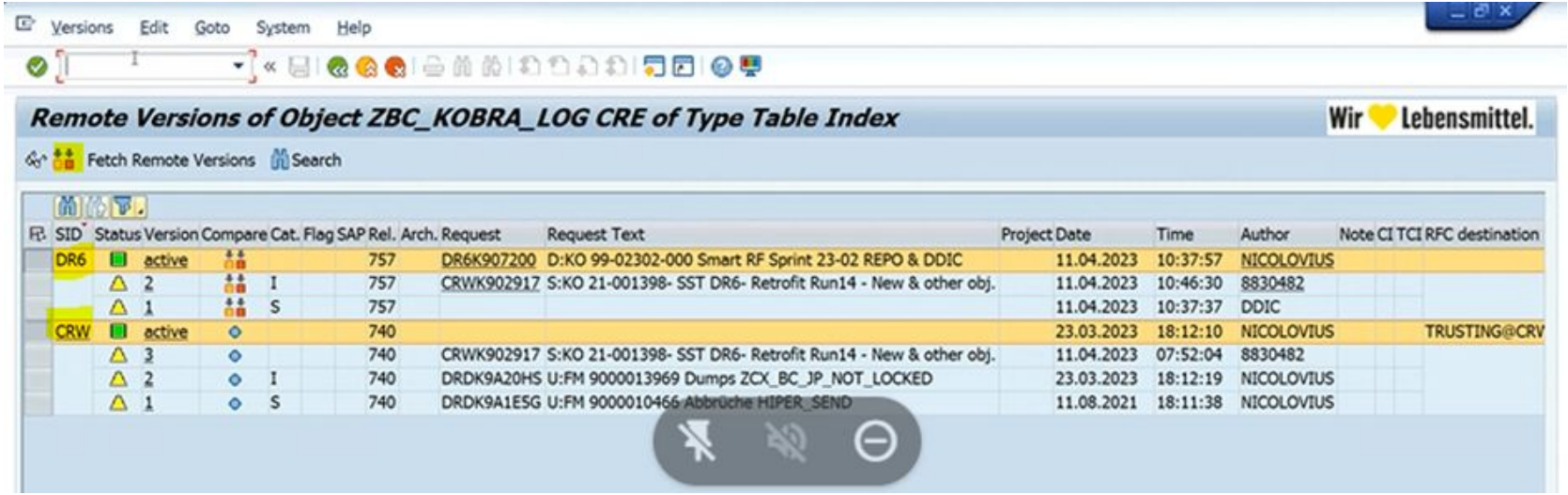


B. Table index will be opened in the next screen.



C. To understand the difference between source and target code, go to Utilities -> Versions -> Version Management -> Fetch Remote version -> RFC Destination and execute

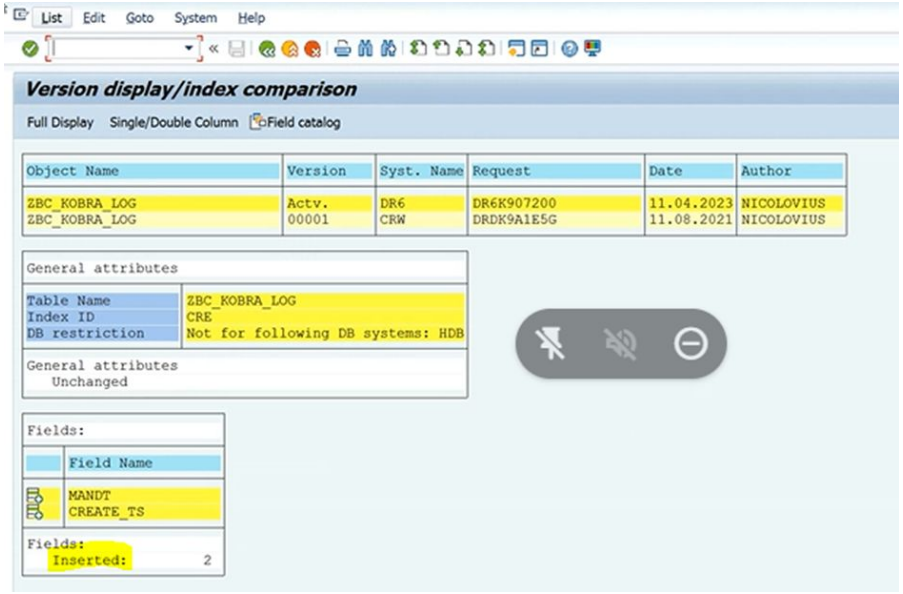
D. Further, select the versions from source and target to compare and understand the differences.



SID	Status	Version	Compare	Cat.	Flag	SAP Rel.	Arch.	Request	Request Text	Project	Date	Time	Author	Note	CI	TCI	RFC destination
DR6		active				757		DR6K907200	D:KO 99-02302-000 Smart RF Sprint 23-02 REPO & DDIC		11.04.2023	10:37:57	NICOLOVIUS				
		2		I		757		CRWK902917	S:KO 21-001398- SST DR6- Retrofit Run14 - New & other obj.		11.04.2023	10:46:30	8830482				
		1		S		757					11.04.2023	10:37:37	DDIC				
CRW		active				740					23.03.2023	18:12:10	NICOLOVIUS				TRUSTING@CRV
		3				740		CRWK902917	S:KO 21-001398- SST DR6- Retrofit Run14 - New & other obj.		11.04.2023	07:52:04	8830482				
		2		I		740		DRDK9A20HS	U:FM 9000013969 Dumps ZCX_BC_JP_NOT_LOCKED		23.03.2023	18:12:19	NICOLOVIUS				
		1		S		740		DRDK9A1E5G	U:FM 9000010466 Abbrüche HIPER_SEND		11.08.2021	18:11:38	NICOLOVIUS				

E. Click on the COMPARE shortcut(beside the specs), to visualize the differences.

LIGHT yellow color refers to Source and DARK yellow color refers to target.



Here, there is NO difference in the general attributes of the INDEX between source and target, but two NEW fields (MANDT, CREATE_TS) has been introduced in the source system with INDEX ID as CRE.

So, these two field needs to created in the target to resolve the conflict between source and target.

Key Points to remember:

- Objects that are not existing in the system are considered as **NEW** object, which can be imported via ToC in the target system from source system.
- For PRE EXISTING objects, if NEW fields are supposed to be introduced as index, then the attributes of the field should be kept same as the source.
- Steps to create an index manually,

  SE11 -> put table name in the Database section -> Indexes(in the toolbar) -> Click NEW index option -> copy the attributes details as provided in the source.

TRAN

Object type: Transaction code

- Transaction codes are taken care as part of Service scope
- TCode SE93 is used to maintain Transaction codes.
- Customised transaction codes are created to make the Module pool / Report Programs, available to the end users.
- New TRAN objects can be imported via ToC.
- For PRE-EXISTING TRAN objects in the system: Do a manual comparison on screen level between source and target to understand the differences.

  And if any difference is found, we have to check via se03 or Version history to understand if its a source change or target change.

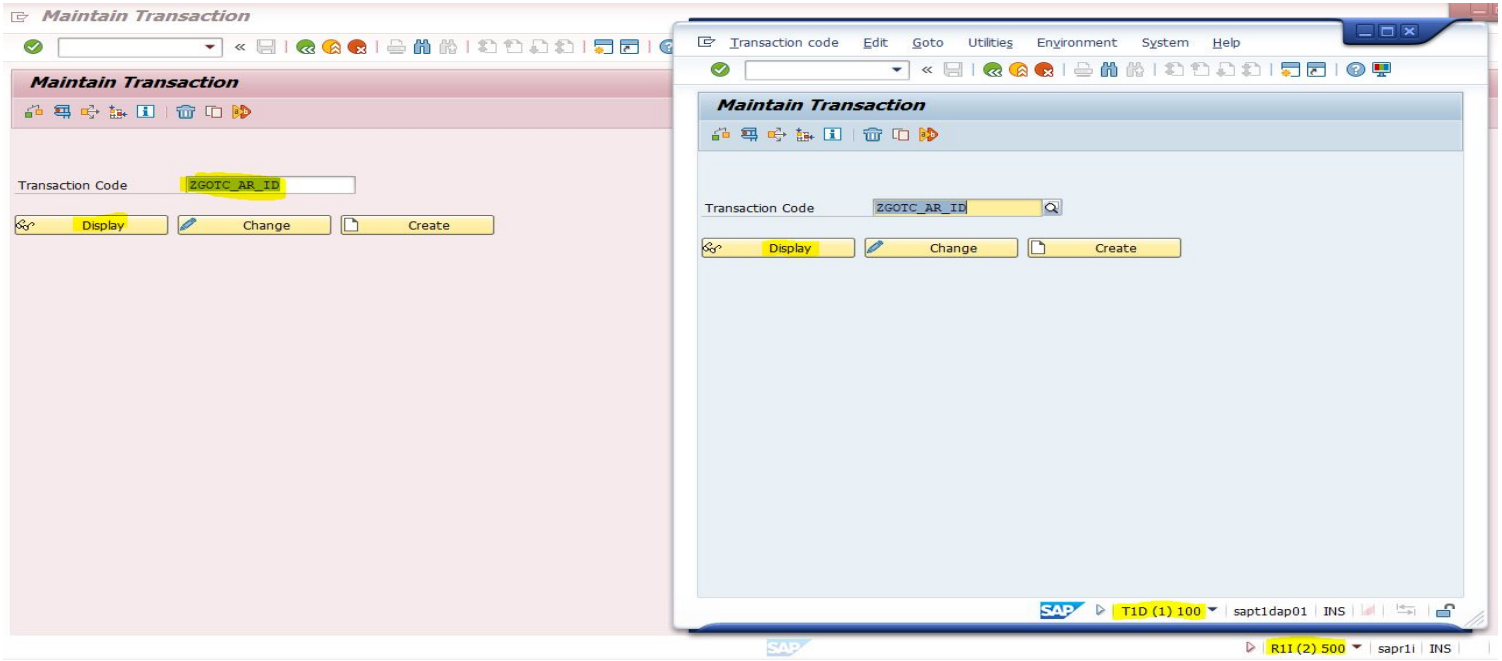
  Note: Version history for Transaction codes may not be available in all systems, in such cases make use of se03 to identify/conclude if change is source specific or target specific change.

  - ➢ If its a source change, we can bring the source changes to target
    and update status/Comments as : Retrofit Done - Added latest source changes to target.
  - ➢ If its a target change, we should not bring the source changes, instead-have to retain target changes.
    and update status/Comments as : No Action - Target has the latest changes..

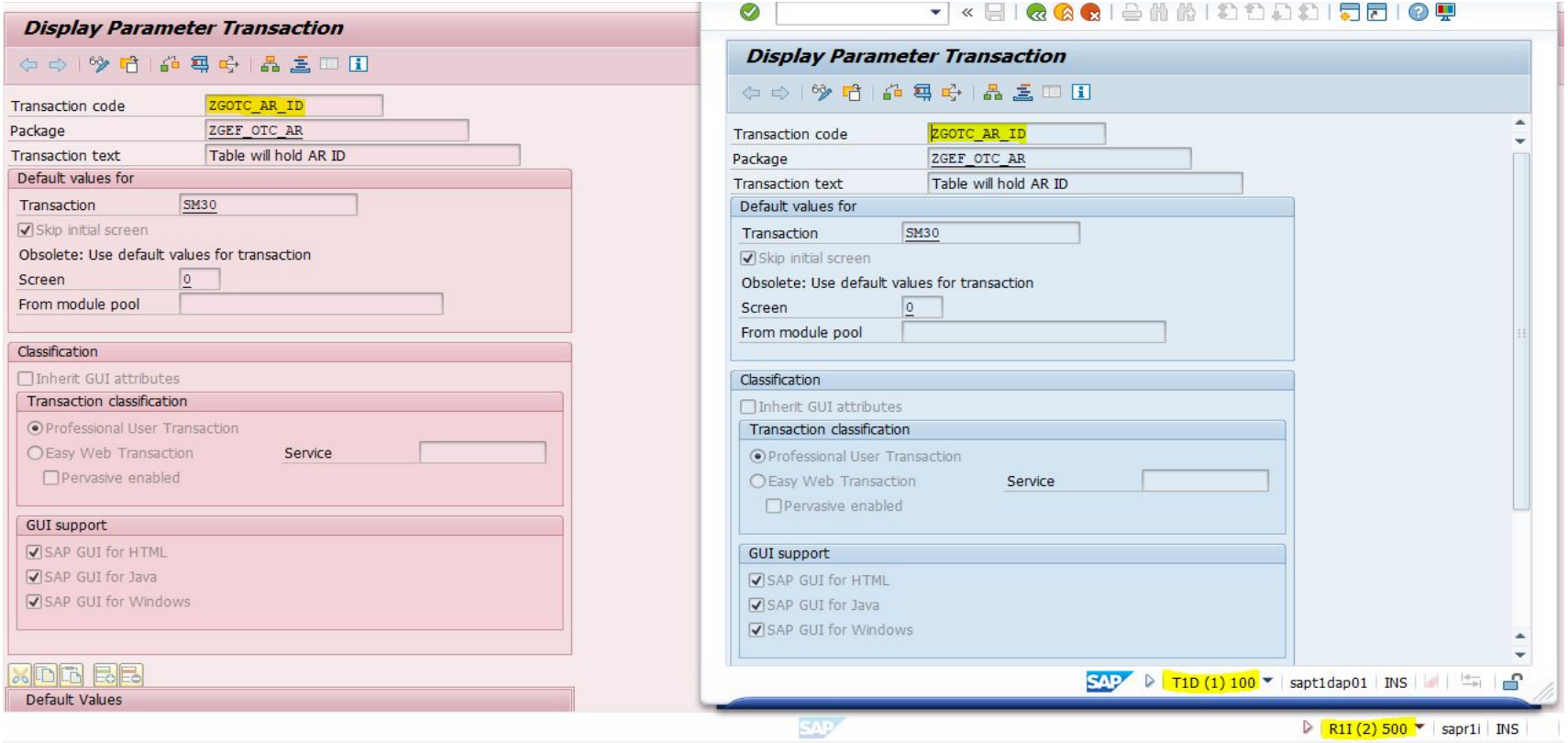
Example: R3TR-TRAN- ZGOTC_AR_ID

Go to t-code SE93 -> Give the object name -> Display in both source and target system.

[In the example, light red side is source and light blue side is target]



- Compare all the tabs manually, to check the difference if any.



In the above example, All looks identical; therefore we can update the scoping sheet status/comments as: No Action-Identical in source and target.

SOBJ

Object type: Business objects

- Tcode: SWO1
- SOBJ object contains the technical definition of a transport object i.e, It contains all the SAP objects.
- For PRE-EXISTING SOBJ objects in the system: Utility program RSSYSCOMP, can be used to compare the active version of target with the source TR version from the source using a RFC (remote function call triggered from target to source or vice-versa). Maximum 8 objects can be executed at once, using this utility.
- Steps to execute the utility program:

1. Open the utility program in SE38 and execute.
2. Give the RFC Dest. and select “Freely selected Objects” radio button

   PgmID: R3TR

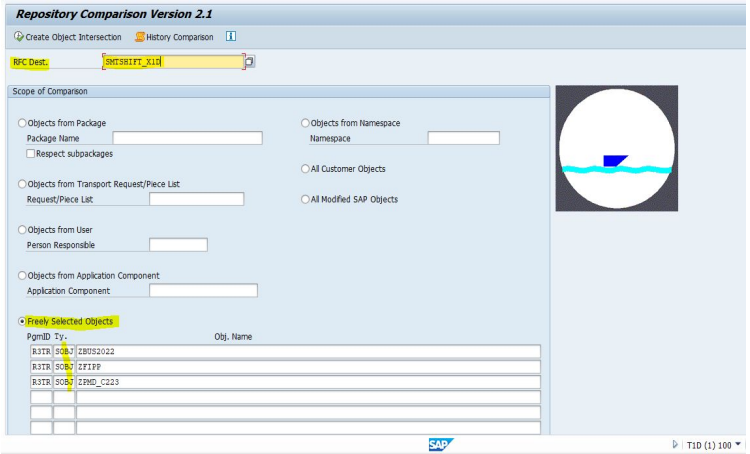
   Ty. : SOBJ

   Obj. Name: SOBJ obj name

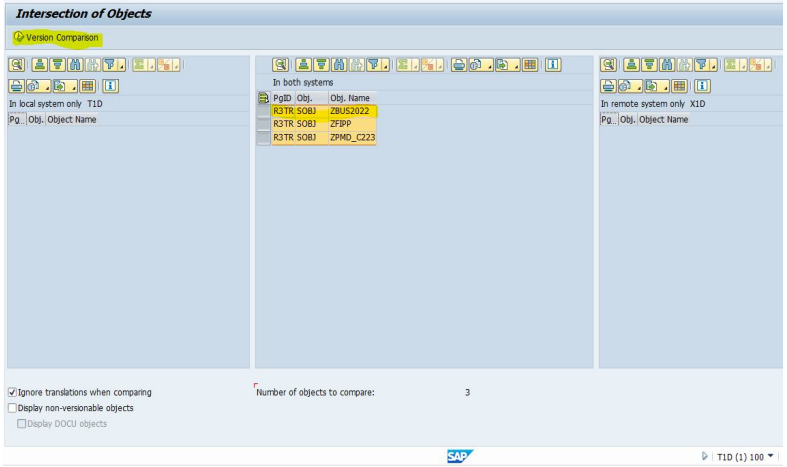
3. Further, click on “Create Object Intersection” and on the NEXT screen click on “Version Comparison”.

4. Here, GREEN color represents that the objects are IDENTICAL between source and target and RED color represents that there is difference between the objects in source and target. Further, to resolve the difference one should compare the objects on screen level in source and target.

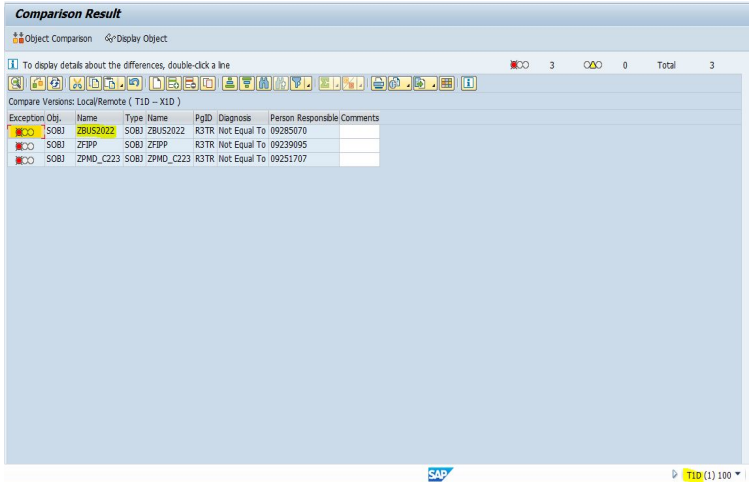
1.


Repository Comparison Version 2.1
Create Object Intersection
History Comparison
RFC Dest.
SMTSHIFT_X1D
Scope of Comparison
Objects from Package
Package Name
Respect subpackages
Objects from Transport Request/Piece List
Request/Piece List
Objects from User
Person Responsible
Objects from Application Component
Application Component
Freely Selected Objects
PgmID Ty.
Obj. Name
R3TR SOBJ ZBUS2022
R3TR SOBJ ZFIPP
R3TR SOBJ ZPMD_C223
Objects from Namespace
Namespace
All Customer Objects
All Modified SAP Objects
T1D (1) 100

2.

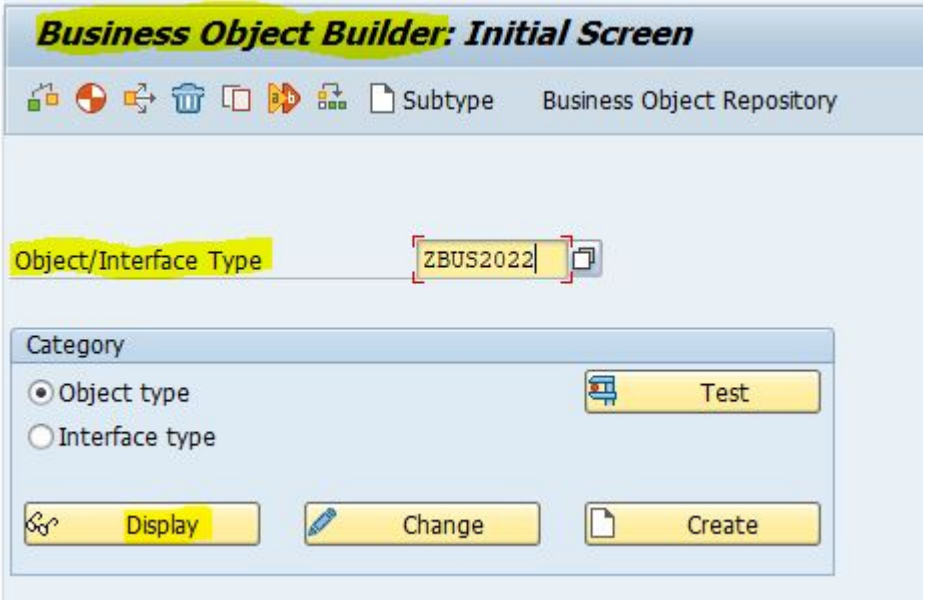

Intersection of Objects
Version Comparison
In local system only T1D
Pg.. Obj. Object Name
In both systems
PgID Obj. Obj. Name
R3TR SOBJ ZBUS2022
R3TR SOBJ ZFIPP
R3TR SOBJ ZPMD_C223
In remote system only X1D
Pg.. Obj. Object Name
Ignore translations when comparing
Display non-versionable objects
Display DOCU objects
Number of objects to compare: 3
T1D (1) 100

3.


Comparison Result
Object Comparison
Display Object
To display details about the differences, double-click a line
3
0
Total 3
Compare Versions: Local/Remote (T1D – X1D)
Exception Obj. Name Type Name PgID Diagnosis Person Responsible Comments
SOBJ ZBUS2022 SOBJ ZBUS2022 R3TR Not Equal To 09285070
SOBJ ZFIPP SOBJ ZFIPP R3TR Not Equal To 09239095
SOBJ ZPMD_C223 SOBJ ZPMD_C223 R3TR Not Equal To 09251707
T1D (1) 100

- When latest TR in source is not the scope TR, then one should do manual comparison on screen level between source and target to understand the differences.

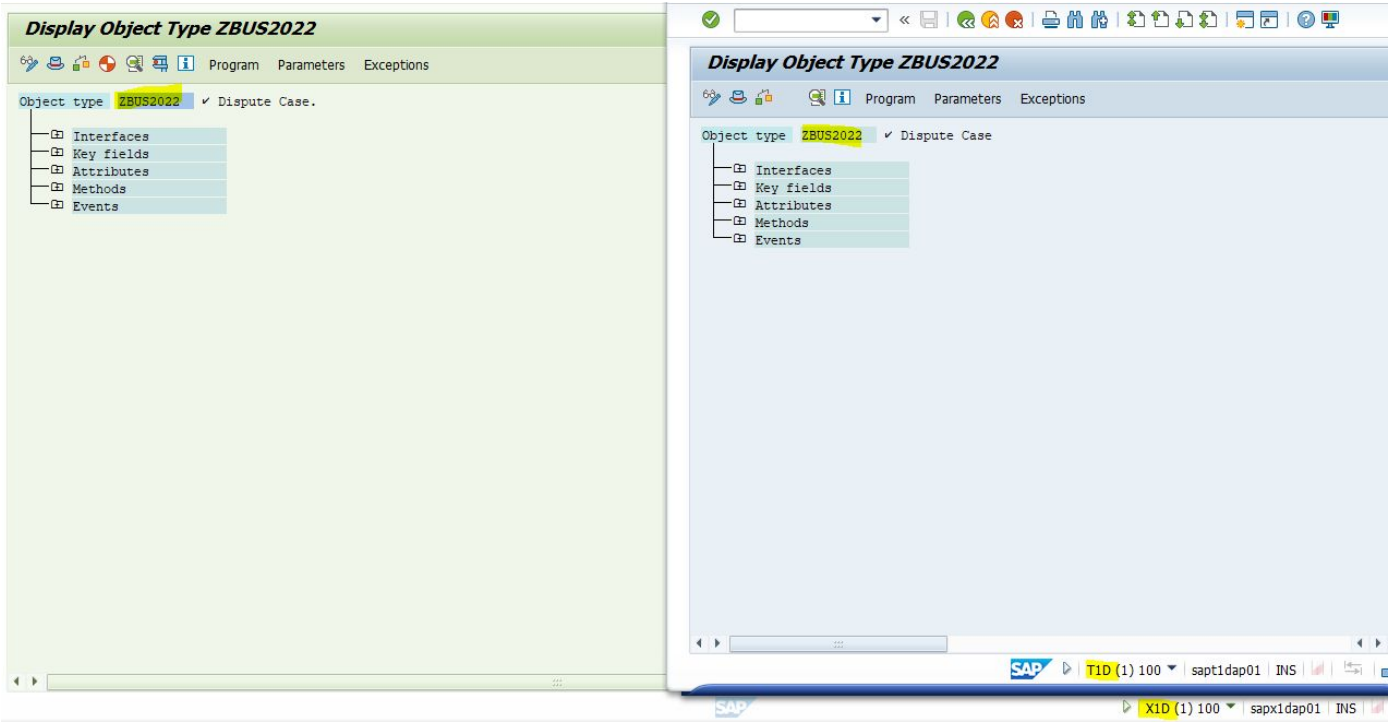
1. T-code SWO1 -> put Object/Interface type -> Display



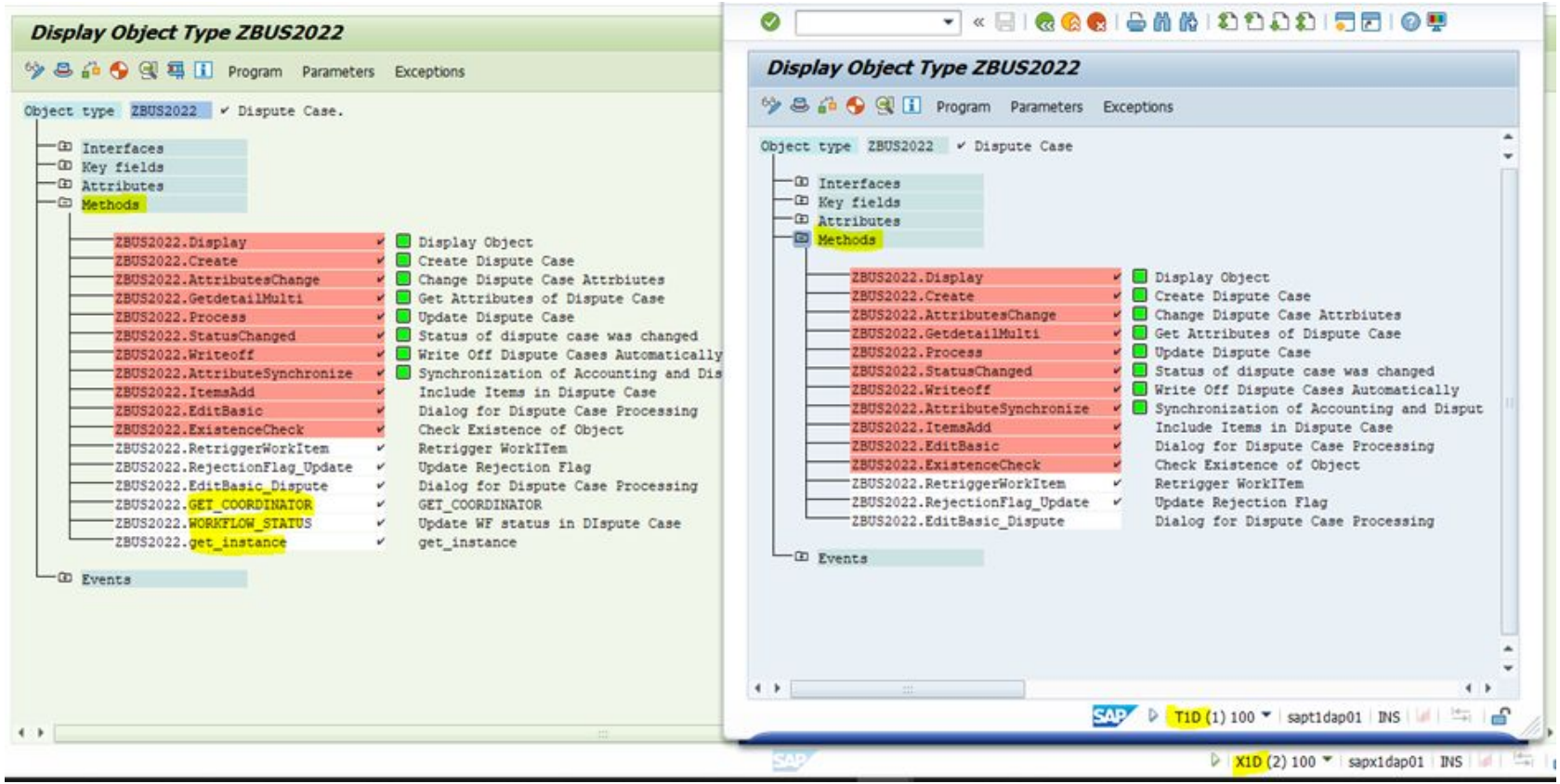
SOBJ obj: ZBUS2022

2. Screenshots of the main screen of SWO1 tcode between source and target:

X1D (100) being the source and T1D (100) being the target,



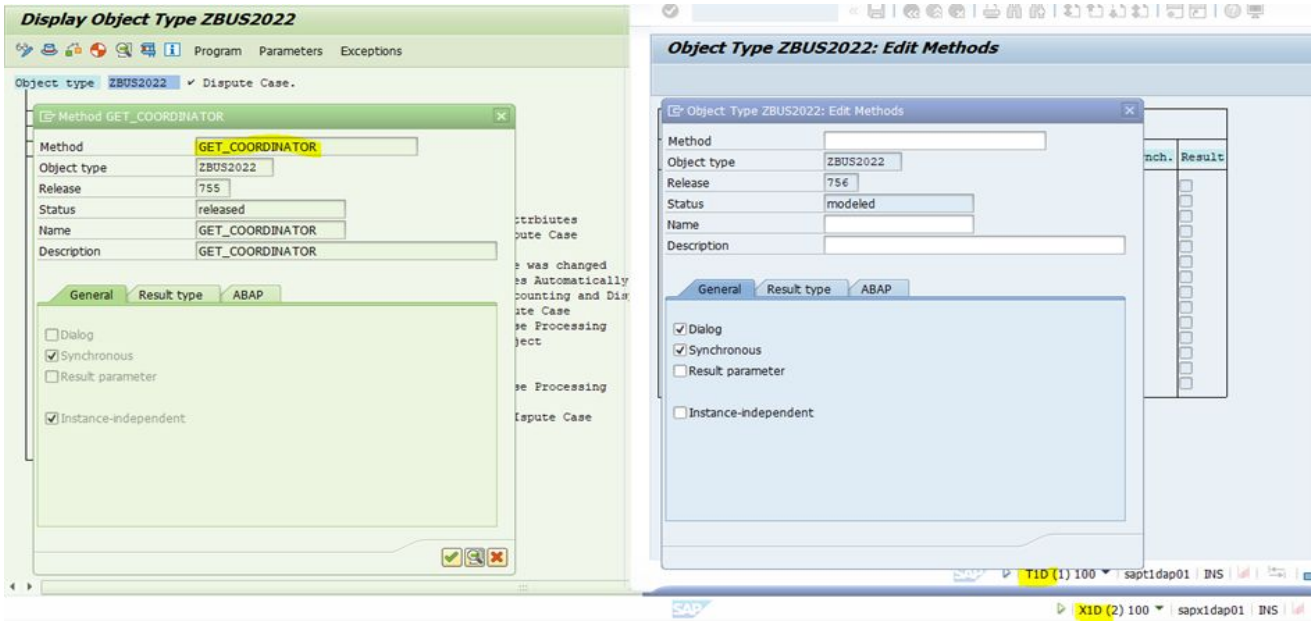
3. Expand all the object types, and check for the difference. Here, there's additional 3 methods in XID(source), so we need to cross check whether these methods are really added in source or have been newly deleted in target. In this case, it has been newly added in source, so we need to add those methods along with their parameters in target as well.



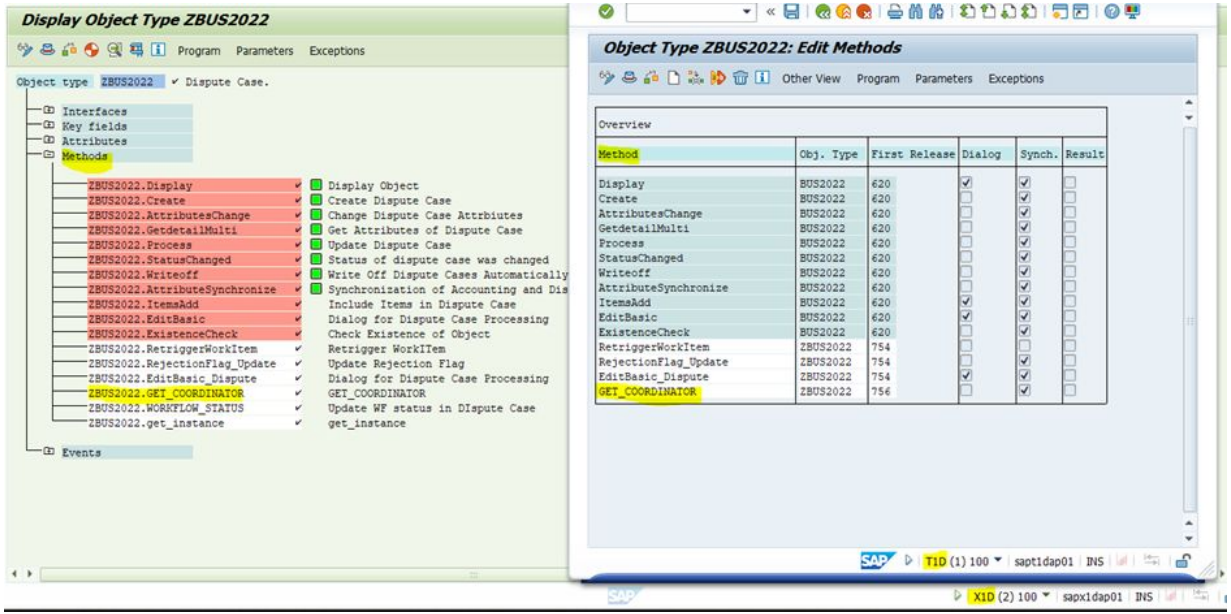
4. To add the method GET_COORDINATOR, select METHODS under OBJECT TYPE and further click on CREATE symbol in toolbar.

5. A pop-up with CREATE METHOD(create with function module as template) will appear, select NO.

6. Now, a new pop-up window will appear, where we can give the details of the method GET_COORDINATOR similar to the details available in source for that particular method and continue.



7. Here, we can see GET_COORDINATOR method got created newly in target.



8. Now, we need to cross check the other attributes of the method, such as PARAMETERS.

Select the method -> Parameters (from tool-bar) -> create -> a pop-up will appear where we need to fill the parameter details similar to the details available in the source (defining importing, exporting parameter details etc) -> Continue

NOTE:

- To make any kind of change in the attributes of a METHOD, one should always check if the method's status is showing as RELEASED(a tick mark symbol will appear beside the method name) or UNRELEASED.
- Only if the method's status is UNRELEASED, proceed with the required changes in the method.
- Steps to change the method status from RELEASED to UNRELEASED:

  Edit -> Change release status -> Object type / Object type component -> To implemented -> Edit -> Change release status -> Object type / Object type component -> To modeled -> now, the method attributes will become editable.

- Steps to change the method status from UNRELEASED to RELEASED:

  Edit -> Change release status -> Object type / Object type component -> To implemented -> Edit -> Change release status -> Object type / Object type component -> To released -> now, the method attributes will become uneditable -> SAVE

Lakshmi

INTF
DYNP
SXCI
SSFO
SSST
SSFT
SFPI
SFPF
SHMA

INTF

Obj Type: INTF(Interface)

Tcode to check the interface object is : SE24

Procedure to do the retrofit:

- If it is a new object we will directly add them into ToC
- If it is already existed object then we will check whether we can import that object(safe to import) or need to do the retrofit manually.

Step 1: Goto SE38 and execute the program “RSSYSCOMP” in target system (Compare Repository of two Systems) as shown below.

Create Object Intersection | History Comparison

RFC Dest.

Scope of Comparison

- (•) Objects from Package
  - Package Name
  - [ ] Respect subpackages
- ( ) Objects from Transport Request/Piece List
  - Request/Piece List
- ( ) Objects from User
  - Person Responsible
- ( ) Objects from Application Component
  - Application Component
- ( ) Freely Selected Objects

PgmID	Ty.	Obj. Name

- ( ) Objects from Namespace
  - Namespace
- ( ) All Customer Objects
- ( ) All Modified SAP Objects

Create Object Intersection | History Comparison

3

RFC Dest. PNRCLNT100_RETRO 1

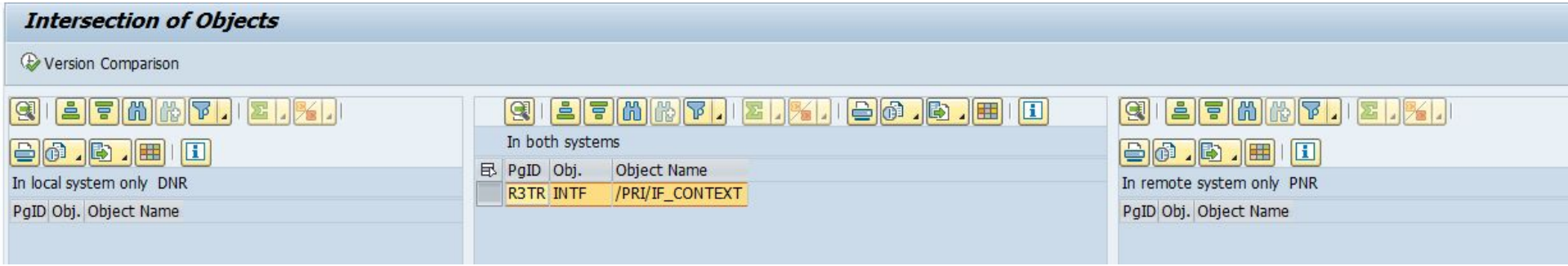
Scope of Comparison

- Objects from Package
  - Package Name
  - Respect subpackages
- Objects from Transport Request/Piece List
  - Request/Piece List
- Objects from User
  - Person Responsible
- Objects from Application Component
  - Application Component
- Freely Selected Objects
  - PgmID Ty. Obj. Name
  - R3TR INTF /PRI/IF_CONTEXT

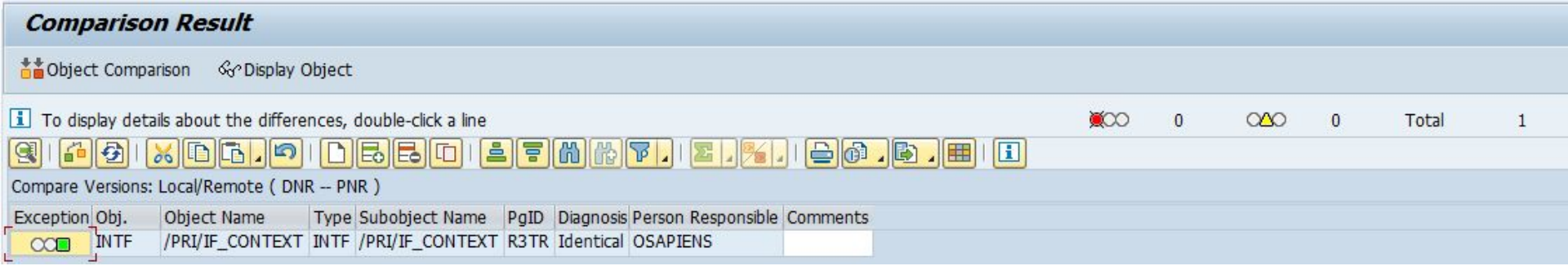
2

- Objects from Namespace
  - Namespace
- All Customer Objects
- All Modified SAP Objects

We need to give RFC Destination and also select the “Freely selected Objects” and provide the PgmID,Type and obj Name as shown in above and click on “Create object interaction” .

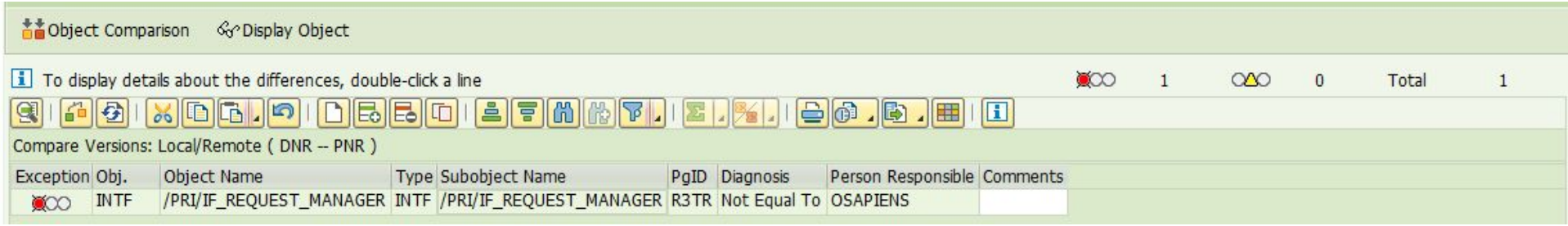


Later we need to click on “Version comparison” button.



This object is identical in both the systems.No Need to take any action as there are no differences in both Source and Target systems.

If the object is not identical in RSSYSCOMP as shown below.

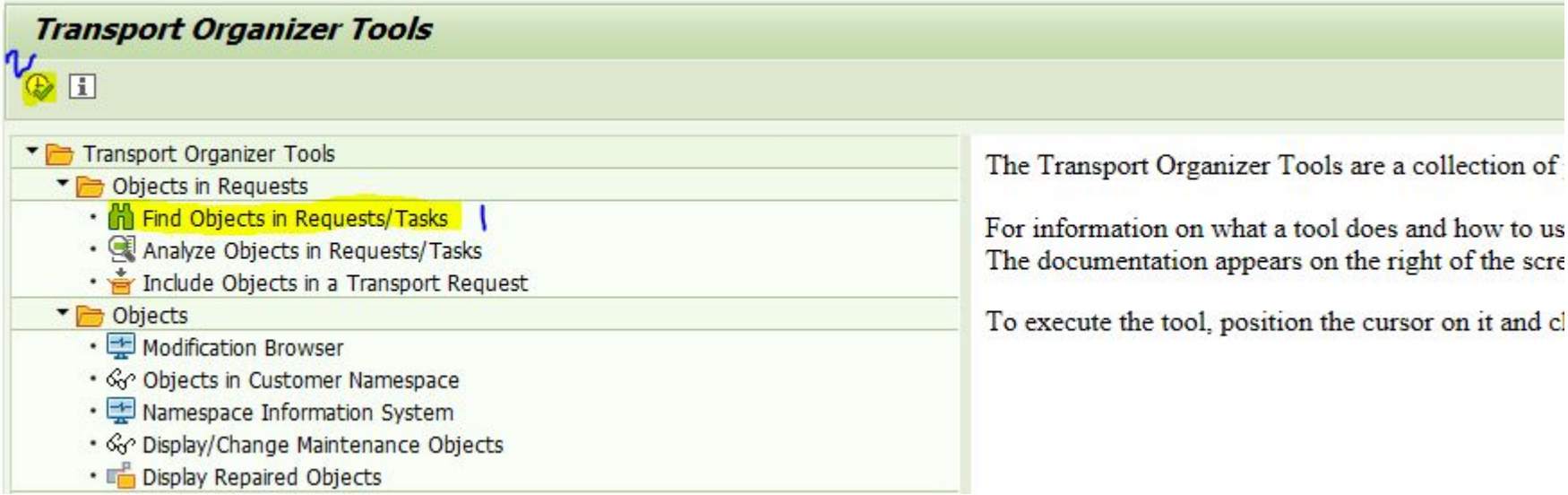


It will show it in a red color exception.To see the differences need to double click on that object.

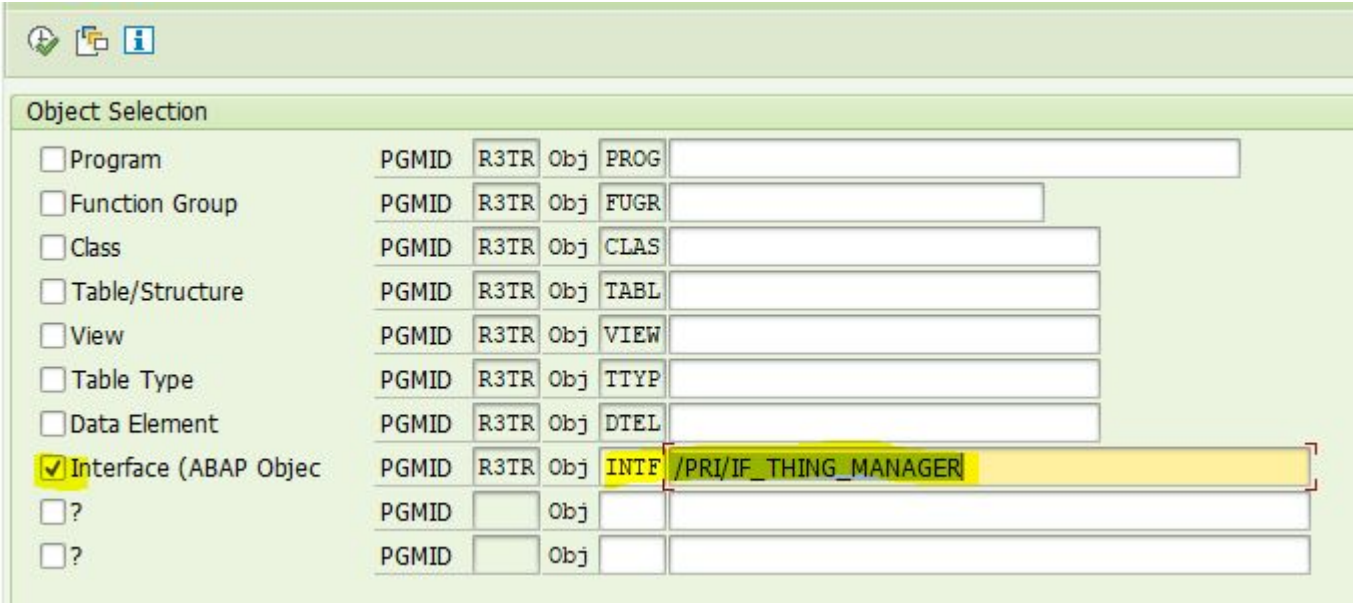
Safe to Import Analysis: When it shows “not identical” in RSSYSCOMP,we need to check for safe to import.

Procedure:

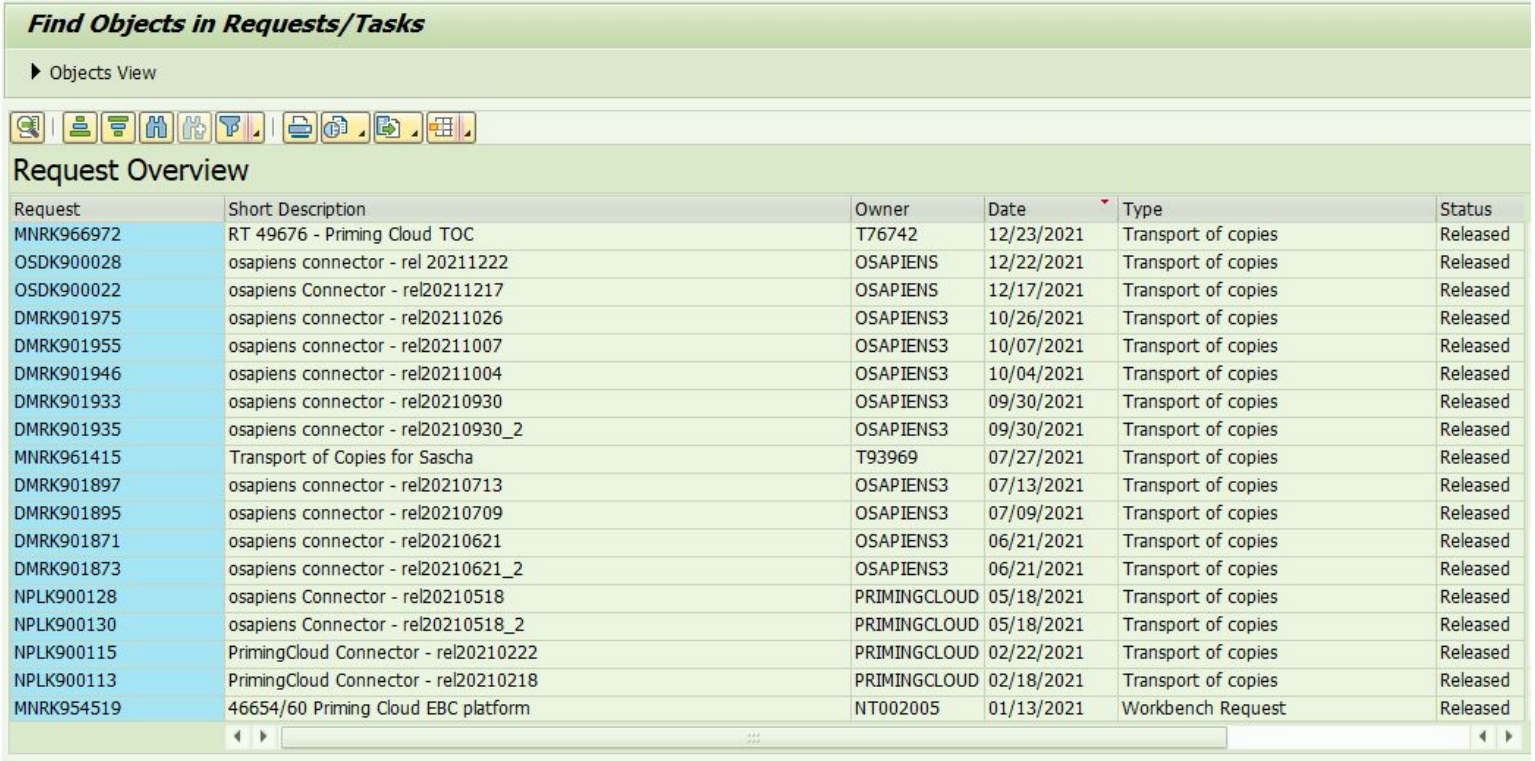
1. Goto Tcode SE03 and select “Find objects in REquests/Tasks and execute



Give Obj Type and Obj Name and select that row and click on execute



Once you get the Transports list,select DATE column and click on Descending to get the latest transports in sorting.



Find Objects in Requests/Tasks

Objects View

Request Overview

Request	Short Description	Owner	Date	Type	Status
MNRK966972	RT 49676 - Priming Cloud TOC	T76742	12/23/2021	Transport of copies	Released
OSDK900028	osapiens connector - rel 20211222	OSAPIENS	12/22/2021	Transport of copies	Released
OSDK900022	osapiens Connector - rel20211217	OSAPIENS	12/17/2021	Transport of copies	Released
DMRK901975	osapiens connector - rel20211026	OSAPIENS3	10/26/2021	Transport of copies	Released
DMRK901955	osapiens connector - rel20211007	OSAPIENS3	10/07/2021	Transport of copies	Released
DMRK901946	osapiens connector - rel20211004	OSAPIENS3	10/04/2021	Transport of copies	Released
DMRK901933	osapiens connector - rel20210930	OSAPIENS3	09/30/2021	Transport of copies	Released
DMRK901935	osapiens connector - rel20210930_2	OSAPIENS3	09/30/2021	Transport of copies	Released
MNRK961415	Transport of Copies for Sascha	T93969	07/27/2021	Transport of copies	Released
DMRK901897	osapiens connector - rel20210713	OSAPIENS3	07/13/2021	Transport of copies	Released
DMRK901895	osapiens connector - rel20210709	OSAPIENS3	07/09/2021	Transport of copies	Released
DMRK901871	osapiens connector - rel20210621	OSAPIENS3	06/21/2021	Transport of copies	Released
DMRK901873	osapiens connector - rel20210621_2	OSAPIENS3	06/21/2021	Transport of copies	Released
NPLK900128	osapiens Connector - rel20210518	PRIMINGCLOUD	05/18/2021	Transport of copies	Released
NPLK900130	osapiens Connector - rel20210518_2	PRIMINGCLOUD	05/18/2021	Transport of copies	Released
NPLK900115	PrimingCloud Connector - rel20210222	PRIMINGCLOUD	02/22/2021	Transport of copies	Released
NPLK900113	PrimingCloud Connector - rel20210218	PRIMINGCLOUD	02/18/2021	Transport of copies	Released
MNRK954519	46654/60 Priming Cloud EBC platform	NT002005	01/13/2021	Workbench Request	Released

If you find the Source TR as the latest TR,that means there are no changes after the Source change.So,we can import this object as “Safe to Import”.

If not,means the latest TR is the target TR,we should check this object manually.

Manual retrofit Process: We can compare the offline results in MELD file by cross checking the versions in both source and target to know the latest changes and do the retrofit.

C:\Users\ladimulam\Downloads\CONA_Run#5\source_par: Browse...

```
interface /PRI/IF_THING_MANAGER
  public .

  type-pools ABAP .

  methods GET_THINGS
    importing
      !IV_CREATE_IF_NOT_EXISTS type ABAP_BOOL default A
      !IT_THING_KEY type /PRI/TT_THING_KEY
    exporting
      !ET_THING type /PRI/TT_THING .
  methods MODIFY_THINGS
    importing
      !IT_THING type /PRI/TT_THING
    exporting
      !ET_THING type /PRI/TT_THING .
endinterface.
```

C:\Users\ladimulam\Downloads\CONA_Run#5\merged\PR Browse...

```
interface /PRI/IF_THING_MANAGER
  public .

  type-pools ABAP .

  methods GET_THINGS
    importing
      !IV_CREATE_IF_NOT_EXISTS type ABAP_BOOL default A
      !IT_THING_KEY type /PRI/TT_THING_KEY
    exporting
      !ET_THING type /PRI/TT_THING .
  methods MODIFY_THINGS
    importing
      !IT_THING type /PRI/TT_THING
    exporting
      !ET_THING type /PRI/TT_THING .
endinterface.
```

C:\Users\ladimulam\Downloads\CONA_Run#5\target\PRI\' Browse...

```
interface /PRI/IF_THING_MANAGER
  public .

  type-pools ABAP .

  methods GET_THINGS
    importing
      !IV_CREATE_IF_NOT_EXISTS type ABAP_BOOL default A

    exporting
      !ET_THING type /PRI/TT_THING .
  methods MODIFY_THINGS
    importing
      !IT_THING type /PRI/TT_THING
    exporting
      !ET_THING type /PRI/TT_THING .
endinterface.
```

DYNP

Obj Type: DYNP(Screen)

Tcode to check DYNP object is : SE51

Procedure to do the retrofit:

- If it is a new object we will directly add them into ToC
- If it is already existed object then we will check whether we can import that object(safe to import) or need to do the retrofit manually.

Step 1: Goto SE38 and execute the program “RSSYSCOMP” in target system (Compare Repository of two Systems) as shown below.

Create Object Intersection | History Comparison

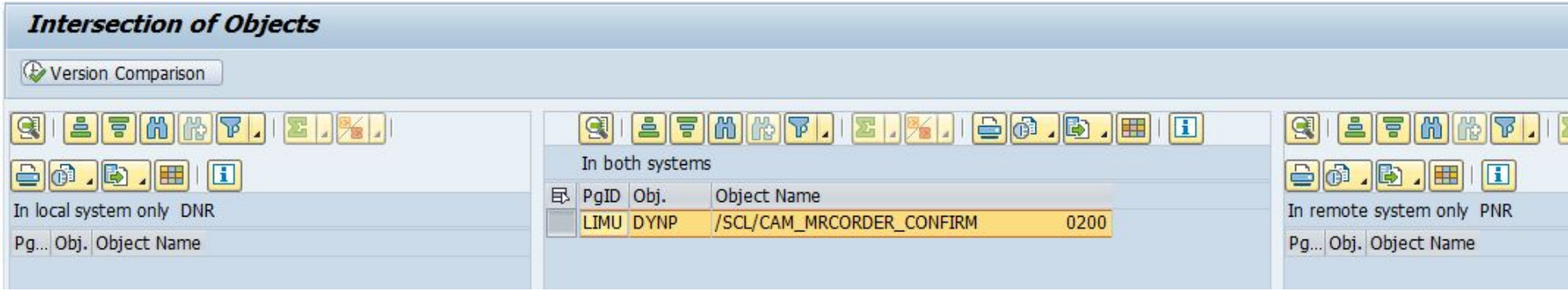
RFC Dest. PNRCLNT100_RETRO

Scope of Comparison

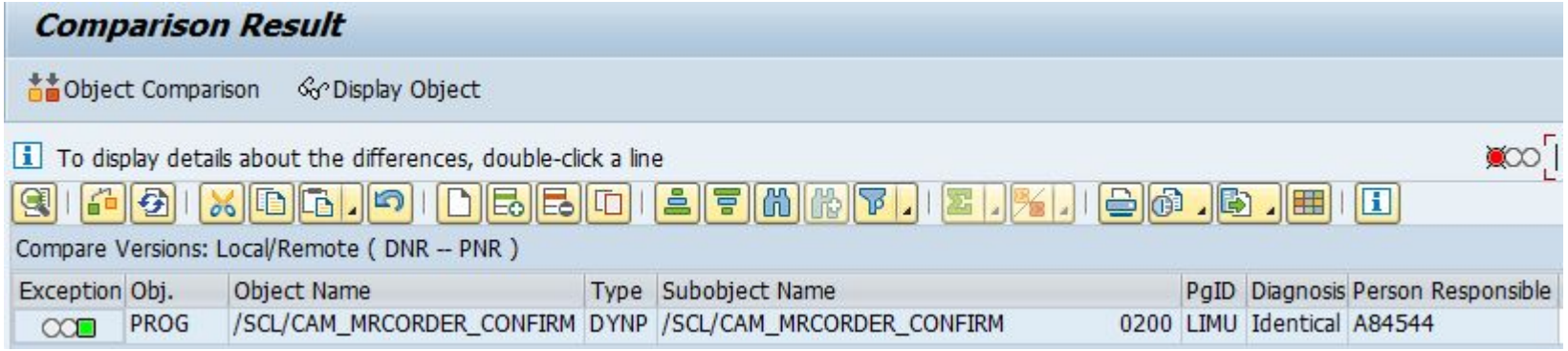
- Objects from Package
  - Package Name
  - Respect subpackages
- Objects from Transport Request/Piece List
  - Request/Piece List
- Objects from User
  - Person Responsible
- Objects from Application Component
  - Application Component
- Freely Selected Objects

PgmID	Ty.	Obj. Name	
LIMU	DYNP	/SCL/CAM_MRCORDER_CONFIRM	0200

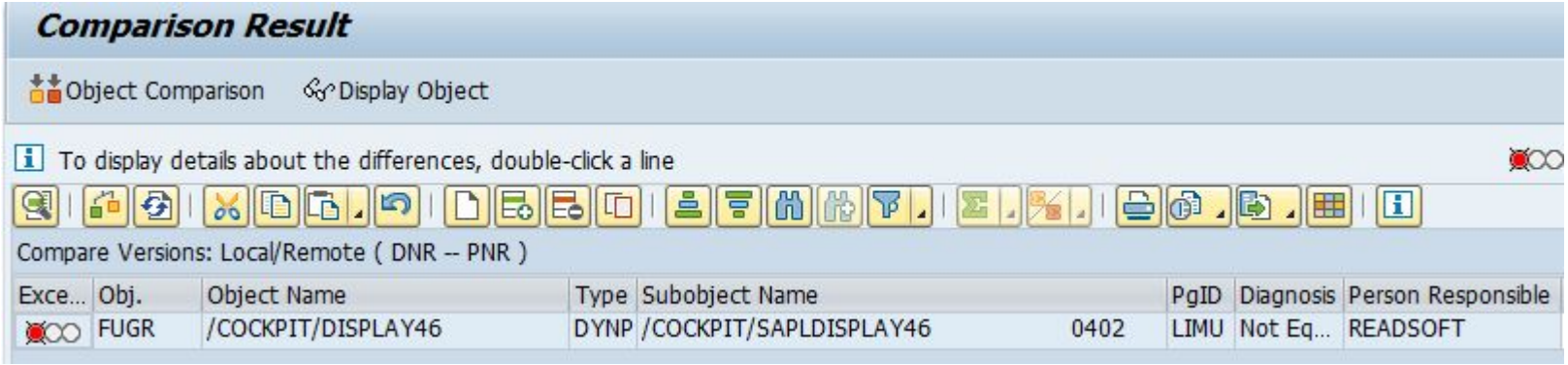
We need to give RFC Destination and also select the “Freely selected Objects” and provide the PgmID,Type and obj Name as shown in above and click on “Create object interaction” .



If it is Identical,No Need to take any action,it is an identical case.



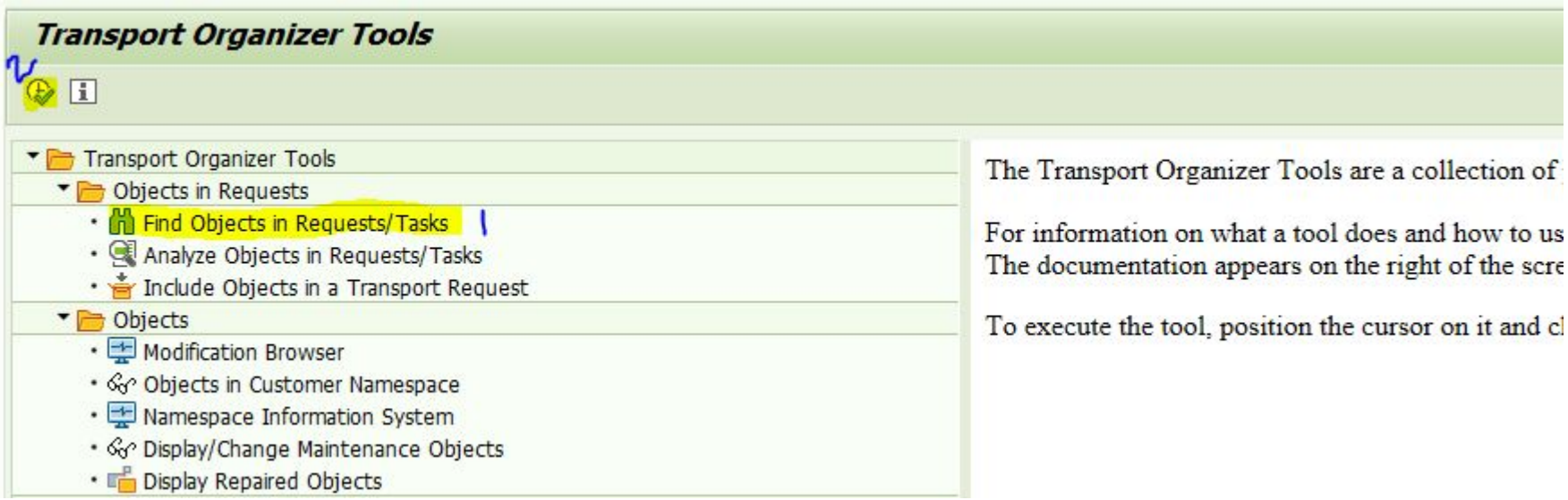
If it is not identical

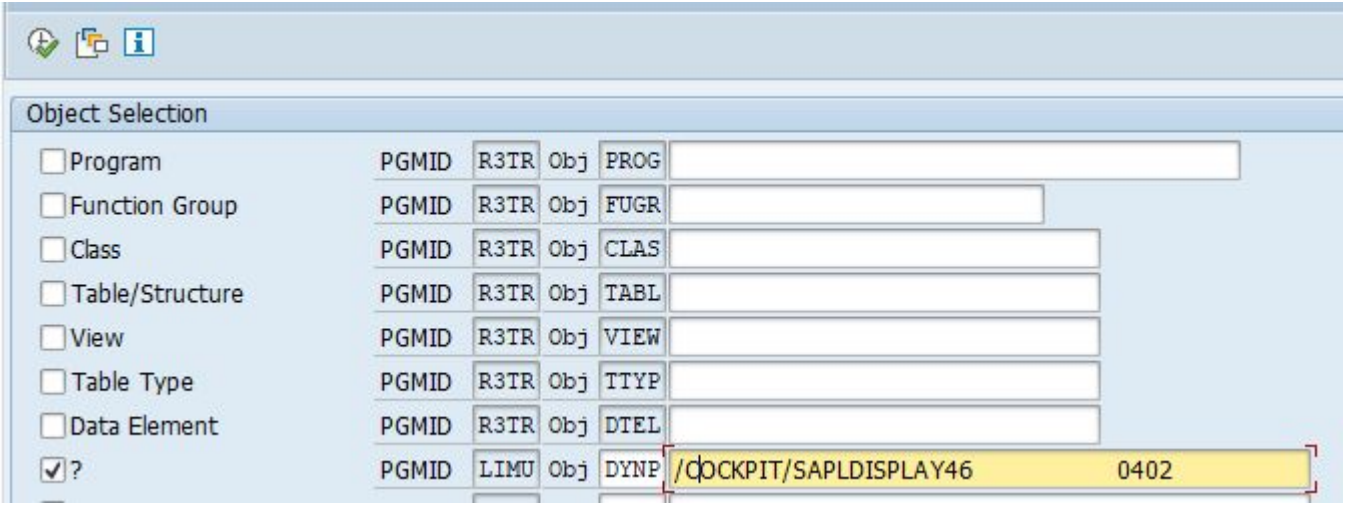


Safe to Import Analysis: When it shows “not identical” in RSSYSCOMP,we need to check for safe to import.

Procedure:

1. Goto Tcode SE03 and select “Find objects in Requests/Tasks and execute





If you find the Source TR as the latest TR,that means there are no changes after the Source change.So,we can import this object as “Safe to Import”.

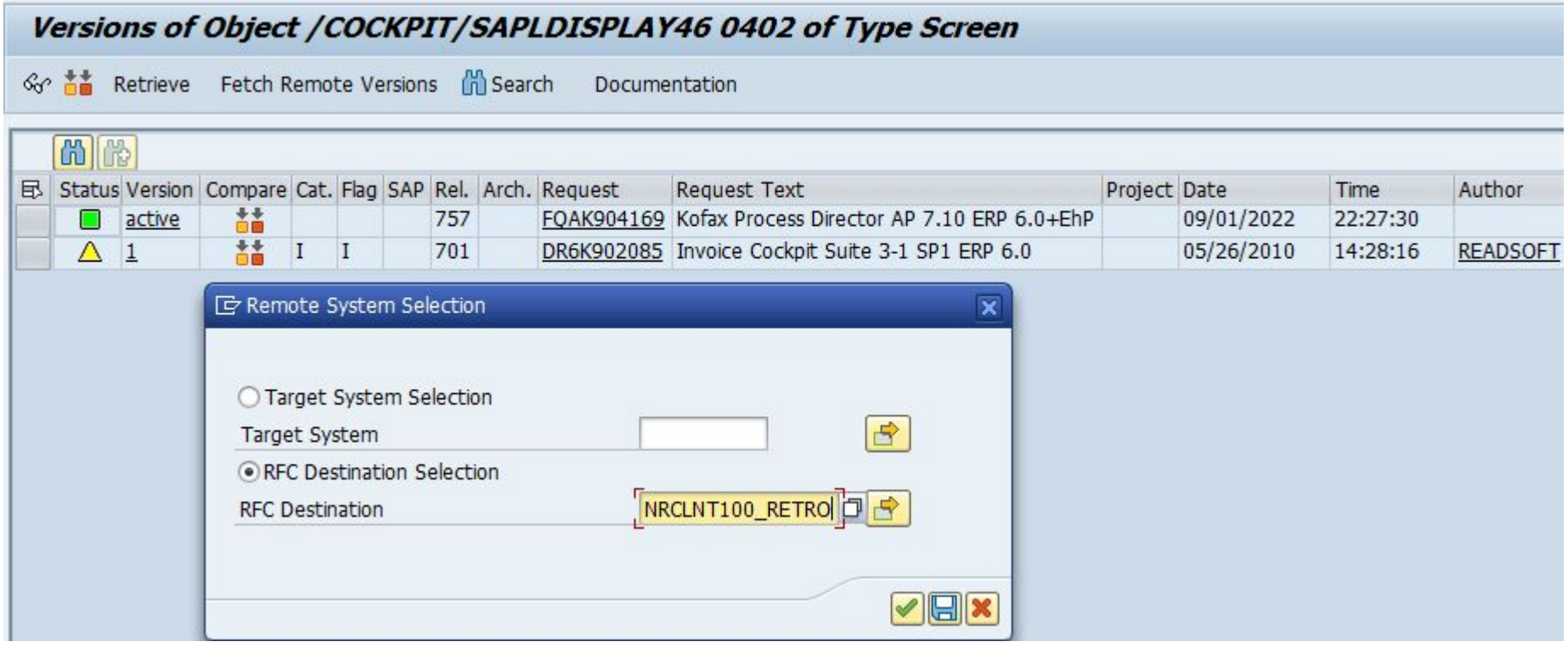
Request Overview

Request	Short Description	Owner	Date	Type	Status
SAPKCCD757	Commandfile for customer export	DDIC	03/27/2023	Piece list	Released
SAPKCCC751	Commandfile for customer export	DDIC	03/25/2023	Piece list	Released
FQAK902031	PROCESS DIRECTOR AP 7.7 ERP 6.0+EhP	READSOFT	05/31/2022	Workbench Request	Released
FQAK904169	Kofax Process Director AP 7.10 ERP 6.0+EhP	READSOFT	05/31/2022	Workbench Request	Released
Q74K900862	PROCESS DIRECTOR AP 7.4 ERP 6.0+EhP	READSOFT	05/20/2016	Workbench Request	Released
S47K903214	PROCESS DIRECTOR AP 7.1 incl. Hotfix 2 R/3 4.7+ Unicode	READSOFT	07/10/2013	Transport of copies	Released
DR6K902085	Invoice Cockpit Suite 3-1 SP1 ERP 6.0	READSOFT	05/26/2010	Transport of copies	Released

If not,means the latest TR is the target TR,we should check this object manually.

Manual retrofit Process:

We will open the DYNP in the system with the transaction SE51 and do the version comparison between the systems using RFC destination.



We should compare the ACTIVE VERSION of the TARGET and SCOPE VERSION of SOURCE systems

Fetch Remote Versions Search

S...	Status	Version	Compare	Cat.	Flag	SAP	Rel.	Arch.	Request	Request Text	Project	Date	Time	Author
PNR		active					750		MNRK979561	52863/10 CHG0096713 : ReadSoft - limit non-USD currency for		05/04/2023	15:01:05	
		1		I	I		701		DR6K902085	Invoice Cockpit Suite 3-1 SP1 ERP 6.0		05/26/2010	14:28:16	READSOFT
DNR		active					757		FQAK904169	Kofax Process Director AP 7.10 ERP 6.0+EhP		09/01/2022	22:27:30	
		1		I	I		701		DR6K902085	Invoice Cockpit Suite 3-1 SP1 ERP 6.0		05/26/2010	14:28:16	READSOFT

And the differences found between the systems should be checked with in the systems version comparison.So that we will get to know where exactly the changes re made.So that we will take the action Accordingly.

EX.
We will do the version comparison in Source system with scope TR version and the below versions of scope TR.

Modifications in Screen

Component	Value/Type
/COCKPIT/SAPLDISPLAY46 0402	Version 00003 <-> Version 00002
New in Version 00003	
Flow Logic	
0139	CHAIN.
0139	FIELD /cockpit/shdr_disp-currency.
0139	MODULE check_currency.
0139	ENDCHAIN.
0139	"End of MOD-001
Modified	
Flow Logic	
0138	
Version 00002	
Version 00003	CHAIN.
Deleted in Version 00003	
Flow Logic	
0060	

The same way we should check in Target system and compare Active version with previous versions and get to know the changes and Complete the retrofit accordingly.

SXCI

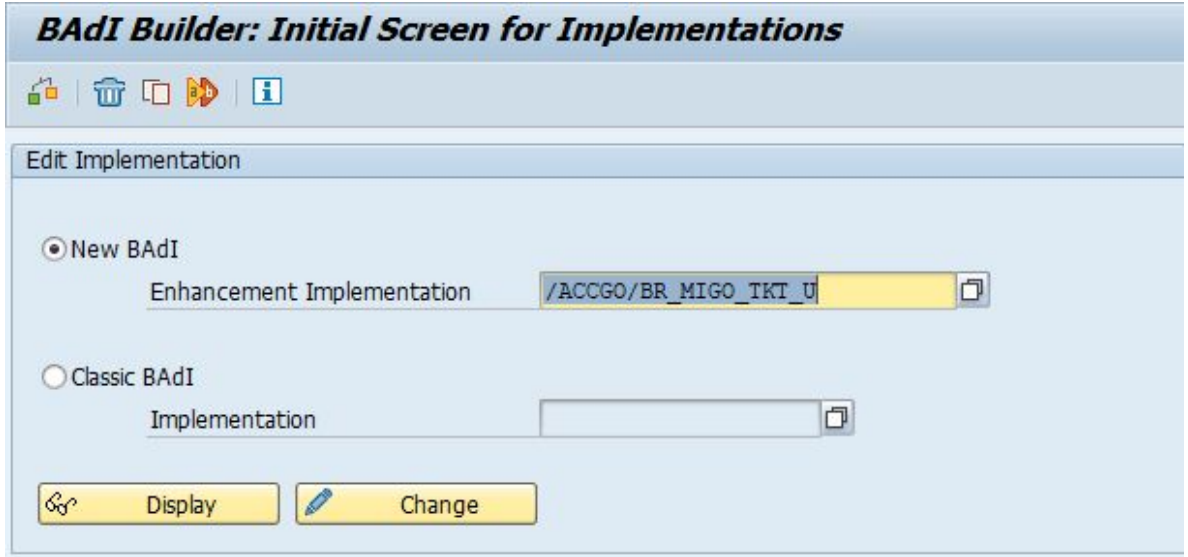
Obj type: Business Add-Ins - Implementations

Transaction: SE19

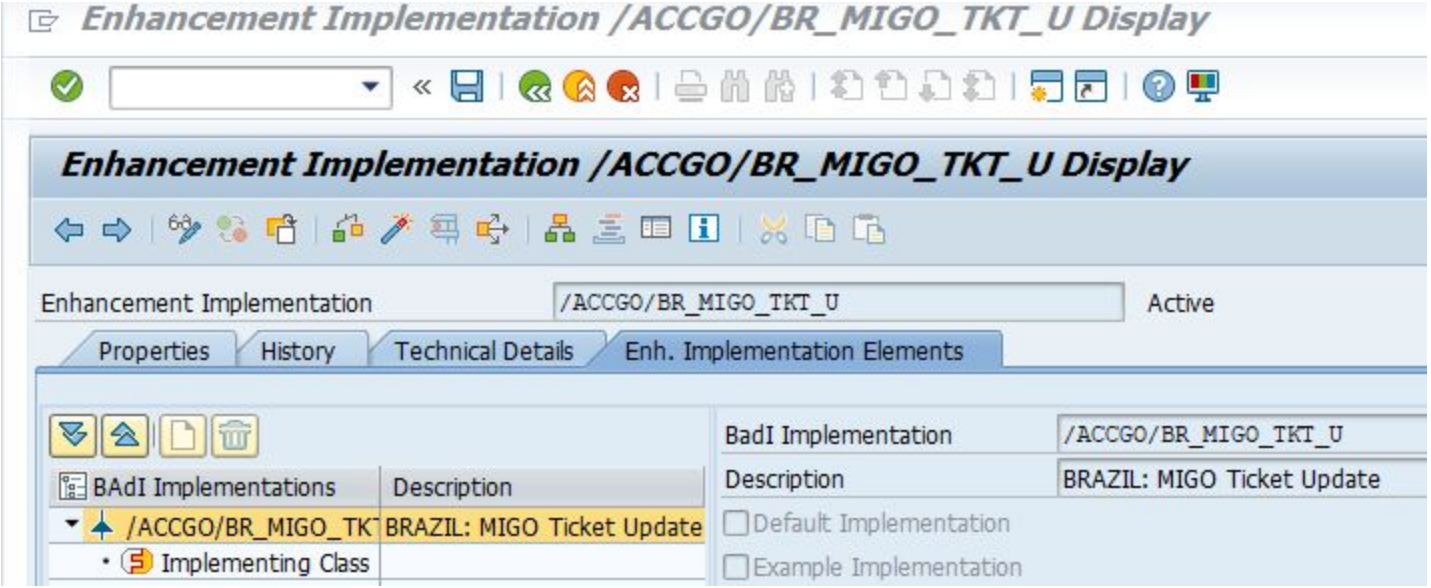
Procedure to do the retrofit:

- If it is a new object we will directly add them into ToC
- If it is already existed object then we need to do the retrofit manually.

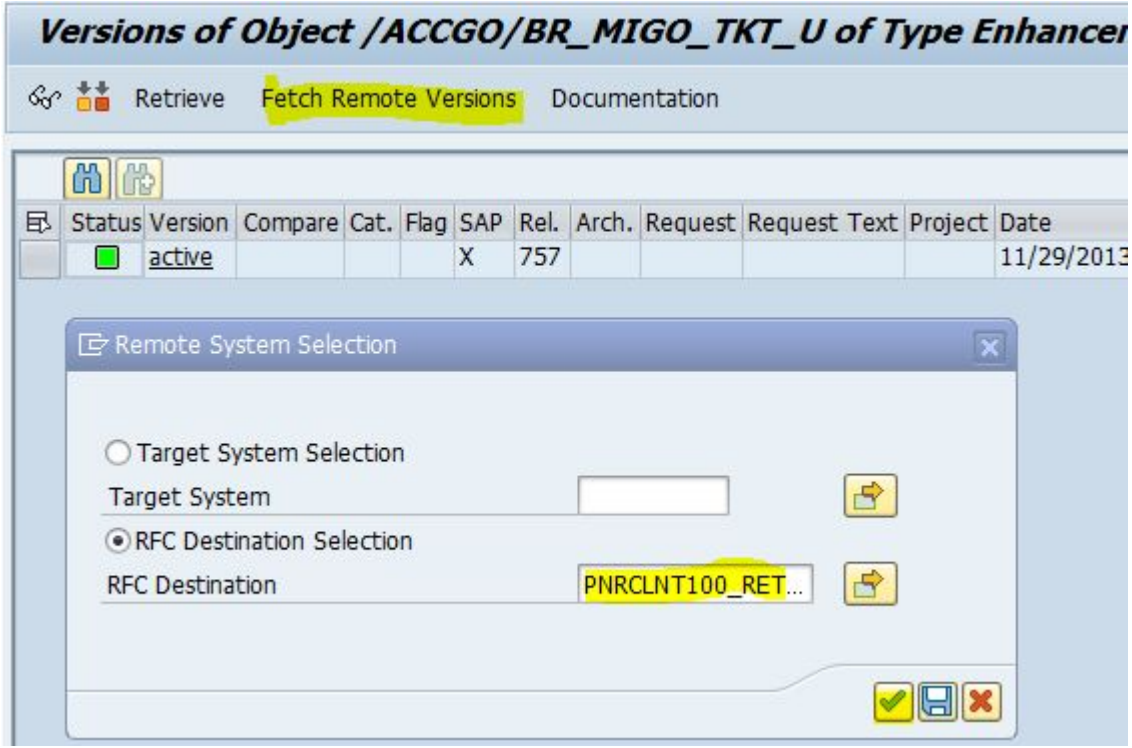
OPen the Object in transaction: SE19 and display



Goto Utilities->Versions->Version comparison



Select Fetch remote versions and provide RFC Destination and do the version comparison between the systems and complete the retrofit.



Praneeth

List of Object types

SPRX	Proxy(Deletion activity)
FUGR	Function Group
FUNC	Function Module
FUGT	Function Group Texts
DDLS	Data Definition Language Source(CDS Views)
PDWS	Workflow template
PDTS	Standard Task
PDAC	Standard rule
FUGX	Function Group with Customer Include: Customer Part
TDAT	Customizing: Table Contents
DSYS	Chapter of a Book Structure

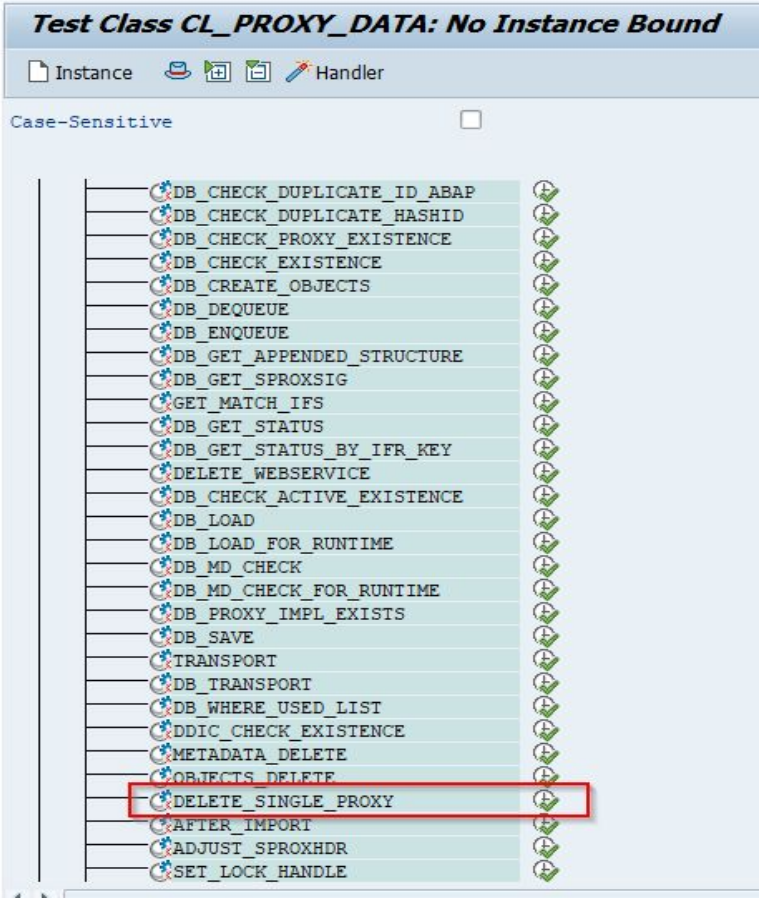
SPRX

Deletion of SPRX objects in Scope

Steps to Delete and capture Proxy and related objects in TR which are IN Scope

1. Go to Transaction SE24. Provide **CL_PROXY_DATA** as Input and click on Execute (F8) button and a Pop-up appears and then press Continue(Enter).
2. Navigate to Method **DELETE_SINGLE_PROXY** and click on Execute Method button besides it.
3.

Execute **DELETE_SINGLE_PROXY** method



Proxy object Before Deletion

Structure: ZDT_PBNA_FREIGHT_UNIT_INFO — Active

Short Description: Data Type for Freight Unit Information-I001049

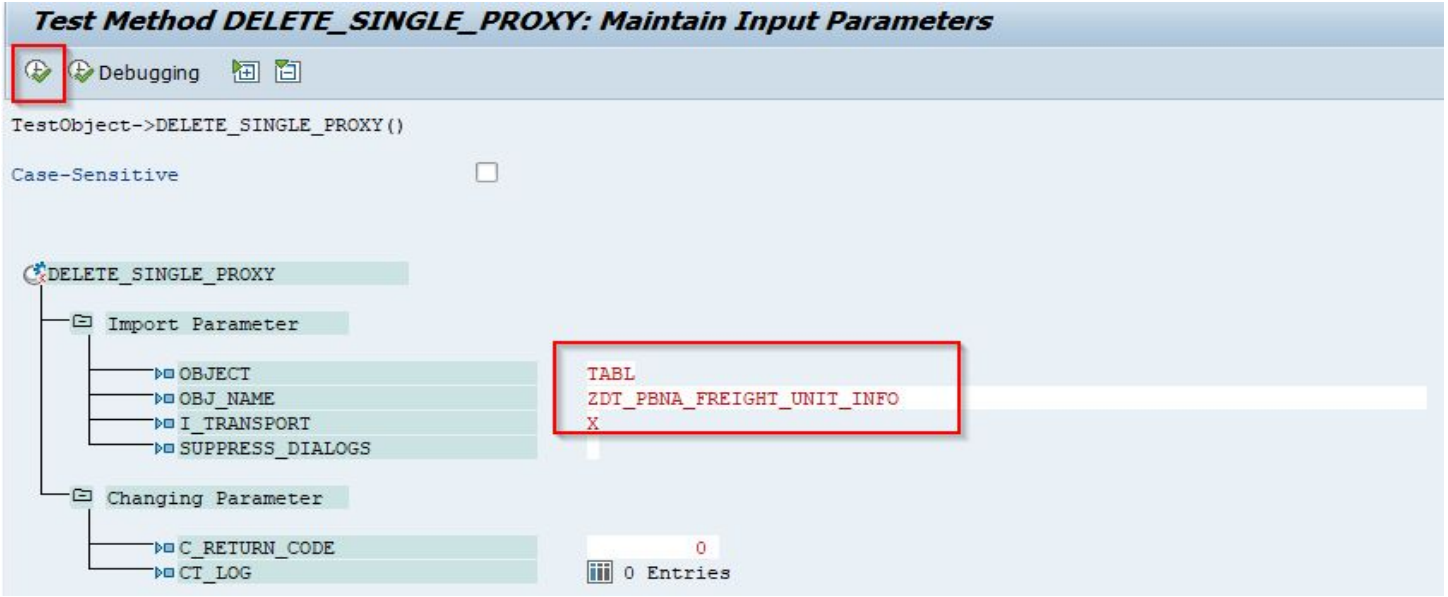
Attributes | Components | Input Help/Check | Currency/quantity fields

Built-In Type — 1 / 3

Component	Typing Method	Component Type	Data Type	Length	Deci...	Coordinate	Short Description
CONTROLLER	Types	PRXCTRLTAB		0	0	0	Control Flags for Fields of a Structure
MESSAGE_HEADER	Types	ZDT_PBNA_FREIGH...		0	0	0	Proxy Structure (generated)
TRANSPORTATION_...	Types	ZDT_PBNA_FREIGH...		0	0	0	

Type ZDT_PBNA_FREIGHT_UNIT_INFO is a generated proxy type (see documentation)

a. Provide the parameters for OBJECT (for example: 'TABL','CLAS', 'DTEL' or 'TTYP'....)
b. Provide the ABAP Name via parameter OBJ_NAME of the proxy.
c. Set TRANSPORT to 'X' to include the deletion also in a transport request.
d. Click on Execute(F8) as highlighted below



Proxy deletion captured in Target TR

Test Method DELETE_SINGLE_PROXY: Maintain Input Parameters

Debugging

TestObject->DELETE_SINGLE_PROXY()

Case-Sensitive

DELETE_SINGLE_PROXY

- Import Parameter
  - OBJECT — TABL
  - OBJ_NAME — ZDT_PBNA_FREIGHT_UNIT_INFO
  - I_TRANSPORT — X
  - SUPPRESS_DIALOGS
- Changing Parameter
  - C_RETURN_CODE — 0
  - CT_LOG — 0 Entries

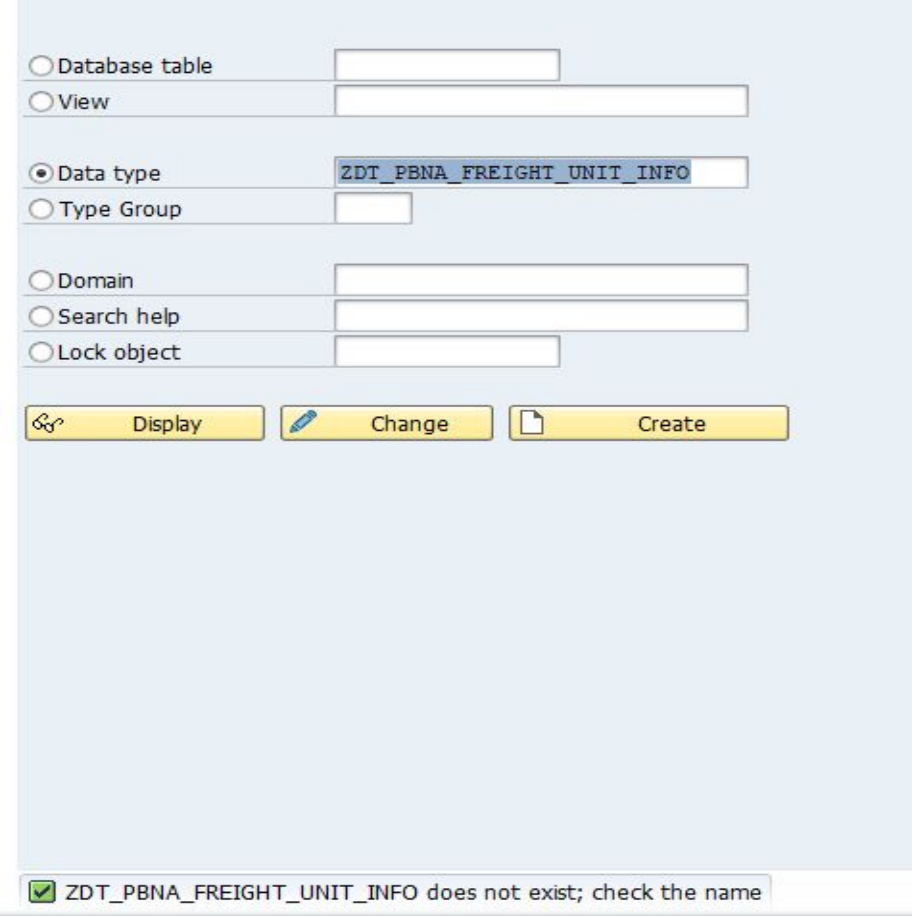
Prompt for transportable workbench request

Table	ZDT_PBNA_FREIGHT_UNIT_INFO
Request	T1DK900746 Workbench request
Short Description	GD_C_DEV_GLB_APD_R_0000001_7000038533: S4 GL...

Own Requests

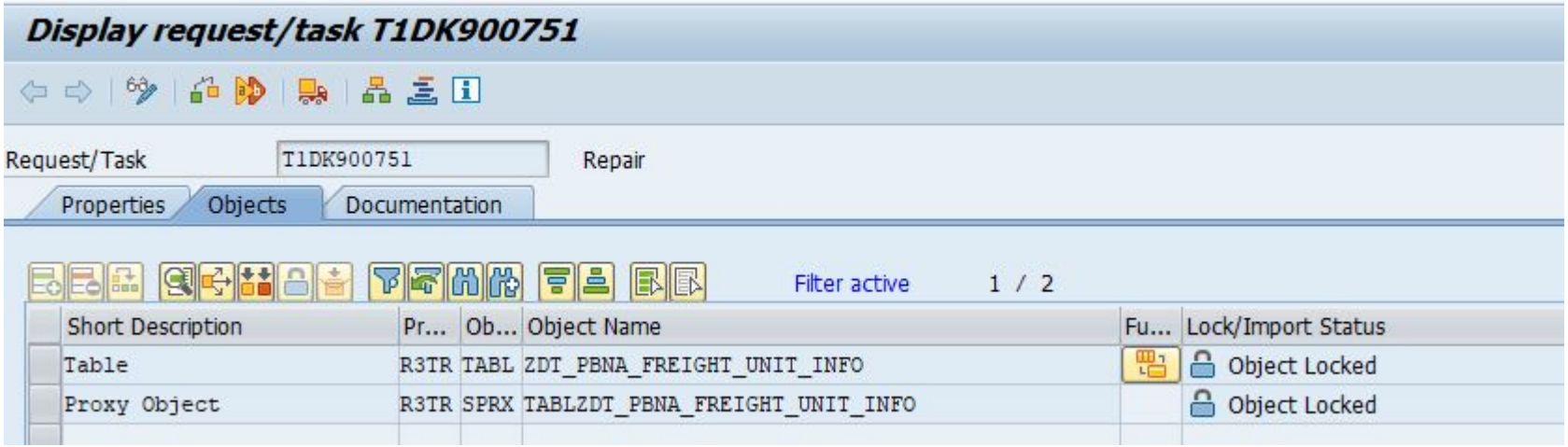
Proxy object Before Deletion

Proxy Object deleted in the system now.



Proxy deletion capture in TR

Proxy object and its related SPRX entry are captured in TR after deletion.



FUGR

Function Group

- In Scope sheet, if FUGR entry exists then all sub objects of respective Function group will also be considered as part of retrofit scope.
- FUGR can have various types of sub objects that can be classified into Function Modules(FUNC),Includes(REPS), Screens(DYNP),GUI Definition(CUAD),DOCU,Function Group Texts(FUGT) and Text elements(REPT).
- RSSYSCOMP program helps to identify the list of sub objects of respective FUGR.
- In Scope sheet in column, the FUGR entry is considered as **Main_Scope** and all respective subobjects of FUGR are listed under **R3TR_Scope**.
- Function group in retrofit scope can be either a normal Custom FUGR or TMG related FUGR.

Retrofit process of FUGR:

Scenario 1: New(ToC)

If FUGR is New in scope,import through ToC from source. As FUGR is imported,all sub objects of respective New FUGR will be imported as well.

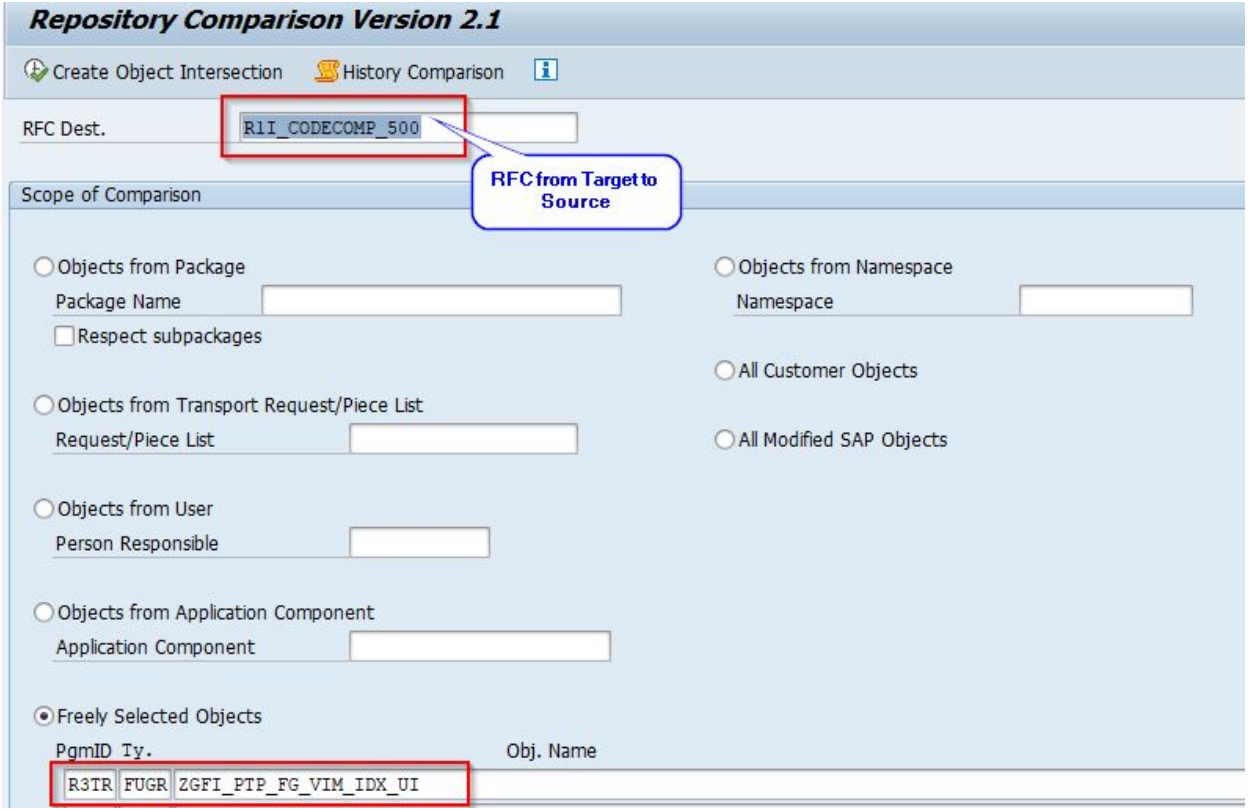
After import, for each new sub-object we have to apply tool rules wherever applicable.

→ You can refer to source_parsed file and see if any tool rule is applied to the object, then we can update the same to target side and maintain the status and comments in scoping sheet as :

- ➢ [in case No tool rules for object in source_parsed file]
  Retrofit Done-Imported via ToC.
- ➢ [in case Tool rules are available for object in source_parsed file]
  Retrofit Done-Imported via ToC, and applied tool rules.

FUGR - RSSYSCOMP

Execute RSSYCOMP in Target system SE38 and provide RFC and FUGR as below.



FUGR - RSSYSCOMP result

It lists out all subobjects of the FUGR

Comparison Result

Object Comparison Display Object

FUGR	ZGFI_PTP_FG_VIM_IDX_UI	DYNP	SAPLZGFI_PTP_FG_VIM_IDX_UI	2005	LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	DYNP	SAPLZGFI_PTP_FG_VIM_IDX_UI	2006	LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	DYNP	SAPLZGFI_PTP_FG_VIM_IDX_UI	2007	LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	DYNP	SAPLZGFI_PTP_FG_VIM_IDX_UI	2008	LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	DYNP	SAPLZGFI_PTP_FG_VIM_IDX_UI	3100	LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	DYNP	SAPLZGFI_PTP_FG_VIM_IDX_UI	3300	LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	DYNP	SAPLZGFI_PTP_FG_VIM_IDX_UI	3400	LIMU	Not Equal To
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	DYNP	SAPLZGFI_PTP_FG_VIM_IDX_UI	6226	LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	DYNP	SAPLZGFI_PTP_FG_VIM_IDX_UI	9100	LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	FUNC	ZGFI_PTP_FM_VIM_IDX_SYN_APPL		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	FUNC	ZGFI_PTP_FM_VIM_IDX_SYN_LABELS		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPS	LZGFI_PTP_FG_VIM_IDX_UIF01		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPS	LZGFI_PTP_FG_VIM_IDX_UIF02		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPS	LZGFI_PTP_FG_VIM_IDX_UII01		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPS	LZGFI_PTP_FG_VIM_IDX_UII02		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPS	LZGFI_PTP_FG_VIM_IDX_UII03		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPS	LZGFI_PTP_FG_VIM_IDX_UIO01		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPS	LZGFI_PTP_FG_VIM_IDX_UITOP		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPS	LZGFI_PTP_FG_VIM_IDX_UIUXX		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPS	SAPLZGFI_PTP_FG_VIM_IDX_UI		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPT	LZGFI_PTP_FG_VIM_IDX_UIF01		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPT	LZGFI_PTP_FG_VIM_IDX_UIF02		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPT	LZGFI_PTP_FG_VIM_IDX_UII01		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPT	LZGFI_PTP_FG_VIM_IDX_UII02		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPT	LZGFI_PTP_FG_VIM_IDX_UII03		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPT	LZGFI_PTP_FG_VIM_IDX_UIO01		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPT	LZGFI_PTP_FG_VIM_IDX_UITOP		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPT	LZGFI_PTP_FG_VIM_IDX_UIUXX		LIMU	Identical
FUGR	ZGFI_PTP_FG_VIM_IDX_UI	REPT	SAPLZGFI_PTP_FG_VIM_IDX_UI		LIMU	Identical

FUGR

Scenario 2: Already Existed - Manual retrofit

- In Scoping sheet, filter Main object with respective FUGR and it lists out FUGR entry along all the sub objects of that FUGR in Scoping sheet.
- For subobjects Function Modules(FUNC) and Includes(REPS) retrofit is done using offline tool results by comparing Scope TR as reference in Source system with Active Target system.
- Subobjects like Screens(DYNP),GUI Definition(CUAD),DOCU,Function Group Texts(FUGT) and Text elements(REPT) can be analyzed under Safe to Import and they fall under service scope.
- If FUGR is a TMG related, then DYNP are adjusted in Target by regenerating TMG for respective Tables/Views that are part of that FUGR in Scope.

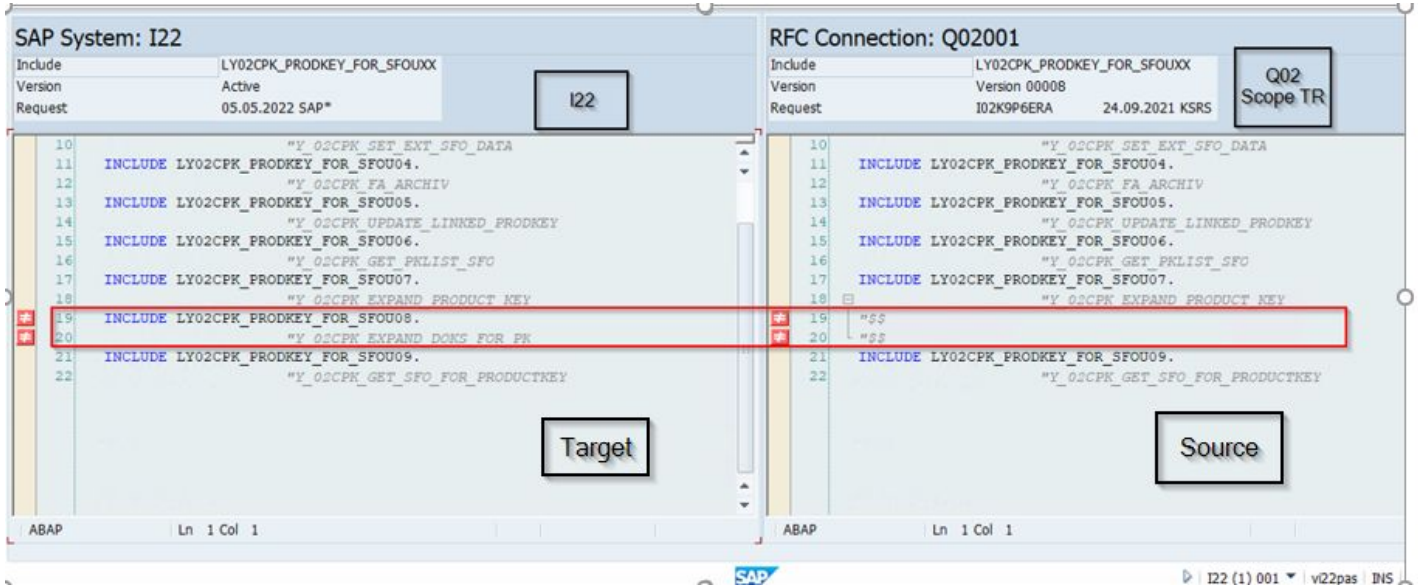
Points to Note:

- In case of TMG related includes/Generated includes - never change them manually/apply the tool rules even if they are suggested in tool results.
- When any differences found in UXX Include of any FUGR, then the following analysis must be done if any FM's are added or deleted in the FUGR.

FM deleted in FUGR UXX Include in current scope

Here FM belongs to FUGR is deleted but Deletion entry of FM is not captured in any Scope TR with deletion flag.But when compared UXX include of the FUGR, the FM is deleted in Source.

The analysis is done on UXX Include of the FUGR and then it has to be marked for Customer Review seeking confirmation to delete FM in the Target.



FUNC

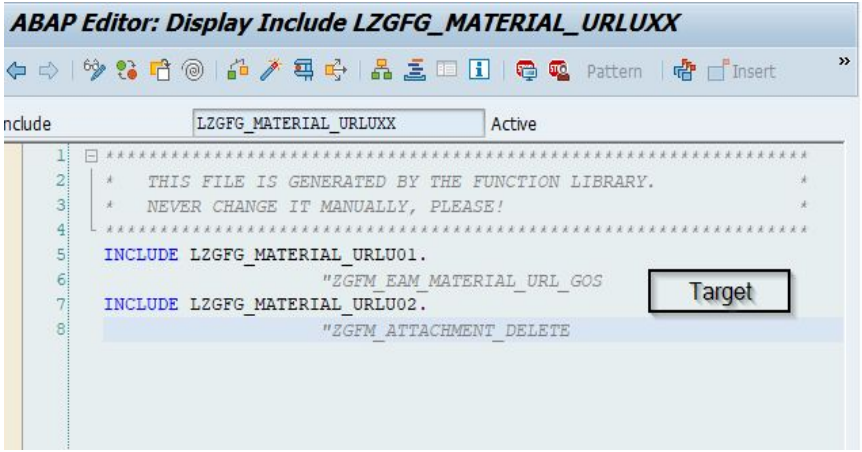
Function Module

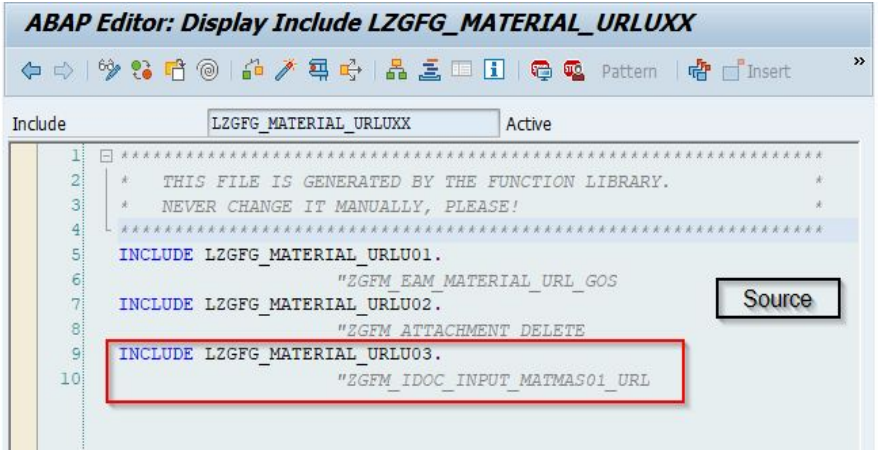
Retrofit process of Function Module:

Scenario 1: New(ToC)

- If FUNC is New in Scope, import through ToC from source. After Import FUGR has to be adjusted using Repair Function group functionality in SE37, then the FM will be added to UXX Include of respective FUGR.

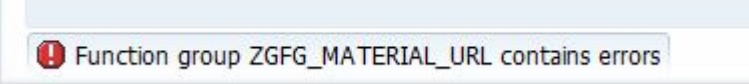
  In Below FUGR **ZGFG_MATERIAL_URL**,highlighted FM in Source is New in scope and imported by ToC to Target.





New FUNC Repair step 1

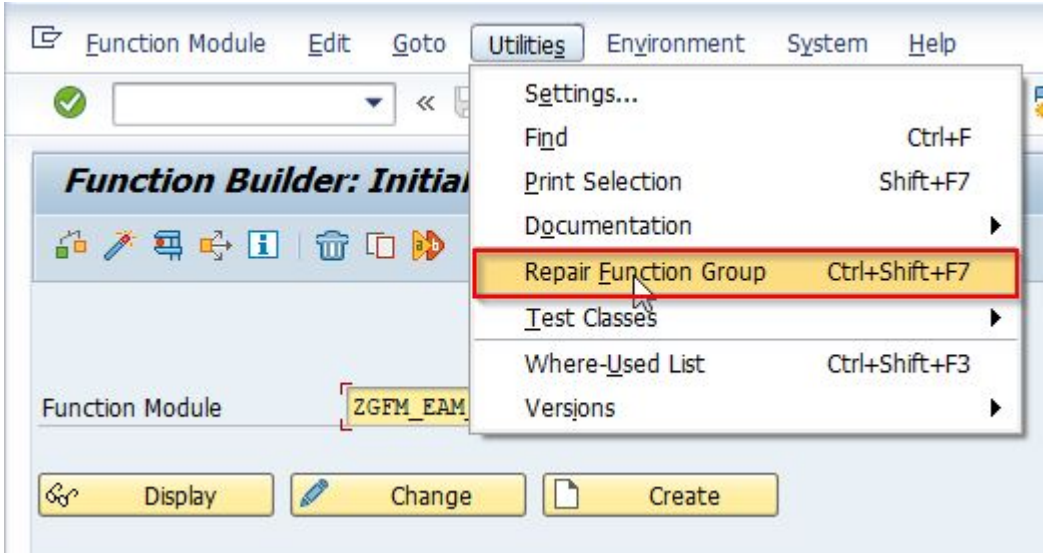
- After ToC import, when tried open FM in SE37 in Target,below error appears. This can be resolved by FUGR repair.



Steps to Repair FUGR in Target after New FM imported:

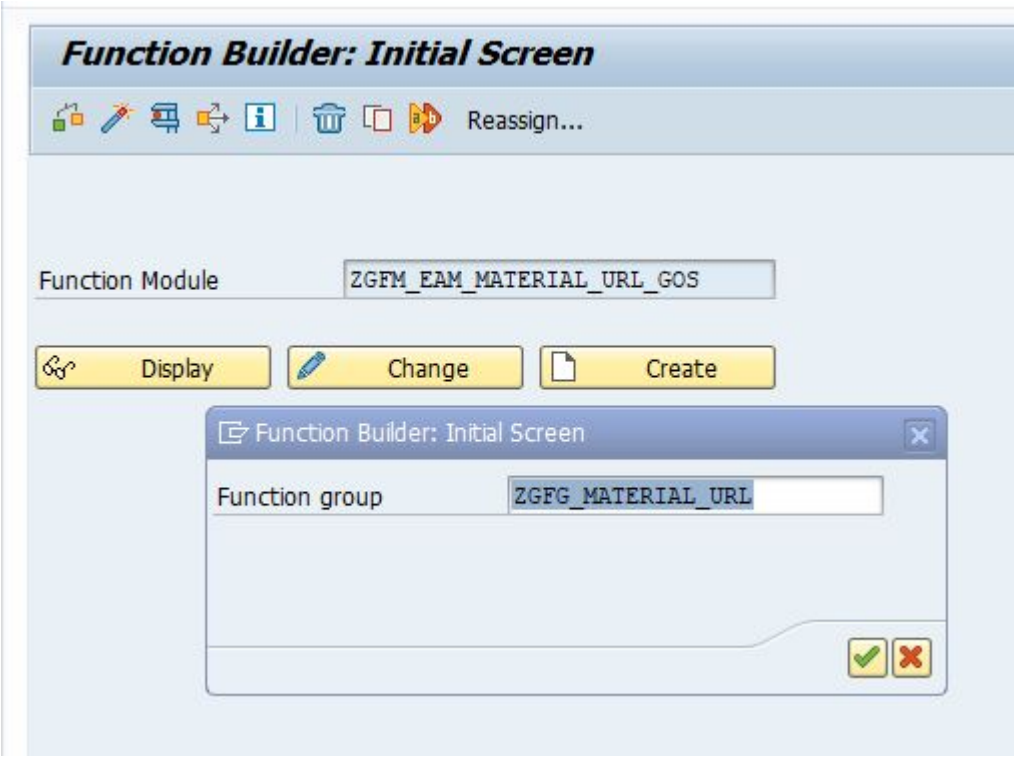
a. Find another existing FM of the same FUGR in Target.Navigate to SE37 and input FM.

b. In Menu, Click on Utilities and press Repair Function Group option.



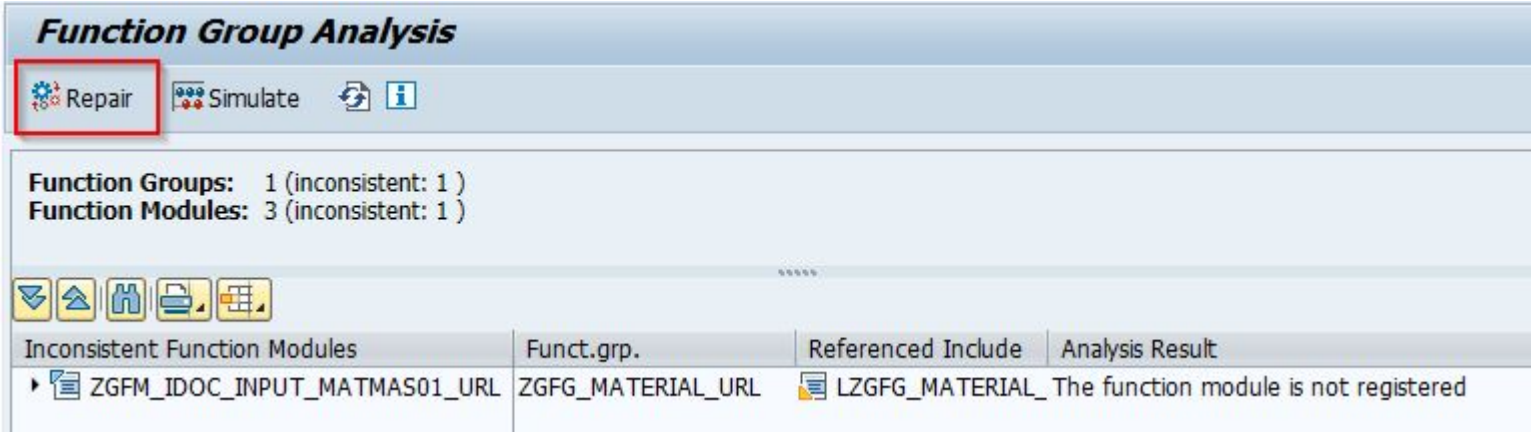
New FUNC Repair step 2

After clicking Repair Function group option,pop up appears with actual FUGR name as below and press Continue.



New FUNC Repair step 3

The below screen appears as it shows New FM listed. Click on Repair button, so that the inconsistencies will be resolved.



Now for each new sub-object we have to apply tool rules wherever applicable.

→ You can refer to source_parsed file and check if any tool rule is applied to the object, then we can update the same to target side and maintain the status and comments in scoping sheet as :

- [in case No tool rules for object in source_parsed file]
  Retrofit Done-Imported via ToC.
- [in case Tool rules are available for object in source_parsed file]
  Retrofit Done-Imported via ToC, and applied tool rules.

FUNC

Scenario 2: Manual merge

Using Offline Tool results,the conflicts and changes in FM are resolved by comparing Scope TR as reference in Source system with Active Target system using Comparison Tool as below..

Source Parsed | Browse...

```
1 FUNCTION /lungh/bw_og_set_plnmg .
2 *"----------------------------------------------------------
3 *"*"Lokale Schnittstelle:
4 *"  IMPORTING
5 *"      VALUE(IV_COMMAND) TYPE  CHAR01 DEFAULT 'R'
6 *"      VALUE(IT_PLNMG) TYPE  /LUNGH/BW_TT_MPT_PLANMN
7 *"  EXCEPTIONS
8 *"       DB_WRITE_ERROR
9 *"----------------------------------------------------------
10  DATA: lv_message_text TYPE string,
11        lv_text1 TYPE symsgv,
12        lv_text2 TYPE symsgv,
13        lv_strlen TYPE i,
14        lr_excptn_ref   TYPE REF TO cx_dynamic_check
15
16  DATA lt_s986 TYPE /lungh/bw_tt_hist_data.
17  DATA: gr_db_table TYPE REF TO data.
18  FIELD-SYMBOLS: <gt_db_table> TYPE ANY TABLE.
19
20  DATA lv_hist_table TYPE tabname VALUE 'S986'.
21
22  IF /lungh/rp_cl_switch=>is_active( iv_switch_id
23                                     iv_switch_subid
24    lv_hist_table = 'ZBW_CDSRIS_HIST'.
25  ENDIF.
26
27  TRY.
28      CREATE DATA gr_db_table TYPE TABLE OF (lv_hist_
29    CATCH cx_sy_create_data_error INTO lr_excptn_ref
30      lv_message_text = lr_excptn_ref->get_text( ).
31      lv_strlen = strlen( lv_message_text ).
32      lv_text1 = lv_message_text+0(lv_strlen).
33      IF lv_strlen > 50.
34        lv_text2 = lv_message_text+50.
35      ENDIF.
36      MESSAGE i000(zbw_og) WITH lv_text1 lv_text2 DI
```

Merged | Browse...

```
1 FUNCTION /lungh/bw_og_set_plnmg .
2 *"----------------------------------------------------------
3 *"*"Lokale Schnittstelle:
4 *"  IMPORTING
5 *"      VALUE(IV_COMMAND) TYPE  CHAR01 DEFAULT 'R'
6 *"      VALUE(IT_PLNMG) TYPE  /LUNGH/BW_TT_MPT_PLANMN
7 *"  EXCEPTIONS
8 *"       DB_WRITE_ERROR
9 *"----------------------------------------------------------
10  DATA: lv_message_text TYPE string,
11        lv_text1 TYPE symsgv,
12        lv_text2 TYPE symsgv,
13        lv_strlen TYPE i,
14        lr_excptn_ref   TYPE REF TO cx_dynamic_check
15
16  DATA lt_s986 TYPE /lungh/bw_tt_hist_data.
17  DATA: gr_db_table TYPE REF TO data.
18  FIELD-SYMBOLS: <gt_db_table> TYPE ANY TABLE.
19
20  DATA lv_hist_table TYPE tabname VALUE 'S986'.
21
22  IF /lungh/rp_cl_switch=>is_active( iv_switch_id
23                                     iv_switch_subid
24    lv_hist_table = 'ZBW_CDSRIS_HIST'.
25  ENDIF.
26
27  TRY.
28      CREATE DATA gr_db_table TYPE TABLE OF (lv_hist_
29    CATCH cx_sy_create_data_error INTO lr_excptn_ref
30      lv_message_text = lr_excptn_ref->get_text( ).
31      lv_strlen = strlen( lv_message_text ).
32      lv_text1 = lv_message_text+0(lv_strlen).
33      IF lv_strlen > 50.
34        lv_text2 = lv_message_text+50.
35      ENDIF.
36      MESSAGE i000(zbw_og) WITH lv_text1 lv_text2 DI
```

Target | Browse...

```
1 FUNCTION /lungh/bw_og_set_plnmg .
2 *"----------------------------------------------------------
3 *"*"Lokale Schnittstelle:
4 *"  IMPORTING
5 *"      VALUE(IV_COMMAND) TYPE  CHAR01 DEFAULT 'R'
6 *"      VALUE(IT_PLNMG) TYPE  /LUNGH/BW_TT_MPT_PLANMN
7 *"  EXCEPTIONS
8 *"       DB_WRITE_ERROR
9 *"----------------------------------------------------------
10  DATA lt_s986 TYPE TABLE OF s986.
11
12  CASE /lunbc/gh_cl_system_check=>get_instance( )->g
13    WHEN 'HR'. "=> HR Aktionen
14
15 *=>  HR muss noch abgeklärt werden, welche Felder re
16
17
18    WHEN OTHERS.
19
20      CHECK it_plnmg[] IS NOT INITIAL.
21
22      SELECT * FROM s986 INTO TABLE lt_s986 FOR ALL
23        WHERE sptag = it_plnmg-sptag
24        AND   werks = it_plnmg-werks
25        AND   matnr = it_plnmg-matnr.
26
27      LOOP AT it_plnmg ASSIGNING FIELD-SYMBOL(<ls_pl
28
29
30
31
32        TRY.
33            ASSIGN lt_s986[ matnr = <ls_plnmg>-matnr
34            IF sy-subrc <> 0.
35              APPEND INITIAL LINE TO lt_s986 ASSIGNI
36              <ls_s986> = CORRESPONDING #( <ls_plnmg
```

Points to note for FUNC object:

FUGT

Function Group Texts

Function group Texts are Text Elements of Function Group. FUGT retrofit process is same as REPT object type.

FUGX

Function Group with Customer Include: Customer Part

- FUGX in the Scope indicates Custom objects related to the standard FUGX to be handled as part of the retrofit.
- These Includes are Customer Exit Includes starts with ‘ZX’ in the Scope under REPS object type.
- The same PROG/REPS retrofit process is followed for these object.

TRKORR	PROGRAM_ID	OBJECT_TYPE	OBJECT NAME	OBJTYPE_SCOPE	Main object
B1DK900059	LIMU	REPS	ZXVEDU13	FUGR	XVED
B1DK902693	LIMU	REPS	ZXVEDU04	FUGR	XVED
B1DK912400	LIMU	REPS	ZXVEDU06	FUGR	XVED
B1DK912400	LIMU	REPS	ZXVEDU11	FUGR	XVED
B1DK912473	LIMU	REPS	ZXVEDU02	FUGR	XVED
B1DK926288	LIMU	REPS	ZXVEDU03	FUGR	XVED
B1DK926639	LIMU	REPS	ZXVEDU15	FUGR	XVED
B1DK935022	LIMU	REPS	ZXVEDU01	FUGR	XVED
H1DK924973	R3TR	REPS	ZXVEDU09	FUGR	XVED
H1DK924973	LIMU	REPS	ZXVEDU09	FUGR	XVED
H1DK924973	LIMU	REPT	ZXVEDU09	FUGR	XVED

PDWS,PDTS,PDAC - Workflow objects

Object Type	Transaction Code
PDWS(Workflow template)	SWDD
PDTS(Standard Task)	PFTC
PDAC(Standard rule)	PFAC

Retrofit process:

Scenario 1: New object(Toc)

- If object is New in Scope, we would add them to ToC and import to Target.

PDWS,PDTS,PDAC - Workflow objects

Scenario 2: Safe to Import(ToC)

Case1: In SE03, for an object identify if there is any common TR exists in both Source and Target systems by Sorting Descending with date. If there is a common TR between Source and Target and in source there are additional TRs exists on above versions of common TR then the object can be considered as Safe to Import.

Case2: If there is a common TR between Source and Target for an object, but the Target has latest TR as smartShift remediation or previous retrofit, then object has to compared manually screen to screen comparison of all tabs for PDTS,PDAC object types and validate if there are any target changes.

Case3: If there are changes identified in manual comparison, object needs to marked for Customer Action as there is no version history available for Workflow objects.

DDLS

Data Definition Language Source(CDS Views)

Prerequisites:

CDS Views retrofit requires Eclipse access for both Source and target systems to identify and retrofit changes in Target system. As there is no remote comparison possible for comparing CDS between source and target, it requires any text comparison tool(Eg:Meld) to perform retrofit..

Retrofit process:

- If DDLS is New, it is imported through ToC.If it is Existed in target,requires Manual code merge.
- Compare DDLS object code of Active Target version with Scope TR version in source system using Tool Comparison tool(Meld).
- Validate each of the identified code conflict whether happened in source or target.
- Compare the versions in both Source and Target respectively and finalize in which system the changes has happened and bring those changes to Target side in comparison tool.
- If Target side has latest changes confirmed after checking versions in both source and target systems then retain Target side changes in the comparison tool.
- Now update the finalized changes in Target system with Syntax check and save in the given Target TR.

Source and Target DDLS in Eclipse

[R1I] ZGCVC_FLASHSALES_MATINFO | [T1D] ZGCVC_FLASHSALES_MATINFO

```
1  /*************************************************************************
2  *       CDS Name        : ZGCVC_FLASHSALES_MATINFO                       *
3  *       [illegible]     : [illegible]                                    *
4  *       [illegible]     : [illegible]                                    *
5  *       Created on      : 02/24/2022                                     *
6  *       RICEF           : E003253                                        *
7  *       PROJECT         : FLNA                                           *
8  *       FD Name         : FSD_RTR_ACO_FLNA_RTR_FLUS_Extraction Program   *
9  *       TR              : H1DK934384                                     *
10 *       Version         : 1.0                                            *
11 *       Description     : FLASH SALES IC STO EXTRACTION                  *
12 *-----------------------------------------------------------------------*
13 *  Modification Log:                                                    *
14 *-----------------------------------------------------------------------*
15 * MOD#    | Date      |Programmer|    CTS    | Description             *
16 *-----------------------------------------------------------------------*
17 * [illegible]                                                           *
18 * [illegible]                                                           *
19 * [illegible]                                                           *
20 * [illegible]                                                           *
21 *-----------------------------------------------------------------------*/
22
23 @AbapCatalog.sqlViewName: 'ZGC_FLS_MATINFO'
24 @AbapCatalog.compiler.compareFilter: true
25 @AbapCatalog.preserveKey: true
26 @AccessControl.authorizationCheck: #CHECK
27 @EndUserText.label: 'Material Information for Flash Sales'
28 define view ZGCVC_FLS_MATINFO
29 as select from matdoc as a
30
31 //left outer join mbew as b        --MOD-001
32 left outer join v_mbew_md as b  //++MOD-001
33 on  a.matnr = b.matnr
34 and a.werks = b.bwkey
35
36 left outer join ekko as c
```

Meld Comparison of DDLS code

Source | Browse...

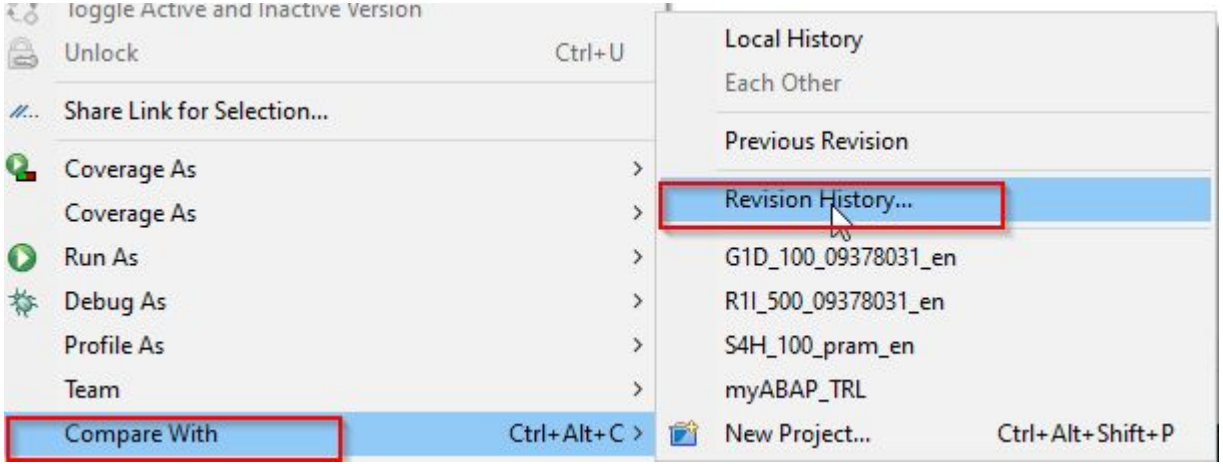
```
23 @AbapCatalog.sqlViewName: 'ZGC_FLS_MATINFO'
24 @AbapCatalog.compiler.compareFilter: true
25 @AbapCatalog.preserveKey: true
26 @AccessControl.authorizationCheck: #CHECK
27 @EndUserText.label: 'Material Information for Flash Sales'
28 define view ZGCVC_FLS_MATINFO
29 as select from matdoc as a
30
31 //left outer join mbew as b       --MOD-001
32 left outer join v_mbew_md as b  //++MOD-001
33 on  a.matnr = b.matnr
34 and a.werks = b.bwkey
35
36 left outer join ekko as c
37 on a.ebeln = c.ebeln
38
39 left outer join mara as d
40 on a.matnr = d.matnr
41
42 left outer join t001w as e
43 on a.werks = e.werks
44
45 left outer join mvke as f
46 on  a.matnr = f.matnr
47 and a.bukrs = f.vkorg
48
49 left outer join t001k as g
50 on g.bwkey = ltrim(a.wempf, '0')
51
```

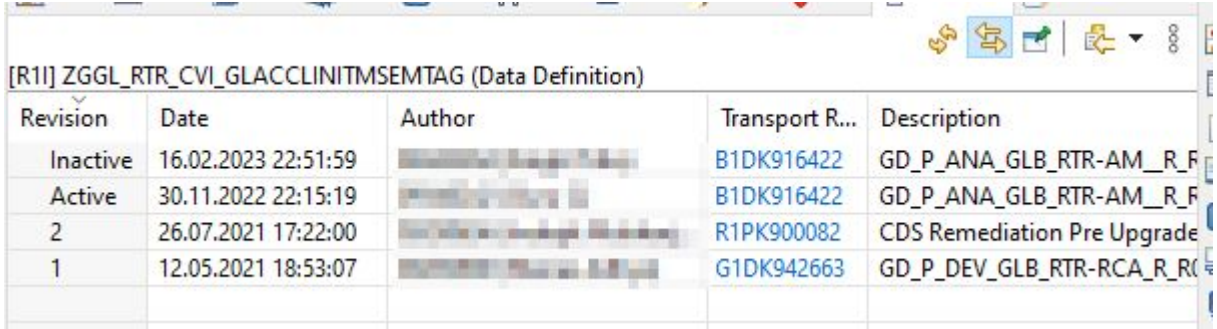
Target | Bro

```
21 *-----------------------------------------------------------------*/
22
23 @AbapCatalog.sqlViewName: 'ZGC_FLS_MATINFO'
24 @AbapCatalog.compiler.compareFilter: true
25 @AbapCatalog.preserveKey: true
26 @AccessControl.authorizationCheck: #CHECK
27 @EndUserText.label: 'Material Information for Flash Sales'
28 define view ZGCVC_FLS_MATINFO
29 as select from matdoc as a
30
31 //left outer join mbew as b       --MOD-001
32 left outer join v_mbew_md as b  //++MOD-001
33 on  a.matnr = b.matnr
34 and a.werks = b.bwkey
35
36 left outer join ekko as c
37 on a.ebeln = c.ebeln
38
39
40 left outer join t001w as e
41 on a.werks = e.werks
42
43 left outer join mvke as f
44 on  a.matnr = f.matnr
45 and a.bukrs = f.vkorg
46
47 left outer join t001k as g
48 on g.bwkey = ltrim(a.wempf, '1')
49
```

DDLS Versions comparison

- For version history of CDS view, right click in the CDS view in the system that requires to be checked. Expand ‘Compare with’ and then click on ‘Revision history’ as displayed below.





TDAT

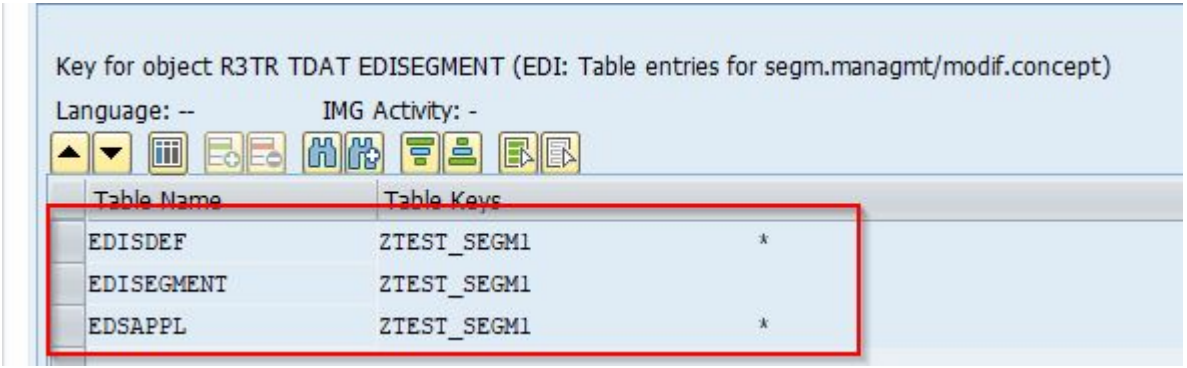
Customizing: Table Contents

TDAT entries are generated as part of IDoc segments creation or changes made to existing IDoc segment and captured in TR.

- IDoc segments are captured in TR as just TABL entry when created or made changes.
  But only when segment is released in WE31,TDAT entries are generated in TR.
- If New IDoc segment to be imported as NEW then IDoc TABL and respective TDAT entries together must be added to ToC.
- If TDAT entries are not added to ToC and just TABL entry is moved to target, then IDoc is imported as only Structure in Target causing failure in IDoc segment creation.

Short Description	Pr...	Ob...	Object Name	Fu...	Lock/Import Status
Table	R3TR	TABL	ZTEST_SEGM1		Object Locked
Customizing: Table Contents	R3TR	TDAT	EDISEGMENT		

TDAT



Note : As of now for both NEW and EXISTING Idoc segments, Idoc segments are handled manually and not as part of ToC import.

Exception : for Syngenta TDAT entries along with the TABL was considered as part of ToC after doing safe to import analysis in the way we do for TABU entries-In case when we found conflict(same segment name, but entirely different fields in source and target), that was reported to the customer- and as part of resolution, customers have created a new segment for existing target segment and we have imported source segment.

DSYS

Chapter of a Book Structure - Tcode - SE18

DSYS object type indicates the BADI documentation of BADI Definition in SE18.
This object type is created when user creates a documentation for any customer BADI Defintion and captured in TR as **R3TR DSYS SIMGZGBDD_STATUS_NOTIF**.

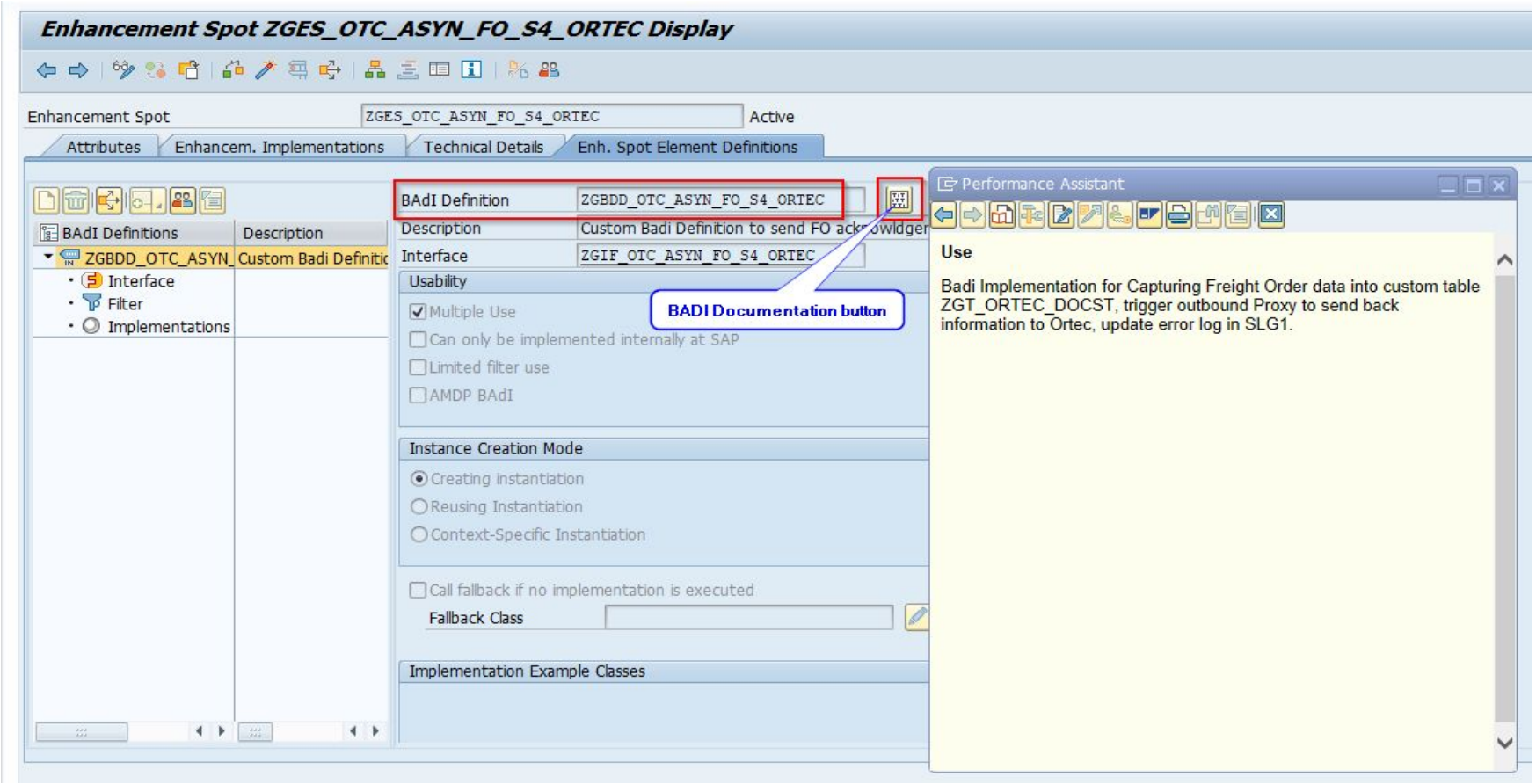
The text after prefix **SIMG** in object name indicates BADI Definition name.

Retrofit process of DSYS:

Scenario 1: New(ToC)

Since BADI Definition is part of Enhancement Spot(ENHS),identify any DSYS entry exists in scope and check if respective BADI Defintion exists in Target system.If BADI definition not exists in Target then identify respective ENHS object exists in scope as New. As ENHS is New then BADI Definition also gets imported along with ENHS. Hence respective DSYS entry can also be added to ToC as New..

DSYS


Enhancement Spot ZGES_OTC_ASYN_FO_S4_ORTEC Display
Enhancement Spot
ZGES_OTC_ASYN_FO_S4_ORTEC
Active
Attributes
Enhancem. Implementations
Technical Details
Enh. Spot Element Definitions
BAdI Definitions
Description
ZGBDD_OTC_ASYN_
Custom Badi Definitic
Interface
Filter
Implementations
BAdI Definition
ZGBDD_OTC_ASYN_FO_S4_ORTEC
Description
Custom Badi Definition to send FO ack owldger
Interface
ZGIF_OTC_ASYN_FO_S4_ORTEC
Usability
Multiple Use
Can only be implemented internally at SAP
Limited filter use
AMDP BAdI
BADI Documentation button
Instance Creation Mode
Creating instantiation
Reusing Instantiation
Context-Specific Instantiation
Call fallback if no implementation is executed
Fallback Class
Implementation Example Classes
Performance Assistant
Use
Badi Implementation for Capturing Freight Order data into custom table ZGT_ORTEC_DOCST, trigger outbound Proxy to send back information to Ortec, update error log in SLG1.

DSYS

Change : SIMGZGBDD_OTC_ASYN_FO_S4_ORTEC Language EN

SAP

Cut | Copy | Paste | Clipboard

Advanced Find | Replace | Select | Can't Undo | Can't Redo | Editing

Paragraph Reset | Reset Character Formatting | Styles | Styles

Spelling & Grammar | Research | Translate | Thesaurus | Set Proofing Language | Word Count | Proofing

Print Layout | Draft | View

&USE&

Badi Implementation for Capturing Freight Order data into custom table ZGT_ORTEC_DOCST, trigger outbound Proxy to send back information to Ortec, update error log in SLG1.

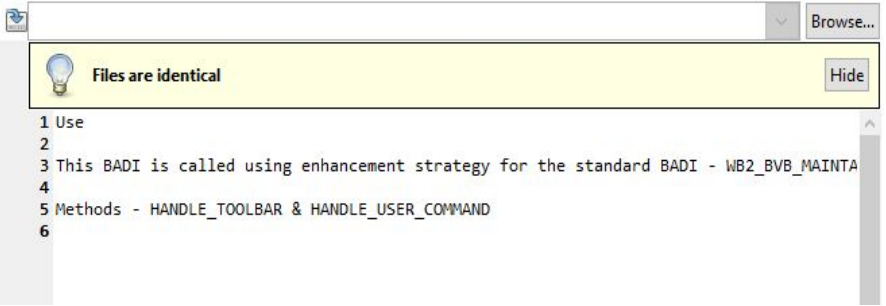
DSYS

Scenario 2: Safe to Import(ToC)

If BADI Definition of DSYS is already exists in Target system,then DSYS can be considered as part of Safe to Import analysis similar to DOCU object type.

Scenario 3: Manual merge

If object is not concluded as Safe to Import, then it requires comparison of documentation and manual merge is done.



Vigneshwari

ENHO - Enhancement Implementation
ENHC - Composite Enhancement Implementation
ENHS - Enhancement Spot
CMOD - Customer enhancement projects
DOCV - Documentation (independent)
SPRX - Proxy Object
DOCU - Documentation
SIRS - Customizing for Content Repositories
DOCT - General Text

ENHO

Object type - ENHANCEMENT IMPLEMENTATION

Procedure :

a) If the object is NEW Scope we will import it via TOC to the respective target system .

b) If the object is in ALREADY_EXISTED Scope then we have to follow the below steps to fix the object

(NOTE : we have to ensure in source system the active version is our scope version . if the active version and Scope version are same then we can proceed the below steps .)

Step 1 : Go to tcode SE38 program – RSSYSCOMP in respective project target system and provide the RFC from target to source and select the radio button “Freely Selected Objects” and provide the object name

Repository Comparison Version 2.1

Create Object Intersection | History Comparison

RFC Dest. TRUSTING@CRW_0020087607

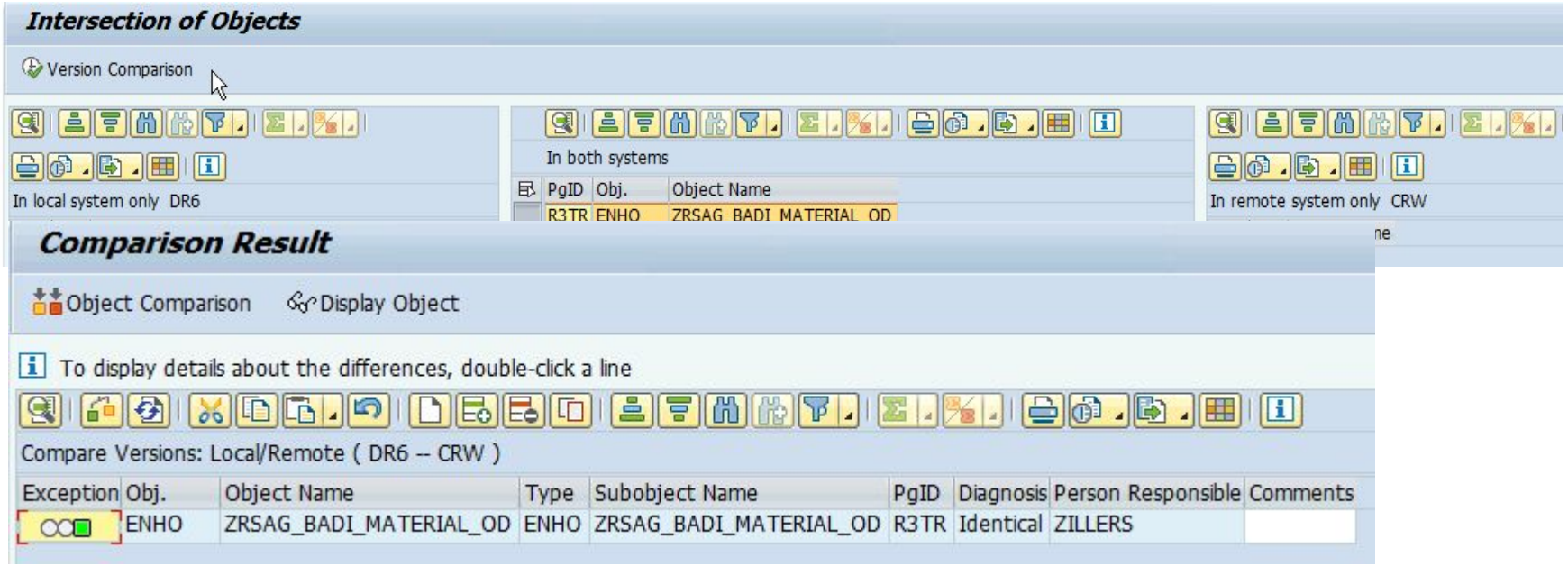
Scope of Comparison

- Objects from Package
  - Package Name
  - Respect subpackages
- Objects from Transport Request/Piece List
  - Request/Piece List
- Objects from User
  - Person Responsible
- Objects from Application Component
  - Application Component
- Objects from Namespace
  - Namespace
- All Customer Objects
- All Modified SAP Objects
- Freely Selected Objects

PgmID	Ty.	Obj. Name
R3TR	ENHO	ZRSAG_BADI_MATERIAL_OD

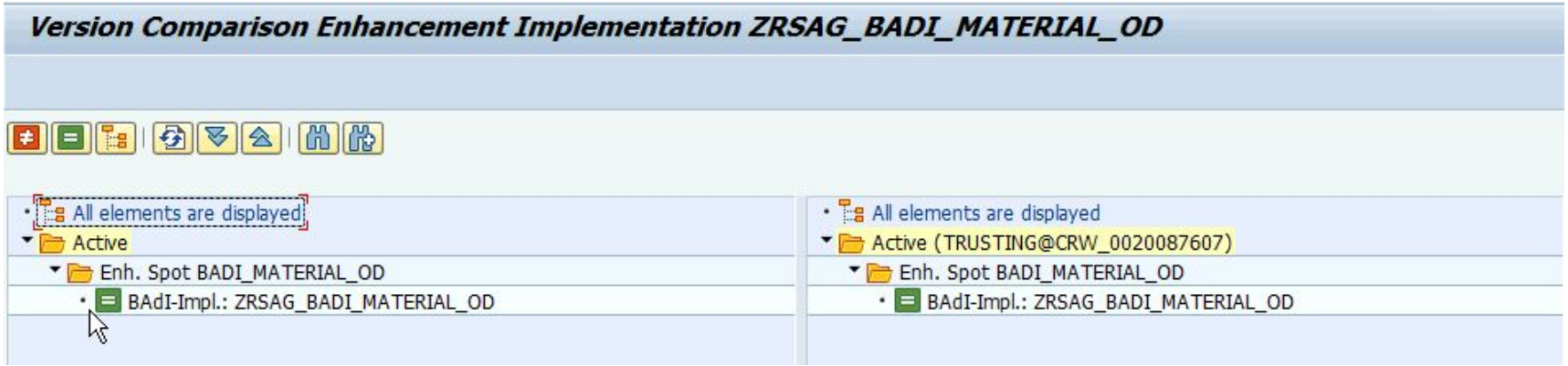
Step 2 : Execute-> version comparison -> then look for status in Diagnosis column

If its IDENTICAL then its NO action case but if it NOT IDENTICAL then we need Fix .
By selecting the object name we will get the comparison screen .



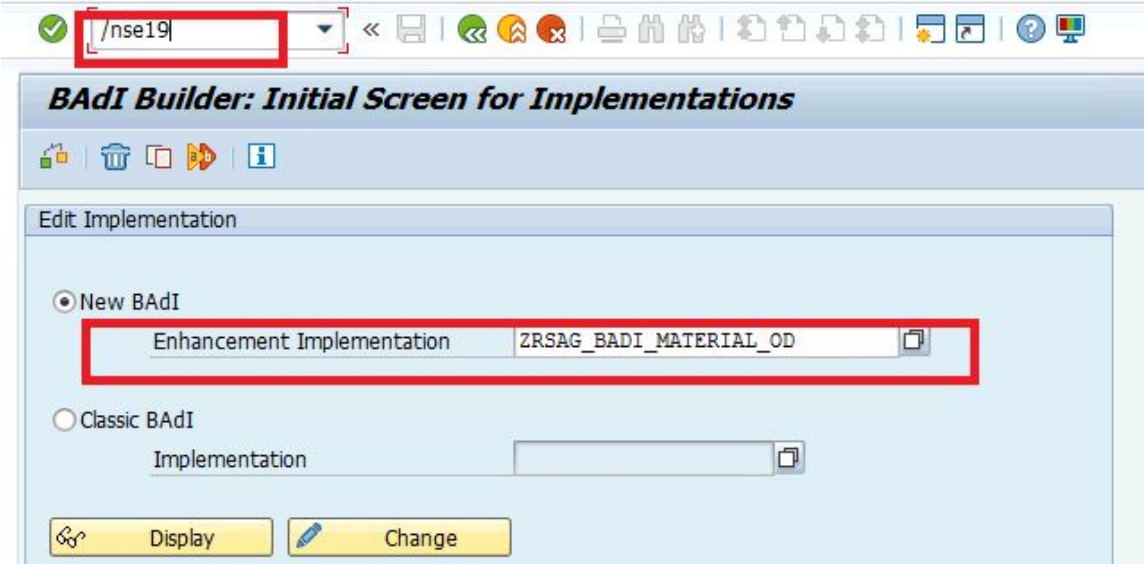
Based on the output of comparison screen we will be adding the NEW changes to the target without over writing the NEW target changes .

Go To TCODE - > SE19 - > enter enhancement implementation name and made the required changes manually



If there any active version TR is not same as our Scope Version TR . then we need to follow the below steps

Go to TCODE - > se19 (in target)-> provide the enhancement name -> select utilities -> version management -> then select “Fetch Remote Comparison” -> provide RFC Destination->then select the target active version and source scope version to compare



Enhancement Implementation | Edit | Goto | Utilities | Environment | System | Help

- Settings...
- Display Object List Ctrl+Shift+F5
- Worklist
- Display Navigation Window
- Where-Used List Ctrl+Shift+F3
- Versions
  - Version Management
  - Generate version
  - Return to active version

Enhancement Implementation Z... Display

Enhancement Implementation ZRSA...

Properties | History | Technical Details | Enh. Implementation Elements

BAdI Implementations	Description
ZRSAG_BADI_MATERIA	Implementierung: Integration
Implementing Class	

BadI Implementation: ZRSAG_BADI_MATERIAL_OD — Documentation

Description: Implementierung: Integration neuer Objekte im Material- bzw. Artikelsta...

☐ Default Implementation

☐ Example Implementation

Customizing: Customizing supported

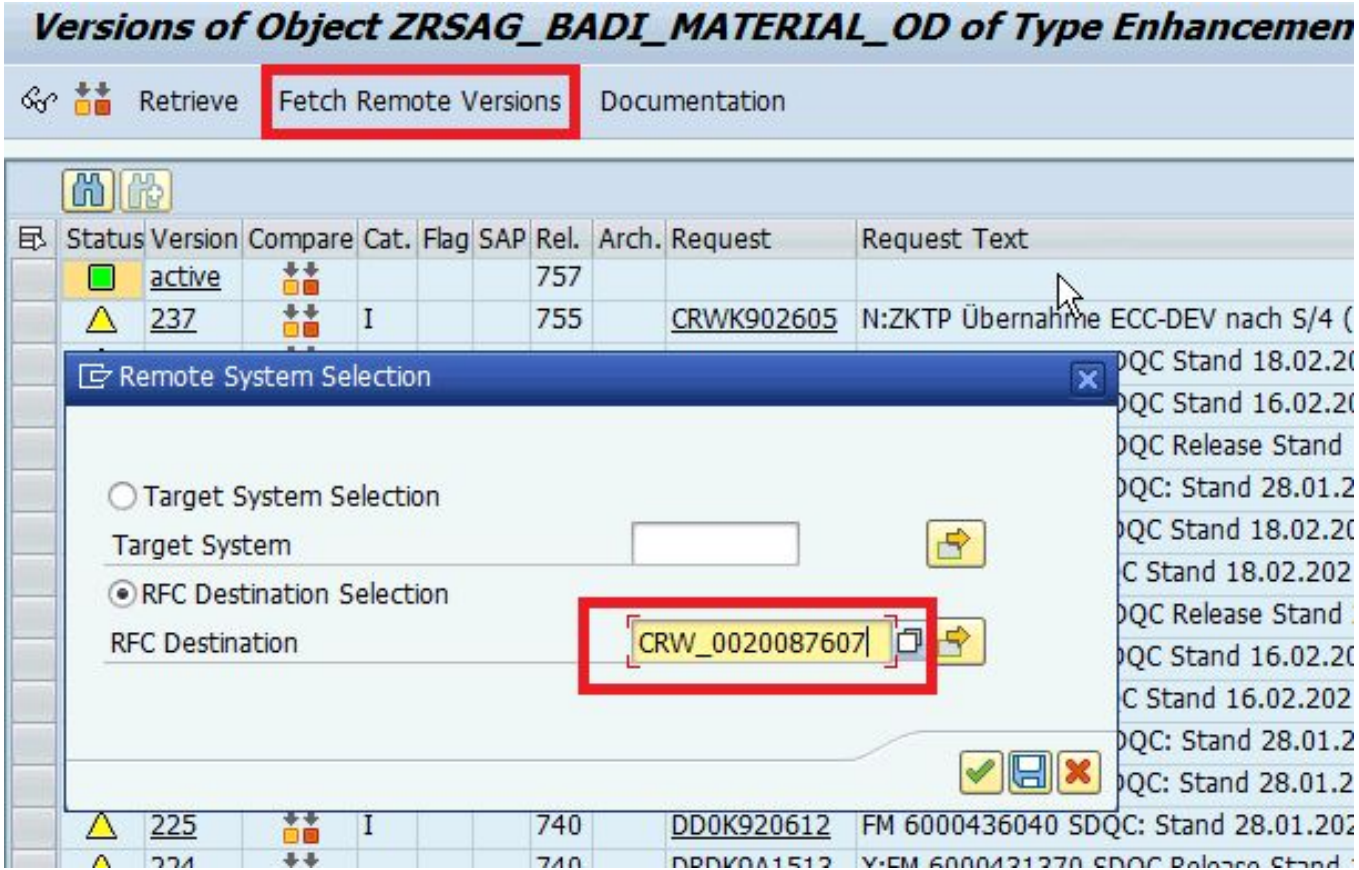
Runtime Behavior

☑ Implementation is active

Runtime Behavior: The implementation will be called

Properties of BAdI Definition

BAdI Definition Name	BADI_MATERIAL_OD
Description	Integration neuer Objekte im Material- bzw. Artikelstamm
Interface	IF_EX_BADI_MATERIAL_OD
Instance Creation Mode	Context-Dependent BAdI


Versions of Object ZRSAG_BADI_MATERIAL_OD of Type Enhancemen
Retrieve
Fetch Remote Versions
Documentation
Status Version Compare Cat. Flag SAP Rel. Arch. Request Request Text
active 757
237 I 755 CRWK902605 N:ZKTP Übernahme ECC-DEV nach S/4 (
Remote System Selection
Target System Selection
Target System
RFC Destination Selection
RFC Destination
CRW_0020087607
225 I 740 DD0K920612 FM 6000436040 SD

Remote Versions of Object ZRSAG_BADI_MATERIAL_OD of Type Enhancement I

Fetch Remote Versions

SID	Status	Version	Compare	Cat.	Flag	SAP	Rel.	Arch.	Request	Request Text	Project	Date	Time	Author	Note	CI
DR6		14					740		DRDK923156	FM 6000025483 RS Auftrag K901678 vom 03.07.2015 /EDM		10.07.2015	08:47:30	HENKEL		
		13					740		DRDK923180	BDHOC FM 6000025483 RS Auftrag K901678 vom 03.07.2015 /EDM		07.07.2015	09:54:19	HENKEL		
		12		I			740		DD0K907715	FM 6000025483 (RS Auftrag K901678 vom 03.07.2015)		06.07.2015	18:12:56	REIMER-KI		
		11					740		DRDK921548	FM 6000020350 Retailsolutions SDQC: Stand 05.06.2015 /EDM		19.06.2015	16:10:37	HENKEL		
		10					740		DRDK921554	BDHOC FM 6000020350 Retailsolutions SDQC: Stand 05.06.2015		15.06.2015	15:13:04	HENKEL		
		9					740		DRDK918861	LOK KO 620-155 RS (Kopie DD0K907402, RS: K901621) /EDM		15.06.2015	15:09:20	HENKEL		
		8		I			740		DD0K907573	retailsolutions SDQC: Stand 5.6.2015 /RS		15.06.2015	14:33:57	REIMER-KI		
		7					740		DRDK918869	BDHOC KO 620-155 RS (Kopie DD0K907402, RS: K901621) /ED		05.05.2015	13:14:25	HENKEL		
		6		I			740		DD0K907404	KO 620-155 RS (Kopie DD0K907402, RS: K901621)		05.05.2015	12:13:48	REIMER-KI		
		5					740		DRDK915178	NIRV FM 6000004941 retailsolutions SDQC: Stand 13.02.2015		27.03.2015	12:53:12	WERNER-ER		
		4					740		DRDK915171	NIRV FM 6000004941 retailsolutions SDQC: Stand 13.02.2015		27.03.2015	12:52:46	WERNER-ER		
		3					740		DRDK915186	FM 6000004941 retailsolutions SDQC: Stand 13.02.2015		20.03.2015	14:53:57	WERNER-ER		
		2					740		DRDK915189	BDHOC FM 6000004941 retailsolutions SDQC: Stand 13.02.2015		19.03.2015	16:46:11	WERNER-ER		
		1		I			740		TESK901581	retailsolutions SDQC: Stand 13.02.2015		17.03.2015	19:02:22	ZILLERS		
CRW		active					740					01.10.2014	01:45:58	TROSSBACH		
		33		I			740		DRDK9A1YUY	D:FM 6000590176 neues Release SD/QC für ECC		03.04.2023	13:14:03	8681628		
		32		I			740		DRDK9A1N5H	D:FM 6000524560 SD/QC Release Stand 31.03.2022		09.05.2022	14:33:17	6499340		
		31		I			740		DRDK9A1IBY	D:FM 6000499552 Release SD/QC 30.11.2021		24.01.2022	15:03:19	6499340		

ENHC

Object type - Composite Enhancement Implementation

Procedure :

a) If the object is NEW Scope we will import it via TOC to the respective target system .

b) If the object is in ALREADY_EXISTED Scope then we have to follow the below steps to fix the object

(NOTE : we have to ensure in source system the active version is our scope version . if the active version and Scope version are same then we can proceed the below steps .)

Step 1 : Go to tcode SE38 program – RSSYSCOMP in respective project target system and provide the RFC

from target to source and select the radio button “Freely Selected Objects” and provide the object name

Repository Comparison Version 2.1

Create Object Intersection | History Comparison

RFC Dest. TRUSTING@CRW_0020087607

Scope of Comparison

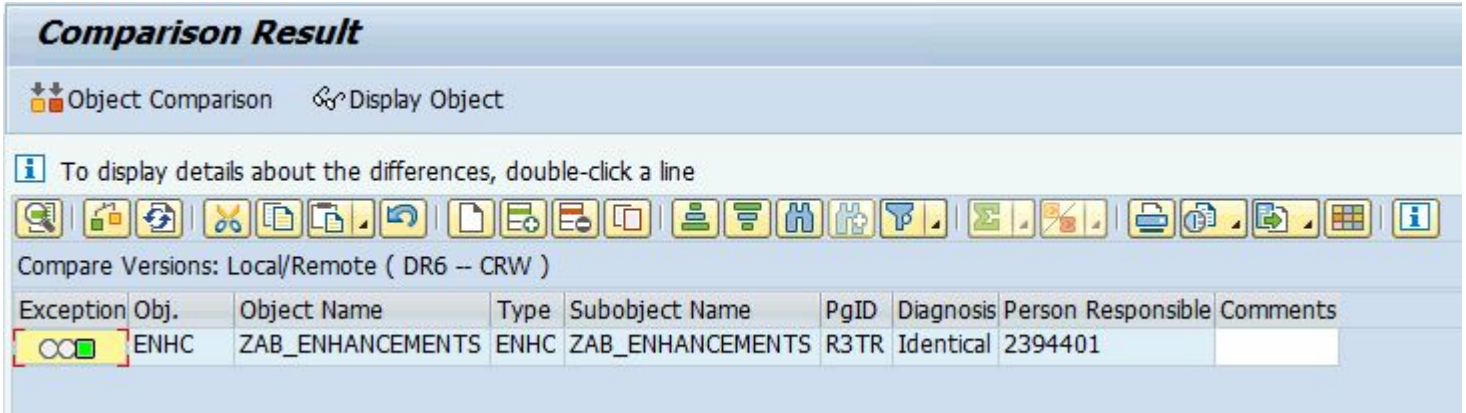
- ○ Objects from Package
  - Package Name
  - ☐ Respect subpackages
- ○ Objects from Transport Request/Piece List
  - Request/Piece List
- ○ Objects from User
  - Person Responsible
- ○ Objects from Application Component
  - Application Component
- ◉ Freely Selected Objects

PgmID	Ty.	Obj. Name
r3tr	ENHC	ZAB_ENHANCEMENTS

- ○ Objects from Namespace
  - Namespace
- ○ All Customer Objects
- ○ All Modified SAP Objects

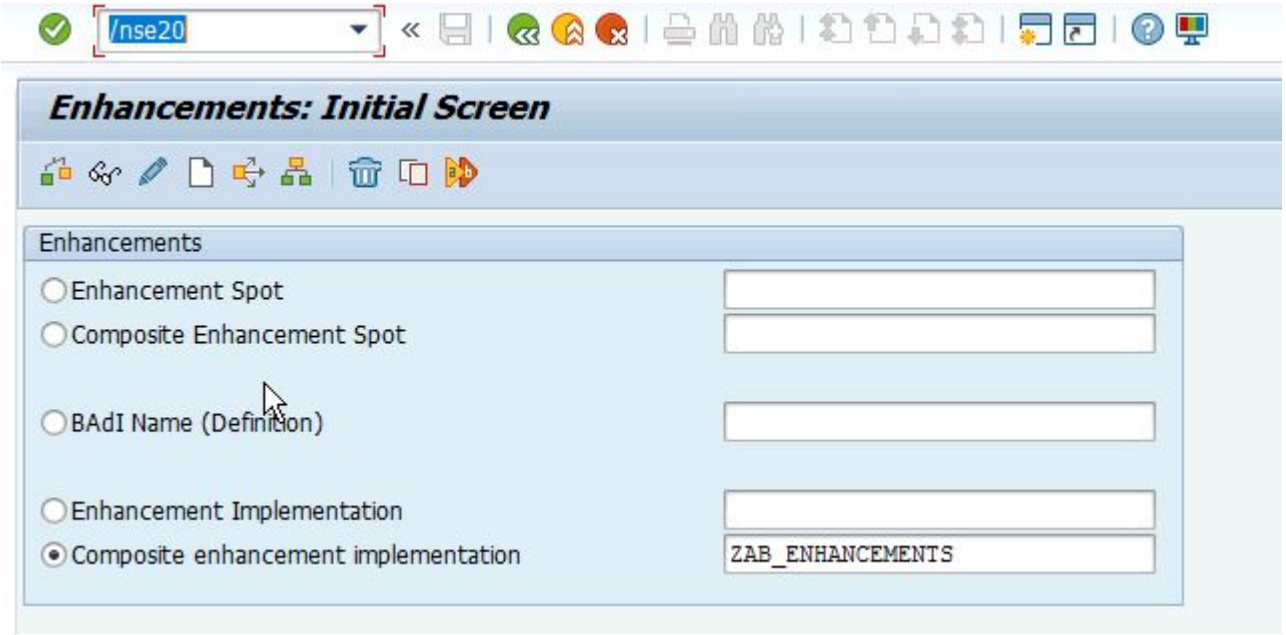
Step 2 : Execute-> version comparison -> then look for status in Diagnosis column

If its IDENTICAL then its NO action case but if it NOT IDENTICAL then we need Fix .
By selecting the object name we will get the comparison screen .



Based on the output of comparison screen we will be adding the NEW changes to the target without over writing the NEW target changes .

Go To TCODE - > SE20- > enter composite enhancement implementation name and made the required changes manually



Enhancement Edit Goto Utilities Environment System Help

Where-Used List Ctrl+Shift+F3

Versions

Version Management

Generate version

Return to active version

Composite Enhancement Implementation: Display

Composite Implementation ZAB_ENHANCEMENTS Active

Attributes | Enhancement Implementations | Composite Implementations

Enhancement Implementation	Short Text	Enhancement Type
ZAB_LKAIPF1T	Edeka-Erweiterung des Includes LKAIPF1T	Redefinition
ZAB_LKAIPF49	Edeka-Erweiterung des Includes LKAIPF49	Redefinition
ZAB_LV50R_VIEWF43	Edeka-Erweiterung des Includes LV50R_VIEWF43	Redefinition
ZAB_LV50R_VIEWH17	Edeka-Erweiterung des Includes LV50R_VIEWH17	Redefinition
ZAB_SAPLV50R_VIEW	Edeka-Erweiterung des Includes LV50R_VIEWG16	Redefinition
ZAB_STATUS_ALV_EH	Status in ALV für EH	Redefinition

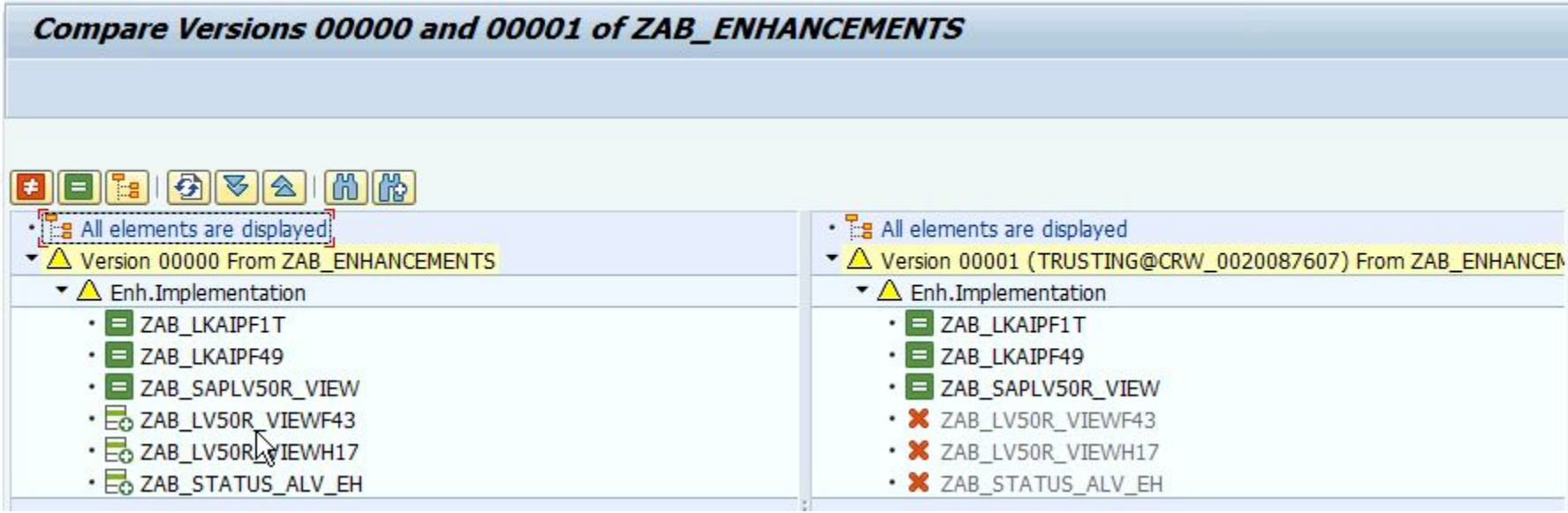
For example we will consider the version 1 TR as our scope version and compared with active version of target then we can find the difference so if the change exist in target is NEW target change by customer where we can conclude that by comparing the active version 3 or 4 earlier version in target

Remote Versions of Object ZAB_ENHANCEMENTS of Type Composite Enhanceme

Fetch Remote Versions

SID	Status	Version	Compare	Cat.	Flag	SAP	Rel.	Arch.	Request	Request Text	Project	Date	Time	Author
DR6		14					740		DRDK9A0TOP	D:KO 19-00328-001 ALV-Einbindung ins EH-Template [REPO]		11.05.2020	14:24:52	KRAEMER-M
		13					740		DRDK9A0V7L	T:KO 19-00328-001 ALV-Einbindung ins EH-Template [REPO]		08.05.2020	09:07:07	KRAEMER-M
		12					740		DRDK9A0V44	T:KO 19-00328-001 ALV-Einbindung ins EH-Template [REPO]		06.05.2020	11:53:37	KRAEMER-M
		11					740		DRDK9A0V40	T:KO 19-00328-001 ALV-Einbindung ins EH-Template [REPO]		06.05.2020	11:32:51	KRAEMER-M
		10					740		DRDK9A0V3W	T:KO 19-00328-001 ALV-Einbindung ins EH-Template [REPO]		06.05.2020	11:08:09	KRAEMER-M
		9					740		DRDK9A0V1O	T:KO 19-00328-001 ALV-Einbindung ins EH-Template [REPO]		04.05.2020	16:50:48	KRAEMER-M
		8					740		DRDK9A0ULP	T:KO 19-00328-001 ALV-Einbindung ins EH-Template [REPO]		21.04.2020	15:44:54	KRAEMER-M
		7					740		DRDK9A0UJ9	T:KO 19-00328-001 ALV-Einbindung ins EH-Template [REPO]		20.04.2020	11:59:23	KRAEMER-M
		6					740		DRDK9A0UIQ	T:KO 19-00328-001 ALV-Einbindung ins EH-Template [REPO]		20.04.2020	10:05:01	KRAEMER-M
		5					740		DRDK9A0UIL	T:KO 19-00328-001 ALV-Einbindung ins EH-Template [REPO]		20.04.2020	09:45:32	KRAEMER-M
		4					740		DRDK9A0UGN	A:KO 19-00328-001 ALV-Einbindung ins EH-Template [REPO]		16.04.2020	17:36:06	KRAEMER-M
		3					740		DRDK9A0REZ	D:FM 6000352364 KA Aufruf VL10(f) - Absprung VA03 [REPO]		02.03.2020	14:18:55	2394401
		2					740		DRDK9A0RND	A:FM 6000352364 KA Aufruf VL10(f) - Absprung VA03 [REPO]		11.02.2020	12:29:24	2394401
		1					740		DRDK9A0RHE	X:FM 6000352364 KA Aufruf VL10(f) - Absprung VA03 [REPO]		06.02.2020	15:37:16	2394401
CRW		active					740					19.05.2020	00:00:00	2394401
		3		I			740		DRDK9A0VND	D:FM 6000352365 KA Fehlerprotokoll Sammellauf Sperre [REPO]		20.07.2020	15:46:06	2394401
		2		I			740		DRDK9A0TOP	D:KO 19-00328-001 ALV-Einbindung ins EH-Template [REPO]		11.05.2020	14:53:42	KRAEMER-M
		1		I			740		DRDK9A0REZ	D:FM 6000352364 KA Aufruf VL10(f) - Absprung VA03 [REPO]		02.03.2020	16:09:37	2394401

If the changes are there in target from initial version then we need ensure the same change exist in source earlier version if it exist in earlier version of source system and removed in active version of source then we need to fix based on the source system .



ENHS

Object type - Enhancement Spot

Procedure :

a) If the object is NEW Scope we will import it via TOC to the respective target system .

b) If the object is in ALREADY_EXISTED Scope then we have to follow the below steps to fix the object

(NOTE : we have to ensure in source system the active version is our scope version . if the active version and Scope version are same then we can proceed the below steps .)

Step 1 : Go to tcode SE38 program – RSSYSCOMP in respective project target system and provide the RFC

from target to source and select the radio button “Freely Selected Objects” and provide the object name

Repository Comparison Version 2.1

Create Object Intersection | History Comparison

RFC Dest. TRUSTING@CRW_0020087607

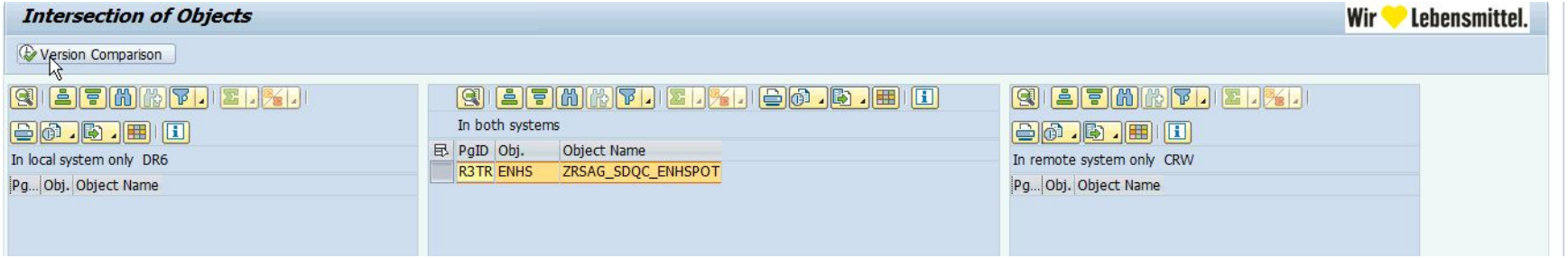
Scope of Comparison

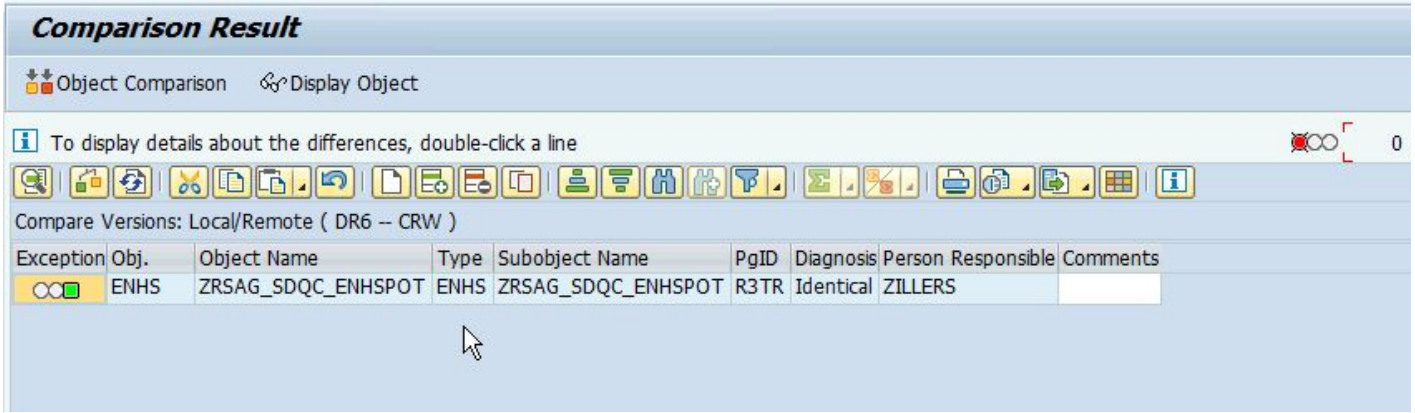
- Objects from Package
  - Package Name
  - Respect subpackages
- Objects from Transport Request/Piece List
  - Request/Piece List
- Objects from User
  - Person Responsible
- Objects from Application Component
  - Application Component
- Objects from Namespace
  - Namespace
- All Customer Objects
- All Modified SAP Objects
- Freely Selected Objects

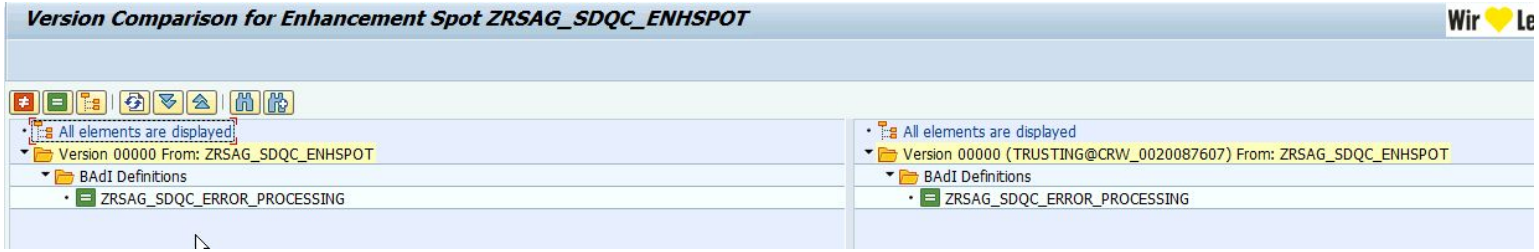
PgmID	Ty.	Obj. Name
R3TR	ENHS	ZRSAG_SDQC_ENHSPOT

Step 2 : Execute-> version comparison -> then look for status in Diagnosis column

If its IDENTICAL then its NO action case but if it NOT IDENTICAL then we need Fix .
By selecting the object name we will get the comparison screen .

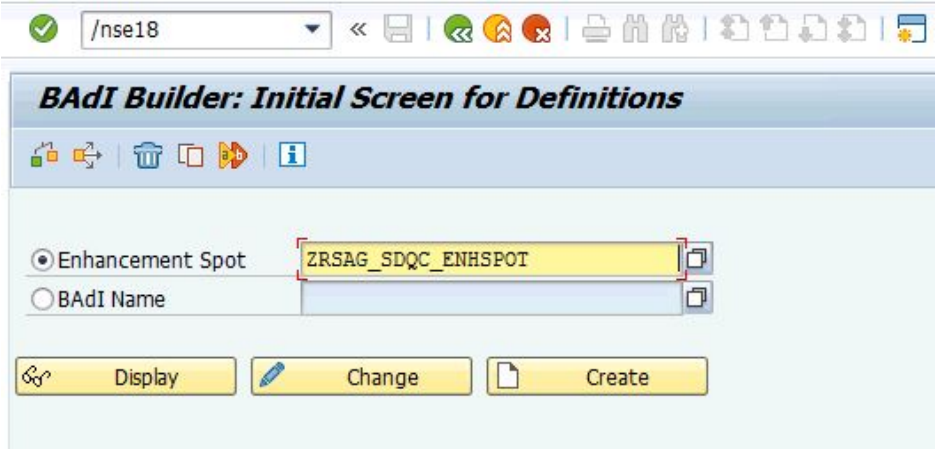






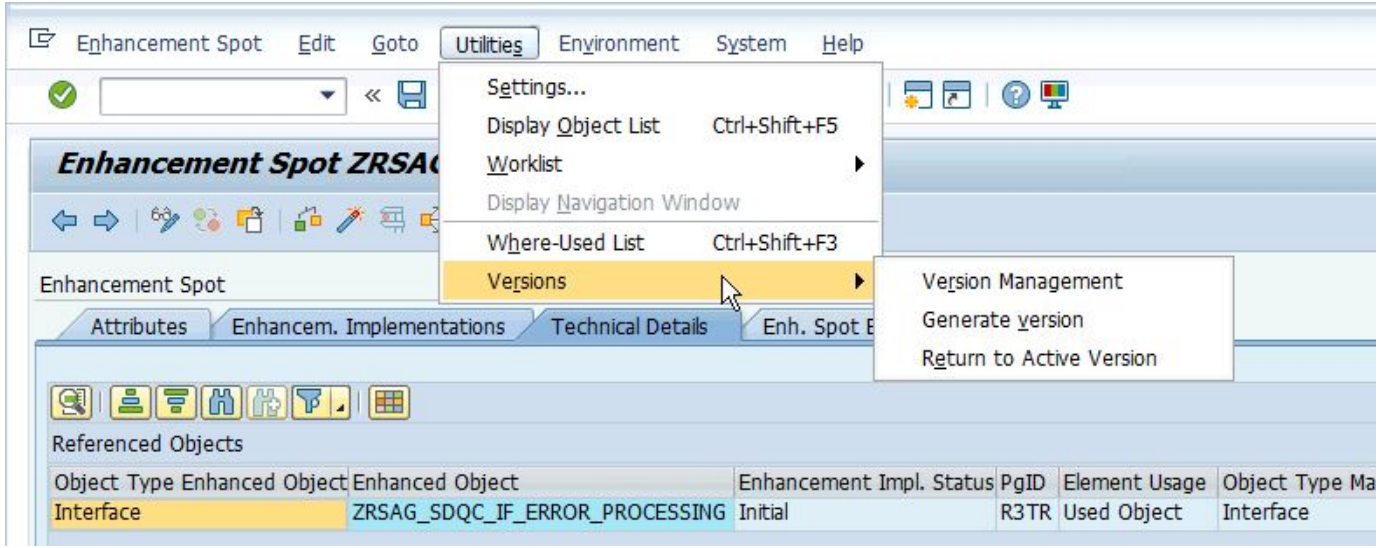
Based on the output of comparison screen we will be adding the NEW changes to the target without over writing the NEW target changes .

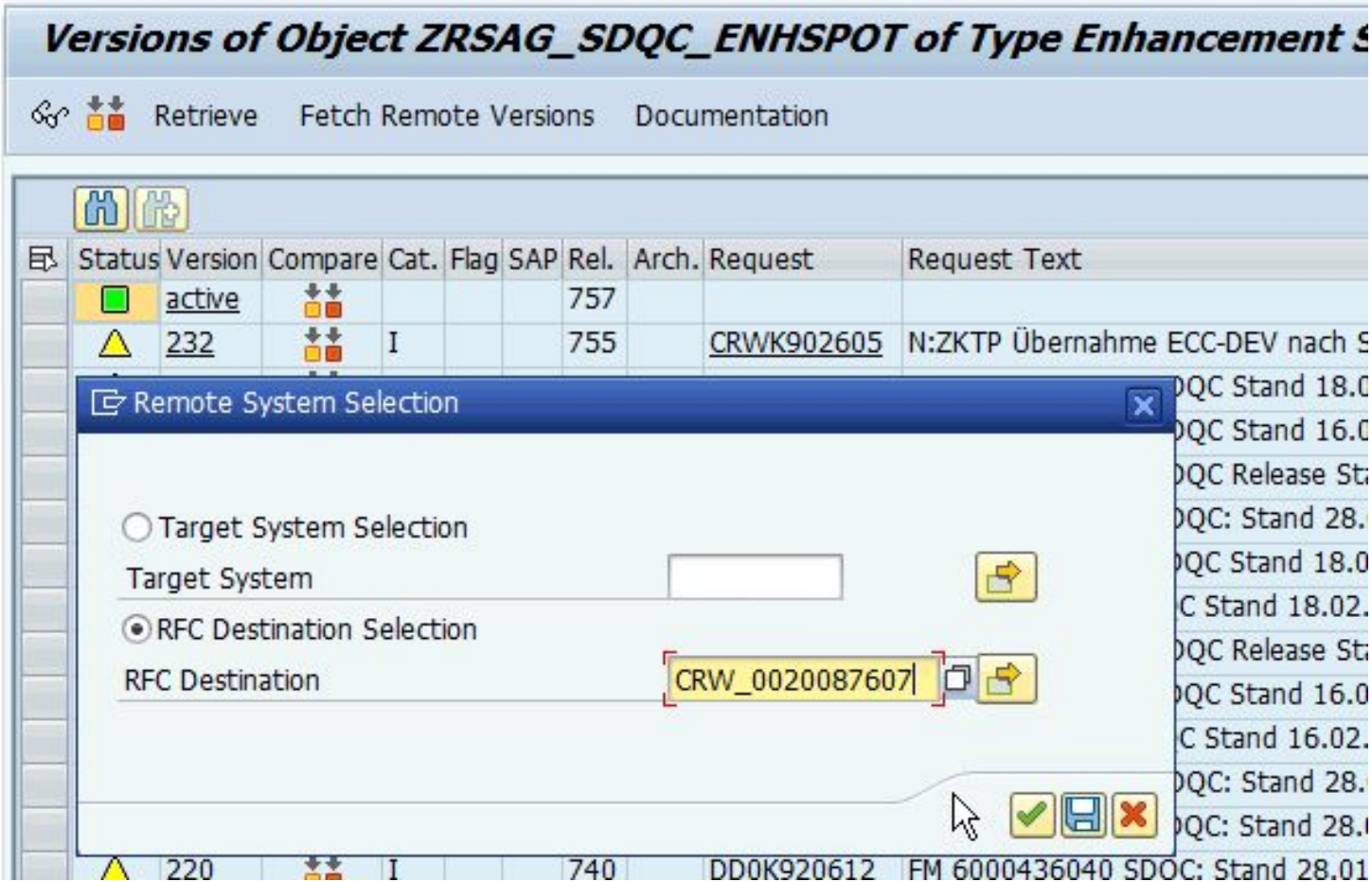
Go To TCODE - > SE18 - > enter enhancement spot name and made the required changes manually



If there any active version TR is not same as our Scope Version TR . then we need to follow the below steps

Go to TCODE - > se18 (in target)-> provide the enhancement name -> select utilities -> version management -> then select “Fetch Remote Comparison” -> provide RFC Destination->then select the target active version and source scope version to compare




Versions of Object ZRSAG_SDQC_ENHSPOT of Type Enhancement S
Retrieve
Fetch Remote Versions
Documentation
Status Version Compare Cat. Flag SAP Rel. Arch. Request Request Text
active 757
232 I 755 CRWK902605 N:ZKTP Übernahme ECC-DEV nach S
Remote System Selection
Target System Selection
Target System
RFC Destination Selection
RFC Destination
CRW_0020087607
220 I 740 DD0K920612 FM 6000436040 SDQC: Stand 28.01

Remote Versions of Object ZRSAG_SDQC_ENHSPOT of Type Enhancement Spot

Fetch Remote Versions

SID	Status	Version	Compare	Cat.	Flag	SAP	Rel.	Arch.	Request	Request Text	Project	Date
DR6		3		I			740		DD0K907573	retailsolutions SDQC: Stand 5.6.2015 /RS		15.0
		2					740		DRDK918869	BDHOC KO 620-155 RS (Kopie DD0K907402, RS: K901621) /ED		05.0
		1		I			740		DD0K907404	KO 620-155 RS (Kopie DD0K907402, RS: K901621)		05.0
CRW		active					740					13.0
		32		I			740		DRDK9A1YUY	D:FM 6000590176 neues Release SD/QC für ECC		03.0
		31		I			740		DRDK9A1N5H	D:FM 6000524560 SD/QC Release Stand 31.03.2022		09.0
		30		I			740		DRDK9A1IBY	D:FM 6000499552 Release SD/QC 30.11.2021		24.0
		29		I			740		DRDK9A1H1Q	D:FM 6000492300 SDQC Release 26.10.2021		29.1
		28		I			740		DRDK9A1GI9	D:FM 6000488320 SDQC Release Stand 07.10.2021		25.1

Version Comparison for Enhancement Spot ZRSAG_SDQC_ENHSPOT

- Differences are displayed
- Version 00000 From: ZRSAG_SDQC_ENHSPOT

- Differences are displayed
- Version 00008 (TRUSTING@CRW_002008

CMOD

Object type - Customer enhancement projects

Procedure

a) If the CMOD object is NEW Scope then we need to move via TOC import from source.
b) If the object exist already we have ensure there are no changes by comparing manually in both systems .
c) For CMOD object type we need to check all the options available like Attributes, Enhancement Assignments and components simultaneously in both systems.
d) If any changes exist then we can move via TOC only after ensuring it is safe to import by checking the version in TCODE -> SE03 .
e) If we didnt find any customer TR in the version list then it is safe to import , but at any customer TR exist then we need to fix the object manually

/nse03

Transport Organizer Tools

- Transport Organizer Tools
  - Objects in Requests
    - Find Objects in Requests/Tasks
    - Analyze Objects in Requests/Tasks
    - Include Objects in a Transport Request
  - Objects
    - Modification Browser
    - Objects in Customer Namespace
    - Namespace Information System
    - Display/Change Maintenance Objects
    - Display Repaired Objects
  - Object Directory

Find Objects in Requests/Tasks

Object Selection

☐ Program	PGMID	R3TR	Obj	PROG		
☐ Function Group	PGMID	R3TR	Obj	FUGR		
☐ Class	PGMID	R3TR	Obj	CLAS		
☐ Table/Structure	PGMID	R3TR	Obj	TABL		
☐ View	PGMID	R3TR	Obj	VIEW		
☐ Table Type	PGMID	R3TR	Obj	TTYP		
☐ Data Element	PGMID	R3TR	Obj	DTEL		
☑ Customer enhancement	PGMID	R3TR	Obj	CMOD	ZRP_IDOC	
☐ ?	PGMID		Obj			
☐ ?	PGMID		Obj			

☐ Search for subobjects too (not if generic entry chosen)

In this Example , Customer changes are there so in spite of importing the via TOC , we need to fix the object manually t, adding the required NEW changes in the target as in Source system.

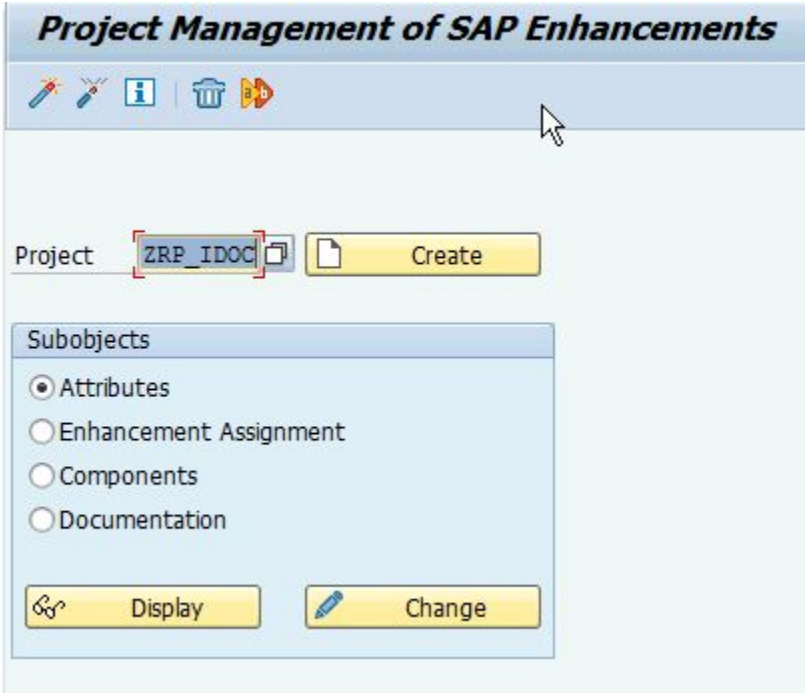
Find Objects in Requests/Tasks

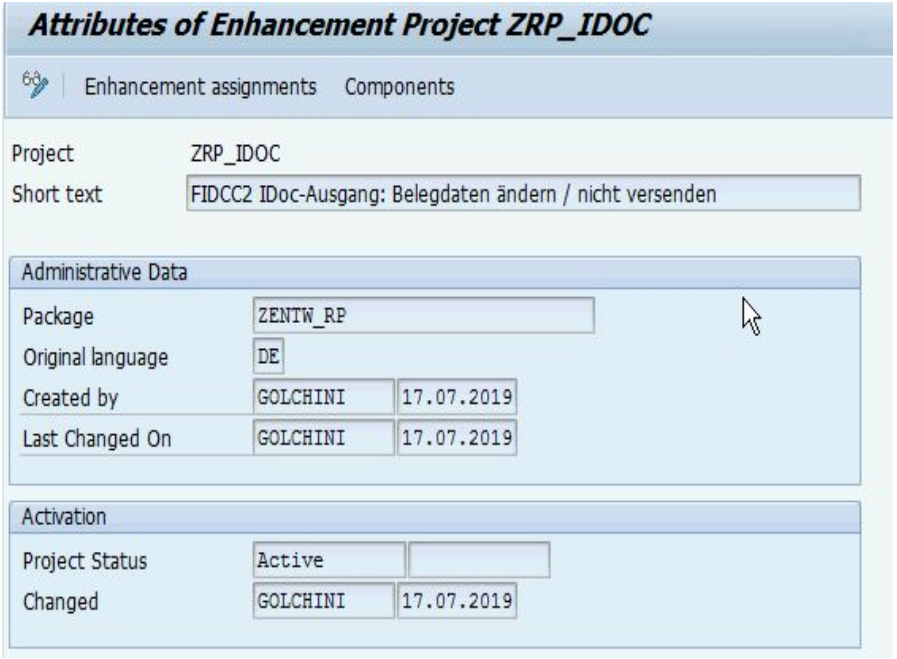
Objects View

Request Overview

Request	Short Description	Owner	Date	Type	Status
DRDK9A0HRY	D:FM 6000321608 Falsches Idoc Rechnungskürzung	8226789	29.07.2019	Workbench Request	Released
DRDK9A0IZJ	T:FM 6000321608 Falsches Idoc Rechnungskürzung	GOLCHINI	17.07.2019	Workbench Request	Released

Go to Tcode-> CMOD in both target and source system and check all properties of the object and add the required changes in to target





SAP Enhancements in Enhancement Project ZRP_IDOC

Enhancement | Components

Enhancement	Text
F050S003	FIDCC2: Change IDoc/do not send

Display ZRP_IDOC

Enhancement assignments | Enhancement

Project				ZRP_IDOC
Enhancement	Impl		Exp	F050S003 FIDCC2: Change IDoc/do not send
Function exit	✓			EXIT_SAPLF050_005 EXIT_SAPLF050_006

DOCT

Object type - General text

Procedure

a) If the object is NEW Scope then we will be importing the object via TOC.
b) For DOCT - ALREADY_EXISTED object also we will import it via TOC .
c) Usually majority of project we will get DOCT object type for message class, reports, class.
d) We can check the object via TCODE->SE61 where we can provide the object type , language and object name .

/nse61

Document Maintenance: Initial Screen

Worklist | Authorizations

Settings

Document Class	Dialog Text
Language	English

Document

Dialog text	

Display | Change | Create

Title (3) 61 Entries found

Restrictions

ID	Cl.	Title
AD		PDF-Based Form
AI		PDF-Based Form - Interface
BF		Business Function + Assignment
BI		Business Add-Ins: Implementations
BS		Business Function Set + Assignment
CA		Class attribute
CC		Configuration Check Group
CE		Class event
CH		Characteristic
CL		Class
CO		Class method
DE		Data element
DT		Dialog Text
DZ		Data element supplement
FU		Function module and parameter
IA		Interface attribute
IE		Interface event
IF		Interface

Title (3) 61 Entries found

Restrictions

ID	Cl.	Title
CA		Class attribute
CC		Configuration Check Group
CE		Class event
CH		Characteristic
CL		Class
CO		Class method
DE		Data element
DT		Dialog Text
DZ		Data element supplement
FU		Function module and parameter
IA		Interface attribute
IE		Interface event
IF		Interface
IO		Interface method
IS		Internet Service
MG		Migration text
MO		Modification text
NA		Message

SPRX

Object type - Proxy Object

a) Usually Proxy objects whether its NEW or Already_Existed both are customer actions
b) The decision might be different based on projects some project we be working on SPRX object which are NEW Scope will be imported via TOC and if it's Already Existing we need to compare between source and target and inform customer to regenerated where we will maintain status as CUSTOMER ACTION.
c) SPRX entries are usually will be having prefix as TABL ,CLAS, TTYP ,INTF which denotes the object type of the proxy object .
d) And the respective dependent object entries in scope sheet will be with IN Scope so we have to ensure those are OUT If we are not performing any changes to SPRX related objects(based projects)

A. CLASS related proxy object entries in scope sheet where we have to ensure all sub objects also consider to perform the necessary steps based on project

B. If it is OUT OF SCOPE and Customer action

C. Will be responsible for NEW Proxy generated objects will be moved and ALREADY_EXISTED objects are maintained as Customer action

PROGR	OBJECT	OBJ_NAME
R3TR	CLAS	ZMEUS_CO_MEBROKER_WEB_SERVICE
R3TR	SPRX	CLASZMEUS_CO_MEBROKER_WEB_SERVICE
LIMU	CINC	ZMEUS_CO_MEBROKER_WEB_SERVICE=CCDEF
LIMU	CINC	ZMEUS_CO_MEBROKER_WEB_SERVICE=CCIMP
LIMU	CINC	ZMEUS_CO_MEBROKER_WEB_SERVICE=CCMAC
LIMU	CLSD	ZMEUS_CO_MEBROKER_WEB_SERVICE
LIMU	CPRI	ZMEUS_CO_MEBROKER_WEB_SERVICE
LIMU	CPRO	ZMEUS_CO_MEBROKER_WEB_SERVICE
LIMU	CPUB	ZMEUS_CO_MEBROKER_WEB_SERVICE
LIMU	METH	ZMEUS_CO_MEBROKER_WEB_SERVICE CONSTRUCTOR
LIMU	METH	ZMEUS_CO_MEBROKER_WEB_SERVICE GET_ANEXO
LIMU	METH	ZMEUS_CO_MEBROKER_WEB_SERVICE GET_ANEXO_ARQUIVO
LIMU	METH	ZMEUS_CO_MEBROKER_WEB_SERVICE GET_ANEXO_DOCUMENTO
LIMU	METH	ZMEUS_CO_MEBROKER_WEB_SERVICE GET_AVISO_EMBARQUE
LIMU	METH	ZMEUS_CO_MEBROKER_WEB_SERVICE GET_CANC_ITEM_REQ
LIMU	METH	ZMEUS_CO_MEBROKER_WEB_SERVICE GET_COLETA_PRECO
LIMU	METH	ZMEUS_CO_MEBROKER_WEB_SERVICE GET_CONTRATO_WF
LIMU	METH	ZMEUS_CO_MEBROKER_WEB_SERVICE GET_ENTREGA
LIMU	METH	ZMEUS_CO_MEBROKER_WEB_SERVICE GET_FILE_CONTENT
LIMU	METH	ZMEUS_CO_MEBROKER_WEB_SERVICE GET_FRS

A. Similarly for Table related proxy object entry in scope sheet will be same action based on project, either it is completely OUT OF SCOPE

B. Or we will be taking responsibility in movement of NEW Objects and for ALREADY_EXISTED will report as customer action

C. There will be situation of taking care of both NEW SCOPE and ALREADY_EXISTED by importing the objects with TOC

Note: The action to be taken entirely depends on after getting confirmation from customer .

PROGR	OBJECT	OBJ_NAME
R3TR	SPRX	TABLZMEUS_ANEXO_NOTA_FISCAL
R3TR	TABL	ZMEUS_ANEXO_NOTA_FISCAL

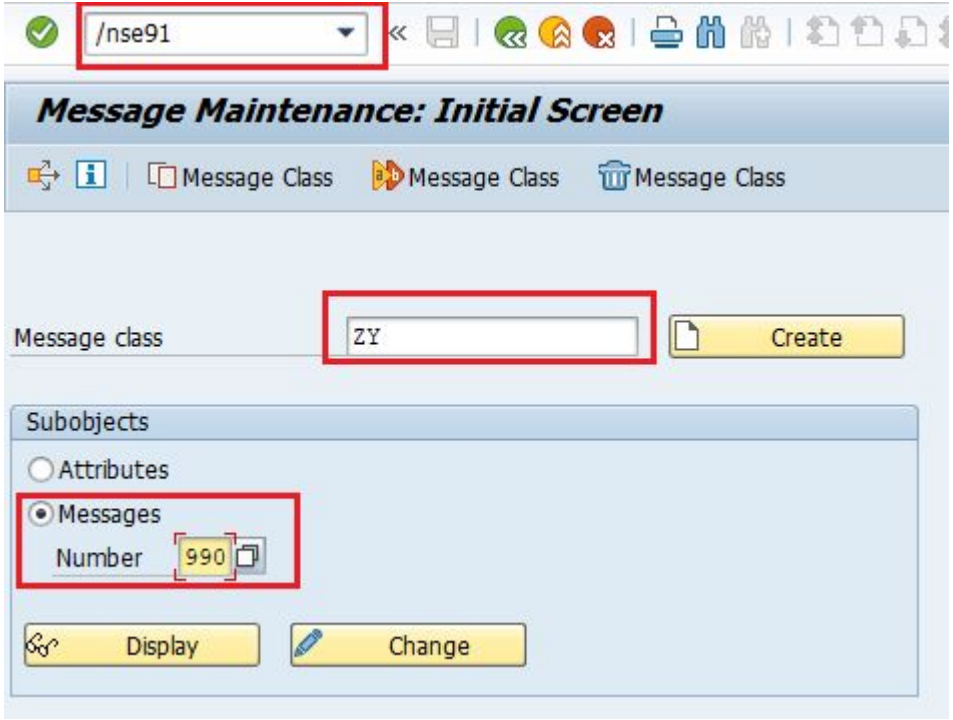
The existing SPRX and its dependent objects should be marked as Customer Action and the comments as below.

This is a generated proxy object and cannot be edited directly in SE11. All the changes or addition/modification of fields have to be first made in middleware
i.e., PO/PI connected to S4 system. As there is an involvement of PO/PI team, the objects are marked under customer action. Once the PO/PI team adjusts these structures, proxy can be re-generated to reflect the changes in SAP.

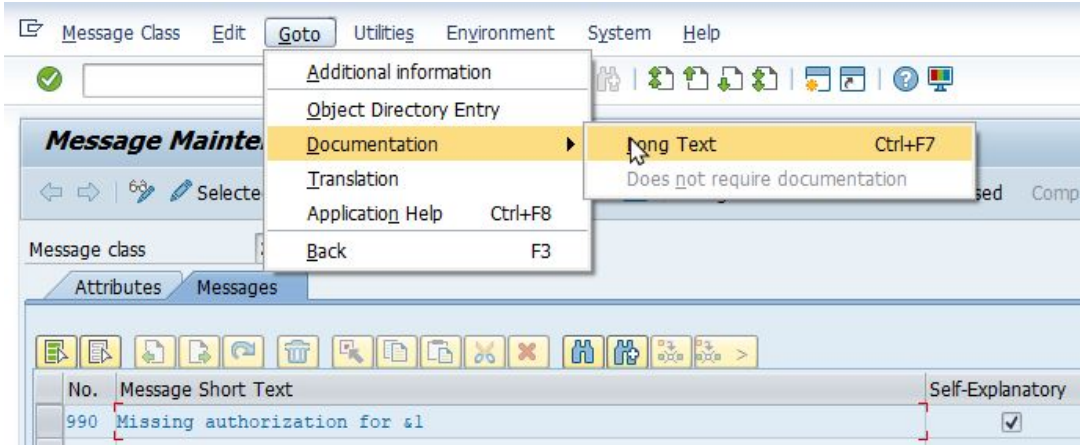
DOCU

Object type - Documentation

a) If the DOCU object is NEW Scope then we will import it via TOC.
b) For DOCU - ALREADY_EXISTED object also we will import it via TOC .
c) Usually object DOCU object in scope list , will be containing prefix like
   CL - class
   DT - Dialog Text
   NA- Message
   RE - Report
d) We can check the object via respective object type tcode's based on prefix which denotes the object type
   For example : NAZY990
   Where NA - denotes documentation of Message class Go to respective TCODE - > SE91 provide message name and number.



Now click on GO TO -> Documentation -> Long Text respective documentation will be displayed has to follow same in both target and source.

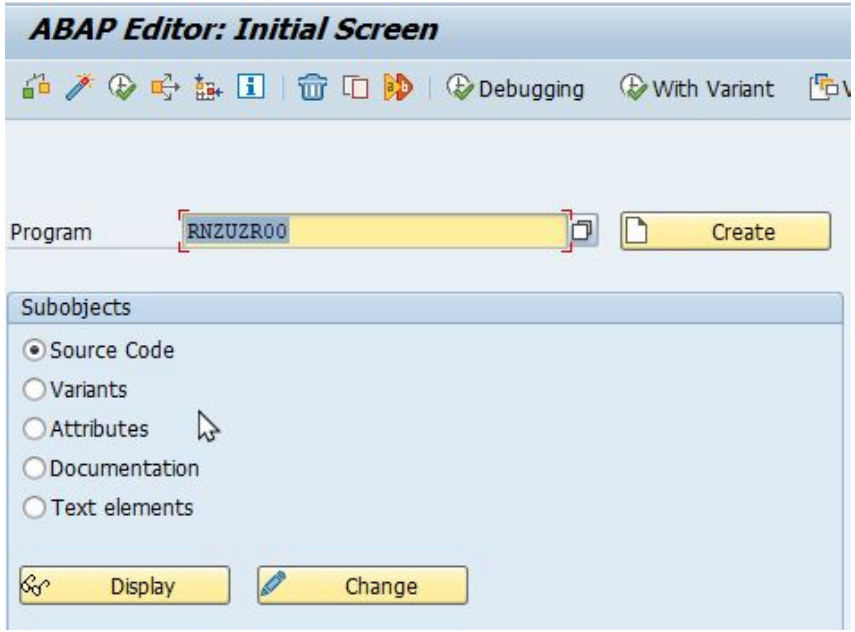

Message Class
Edit
Goto
Utilities
Environment
System
Help
Additional information
Object Directory Entry
Documentation
Translation
Application Help Ctrl+F8
Back F3
Long Text Ctrl+F7
Does not require documentation
Message Mainte
Selecte
Message class
Attributes
Messages
No.
Message Short Text
Self-Explanatory
990
Missing authorization for &1

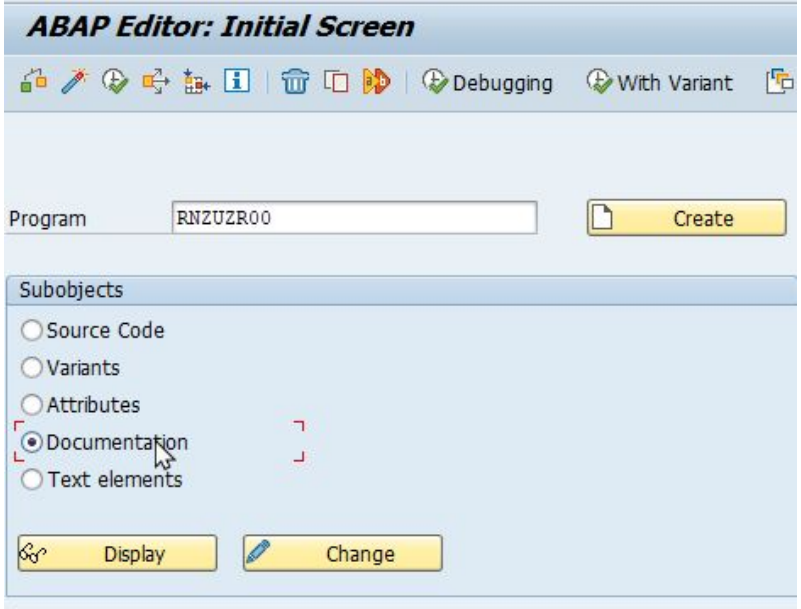
Information
The short text describes the object sufficiently

Example : RERNZUZR00 - > where RE -prefix denotes Report

Go to TCODE - > SE38->Program Name -> RNZUZR00-> select Documentation radio button to display the document

Or we can select source code radio button -> select GO TO -> Documentation





Program Edit Goto Utilities Environment System Help

Attributes
Text Elements
Messages
Variants
Object Directory Entry
Documentation
Translation
Application Help
Back F3

ABAP Editor: ... R00

Pattern Insert Replace

Report Active

```
1   *-----------------------------------------------------------------------*
2   * Aut                                   ( C5056158)                     *
3   * Dat                                                                   *
4   *-----------------------------------------------------------------------*
5   * Change History of ALV Development                                     *
6   ************************************************************************
7   * Program Description : IS-H: Post Back Dunned Copayment Receivables    *
8   *                                                                       *
9   * Short Description :                                                   *
10  *  1) There is no change in the report logic.All the changes are       *
11  *     related to the output format. The ALV object model is used       *
12  *     to show the output.                                               *
13  *  2)  The changes are marked with : Specified User name               *
14  *                                    and with Description              *
15  *  3) Final OutPut :                                                    *
```

ABAP Editor: Display Report RNZ

Report RNZUZR00

```
1  *-------------------------
2  * Author      : Rajesh kuma
3  * Date:        19/08/2004
4  *-------------------------
5  * Change History of ALV De
6  ***************************
7  * Program Description : IS
8  *
9  * Short Description :
10 *  1) There is no change i
11 *     related to the outpu
12 *     to show the output.
13 *  2)  The changes are mar
14 *
15 *  3) Final OutPut :
16 *         a) OutPut is Sho
17 *         b) It uses the
18 ***************************
19
20
21 REPORT rnzuzr00 MESSAGE-ID
22 *
23 * Liste gemahnter Zuzahlun
```

Performance Assistant

RE RNZUZR00

Short Text

IS-H: Post Back Dunned Copayment Receivables

Description

You use this report to post dunned and unpaid copayment requests made of patients to the corresponding insurance provider, and to print out a list.

Normally, hospitals must issue one dunning notice for the copayment request. If the patient still does not make a copayment, you can use this report to transfer the request to the insurance provider (providing there is an invoice for the case, and this invoice can be used to deduct the copayment amount).

On the selection screen, you can limit the number of copayment requests that have to be checked.

This report has an interactive mode, and a mode that is suitable for batch processing.

If you are working in interactive mode, the report outputs a list of copayment requests. You can select

ABAP Editor: Initial Screen

Program RNZUZR00

Subobjects

- Source Code
- Variants
- Attributes
- Documentation
- Text elements

Display | Change

Performance Assistant

RE RNZUZR00

Short Text

IS-H: Post Back Dunned Copayment Receivables

Description

You use this report to post dunned and unpaid copayment requests made of patients to the corresponding insurance provider, and to print out a list.

Normally, hospitals must issue one dunning notice for the copayment request. If the patient still does not make a copayment, you can use this report to transfer the request to the insurance provider (providing there is an invoice for the case, and this invoice can be used to deduct the copayment amount).

On the selection screen, you can limit the number of copayment requests that have to be checked.

This report has an interactive mode, and a mode that is suitable for batch processing.

If you are working in interactive mode, the report outputs a list of copayment requests. You can select

SIRS

Object type - Customizing for Content Repositories

a) If the SIRS object in scope is NEW Scope object then we need to move via TOC .
b) If the object is with ALREADY_EXISTED status then in Go to TCODE -> SE11 -> table - E071 check for existence of object n target ,if exist Go to the transaction code **OAC0** to open the Display Content Repositories: Overview window.check for the respective object in scope sheet.
c) In source system also Go to TCODE-> **OAC0** -> check for the respective object and compare parallel is there any difference found between source and target we need to move the object via TOC
d) If there are no changes found then its an ‘NO ACTION” case.

Data Browser: Table E071 Select Entries 10

Check Table...

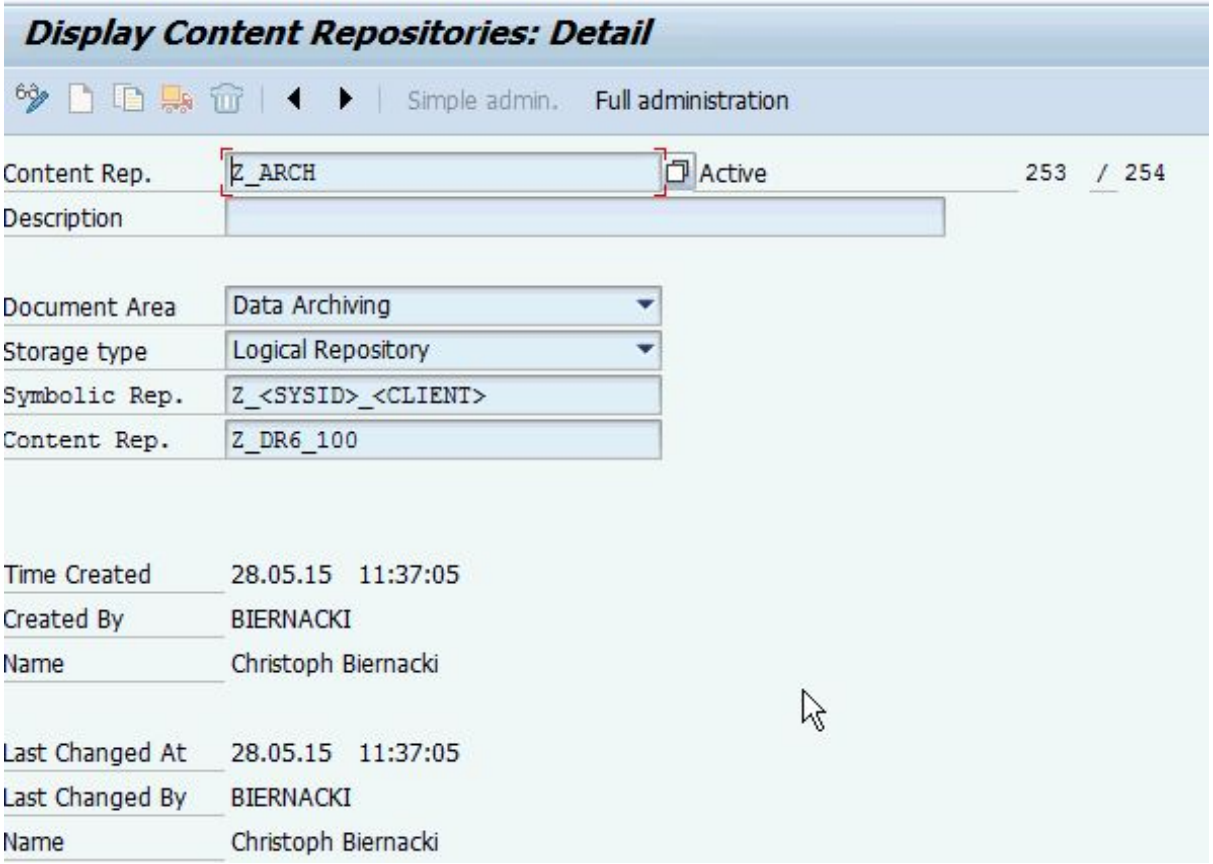
TRKORR	AS4POS	PGMID	OBJECT	OBJ_NAME	OBJFUNC	LOCKFLAG	GENNUM	LANG	ACTIVITY
DRAK900015	000364	R3TR	SIRS	Z1		3			
DRDK914002	000356	R3TR	SIRS	Z1		3			
DRDK920407	000005	R3TR	SIRS	Z_ARCH		3			
DRDK920408	000004	R3TR	SIRS	Z_ARCH					

Display Content Repositories: Overview

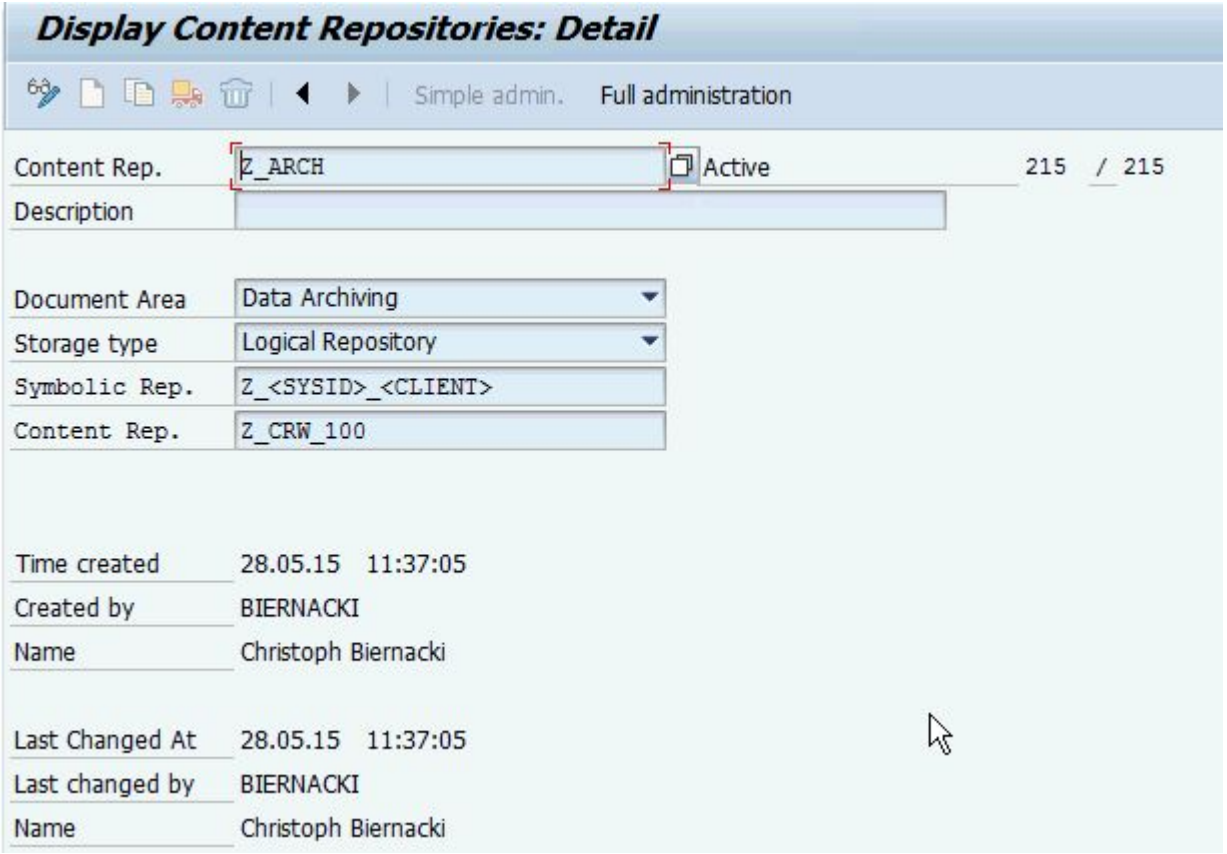
Content Repository

Content Repository	Document...	Storage type	Ver...	Description
ZC		HTTP content se...	0045	
ZT	ARCHLINK	SAP System Data...	0047	
Z_ARCH	DATAARCH	Logical Repository		
Z_DRD_100		HTTP content se...	0045	

Target system



Source system



DOCV

Object type - Documentation (Independent)

a) If the object is NEW Scope then we will import it via TOC.
b) For DOCV - ALREADY_EXISTED object also we will import it via TOC .
c) Usually object DOCV object in scope list , will be containing prefix like
   CL - class
   DT - Dialog Text
   NA- Message
   RE - Report
d) Usually majority of project we will get DOCV object type for message class, reports, class.

Object T	Sco	Scope	SPOC	St
TABL	IN	Service	Shubham	
VIEW	IN	Service	Shubham	
TABD	IN	Service	Shubham	
TABT	IN	Service	Shubham	
VIED	IN	Service	Shubham	
PARA	IN	Service	Shubham	
SOTU	IN	Service	Shubham	
SOTR	IN	Service	Shubham	
SOTS	IN	Service	Shubham	
SOTT	IN	Service	Shubham	

TABL/TABD/TABT

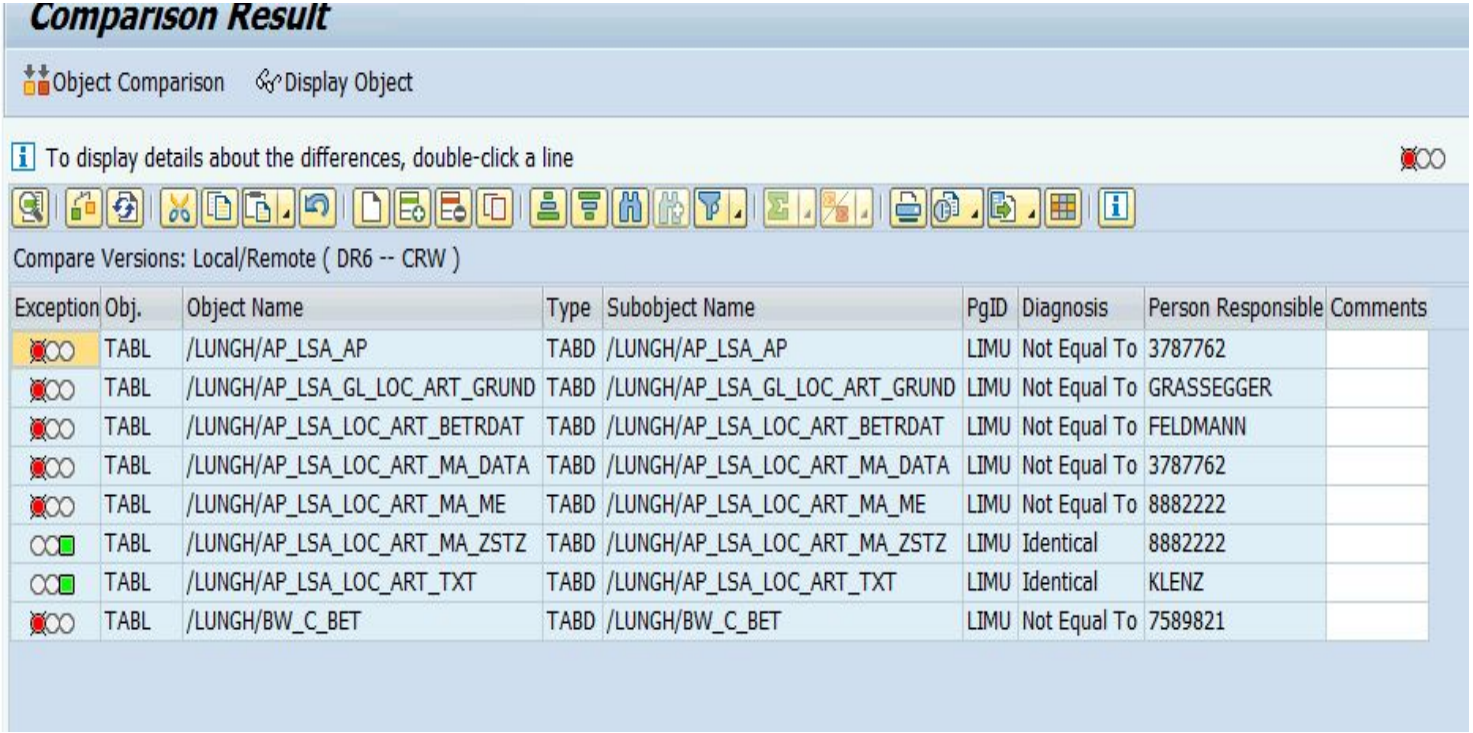
Object Type: Table/Table definition/Table technical settings

- Retrofit of TABL, TABD, TABT have same procedure.
- Tcode: SE11
- Either Go to the RSSYSCOMP and do it using RFC connection or do it manually by version comparison.
- However, doing retrofit check by using RSSYSCOMP has one benefit i.e. we can check for eight object at once instead one by one in version comparison.
- Below snip is the RSSYSCOMP where I've added 8 object at once

Freely Selected Objects

PgmID	Ty.	Obj. Name
LIMU	TABD	/LUNGH/BW_C_BET
LIMU	TABD	/LUNGH/AP_LSA_AP
LIMU	TABD	/LUNGH/AP_LSA_GL_LOC_ART_GRUND
LIMU	TABD	/LUNGH/AP_LSA_LOC_ART_BETRDAT
LIMU	TABD	/LUNGH/AP_LSA_LOC_ART_MA_DATA
LIMU	TABD	/LUNGH/AP_LSA_LOC_ART_MA_ME
LIMU	TABD	/LUNGH/AP_LSA_LOC_ART_MA_ZSTZ
LIMU	TABD	/LUNGH/AP_LSA_LOC_ART_TXT

- After the inputting all the value, execute. Below is the output screen.



Comparison Result

Object Comparison Display Object

To display details about the differences, double-click a line

Compare Versions: Local/Remote (DR6 -- CRW)

Exception	Obj.	Object Name	Type	Subobject Name	PgID	Diagnosis	Person Responsible	Comments
	TABL	/LUNGH/AP_LSA_AP	TABD	/LUNGH/AP_LSA_AP	LIMU	Not Equal To	3787762	
	TABL	/LUNGH/AP_LSA_GL_LOC_ART_GRUND	TABD	/LUNGH/AP_LSA_GL_LOC_ART_GRUND	LIMU	Not Equal To	GRASSEGGER	
	TABL	/LUNGH/AP_LSA_LOC_ART_BETRDAT	TABD	/LUNGH/AP_LSA_LOC_ART_BETRDAT	LIMU	Not Equal To	FELDMANN	
	TABL	/LUNGH/AP_LSA_LOC_ART_MA_DATA	TABD	/LUNGH/AP_LSA_LOC_ART_MA_DATA	LIMU	Not Equal To	3787762	
	TABL	/LUNGH/AP_LSA_LOC_ART_MA_ME	TABD	/LUNGH/AP_LSA_LOC_ART_MA_ME	LIMU	Not Equal To	8882222	
	TABL	/LUNGH/AP_LSA_LOC_ART_MA_ZSTZ	TABD	/LUNGH/AP_LSA_LOC_ART_MA_ZSTZ	LIMU	Identical	8882222	
	TABL	/LUNGH/AP_LSA_LOC_ART_TXT	TABD	/LUNGH/AP_LSA_LOC_ART_TXT	LIMU	Identical	KLENZ	
	TABL	/LUNGH/BW_C_BET	TABD	/LUNGH/BW_C_BET	LIMU	Not Equal To	7589821	

- When we get result from RSSYSCOMP, we can see which are identical and which are not identical.
- Then we can check from the results and from SE11 we can edit the objects and add the missing components from source to target .
- After adding or changing the component we can again run the version comparison where we can see both source and target changes are in sync or not.

Version Display/Comparison of Tables and Structures

Full Display Single/Double Column Field catalog

Object Name	Version	Syst. Name	Request	Date	Author
/LUNGH/AP_LSA_AP	Actv.	DR6	DR6K907201	20230412	1874355

General Attributes:

Table Name	/LUNGH/AP_LSA_AP	/LUNGH/AP_LSA_AP

General Attributes:
Unchanged

Fields

Field Name	Key	Data element	Check table	N	DTyp	Lngth	Group	R

SRS ID

List does not contain any data

Fields
Unchanged

TABT -

- TABT is nothing but the Technical setting of the table
- For checking TABT we need to simply open technical setting of the table in both the system (Target & Source)
- Comparing to slides we can figure out any changes are present or not.
- If changes are present then we can make the changes in TABT.

TABT (Technical setting) comparing in two systems-

Dictionary: Display Technical Settings

Revised<->Active

Name	ZEDM_NACHERNTE_P	Transparent Table	
Short Descript.			
Last Changed	KWIATKOWSKI	15.11.2022	
Status	Actv.	saved	

General Properties | DB-Specific Properties

Logical Memory Parameters

Data Class	APPL0	Master Data, Transparent Tables
Size Category	0	Expected Data Records 0 to 30.000

Table Sharing

Sharing Type — Not classified

Display

Buffering

- (•) Buffering Not Allowed
- ( ) Buffering allowed but switched off
- ( ) Buffering Activated

Dictionary: Display Technical Settings

Revised<->Active

Name	ZEDM_NACHERNTE_P	Transparent Table	
Short Descript.			
Last Changed	KWIATKOWSKI	15.11.2022	
Status	Actv.	saved	

General Properties | DB-Specific Properties

Logical Memory Parameters

Data Class	APPL0	Master Data, Transparent Tables
Size Category	0	Expected Data Records 0 to 30.000

Table Sharing

Sharing Type — Not classified

Display

Buffering

- (•) Buffering Not Allowed
- ( ) Buffering allowed but switched off
- ( ) Buffering Activated

Object Type - VIEW & VIED

- Tcode: SE11
- Either Go to the RSSYSCOMP and do it using RFC connection or do it manually by version comparison.
- However, doing retrofit check by using RSSYSCOMP has one benefit i.e. we can check for eight object at once instead one by one in version comparison.
- Below snip is the RSSYSCOMP where I've added 8 objects at once

Freely Selected Objects

PgmID	Ty.	Obj. Name
R3TR	VIEW	ZGP_V_NO_SYNC_BP
R3TR	VIEW	ZRP_POS_T100C
R3TR	VIEW	ZRP_POS_V_T100C
R3TR	VIEW	ZRSAG_SQ_V_ART1
R3TR	VIEW	/LUNEH/GP_V_BREO
R3TR	VIEW	/LUNEH/GP_V_BRTA
R3TR	VIEW	/LUNEG/BI_V_RBKP
LIMU	VIED	/LUNEH/FG_V_RPA

RSSYSCOMP Results of VIEW/VIED

Comparison Result

Object Comparison Display Object

To display details about the differences, double-click a line

Compare Versions: Local/Remote (DR6 -- CRW)

Exception	Obj.	Object Name	Type	Subobject Name	PgID	Diagnosis	Person Responsible	Comments
	VIEW	/LUNEH/FG_V_RPA	VIED	/LUNEH/FG_V_RPA	LIMU	Not Equal To	7640046	
	VIEW	/LUNEG/BI_V_RBKP	VIED	/LUNEG/BI_V_RBKP	LIMU	Not Equal To	6377588	
	VIEW	/LUNEH/GP_V_BRTA	VIED	/LUNEH/GP_V_BRTA	LIMU	Not Equal To	TIPOGRAFOV	
	VIEW	ZGP_V_NO_SYNC_BP	VIED	ZGP_V_NO_SYNC_BP	LIMU	Not Equal To	3148812	
	VIEW	ZRP_POS_V_T100C	VIED	ZRP_POS_V_T100C	LIMU	Not Equal To	GOLCHINI	
	VIEW	ZRSAG_SQ_V_ART1	VIED	ZRSAG_SQ_V_ART1	LIMU	Not Equal To	ZILLERS	

- When we get result from RSSYSCOMP, we can see which are identical and which are not identical.
- Then we can check from the results and from SE11 we can edit the objects and add the missing components from source to target .
- After adding or changing the component we can again run the version comparison where we can see both source and target changes are in sync or not.

After making changes as per source in the VIEW/VIED -

Version Display/Comparison for Views

Full Display Single/Double Column Field catalog

Object Name	Version	Syst. Name	Request	Date	Author	Object N
/LUNEH/FG_V_RPA	Actv.	DR6	DR6K907237	20230413	2782798	/LUNEH/F

General Attributes:
Unchanged

Base Tables:

Basis Table	Predecess.	FRK field		Basis Table	Predecess.	FRK field
List does not contain any data						

Base Tables:
Unchanged

Fields:

View field	Table Name	Field Name	Key	Data element
List does not contain any data				

Fields:
Unchanged

Object Type PARA -

- if existing we will check with customer for safe to import
- If new we can import it manually or TOC
- Either in TPARA table or se80 by selecting development object
- We usually don’t retrofit this object type instead we import it.

Object Types SOTU, SOTR, SOTS, SOTT -

- These types are nothing but OTR Texts - Online Text Repository
- Online Text Repository - a central storage location for texts, and provides services for editing and managing these texts
- Each text is stored once per package. General, frequently recurring texts are included in the OTR basic vocabulary and can be used across all packages. Texts are saved independently of the object that references them.
- The main task of the OTR is to support the creation and translation of texts.
- Main advantage is texts can easily be reused.
- There are two options for saving texts in the OTR:
    - OTR short text (up to 255 characters in length)
    - OTR long text (unlimited length)

- If the transport request for the texts for translation is not released, the texts are not transported.
- transport all texts that belong to a package, independently of an object. To do this, create a transport request:
    - For OTR short texts, use the object type R3TR SOTR
    - For OTR long texts, use the object type R3TR SOTS

- whenever we a create a OTR TEXT entry the corresponding entries is generated in the transport request.
    - OTR short texts: object type LIMU SOTT
    - OTR long texts: object type LIMU SOTU
- If, however, you only want to transport texts in a particular language, create a transport request as follows:
    - For OTR short texts, use the object type LANG SOTR
    - For OTR long texts, use the object type **LANG SOTS**

OTR is currently supported by the following applications:

- Web Dynpro for ABAP (see Internationalization and Translation)
- Business Server Pages
- Exception Builder
- HTTP Services
- Web Services
- Enhancement Tools

Applications can have only short texts (currently Exception Builder), only long texts (HTTP services), or both (BSP applications, see also Business Server Pages).

Retrofit of SOTU, SOTR, SOTS, SOTT

- We check **SOTR** using **TCODE – SOTR_EDIT**
- For New Objects, we add them in the TOC & for the existing ones we manually check
- Mostly existing objects are No Action case s, Customer rarely make changes to it where we can make it safe to import or manually fix .

/nsotr_edit

OTR: Initial Editor Screen

Text Search | Review | Reassign | Text | Alias

Text Type

- (•) Short Text
- ( ) Long Text

Text

Language	English
Text	
Alias	
Concept	

Display | Create | Change

Object Type : VIEW(External View)

Prerequisites :

- Install SAP HANA modeler View in eclipse
- Get access to HANA DB

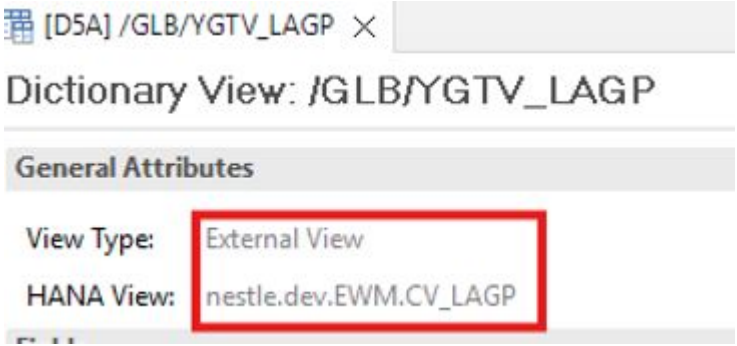
Steps for already existed object.

1. Find the HANA view from external view in eclipse(if already existed)
2. Login to HANA DB from SAP HANA modeler perspective.
3. Navigate to HANA view under content folder and make the necessary changes.
4. To update the HANA view changes in external view,switch to ABAP perspective and click on “Synchronize” and save the changes in TR

Steps for New object.

1. Login to HANA DB from SAP HANA modeler perspective.
2. Create the HANA view under content folder and make the necessary changes.
3. Switch to ABAP perspective to create external view with respective HANA view.

1. Find the HANA view from external view in eclipse(if already existed)



2. Login to HANA DB from SAP HANA modeler perspective.

3. Navigate to HANA view under content folder and make the necessary changes.
4. To update the HANA view changes in external view,switch to ABAP perspective and click on “Synchronize” and save the changes in TR

Steps for New object.

1. Login to HANA DB from SAP HANA modeler perspective.
2. Create the HANA view under content folder and make the necessary changes.
3. Switch to ABAP perspective to create external view with respective HANA view.