

# ZSUIM: Mass Role Comparison



## Applies to:

SAP 4.6C

For more information, visit the [Security homepage](#).

## Summary

Role comparison is an important activity for any SAP Security Consultant. It's performed very extensively while transporting roles from Dev to Integration or from Integration system to Production. The drawback with standard SUIM role comparison is, it does not give you the flexibility to compare mass roles across the systems, this document refers to the customized version of SUIM role comparison or report RSUSR050 which extends the SUIM functionality to multiple role comparison at one go.

**Author:** Akshay Agarwal

**Company:** IBM India Pvt Ltd.

**Created on:** 31 May 2010

## Author Bio



I am Akshay Agarwal, working in IBM. I am having more than 3.5 years of experience in SAP Security, where I have worked on MDM, NW Portal, XI, GRC, Vendavo.

## Table of Contents

Why do we need to perform Role Comparison?.....	3
What is the drawback of the current SUIM Role Comparison functionality? .....	4
ZSUIM: Mass Role Comparison .....	5
Implementation Logic.....	7
Related Content.....	8
Disclaimer and Liability Notice.....	9

## Why do we need to perform Role Comparison?

Role comparison is a very important step, which has to be performed especially while transporting derived roles (Orglevel Changes) from Development system to Integration systems for testing or from Integration system to Production system.

As we all know that while transporting the derived roles, automatically its parent roles are included in the same transport. Hence we need to ensure that the parent role is in sync throughout the landscape before we transport its associated derived role. If there has been any change found in the parent role (While comparing it to Int./Prod. system) then we need to ensure that we push the changes of the parent role to all its derived role (In Dev.) and ensure to transport all its derived role to integration system (For Testing) and later to Production (If test passes).

If we miss/ignore the role comparison step then we might end up transporting untested changes to the production system, or it might lead to the inconsistency between the derived roles and its parent role which in turn will lead to "Red" authorization tab of the roles.

Also this role comparison can help resolve the issues of multiple changes being performed on the same role but by different people. This happens when same role is used by multiple businesses and the update is being carried out by different teams simultaneously. Before updating the role, it's a good practice to first check if the roles are in sync throughout the landscape or not. If there is any difference then by checking out the change history we need to find out who has made the last change and co-ordinate with him/her and ensure that your changes do not overwrite someone else's changes.

Role comparison can also ensure that the SU24 changes are corrected in development system and additional/unwanted changes are not transported further. After doing the role changes in development system it's always recommended to perform the role comparison from dev-test or dev to production and ensure the changes which you have done are only reflected in the role comparison output. Any additional change/difference should be re-checked before transporting the changes to Integration system for testing.

## What is the drawback of the current SUIM Role Comparison functionality?

Below is the standard role comparison screen which you can reach by executing report RSUSR050 through SE38 or by running transaction SUIM >> Comparison >> From Roles

The screenshot shows the 'Comparisons' screen with the following fields:

- Remote comparison: RFC destination for system A (UDR), RFC destination for system B (GB2590)
- Compare roles: Role A (TEST\_1), Role B (TEST\_1)

As shown above only one roles i.e TEST\_1 is compared between the two systems to get below output:

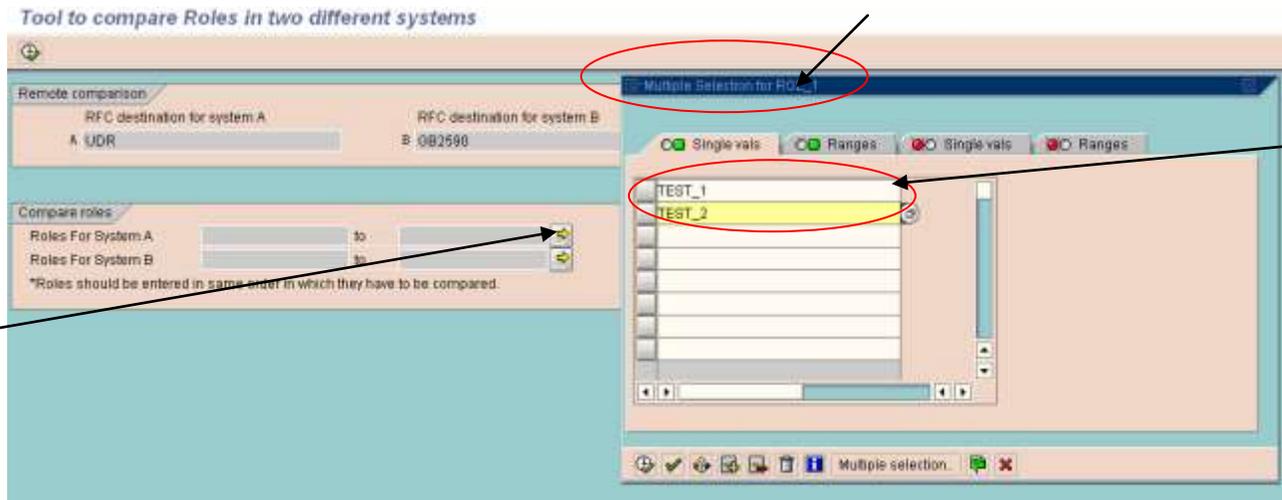
Role	TEST_1	TEST_1
Client:	510	590
System:	UDR	GB2
Same values	K_REPO_USR	K_REPO_USR
Same values	PLOG	PLOG
Same values	S_ADMI_FCD	S_ADMI_FCD
Same values	S_ARCHIVE	S_ARCHIVE
Same values	S_BDS_DS	S_BDS_DS
Same values	S_BTCH_ADM	S_BTCH_ADM
Same values	S_BTCH_JOB	S_BTCH_JOB
Same values	S_BTCH_NAM	S_BTCH_NAM
Same values	S_C_FUNCT	S_C_FUNCT
Same values	S_DATASET	S_DATASET
Same values	S_DOKU_AUT	S_DOKU_AUT

Issue here is if you have multiple roles to be compared (For e.g say 100 roles) across systems then you have to perform the above step manually for each role again and again (Repeating the above step 100 times).

Currently there is not such functionality which gives you the advantage of comparing multiple roles on a single hit.

## ZSUIM: Mass Role Comparison

Idea is to customize the SUIM tcode such that it provides the flexibility of comparing multiple roles across the systems. Refer to the below screen shot:

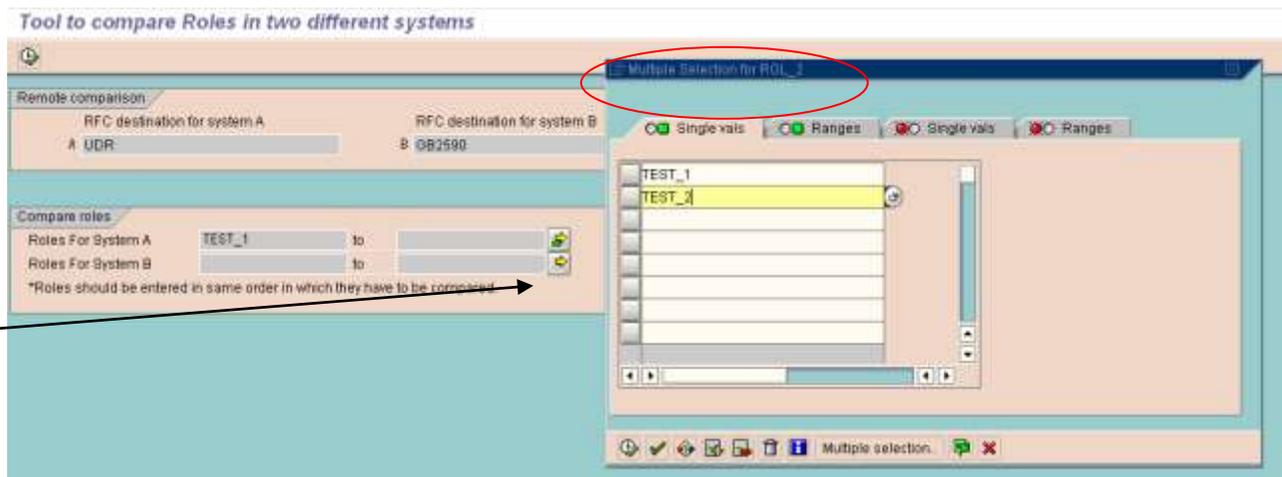


In the above screen you can see, we have customized the SUIM role comparison functionality such that it gives you an option to provide multiple roles as the input.

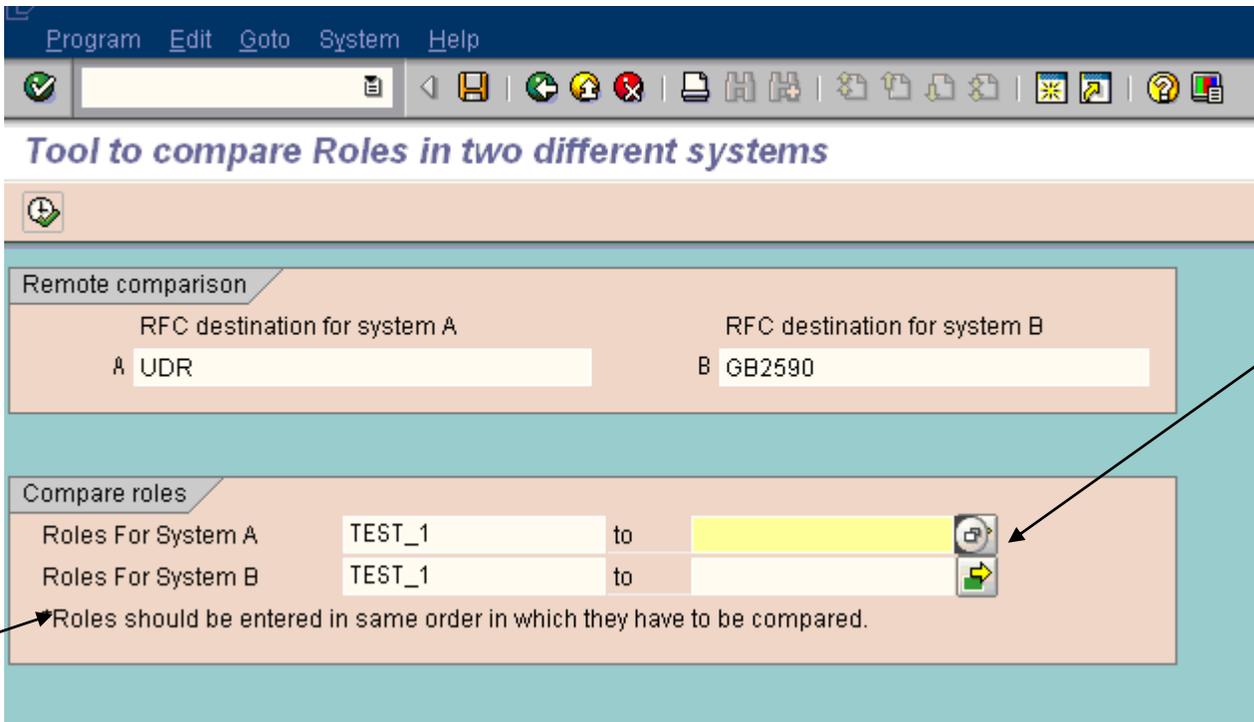
In the above example TEST\_1 and TEST\_2 are the two roles (Input of "Roles for System A") which are being compared between two systems A and B.

Below screen provides you the input of "Roles for System B".

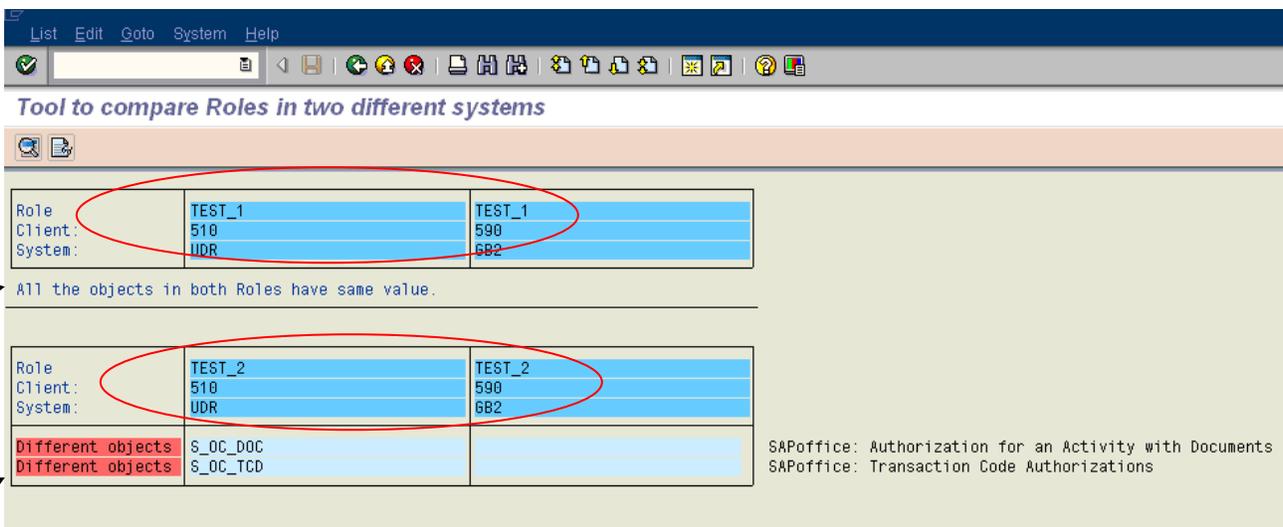
**Note:** Same set of roles are used i.e TEST\_1 and TEST\_2. Also the customized program is written such that we should enter the roles in same order in which they have to be compared.



All the inputs proved as shown below and explained above:



Below is the customized output which shows that TEST\_1 is in sync and no difference has been found while comparing it between the two systems. TEST\_2 is the role which is **not** in sync and objects which have different values are displayed as shown in the screen shot below:



The above screen gives you a very simplified output which matches our requirement. Unlike the regular SUIM role comparison output which displays complete list of the same objects and as well as different, here we have restricted the list to only different objects.

This is how you we can customize the SUIM screen to get the extended functionality of mass role(s) comparison.

## Implementation Logic

- If you are not an expert in ABAP programming then you can take help of a developer/ABAPer, to get this customization done.
- At the first place you need to customize the screen of the report RSUSR050 to include multiple selection options → . These multiple selection options can be seen at screens of various other tcodes for example PFUD or run a report RSUSR070. Hence customize the input screen of RSUSR050 such that it takes multiple inputs.

Below is the piece of code as an example:

```
* -----*
* Selection screen
* -----*
* Remote comparison
SELECTION-SCREEN BEGIN OF BLOCK dest_CMP WITH FRAME TITLE TEXT-010.
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 8(30) TEXT-201  MODIF ID RFC.
SELECTION-SCREEN COMMENT 50(30) TEXT-202  MODIF ID RFC.
SELECTION-SCREEN END OF LINE.
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT 6(1) TEXT-203  MODIF ID RFC.
SELECTION-SCREEN POSITION 8.
PARAMETERS SYST_1 LIKE RFCDES-RFCDEST  MODIF ID RFC.
SELECTION-SCREEN POSITION 50.
SELECTION-SCREEN COMMENT 48(1) TEXT-204 MODIF ID RFC.
PARAMETERS SYST_2 LIKE RFCDES-RFCDEST  MODIF ID RFC.
SELECTION-SCREEN END OF LINE.
SELECTION-SCREEN END OF BLOCK dest_CMP.
SELECTION-SCREEN SKIP 1.
* Role
SELECTION-SCREEN BEGIN OF BLOCK ro11_CMP WITH FRAME TITLE TEXT-004.
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT (19) TEXT-043  MODIF ID ROL.
SELECTION-SCREEN POSITION 20.
SELECT-OPTIONS ro1_1 FOR AGR_DEFINE-AGR_NAME MEMORY ID ro11 MODIF ID ROL
.
SELECTION-SCREEN END OF LINE.
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT (19) TEXT-042 MODIF ID ROL.
SELECTION-SCREEN POSITION 20.
SELECT-OPTIONS ro1_2 FOR AGR_DEFINE-AGR_NAME MEMORY ID ro11 MODIF ID
ROL
.
SELECTION-SCREEN END OF LINE.
SELECTION-SCREEN BEGIN OF LINE.
SELECTION-SCREEN COMMENT (75) TEXT-048  MODIF ID ROL.
SELECTION-SCREEN END OF LINE.
SELECTION-SCREEN END OF BLOCK ro11_CMP.
```

```

* -----*
*  INITIALIZATION
* -----*
INITIALIZATION.
  IMPORT RFC      FROM MEMORY ID 'RSUSR050_RFC'.
  MOVE TEXT-102 TO SSCRFIELDS-FUNCTXT_02 .  "Rolle
  IF RFC IS INITIAL.
    MOVE TEXT-103 TO SSCRFIELDS-FUNCTXT_05 .  "uber systeme
  ELSE.
    MOVE TEXT-104 TO SSCRFIELDS-FUNCTXT_05 .  "intern
  ENDIF.

```

- Now using the same logic of RSUSR050 for role comparison and functionality of nested loops you can run the same role comparison for “N” number of different roles, such that it keep on storing the output for each role.
- After comparing all the roles one by one(Nested loop) and storing the output data, the last part is to display it, which can further be customized/restricted to only “Different Objects”.

## Related Content

<http://service.sap.com/securityguide>

<http://service.sap.com/security>

<http://www.sdn.sap.com/irj/sdn/security>

<http://help.sap.com/>

## Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.