460089 - Minimum authorization profiles for external RFC programs

Version	13	**Type**	SAP Note
Language	English	**Master Language**	German
Priority	Recommendations / Additional Info	**Category**	Installation information
Release Status	Released for Customer	**Released On**	06.07.2018
Component	BC-MID-RFC (RFC)		

Please find the original document at **https://launchpad.support.sap.com/#/notes/ 460089**

Symptom

In the RFC communication between an external program and an SAP system, you want to use a user who only has authorizations in the SAP system that are absolutely necessary. Depending on the technology that your program uses for communication, the release of the attached R/3 or SAP system and the type of the calls that you want to execute, the SAP user set in the external program requires different authorizations.
For the sake of clarity, these authorizations are summarized in this note.

Other Terms

RFC, RFC SDK, NW RFC SDK, JCo, SAP Java Connector, .NET Connector, Business Connector, authorization, authorization objects

Reason and Prerequisites

External programs can essentially use the following technologies for RFC communication:

1. RFC SDK (C/C++)
2. NW RFC SDK (C/C++)
3. Java Connector (Java)
4. .NET Connector (C#)
5. Business Connector (XML,HTTP,FTP or SMTP interface)

Most of these components provide a repository service, which dynamically reads the interface definition of a remote-enabled function module (abbreviated as RFM) from the ABAP Data Dictionary. This service in turn calls several function modules in the ABAP application server internally. For this reason it requires specific authorizations in function groups.

Solution

Firstly some general comments:

- You can use the auth/rfc_authority_check=0 profile parameter to deactivate the authorization check in the RFC. This is set by default in all R/3 releases up to and including Release 3.1l. To protect the system against

unauthorized RFC accesses, you should check that this parameter is set to at least 1.

- In Releases 3.1I to 4.6D there were some bugs at the beginning in the RFC authorization check. Ensure that your R/3 kernel has at least the patch level referred to in Note 93254.
- Some application function modules and BAPIs perform an internal check again for a "business" authorization object. In this case the user, who wants to execute the RFC call, must also have authorization for this "application authorization object" as well as the "technical" authorizations described in the following section.
- If the module that you called wants to execute a transaction code, the user also requires the S_TCODE authorization object (with the relevant transaction codes). If you start a report within the module, the user also requires the S_PROGRAM authorization object (with the relevant program groups).
- In general it is sufficient if the user is of the "CPIC" type (R/3 Release 3.1I to 4.6B) or "communication" (as of R/3 Release 4.6C). The user must only be of the "dialog" type if you want to debug function module calls from the external program in the ABAP debugger. For the additional authorizations required to carry out the debugging, please see Notes 905364, 668256 and 668252.

Some standard scenarios are described in the following section. In the case of scenarios in which a dynamic repository is used, it is assumed that two different types of users are used: A special user that is responsible for the repository accesses and the application user that should execute the actual application RFMs. This is advisable for security reasons. If you only want to use one user in the external program, simply assign the user the union of both authorizations.

The authorization profile of the user must contain the S_RFC authorization object, whereby the fields are filled as follows:
ACTVT: 16
RFC_TYPE: FUGR
RFC_NAME: The list of the function groups executed below.

In the following list X is the name of the function group for which you want to call function modules.

1. Direct call of a function module

(for example, via the RFC library or via NW RFC SDK/JCo/NCo with static repository)

Application user:

R/3 release	Function groups
3.1I	SYST, X
As of 4.0A	SYST, SYSU, X

2. Direct call of function module via tRFC or qRFC

(for example, via the RFC library or via NW RFC SDK/JCo/NCo with static repository)

Application user:

R/3 release	Function groups
3.1I	SYST, ARFC, ERFC, X
As of 4.0A	SYST, SYSU, ARFC, ERFC, X

3. Call of function module using dynamic repository

(For example, using NW RFC SDK, JCo, .NET Connector or Business Connector)

Application user:

R/3 release	Function groups
3.1I	SYST, X
As of 4.0A	SYST, SYSU, X

Repository user:

R/3 release	Function groups
3.1I	SYST, RFC1, SUNI
4.0A - 4.5B	SYST, RFC1, SYSU, SUNI, SDIF
4.6A - 4.6C	SYST, RFC1, SYSU, SDIF
As of 4.6D	SYST, RFC1, SYSU, SDIFRUNTIME

4. Call of tRFC or qRFC using dynamic repository

(For example, using NW RFC SDK, JCo, .NET Connector or Business Connector)

Application user:

R/3 release	Function groups
3.1I	SYST, ARFC, ERFC, X
As of 4.0A	SYST, SYSU, ARFC, ERFC, X

Repository user: As in previous point

5. Sending and receiving IDocs

(For example, with the SAP Java IDoc Library or the Business Connector)

Application user (for sending IDocs):

R/3 release	Function groups
3.1I	SYST, ARFC, ERFC, BD11
As of 4.0A	SYST, SYSU, ARFC, ERFC, EDIN

In addition, the user still requires the authorization object B_ALE_RECV, whereby the field EDI_MESTYP is to be filled with the list of message types of the IDocs to be processed.

Repository user:

R/3 release	Function groups
3.1I	SYST, RFC1, SUNI, EDI6, EDI8, EDIF
4.0A - 4.5B	SYST, RFC1, SYSU, SUNI, SDIF, EDIMEXT, EDI6
4.6A - 4.6C	SYST, RFC1, SYSU, SDIF, EDIMEXT, EDI6
As of 4.6D	SYST, RFC1, SYSU, SDIFRUNTIME, EDIMEXT, EDI6

The user also requires the authorization object S_IDOCDEFT via the authorization "S_IDCDFT_DIS", for example.

Note: If the profile parameter auth/rfc_authority_check has a value greater than "2", all users also require the authorization for the function group SRFC.

Authorization check as of SAP Release 7.10
As of Release 7.10 you can execute the RFC authorization check on individual function modules, instead of on entire function groups. You can also use the procedure described above, but if you want to refine the authorization check even further, fill the fields of the S_RFC authorization obect as follows:
ACTVT: 16
RFC_TYPE: FUNC
RFC_NAME: The list of the function modules executed below.

In the following section Y is the name of the function module that you want to call.

1. Direct call of a function module

Application user: RFCPING, SYSTEM_RESET_RFC_SERVER, Y

2. Direct call of function module via tRFC or qRFC

Application user: RFCPING, SYSTEM_RESET_RFC_SERVER, API_CHECK_TID, API_CREATE_TID, API_CLEAR_TID, ARFC_DEST_SHIP, ARFC_DEST_CONFIRM, Y

3. Call of function module using dynamic repository

Application user: RFCPING, SYSTEM_RESET_RFC_SERVER, Y

Repository user: RFCPING, SYSTEM_RESET_RFC_SERVER, RFC_GET_FUNCTION_INTERFACE, DDIF_FIELDINFO_GET

4. Call of tRFC or qRFC using dynamic repository

Application user: RFCPING, SYSTEM_RESET_RFC_SERVER, API_CHECK_TID, API_CREATE_TID, API_CLEAR_TID, ARFC_DEST_SHIP, ARFC_DEST_CONFIRM, Y

Repository user: As in previous point

5. Sending and receiving IDocs

Application user (for sending IDocs): RFCPING, SYSTEM_RESET_RFC_SERVER, API_CHECK_TID, API_CREATE_TID, API_CLEAR_TID, ARFC_DEST_SHIP, ARFC_DEST_CONFIRM, IDOC_INBOUND_ASYNCHRONOUS, IDOC_INBOUND_IN_QUEUE

In addition, the user still requires the authorization object B_ALE_RECV, whereby the field EDI_MESTYP is to be filled with the list of message types of the IDocs to be processed.

Repository user: RFCPING, SYSTEM_RESET_RFC_SERVER, RFC_GET_FUNCTION_INTERFACE, DDIF_FIELDINFO_GET, IDOCTYPE_READ_COMPLETE, EDI_AGREE_OUT_MESSTYPE_READ
The user also requires the S_IDOCDEFT authorization object, for example, using the "S_IDCDFT_DIS" authorization.

Reduction of the roundtrips when using the dynamic repositories
As of SAP System Release 7.00, you can use the procedure described in Note

1456826 to reduce the roundtrips required for the dynamic repository between the external connector and the SAP system. If you want to do this, the repository user that is used also requires the authorization for the following function group: RFC_METADATA

If the alternative authorization check at function module level is to be used, then the repository user requires the authorization for the following function modules instead: RFC_METADATA_GET, RFC_METADATA_GET_TIMESTAMP

Other Components

Component	Description
BC-MID-BUS	Business Connector
BC-MID-CON-JCO	Java-Connector
BC-MID-CON-NCO	SAP .Net Connector
BC-MID-RFC-SDK	NetWeaver RFC SDK, classical RFC SDK

This document refers to

SAP Note/KBA	Title
20534	Authorization check - a short introduction
1853904	RFC authorizations for Java Connector
1820917	Unable to cast object of type 'System.InvalidCastException'
1553144	Authorization error during inserting query
1168772	Minimum authorization profile for JRA

This document is referenced by

SAP Note/KBA	Title
2577916	Error "BI system has version; 70000 is required" or "System is not a BI system" occurs in Bex Analyzer after update to SAP GUI 750
2531301	No authorization to send IDocs with message type

1853904	RFC authorizations for Java Connector
1820917	Unable to cast object of type 'System.InvalidCastException'
1553144	Authorization error during inserting query
1168772	Minimum authorization profile for JRA
20534	Authorization check - a short introduction

