Configuration Guide

Configuring the ABAP Back-end for ABAP Development Tools

THE BEST-RUN BUSINESSES RUN SAP

SAP®

Contents

Configuring the ABAP Back-End

This documentation describes how to configure the ABAP server for each back-end system in order to connect them with ABAP Develeopment Tools for SAP NetWeaver.

The configuration from ABAP Develeopment Tools for SAP NetWeaver to a back-end system includes the completion of the following activities for each back-end system you want to work with:

1. Establishing Secure Network Communication
2. Providing Roles and User Authorizations
3. Activating Services
4. Configuring Profile Parameters

1 Establishing Secure Network Communication

In ABAP Development Tools, developers always work with ABAP projects to access repository objects from back-end systems. An ABAP project represents a real system connection on the front-end client and, therefore, it requires an authorized user to access the back-end system. With the standard authentication method, the user enters his or her user ID and password at the front-end client to log on to the back-end system.

Prerequisites

For security reasons, however, you must ensure that the **Secure Network Communication (SNC)** is implemented for the corresponding system and enabled for the selected system connection.

Procedure

To enable SNC for an ABAP system …

1. Add the relevant ABAP system to the **SAP Logon** Pad - if not already done.
2. In the SAP Logon Pad, open the **Properties** page for the ABAP system.
3. Enable the **Secure Network Communication** for the selected system - if not already done.

RECOMMENDATION

For the sake of convenience, and again for security reasons, use also the **single sign-on (SSO)** option for system authentication (if it is available in your SAP NetWeaver system landscape). Compared with SNC, SSO meets the security requirements for working with ABAP projects even more. Using SSO, the user does not need to enter a user ID and password, but can access the specific system directly after the system has checked the validity of the logon ticket.

Procedure

To enable SSO for an ABAP system...

1. Install the SAP NetWeaver **Single Sign-On 1.0 SP03 or higher** (either "Secure LoginClient" or "Enterprise Single Sign-On") for the corresponding platform (either for 32- or 64-Bit).
2. Configure the Secure Network Communication between ABAP Development Tools client and the ABAP back-end system.
For more information, see http://scn.sap.com/community/netweaver-sso.

2 Providing Roles and User Authorizations

The assignment of authorizations to back-end system users is based on roles that are predefined in the ABAP system. As system administrator, you will assign to such back-end system users one or multiple roles that, in turn, are based – at a technical level – on authorization objects.

When you are assigning user authorizations in an ABAP system for access to repository objects from within an ABAP project, we strongly recommend using the standard roles (see Table below) and authorization default values that are provided for working with ABAP Development Tools.

Procedure

Use one of the standard SAP roles below as a master copy to assign users of the ABAP Development Tools in your organization.

See also: [Reference Documentation for User and Role Administration](#)

Standard Roles and Authorization Objects

The table below shows the standard roles. One of them should be assigned to the users of ABAP Development Tools depending on the needed authorizations:

Role	Description
SAP_BC_DWB_ABAPDEVELOPER	Role that contains all authorizations relevant for ABAP developers
SAP_BC_DWB_WBDISPLAY	Role that contains all authorizations to display and browse ABAP development objects. NOTE The users are not allowed to modify ABAP development objects.

Table: List of predefined standard roles

These standard roles are linked with the following authorization objects that are additionally required for the usage of ABAP Development Tools:

- S_ADT_RES: Resource access by ABAP Development Tools
- S_RFC: Authorization check for RFC access
- S_ TCODE: Transaction codes check at transaction start

URI Prefixes for the Authorization Object S_ADT_RES

ABAP Development Tools uses the following URI prefixes for the authorization object **S_ADT_RES**:

- /sap/bc/adt/abapsource/*
- /sap/bc/adt/abapunit/*
- /sap/bc/adt/activation
- /sap/bc/adt/activation/*
- /sap/bc/adt/atc/*
- /sap/bc/adt/checkruns
- /sap/bc/adt/checkruns/*

- /sap/bc/adt/compatibility/*
- /sap/bc/adt/core/discovery
- /sap/bc/adt/datapreview
- /sap/bc/adt/datapreview/*
- /sap/bc/adt/dataproviders
- /sap/bc/adt/dataproviders/*
- /sap/bc/adt/ddic/*
- /sap/bc/adt/debugger
- /sap/bc/adt/debugger/*
- /sap/bc/adt/discovery
- /sap/bc/adt/docu/*
- /sap/bc/adt/documentation/*
- /sap/bc/adt/error/*
- /sap/bc/adt/feeds
- /sap/bc/adt/feeds/*
- /sap/bc/adt/filestore/ui5-bsp/*
- /sap/bc/adt/fpm/*
- /sap/bc/adt/functions/*
- /sap/bc/adt/includes/*
- /sap/bc/adt/logs
- /sap/bc/adt/logs/*
- /sap/bc/adt/messageclass
- /sap/bc/adt/messageclass/*
- /sap/bc/adt/navigation/*
- /sap/bc/adt/nhi/*
- /sap/bc/adt/oo/*
- /sap/bc/adt/packages
- /sap/bc/adt/packages/*
- /sap/bc/adt/programs
- /sap/bc/adt/programs/*
- /sap/bc/adt/quickfixes/*
- /sap/bc/adt/refactorings
- /sap/bc/adt/refactorings/*
- /sap/bc/adt/repository/*
- /sap/bc/adt/runtime/dumps
- /sap/bc/adt/runtime/dumps/*
- /sap/bc/adt/runtime/systemmessages
- /sap/bc/adt/runtime/systemmessages/*
- /sap/bc/adt/runtime/traces

- /sap/bc/adt/runtime/traces/*
- /sap/bc/adt/security/*
- /sap/bc/adt/sscr/*
- /sap/bc/adt/system/clients
- /sap/bc/adt/system/users
- /sap/bc/adt/wdy
- /sap/bc/adt/wdy/*
- /sap/bc/cts/*
- /sap/bc/esproxy
- /sap/bc/esproxy/*

NOTE

The wildcard "*" is used for the subpathes of the URI.

Remote Access to Function Modules (Specified in the Authorization Object S_ RFC)

ABAP Development Tools requires remote access to the following function modules that are specified for the authorization object **S_ RFC**:

Activity [ACTVT]	Name of RFC object to that can be accessed [RFC_NAME]	Type of RFC object that can be accessed [RFC_TYPE]
16 (Execute)	DDIF_FIELDINFO_GET	FUNC (Function module)
	RFCPING	
	RFC_GET_FUNCTION_INTERFACE	
	SADT_REST_RFC_ENDPOINT	
	SUSR_USER_CHANGE_PASSWORD_RFC	
	SYSTEM_RESET_RFC_SERVER	

Table: List of function modules, specified for the authorization object S_RFC

Transaction Code Check at Transaction Start (Specified in the Authorization Object S_ TCODE)

ABAP Development Tools has to start certain transactions in order to enable the SAP GUI integration in Eclipse. Therefore, the tools require access to the following transaction codes that are specified in the authorization object **S_ TCODE**:

- SADT_START_TCODE
- SADT_START_WB_URI

3 Activating Services

Using HTTP services that are predefined in 7.31 systems (and higher) you can access the ABAP keyword help and the ABAP problem help that is available in the current back-end system.

If you also plan to use the **Web Dynpro ABAP Development** tool, then to access the **Layout** page of the Web Dynpro View Controller Editor you have to activate the preview service. The Web Dynpro View Controller Editor provides a preview of the user interface at design time only if the preview service is active.

In the ABAP system the HTTP and the preview services appear disabled by default. Therefore, as system administrator you have to activate them for each system in question.

Procedure

1. Call the transaction **SICF** in the relevant back-end system.
2. As a filter, select **DEFAULT_HOST** for the Virtual Host entry. Then press F8 or use the corresponding toolbar icon to display the services tree.
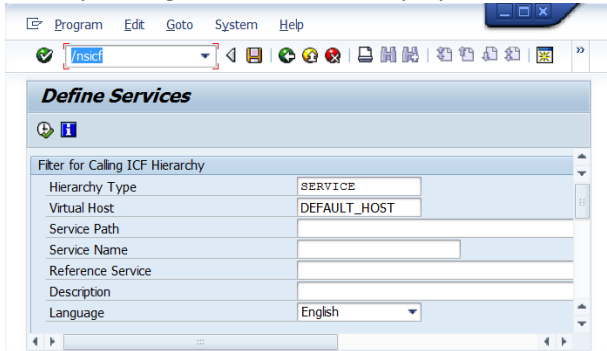

Figure: *Entry page of the SICF transaction*

3. To activate the **docu** and **ToolsDocu** service items, use the sub-steps below:

   a. Expand the node **default_host** > **sap** > **bc** > **abap**.
   The subtree displays the service items **docu** and **ToolsDocu**.

   b. Select the service **docu** and choose **Activate Service** from the context menu.
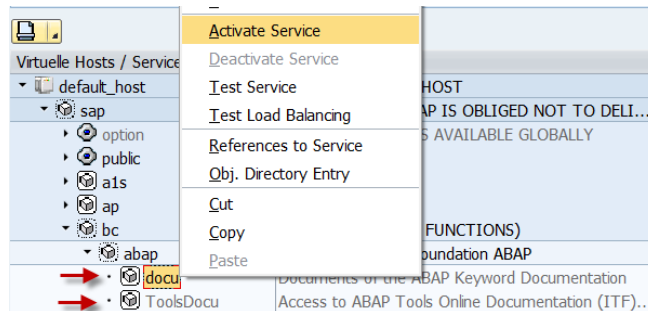
Figure: *Activating **docu** service*

   c. Repeat the last step for the service **ToolsDocu**.

   d. In addition, expand the link node in the public subtree **default_host** > **sap** > **public** > **bc** > **abap.**

   e. Select the link node **docu** and choose **Activate Link** from the context menu.
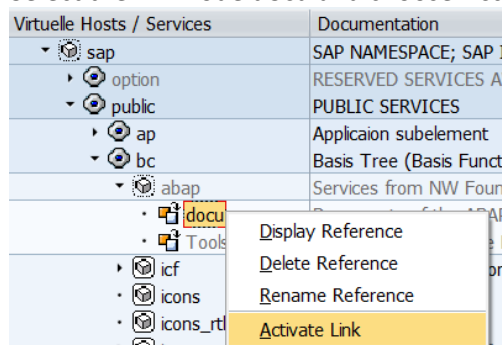
Figure: *Activating link to **docu** service.*

   f. Repeat the last step for the link item **ToolsDocu.**

g. To test the link to the service, choose **Test Link** from the context menu.
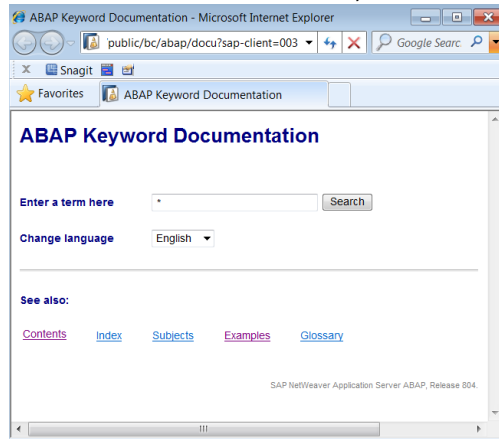
Figure: *Default page that is displayed after successful service call*

4. If you want to activate the preview service, use the sub-steps below:

a. Expand the node **default_host > sap > bc > webdynpro > sap**.

The subtree displays the service item wdy_aie_vd_preview.

b. Select the service **wdy_aie_vd_preview**.

c. From the context menu, choose **Activate Service**.
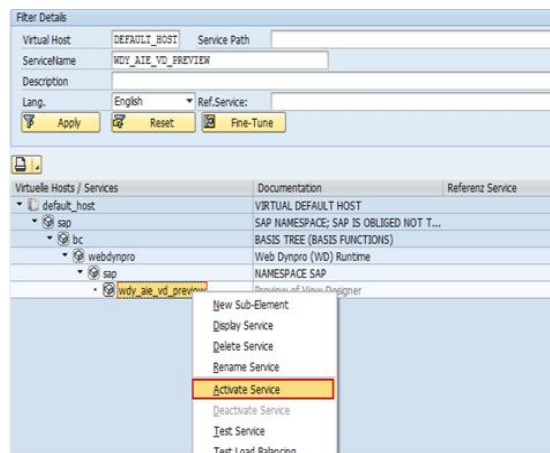
Figure: *Activating wdy_aie_vd_preview service*

4 Configuring Profile Parameters

In addition to logon tickets, ABAP systems can also issue the more restricted assertion tickets when accessing system services. This might be important in order to meet maximum security requirements when you are using the integrated SAP GUI in ABAP Development Tools.

Instead of asking for the password, the back-end system checks the validity of the assertion ticket to allow the user access to system services.

Procedure

Use the following settings to configure your ABAP system in such a way that it issues the assertion tickets only (without logon).

Profile Parameters

Parameter	Value	Description
login/create_sso2_ticket	3	The parameter with this value causes the ABAP system to issue assertion tickets only (no logon tickets).
login/accept_sso2_ticket	1	The parameter with this value configures the ABAP system to accept assertion tickets.

Table: *List of profile parameters*

Copyright