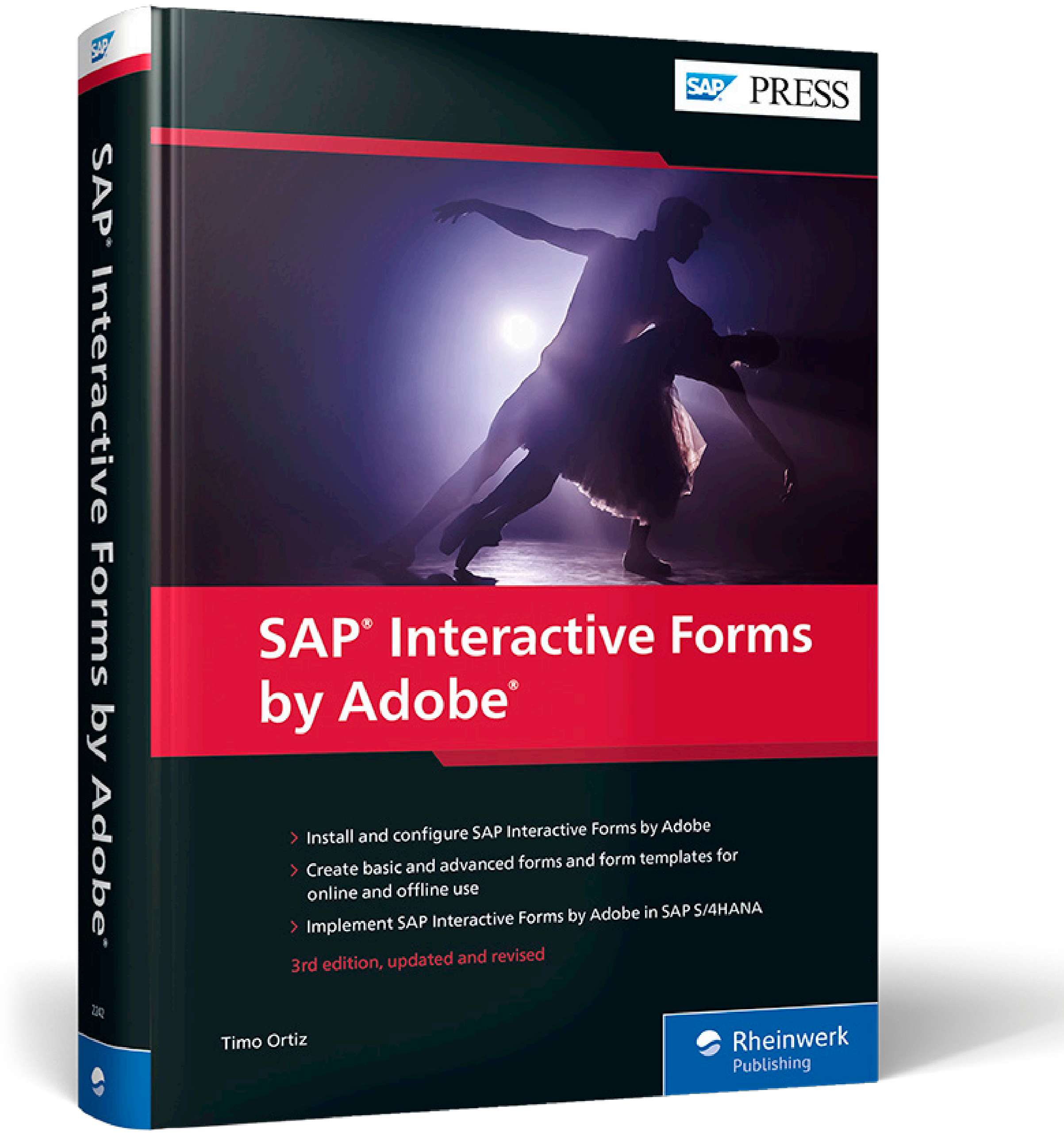
SAP PRESS First-hand knowledge.

Rheinwerk Publishing


SAP PRESS
SAP® Interactive Forms by Adobe®
› Install and configure SAP Interactive Forms by Adobe
› Create basic and advanced forms and form templates for online and offline use
› Implement SAP Interactive Forms by Adobe in SAP S/4HANA
3rd edition, updated and revised
Timo Ortiz
Rheinwerk Publishing
SAP® Interactive Forms by Adobe®

Browse the Book

This sample chapter guides you through the setup and installation of Adobe Document Services (ADS). You'll get step-by-step instructions for each configuration process, including automatic, basic, scenario-dependent, parallelization, and optional configuration. Then, you'll learn how to operate ADS, including monitoring individual components and settings. The chapter concludes with an overview of the ADS hub concept, which allows ADS web services to operate as a standalone system compatible with both ABAP and Java.

"Installing and Configuring Adobe Document Services"

Contents

Index

The Author

Timo Ortiz

SAP Interactive Forms by Adobe

724 pages, 3rd, updated and revised edition 2022, $99.95
ISBN 978-1-4932-2242-1

www.sap-press.com/5494

Chapter 3

Installing and Configuring Adobe Document Services

In this chapter, we show you how to set up Adobe Document Services and prepare it for forms output. Furthermore, you'll get an overview of the scenario-dependent configuration of the runtime components as well as optional setting options.

To work with SAP Interactive Forms by Adobe, you need Adobe Document Services (ADS) in addition to Adobe LiveCycle Designer. These are integrated into SAP NetWeaver Application Server Java (SAP NetWeaver AS Java). You also need SAP NetWeaver Application Server ABAP (SAP NetWeaver AS ABAP) to run other scenarios described in this book. It's assumed in the following that the ABAP and Java application servers are already installed, namely in the versions SAP NetWeaver AS ABAP 7.54 (SAP_BASIS 754) or higher and SAP NetWeaver AS Java 7.54 or higher.

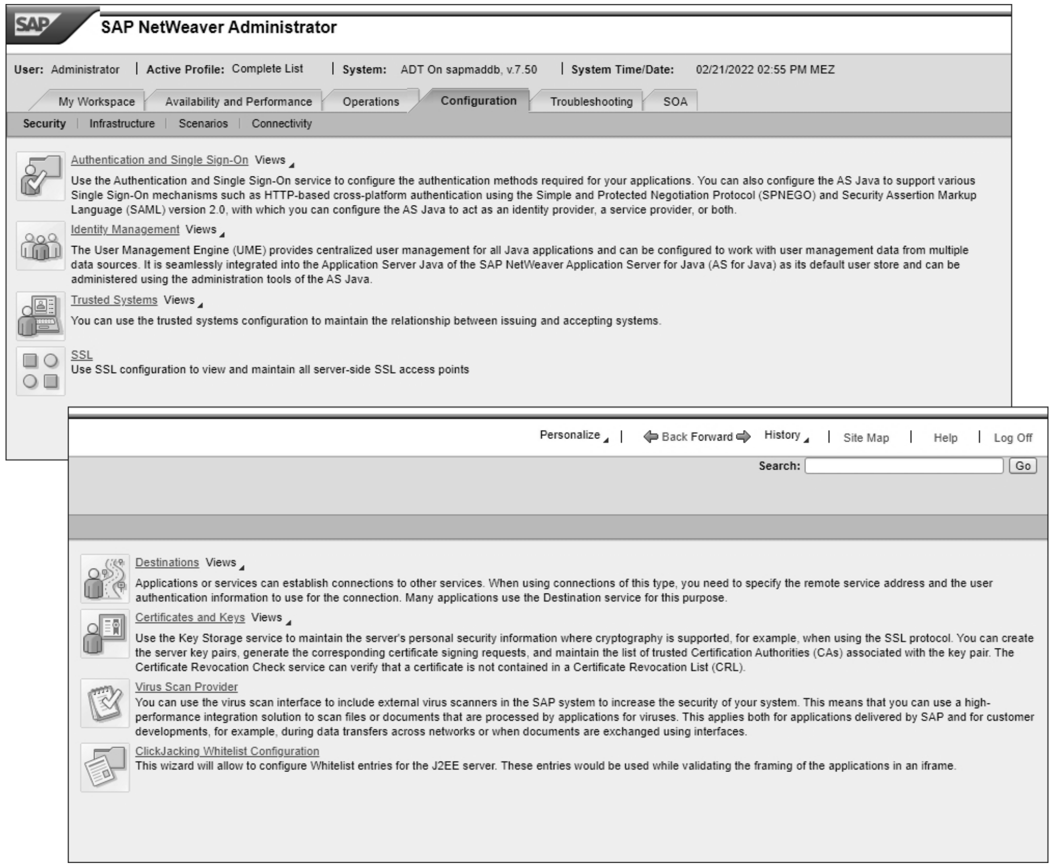
The installation and configuration of ADS usually isn't performed by form developers, but by the Basis team. This chapter shows the special features to be considered for the installation and operation of ADS as well as the necessary configuration.

3.1 Automatic Configuration

We describe in detail the configuration required to operate ADS in an ABAP environment in the following sections. It's important to know the basic concepts to find errors quickly. This also enables you to adjust the necessary settings in case there are changes in the system landscape. However, you can also have the configuration performed automatically. SAP delivers a Central Technical Configuration (CTC) template. This can be used to set up an ABAP system and a Java system so that ADS can be used directly. This template can also be called several times, for example, if you want to operate a second ABAP system with the same ADS. These ADSs are then operated as a hub (Section 3.7). Some configuration steps can then be omitted because the configuration is based on a wizard.

You start the automatic configuration in SAP NetWeaver Administrator using the following steps:

1. Select the **Configuration** tab and then the **Scenarios** tab below it (see Figure 3.1).


SAP NetWeaver Administrator
User: Administrator | Active Profile: Complete List | System: ADT On sapmaddb, v.7.50 | System Time/Date: 02/21/2022 02:55 PM MEZ
My Workspace
Availability and Performance
Operations
Configuration
Troubleshooting
SOA
Security | Infrastructure | Scenarios | Connectivity
Authentication and Single Sign-On Views
Identity Management Views
Trusted Systems Views
SSL
Personalize | Back Forward History | Site Map | Help | Log Off
Search: Go
Destinations Views
Certificates and Keys Views
Virus Scan Provider
ClickJacking Whitelist Configuration

Use the Authentication and Single Sign-On service to configure the authentication methods required for your applications. You can also configure the AS Java to support various Single Sign-On mechanisms such as HTTP-based cross-platform authentication using the Simple and Protected Negotiation Protocol (SPNEGO) and Security Assertion Markup Language (SAML) version 2.0, with which you can configure the AS Java to act as an identity provider, a service provider, or both.

The User Management Engine (UME) provides centralized user management for all Java applications and can be configured to work with user management data from multiple data sources. It is seamlessly integrated into the Application Server Java of the SAP NetWeaver Application Server for Java (AS for Java) as its default user store and can be administered using the administration tools of the AS Java.

You can use the trusted systems configuration to maintain the relationship between issuing and accepting systems.

Use SSL configuration to view and maintain all server-side SSL access points

Applications or services can establish connections to other services. When using connections of this type, you need to specify the remote service address and the user authentication information to use for the connection. Many applications use the Destination service for this purpose.

Use the Key Storage service to maintain the server's personal security information where cryptography is supported, for example, when using the SSL protocol. You can create the server key pairs, generate the corresponding certificate signing requests, and maintain the list of trusted Certification Authorities (CAs) associated with the key pair. The Certificate Revocation Check service can verify that a certificate is not contained in a Certificate Revocation List (CRL).

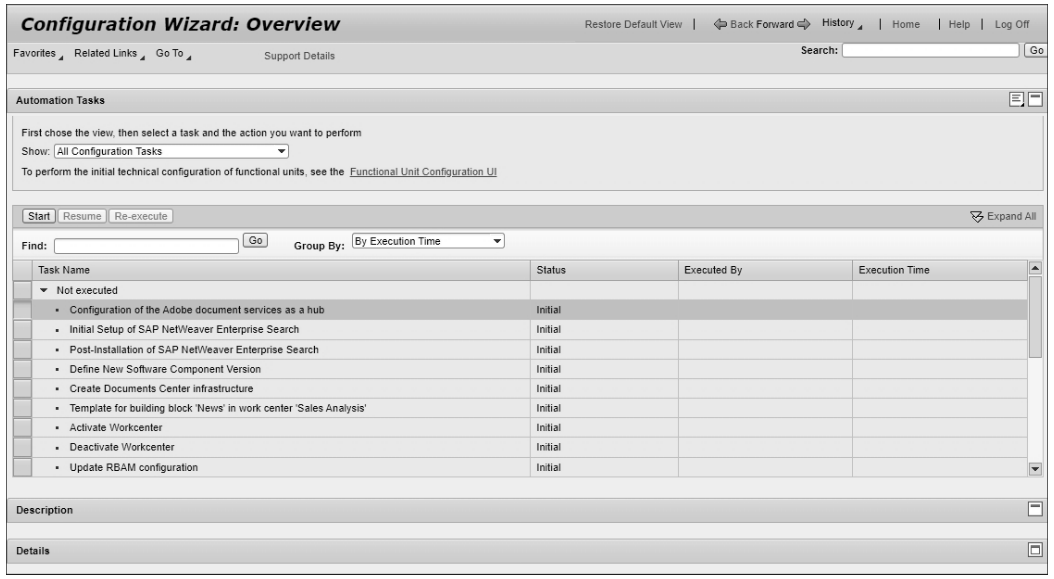
You can use the virus scan interface to include external virus scanners in the SAP system to increase the security of your system. This means that you can use a high-performance integration solution to scan files or documents that are processed by applications for viruses. This applies both for applications delivered by SAP and for customer developments, for example, during data transfers across networks or when documents are exchanged using interfaces.

This wizard will allow to configure Whitelist entries for the J2EE server. These entries would be used while validating the framing of the applications in an iframe.

Figure 3.1 SAP NetWeaver Administrator: Configuration

2. Start the configuration wizard.
3. Select the **Configuration of the Adobe document services as a hub** entry, and click the **Start** button (see Figure 3.2).

The system then guides you through the configuration in 14 steps.


Configuration Wizard: Overview
Restore Default View | Back Forward History | Home | Help | Log Off
Favorites Related Links Go To
Support Details
Search: Go
Automation Tasks
Functional Unit Configuration UI
Start
Resume
Re-execute
Expand All
Find: Go
Description
Details

First chose the view, then select a task and the action you want to perform

Show: All Configuration Tasks

To perform the initial technical configuration of functional units, see the Functional Unit Configuration UI

Group By: By Execution Time

Task Name	Status	Executed By	Execution Time
▼ Not executed			
• Configuration of the Adobe document services as a hub	Initial		
• Initial Setup of SAP NetWeaver Enterprise Search	Initial		
• Post-Installation of SAP NetWeaver Enterprise Search	Initial		
• Define New Software Component Version	Initial		
• Create Documents Center infrastructure	Initial		
• Template for building block 'News' in work center 'Sales Analysis'	Initial		
• Activate Workcenter	Initial		
• Deactivate Workcenter	Initial		
• Update RBAM configuration	Initial		

Figure 3.2 Configuring Adobe Document Services with the Configuration Wizard

3.2 Basic Configuration

In this section, we describe the steps required to set up ADS. Here we distinguish between basic, scenario-dependent, and optional configuration, and then we present the tools and verification options used for each configuration aspect.

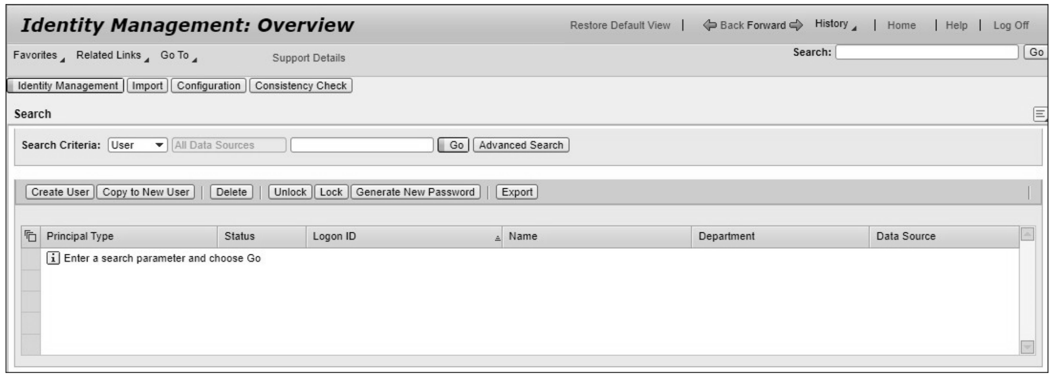
ADS can be called via a web service interface. The exposed functionality is encapsulated in the PDF object, which is available in both ABAP and Java environments. Consequently, to use ADS, access to the web service must be configured. The following steps are necessary for this.

3.2.1 Creating a Technical User

First, you create a user in the Java system through which ADS can be called. This user is usually called `ADSUser` and is created through the following steps:

1. Start SAP NetWeaver Administrator via the URL *http://<server>:<port>/nwa*, for example, *http://localhost:50000/nwa*, if you're locally on the Java system. The logon page opens.
2. Enter "Administrator" as the user name, enter the corresponding password, and click **Log in.**
3. In the following dialog, select the **Configuration** tab, and click the **Identity Management** link (refer to Figure 3.1).

4. You're now in the user administration of the Java system (see Figure 3.3). First check whether the user `ADSUser` already exists by entering "ADSUser" in the input field in Figure 3.3 and clicking the **Go** button. If the user doesn't exist, create it using the following steps. If the user already exists, you can check the settings analogously.

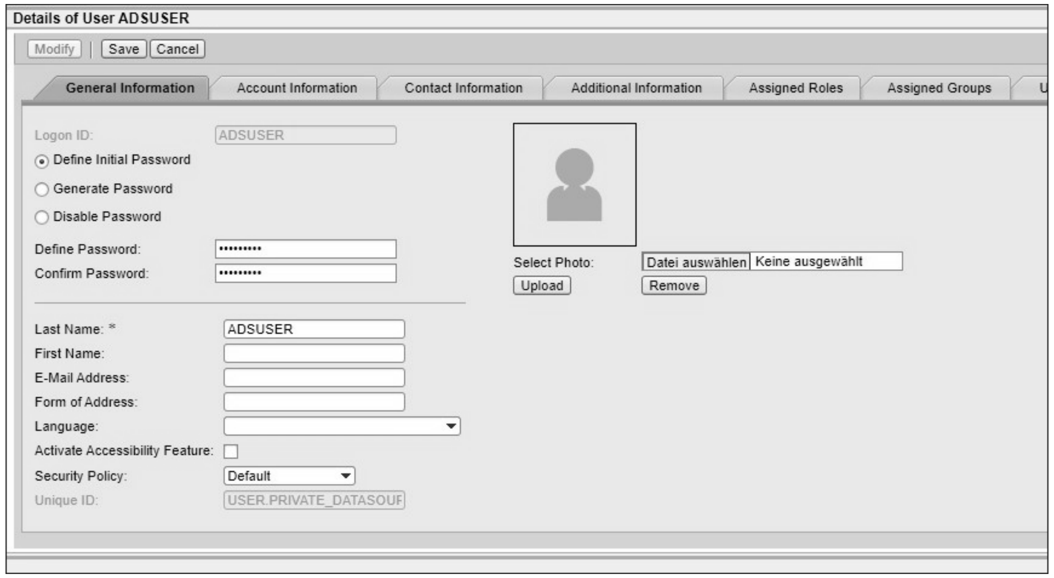


Search Criteria: User ▾ All Data Sources [ ] Go Advanced Search

Principal Type	Status	Logon ID	Name	Department	Data Source
Enter a search parameter and choose Go					

Figure 3.3 Identity Management (User Administration)

5. Click the **Create User** button.
6. In the next screen, set the password and other technical settings (see Figure 3.4). You must remember the password because you'll need it later during the web service configuration.



Logon ID: ADSUSER

● Define Initial Password
○ Generate Password
○ Disable Password

Define Password: •••••••••
Confirm Password: •••••••••

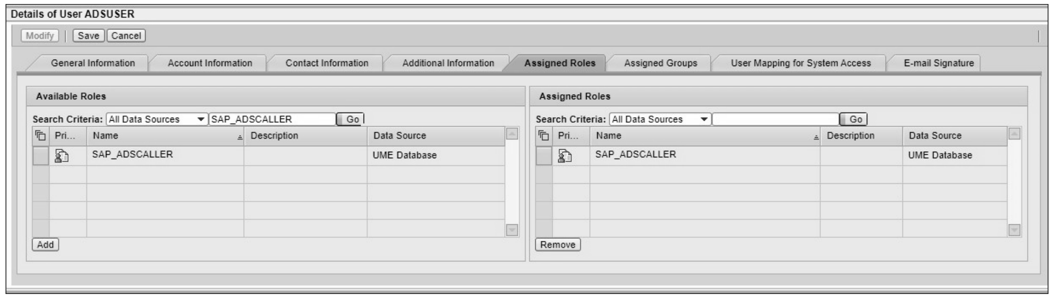
Select Photo: Datei auswählen Keine ausgewählt

Last Name: * ADSUSER
First Name:
E-Mail Address:
Form of Address:
Language:
Activate Accessibility Feature: □
Security Policy: Default
Unique ID: USER.PRIVATE_DATASOUF

Figure 3.4 Creating User ADSUser

7. Enter any name in the **Last Name** field, as this is a mandatory field. In the **Security-Policy** selection field, select the **Technical User** option.

8. Switch to the **Assigned Roles** tab (see Figure 3.5). Enter "SAP_ADSCALLER" as the role on the left of the screen, and click the **Go** button.



Available Roles

Search Criteria: All Data Sources ▾ SAP_ADSCALLER Go

Pri...	Name	Description	Data Source
	SAP_ADSCALLER		UME Database

Assigned Roles

Search Criteria: All Data Sources ▾ [ ] Go

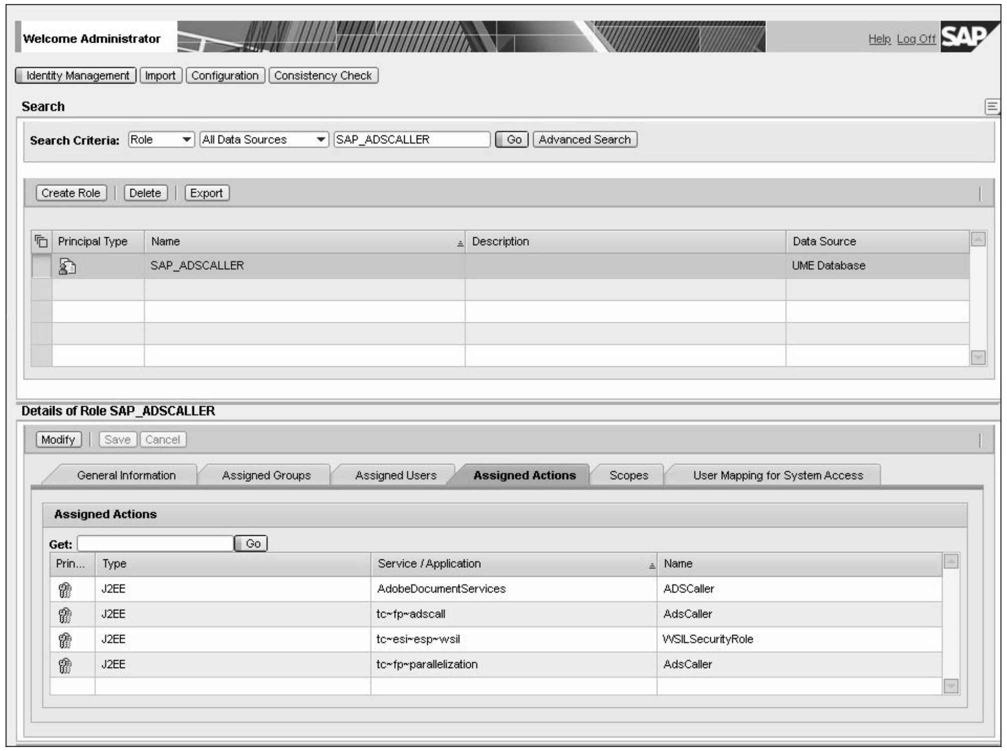
Pri...	Name	Description	Data Source
	SAP_ADSCALLER		UME Database

Figure 3.5 Role Maintenance in User Administration

9. Highlight this role in the table, and click **Add** to assign it to the `ADSUser` user. Save the user settings.

To access ADS, the role `SAP_ADSCALLER` is linked to various applications. You can check these links in SAP NetWeaver Administrator in **Identity Management** with these steps:

1. Select **Role** in the **Search Criteria** field, and enter "SAP_ADSCALLER" in the input field to the right. Click the **Go** button (see Figure 3.6).



Search Criteria: Role ▾ All Data Sources ▾ SAP_ADSCALLER Go

Principal Type	Name	Description	Data Source
	SAP_ADSCALLER		UME Database

Get: [ ] Go

Prin...	Type	Service / Application	Name
	J2EE	AdobeDocumentServices	ADSCaller
	J2EE	tc~fp~adscall	AdsCaller
	J2EE	tc~esi~esp~wsil	WSILSecurityRole
	J2EE	tc~fp~parallelization	AdsCaller

Figure 3.6 Linking the Role

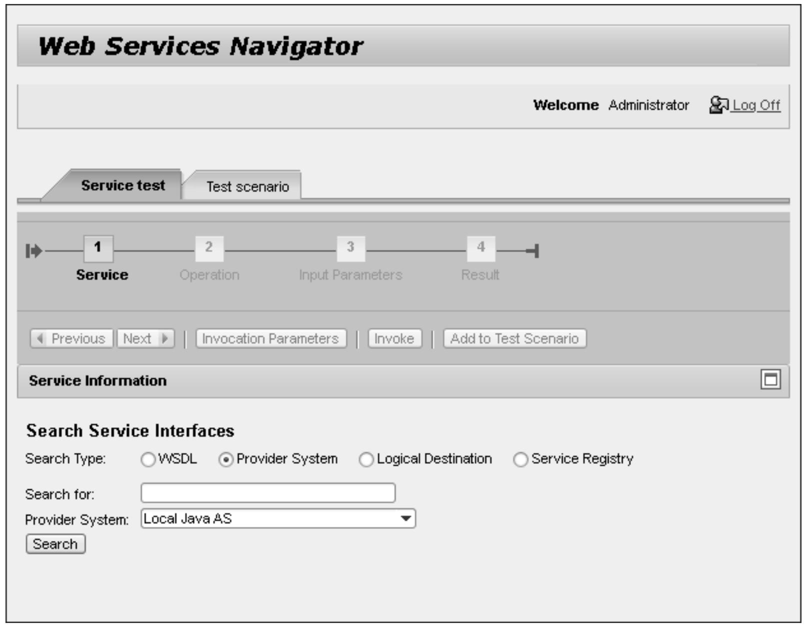
2. The `SAP_ADSCALLER` role is displayed as a hit. Select this line.
3. In the **Details** area **for role SAP_ADSCALLER,** you can see the various applications that are linked to the role under the **Assigned Actions** tab. Check whether the selection matches the representation in Figure 3.6.

The first steps to configure ADS are now complete. You can close SAP NetWeaver Administrator via the **Logout** button.

3.2.2 Web Service Test

The previously described configuration steps enable the password-protected call of ADS. You can test this using the *Web Services Navigator* as described in the following steps:

1. Call the URL *http://<server>:<port>/wsnavigator*, and log in as administrator with the appropriate password. Use the same password as before when calling SAP NetWeaver Administrator. The **Web Services Navigator** screen appears (see Figure 3.7).



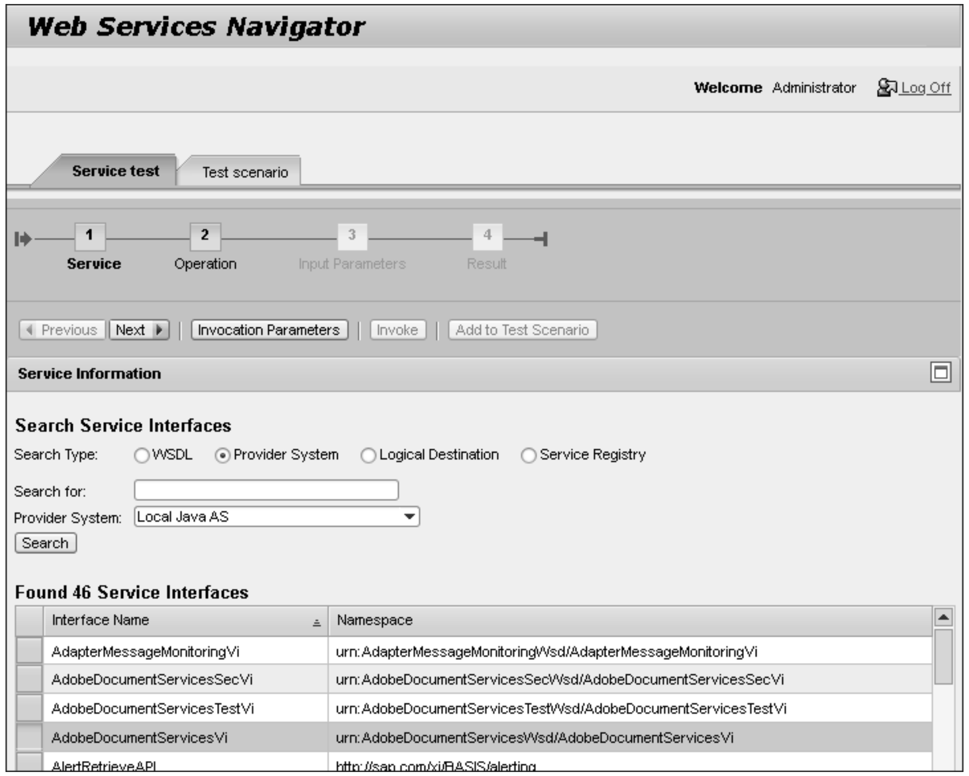
Search Type: ○ WSDL ● Provider System ○ Logical Destination ○ Service Registry

Search for:

Provider System: Local Java AS

Figure 3.7 Web Services Navigator

2. Next to **Search Type**, select the **Provider System** radio button, and click the **Search** button.
3. From the displayed list of available web services (see Figure 3.8), select **AdobeDocumentServicesVi**, and confirm with **Next**.
4. Under **Operation**, you'll see the `rpData` method for testing ADS. Select **rpData**, and click **Next** (see Figure 3.9).



Search Type: ○ WSDL ● Provider System ○ Logical Destination ○ Service Registry

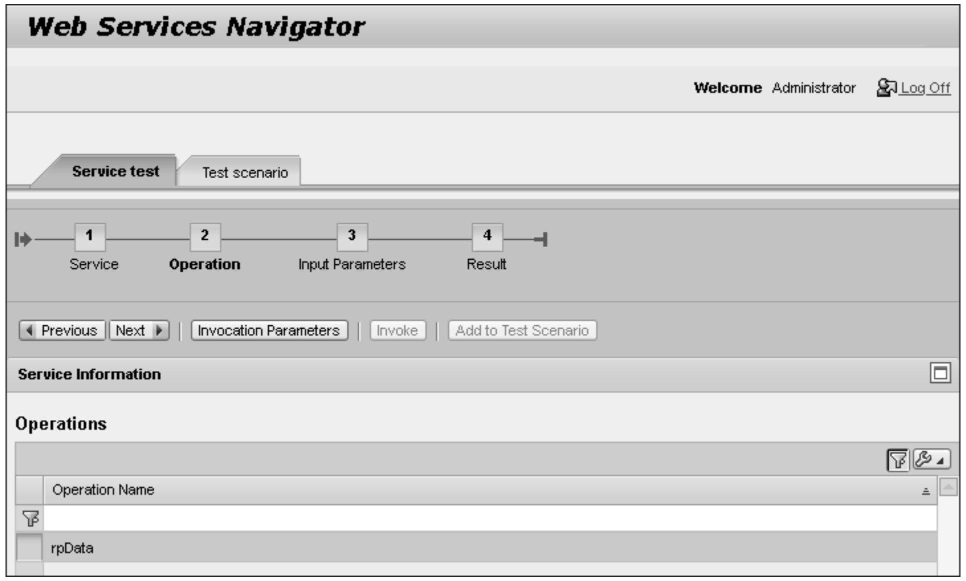
Search for:

Provider System: Local Java AS

Found 46 Service Interfaces

Interface Name	Namespace
AdapterMessageMonitoringVi	urn:AdapterMessageMonitoringWsd/AdapterMessageMonitoringVi
AdobeDocumentServicesSecVi	urn:AdobeDocumentServicesSecWsd/AdobeDocumentServicesSecVi
AdobeDocumentServicesTestVi	urn:AdobeDocumentServicesTestWsd/AdobeDocumentServicesTestVi
AdobeDocumentServicesVi	urn:AdobeDocumentServicesWsd/AdobeDocumentServicesVi
AlertRetrieveAPI	http://sap.com/xi/BASIS/alerting

Figure 3.8 Selecting the Web Service

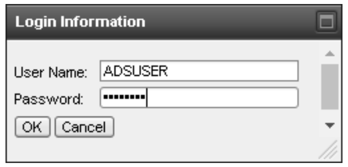


Operations

Operation Name
rpData

Figure 3.9 Adobe Document Services Testing Method

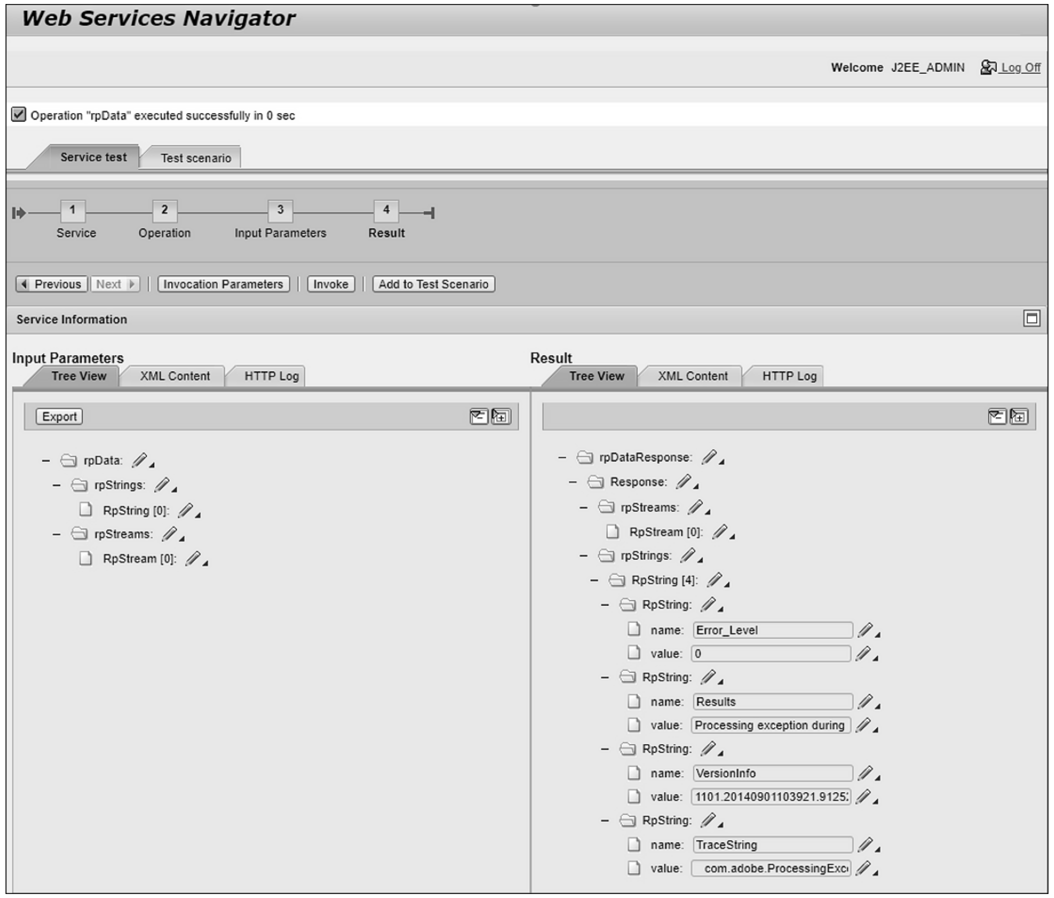
5. In the following step, you can still set call parameters for the execution of the method. However, this isn't necessary for our test. Click on the **Next** button.
6. A window for entering the user parameters will appear (see Figure 3.10). Enter the user "ADSUSER" and the password you chose in the previous section.



User Name: ADSUSER
Password: ••••••••

Figure 3.10 Login for the Web Service Test

7. After clicking **OK**, you'll receive the response of the web service call (see Figure 3.11).



☑ Operation "rpData" executed successfully in 0 sec

```
– rpData:
  – rpStrings:
      RpString [0]:
  – rpStreams:
      RpStream [0]:
```

```
– rpDataResponse:
  – Response:
    – rpStreams:
        RpStream [0]:
    – rpStrings:
      – RpString [4]:
        – RpString:
            name:  Error_Level
            value: 0
        – RpString:
            name:  Results
            value: Processing exception during
        – RpString:
            name:  VersionInfo
            value: 1101.20140901103921.9125:
        – RpString:
            name:  TraceString
            value: com.adobe.ProcessingExc
```

Figure 3.11 Result of the Web Service Test

You don't specify any further data in this test and thus can't simulate correct processing by ADS. You only want to check whether the communication could be established

successfully. Without any data, you'll see an error message (**com.adobe.ProcessingException: Required stream: "PDFDocument" not found in request OR its length is zero**) in the **Result** area, as expected. However, this indicates that the web service call was successful—after all, it came from ADS.

3.3 Scenario-Dependent Configuration

In the previous section, we described the basic configuration of ADS required for all scenarios. In the following, we'll now go into the configuration steps required for the different uses. Figure 3.12 shows the components relevant for form processing and discussed in this book, as well as the communication channels they use.

The web service functions as an interface to ADS. Two different client types access this interface: the *RFC destination* (the ABAP application server) and the *web service proxy* (the Java application server).

The Simple Object Access Protocol (SOAP) framework enables the creation and use of web services based on SOAP (for more information, visit *http://s-prs.co/v224206*).

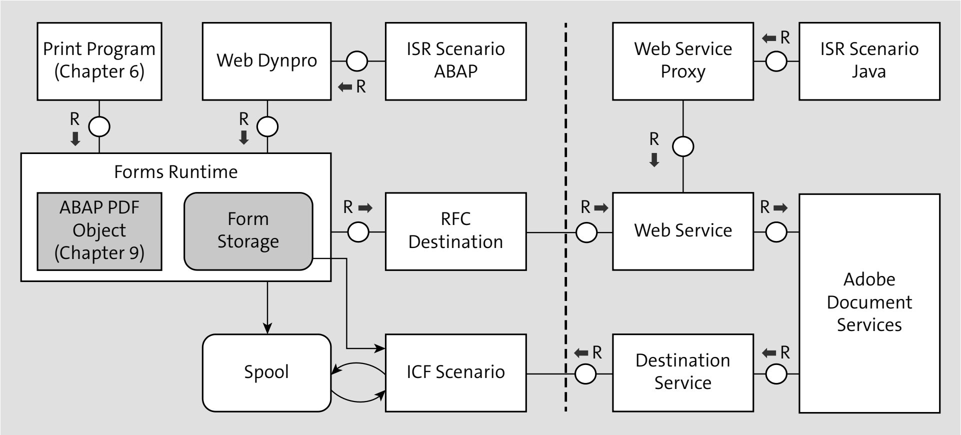


Figure 3.12 SAP Interactive Forms by Adobe Architecture

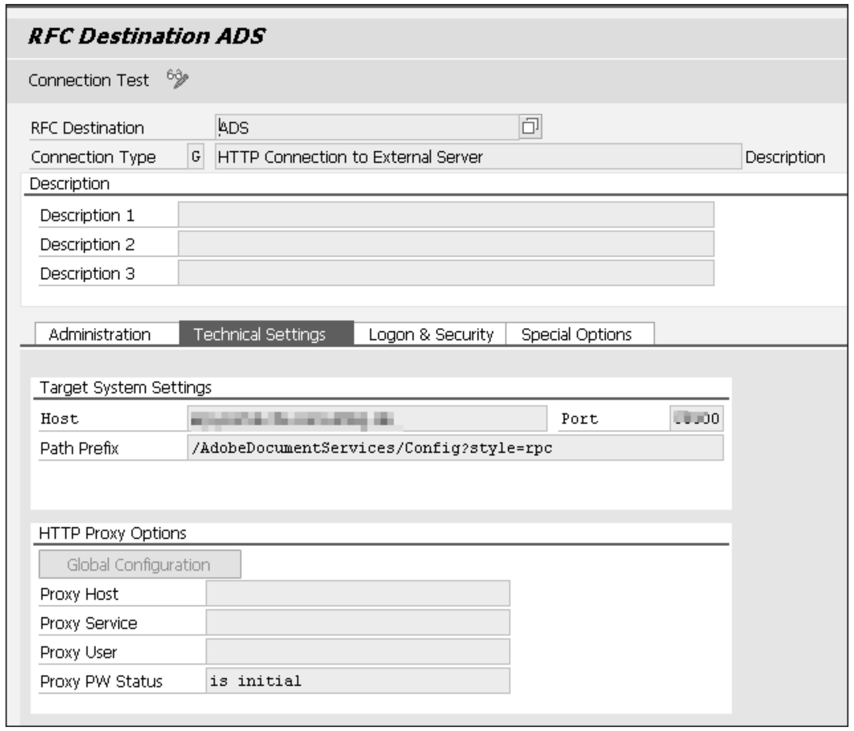
Both ABAP print programs and applications in Web Dynpro ABAP use the ABAP PDF object at runtime, which requires a configured remote function call (RFC) connection to communicate with ADS (the RFC destination). Form scenarios that run on the Java application server, such as applications in Web Dynpro Java, access the ADS web service through a proxy (the web service proxy).

There is an additional return channel for the ABAP form runtime: the Internet Communication Framework (ICF) service. To access this interface, a *destination* must be configured for ADS. This return channel is used for reloading form templates because these are cached by ADS and read only when needed. In addition, application data and documents generated by ADS are transferred via this communication path during mass printing; that is, the web service call itself only contains the control data in these scenarios.

3.3.1 Setting Up the RFC Destination

To be able to call ADS from an ABAP application server, an RFC connection must be set up via Transaction SM59 (Configuration of RFC Connections) as follows:

1. Click the **Create** button, and enter a name for the connection (see Figure 3.13). The default connection must be named "ADS", but other connections can be configured. The application program controls the ADS call by passing the connection name. For the connection type, select **G HTTP Connection to External Server**.



RFC Destination ADS

RFC Destination	ADS
Connection Type	G HTTP Connection to External Server
Description 1	
Description 2	
Description 3	

Target System Settings

Host	[redacted]	Port	[redacted]
Path Prefix	/AdobeDocumentServices/Config?style=rpc		

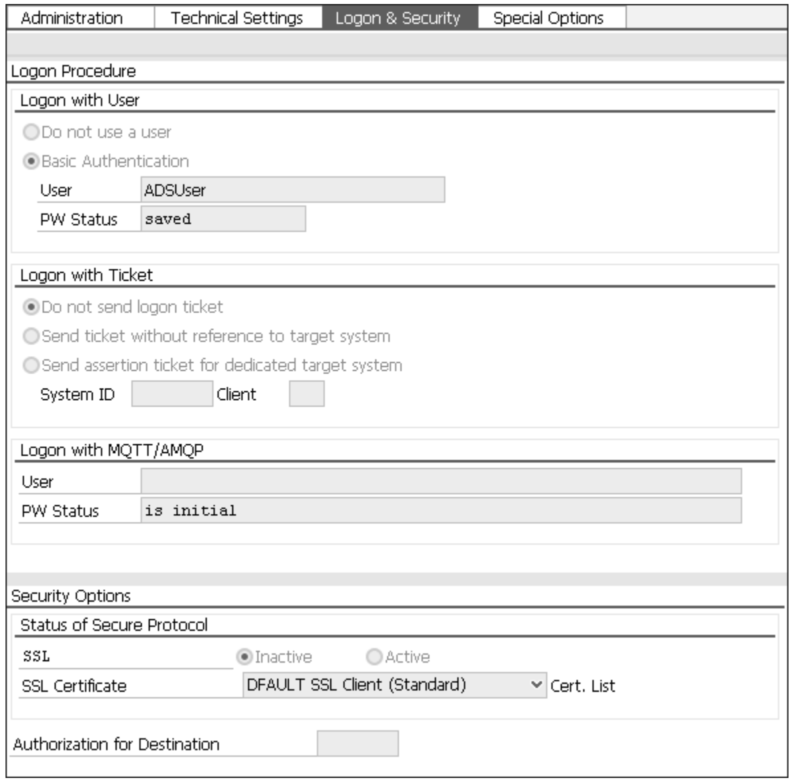
HTTP Proxy Options

Proxy Host	
Proxy Service	
Proxy User	
Proxy PW Status	is initial

Figure 3.13 RFC Connection Configuration

2. On the **Technical Settings** tab in block **Target System Setting**, fill in the **Host** field with the System name, **Port** with the HTTP port, and **Path Prefix** with the path of the ADS web service ("/AdobeDocumentServices/Config?style=rpc"). After entering the path, you get a warning that you can skip by pressing Enter.

3. Switch to the **Logon & Security** tab, and activate the **Basic Authentication** radio button. In the **User** field, enter "ADSUser" and the password you've chosen (see Figure 3.14). To finish the configuration, click the **Save** button.



Logon Procedure

Logon with User

○ Do not use a user
● Basic Authentication

User	ADSUser
PW Status	saved

Logon with Ticket

● Do not send logon ticket
○ Send ticket without reference to target system
○ Send assertion ticket for dedicated target system

System ID		Client	

Logon with MQTT/AMQP

User	
PW Status	is initial

Security Options

Status of Secure Protocol

SSL ● Inactive ○ Active

SSL Certificate	DFAULT SSL Client (Standard)	Cert. List
Authorization for Destination		

Figure 3.14 Adobe Document Services Login Procedure

You can then check the RFC connection using program FP_PDF_TEST_00. The connection test offered in Transaction SM59 doesn't work because it doesn't support SOAP. Therefore, call Transaction SE38. If you don't have authorization to do this, you can use Transaction SA38 to execute it if necessary. Execute program FP_PDF_TEST_00 there, specifying the RFC connection you've configured (ADS in our example). If successful, the internal version number of ADS is returned.

3.3.2 Setting Up the Destination and Internet Communication Framework Service

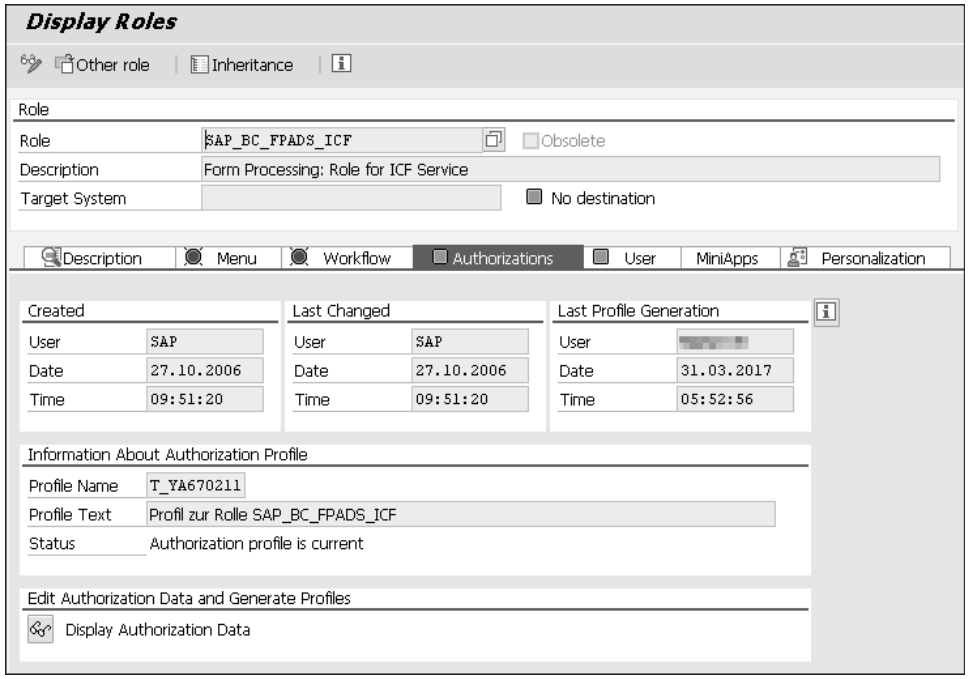
While ADS is called using SOAP via the RFC connection configured in Transaction SM59, the actual payload data—such as form template and business data (large amounts of data in some scenarios)—is transported via a separate HTTP connection. This enables form template caching by ADS because the call contains only a reference to the template used. In addition, a time stamp is used to decide whether an update of the local cache is necessary or whether the already-cached template can be used. Furthermore, the separate connection circumvents limitations of SOAP because this protocol wasn't designed for the exchange of very large amounts of data. In this section, we'll discuss how to set up the HTTP connection, including creating a service user, activating and testing the relevant ICF services, and creating and testing the destination on the Java application server.

Create Service User for HTTP Communication

You must create a service user through which the communication between ADS and the ABAP application server takes place. This user is assigned the role `SAP_BC_FPADS_ICF`.

First, you need to generate the profile associated with this role, if this hasn't already been done in your system, with the following steps:

1. Call Transaction PFCG (Role Maintenance) to display the role.
2. Switch to the **Authorizations** tab, and click the **Display Authorization Data** button in the **Edit Authorization Data and Generate Profiles** area (see Figure 3.15).
3. You'll now see all the permissions assigned to this role and can generate a current profile using the **Generate** button.



Role

Role SAP_BC_FPADS_ICF ☐ Obsolete

Description Form Processing: Role for ICF Service

Target System ☐ No destination

	Created	Last Changed	Last Profile Generation
User	SAP	SAP	[redacted]
Date	27.10.2006	27.10.2006	31.03.2017
Time	09:51:20	09:51:20	05:52:56

Information About Authorization Profile

Profile Name T_YA670211

Profile Text Profil zur Rolle SAP_BC_FPADS_ICF

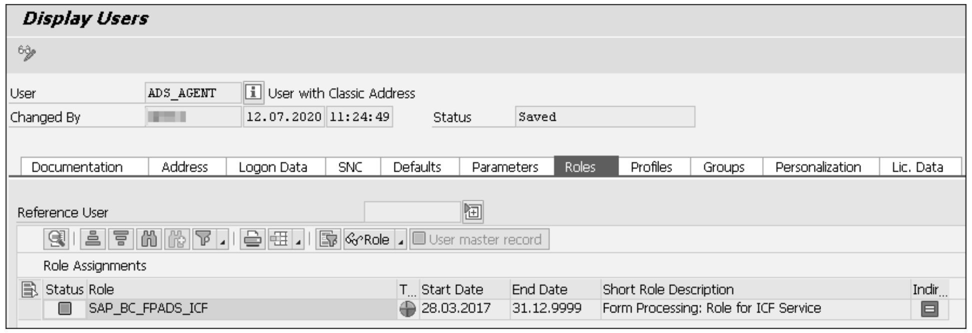
Status Authorization profile is current

Edit Authorization Data and Generate Profiles

Figure 3.15 Display Authorization Data in Role Maintenance

After the profile has been regenerated, you can create the service user:

1. Call Transaction SU01 (User Maintenance). Enter the name "ADS_AGENT", and click the **Create** button.
2. On the **Address** tab, fill in the **Last Name** field; it's recommended to repeat the user name here.
3. Switch to the **Logon Data** tab, and select **Service** as the **user type** there. In addition, set a password.
4. Switch to the **Roles** tab, and assign the `SAP_BC_FPADS_ICF` role to the user. Finally, click the **Save** button (see Figure 3.16).



User ADS_AGENT User with Classic Address

Changed By [redacted] 12.07.2020 11:24:49 Status Saved

Reference User

Status	Role	T...	Start Date	End Date	Short Role Description	Indir...
■	SAP_BC_FPADS_ICF	⊕	28.03.2017	31.12.9999	Form Processing: Role for ICF Service	⊟

Figure 3.16 Maintain User

Role SAP_BC_FP_ICF Is No Longer Required

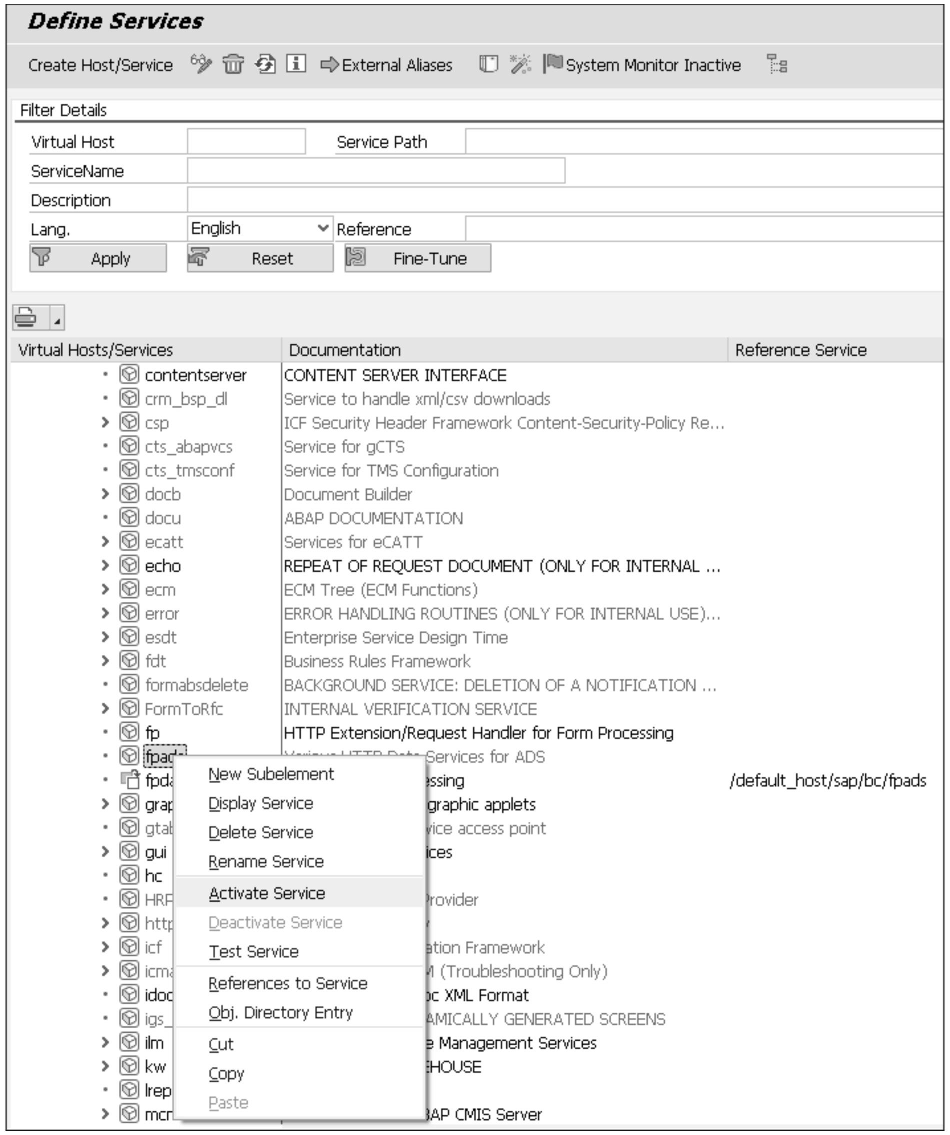
In older releases, user `ADS_AGENT` is also still assigned to role `SAP_BC_FP_ICF`, which was introduced before `SAP_BC_FPADS_ICF`. Because the new role contains all authorizations of the first role, `SAP_BC_FP_ICF` is no longer needed.

Activate Internet Communication Framework Services

Next, you have to activate the two ICF services that are called by ADS. To do this, call Transaction SICF (Maintenance of Services), and click the **Execute** button. Navigate to the **default_host/sap/bc/fp** node, and select the **Activate Service** option from the context menu (see Figure 3.17). Repeat this for the **default_host/sap/bc/fpads** node.

Test the Service User and the Internet Communication Framework Services

You can test the ICF services that are used to access the ABAP form store, among other things, using a web browser. You must first determine the PC name and the HTTP port of the ABAP application server. To do this, call Transaction SICF, click the **Run** button, and then select **Port Information** from the **Goto** menu. A table is displayed in a dialog box. In the **Protocol** column, find the **HTTP** entry, and note or copy the corresponding entries in the **Hostname** and **Service** columns.

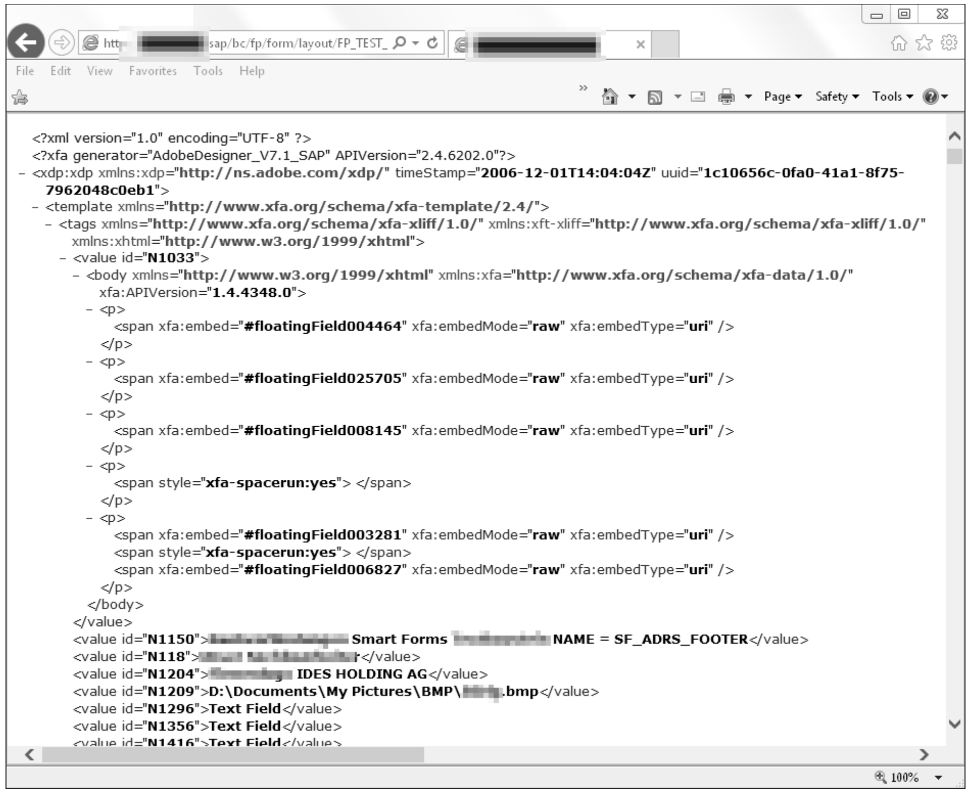


Virtual Host		Service Path	
ServiceName			
Description			
Lang.	English	Reference	

Virtual Hosts/Services	Documentation	Reference Service
contentserver	CONTENT SERVER INTERFACE	
crm_bsp_dl	Service to handle xml/csv downloads	
csp	ICF Security Header Framework Content-Security-Policy Re…	
cts_abapvcs	Service for gCTS	
cts_tmsconf	Service for TMS Configuration	
docb	Document Builder	
docu	ABAP DOCUMENTATION	
ecatt	Services for eCATT	
echo	REPEAT OF REQUEST DOCUMENT (ONLY FOR INTERNAL …	
ecm	ECM Tree (ECM Functions)	
error	ERROR HANDLING ROUTINES (ONLY FOR INTERNAL USE)…	
esdt	Enterprise Service Design Time	
fdt	Business Rules Framework	
formabsdelete	BACKGROUND SERVICE: DELETION OF A NOTIFICATION …	
FormToRfc	INTERNAL VERIFICATION SERVICE	
fp	HTTP Extension/Request Handler for Form Processing	
fpads	Services for ADS	
fpda	ssing	/default_host/sap/bc/fpads
grap	graphic applets	
gtab	ice access point	
gui	ices	
hc		
HRP	Provider	
http		
icf	ation Framework	
icma	M (Troubleshooting Only)	
idoc	c XML Format	
igs_	AMICALLY GENERATED SCREENS	
ilm	e Management Services	
kw	EHOUSE	
lrep		
mcr	AP CMIS Server	

Figure 3.17 Activate ICF Services

Now start a browser, and enter the following URL: "http://<server>:<port>/sap/bc/fp/form/layout/FP_TEST_03_TABLE.XDP". Here, *<server>* corresponds to the hostname determined earlier, and *<port>* corresponds to the service number. You'll now be prompted for a user name and password. Enter "ADS_AGENT" here and the password you specified. After successful authentication, you'll see an XML file, which is template `FP_TEST_03_TABLE` stored in the ABAP form repository in the original language (see Figure 3.18).



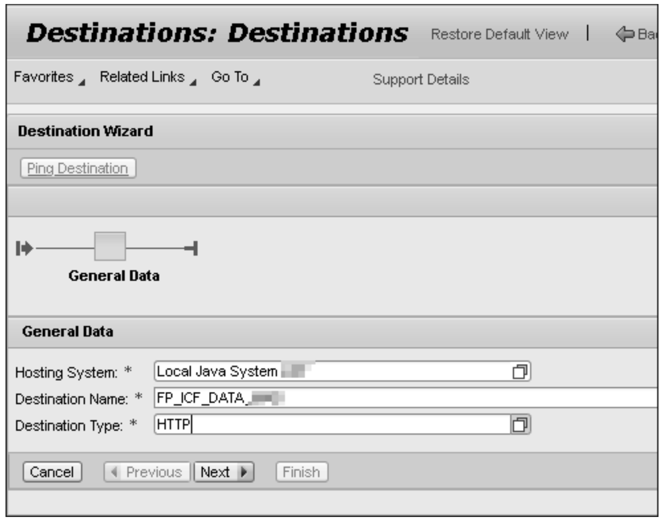
```
  <?xml version="1.0" encoding="UTF-8" ?>
  <?xfa generator="AdobeDesigner_V7.1_SAP" APIVersion="2.4.6202.0"?>
- <xdp:xdp xmlns:xdp="http://ns.adobe.com/xdp/" timeStamp="2006-12-01T14:04:04Z" uuid="1c10656c-0fa0-41a1-8f75-
    7962048c0eb1">
  - <template xmlns="http://www.xfa.org/schema/xfa-template/2.4/">
    - <tags xmlns="http://www.xfa.org/schema/xfa-xliff/1.0/" xmlns:xft-xliff="http://www.xfa.org/schema/xfa-xliff/1.0/"
        xmlns:xhtml="http://www.w3.org/1999/xhtml">
      - <value id="N1033">
        - <body xmlns="http://www.w3.org/1999/xhtml" xmlns:xfa="http://www.xfa.org/schema/xfa-data/1.0/"
            xfa:APIVersion="1.4.4348.0">
          - <p>
              <span xfa:embed="#floatingField004464" xfa:embedMode="raw" xfa:embedType="uri" />
            </p>
          - <p>
              <span xfa:embed="#floatingField025705" xfa:embedMode="raw" xfa:embedType="uri" />
            </p>
          - <p>
              <span xfa:embed="#floatingField008145" xfa:embedMode="raw" xfa:embedType="uri" />
            </p>
          - <p>
              <span style="xfa-spacerun:yes"> </span>
            </p>
          - <p>
              <span xfa:embed="#floatingField003281" xfa:embedMode="raw" xfa:embedType="uri" />
              <span style="xfa-spacerun:yes"> </span>
              <span xfa:embed="#floatingField006827" xfa:embedMode="raw" xfa:embedType="uri" />
            </p>
          </body>
        </value>
        <value id="N1150">[redacted] Smart Forms [redacted] NAME = SF_ADRS_FOOTER</value>
        <value id="N118">[redacted]r</value>
        <value id="N1204">[redacted] IDES HOLDING AG</value>
        <value id="N1209">D:\Documents\My Pictures\BMP\[redacted].bmp</value>
        <value id="N1296">Text Field</value>
        <value id="N1356">Text Field</value>
        <value id="N1416">Text Field</value>
```

Figure 3.18 Performing a Test Call of the FP_TEST_03_TABLE Form

Create a Destination on the Java Application Server

After you've configured the ABAP application server for access by ADS, you must create a destination on the Java application server that refers to the enabled services. The ADS web services then use this destination to access the ICF service in the ABAP application server. To do this, follow these steps:

1. Start SAP NetWeaver Administrator, and select the **Destinations** link on the **Configuration** tab (refer to Figure 3.1).
2. In the next window, **List of Destinations**, click the **Create** button to create a new destination.
3. In the **Destination Name** field, enter "FP_ICF_DATA_<SID>", replacing <SID> with the system ID of the ABAP system.
4. Select **HTTP** in the **Destination Type** field, and click the **Previous** button (see Figure 3.19).
5. For the **URL** field, enter the same server and port in the form *http://<server>:<port>* that you used in the connection test (refer to Figure 3.18). However, this time don't specify a path, as shown in Figure 3.20.

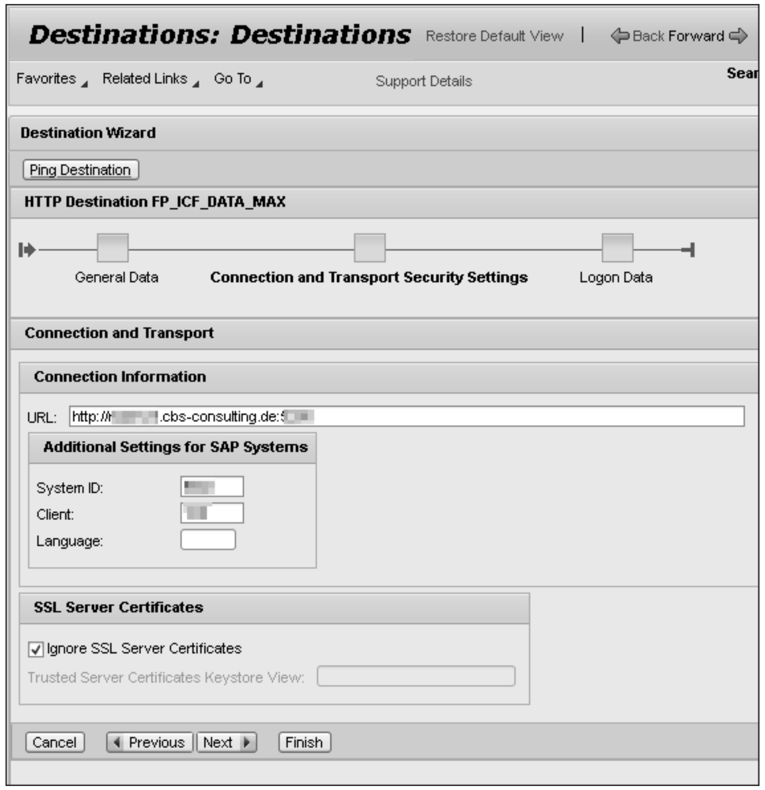


Hosting System: * Local Java System [redacted]

Destination Name: * FP_ICF_DATA_[redacted]

Destination Type: * HTTP

Figure 3.19 Create Destination FP_ICF_DATA_<SID>



HTTP Destination FP_ICF_DATA_MAX

URL: http://[redacted].cbs-consulting.de:[redacted]

System ID: [redacted]

Client: [redacted]

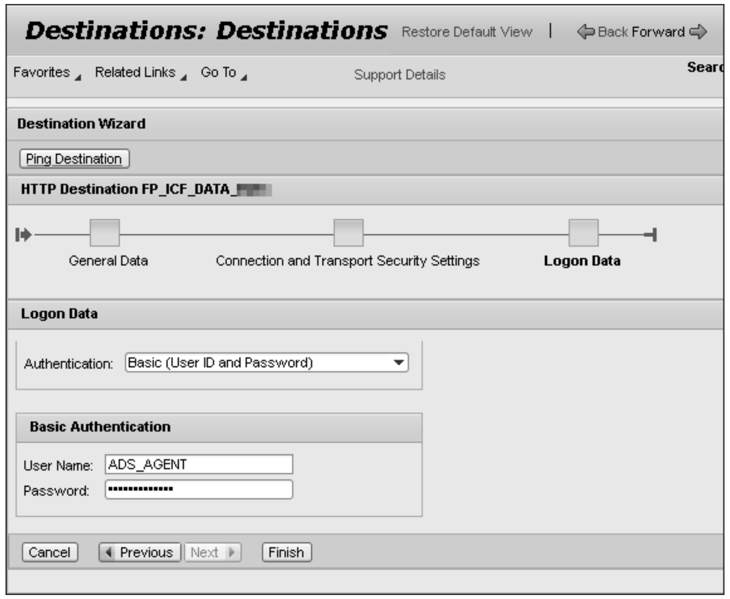
Language:

☑ Ignore SSL Server Certificates

Trusted Server Certificates Keystore View:

Figure 3.20 Connection Settings

6. Make another entry in the **System ID** field, and specify the client in which you've configured the ADS_AGENT user in the **Client** field. Click on the **Previous** button.
7. In the **Authentication** field, select **Basic (User ID and Password)**, as shown in Figure 3.21.



HTTP Destination FP_ICF_DATA_[redacted]

Authentication: Basic (User ID and Password)

User Name: ADS_AGENT

Password: ••••••••••••••

Figure 3.21 Login Information

8. In the **User Name** field, enter "ADS_AGENT" and the corresponding password, and click the **Finish** button.

Destination Test

The connection test from SAP NetWeaver Administrator generally generates an error message for this connection because only the PC name and HTTP port are stored in the URL. The path information to the respective service is only appended at runtime. To be able to carry out a test nevertheless, it's advisable to temporarily extend the stored URL. To do this, append the path */sap/bc/fp/form/layout/FP_TEST_03_TABLE.XDP* to the URL, and click **Ping Destination** without saving the changes first. If this test could be executed successfully, the message appears: **Successfully connected to HTTP destination FP_ICF_DATA_<SID> with response code 200. Content type text/html. . . .** Then, click the **Cancel** button because you've attached the path only for a test. This test simulated reading the FP_TEST_03_TABLE form from the ABAP form store.

In addition, you can test this configuration from the ABAP application server as follows:

1. Call test program FP_CHECK_DESTINATION_SERVICE in Transaction SE38 or Transaction SA38.

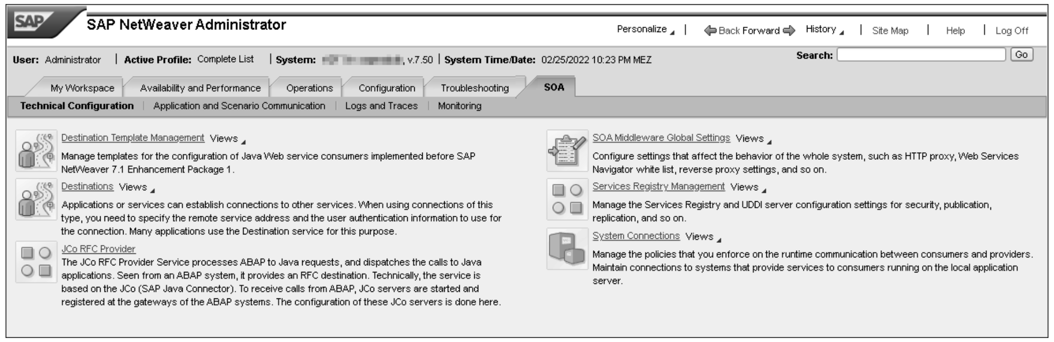
2. There, enter the name of the configured RFC connection (“ADS”), and select the **With Destination Service** checkbox.
3. Click **Run**, and ADS will be called and generate a PDF form. Here the return channel you configured before will be used. If successful, you’ll get a short informational message (i.e., **. . . bytes transferred**).

3.3.3 Configuring the Web Service Proxy

The Java PDF object accesses ADS via a web service proxy. To use applications in Web Dynpro Java, the proxy must be configured on the respective client system. The call can be either a local call to the same system or a call to a remote Java application server.

In the following, we assume that the web service proxy isn’t yet configured in your system. In any case, it’s important to know the configuration, for example, in case you need to change the connection. Follow these steps:

1. Start SAP NetWeaver Administrator again, as described earlier in Section 3.2.1.
2. Select the **SOA** tab, and click the **Destination Template Management** link (see Figure 3.22).
3. Click the **New** button to create a new entry.
4. In the following screen, select **WSIL** in the **Destination Type** field, and enter “Config-Port_Document1” in the **Destination Name** field (see Figure 3.23).
5. Complete the **URL** field in the form *http://<server>:<port>/inspection.wsil*, and then click **Next**.
6. Now you have to take care of the authentication because every user (also a technical user) has to log in to the system to access ADS. Under **Authentication,** select the **HTTP Authentication** entry with the **User ID/Password (Basic)** option, as shown in Figure 3.24.
7. In the **User ID** field, enter “ADSUser” with your chosen password.


SAP
SAP NetWeaver Administrator
Personalize | Back Forward | History | Site Map | Help | Log Off
Search: Go
My Workspace
Availability and Performance
Operations
Configuration
Troubleshooting
SOA
Technical Configuration | Application and Scenario Communication | Logs and Traces | Monitoring

User: Administrator | Active Profile: Complete List | System: [redacted], v.7.50 | System Time/Date: 02/25/2022 10:23 PM MEZ

Destination Template Management Views
Manage templates for the configuration of Java Web service consumers implemented before SAP NetWeaver 7.1 Enhancement Package 1.

Destinations Views
Applications or services can establish connections to other services. When using connections of this type, you need to specify the remote service address and the user authentication information to use for the connection. Many applications use the Destination service for this purpose.

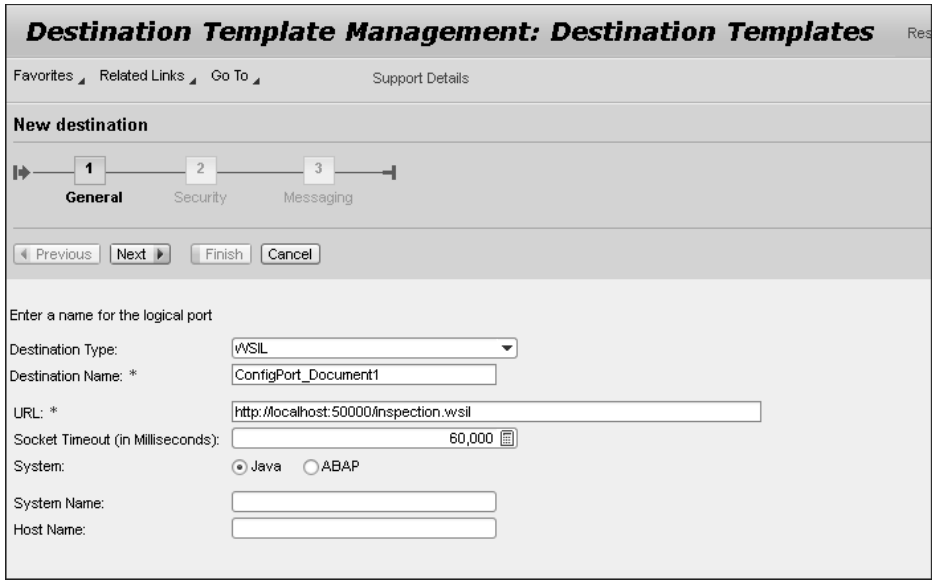
JCo RFC Provider
The JCo RFC Provider Service processes ABAP to Java requests, and dispatches the calls to Java applications. Seen from an ABAP system, it provides an RFC destination. Technically, the service is based on the JCo (SAP Java Connector). To receive calls from ABAP, JCo servers are started and registered at the gateways of the ABAP systems. The configuration of these JCo servers is done here.

SOA Middleware Global Settings Views
Configure settings that affect the behavior of the whole system, such as HTTP proxy, Web Services Navigator white list, reverse proxy settings, and so on.

Services Registry Management Views
Manage the Services Registry and UDDI server configuration settings for security, publication, replication, and so on.

System Connections Views
Manage the policies that you enforce on the runtime communication between consumers and providers. Maintain connections to systems that provide services to consumers running on the local application server.

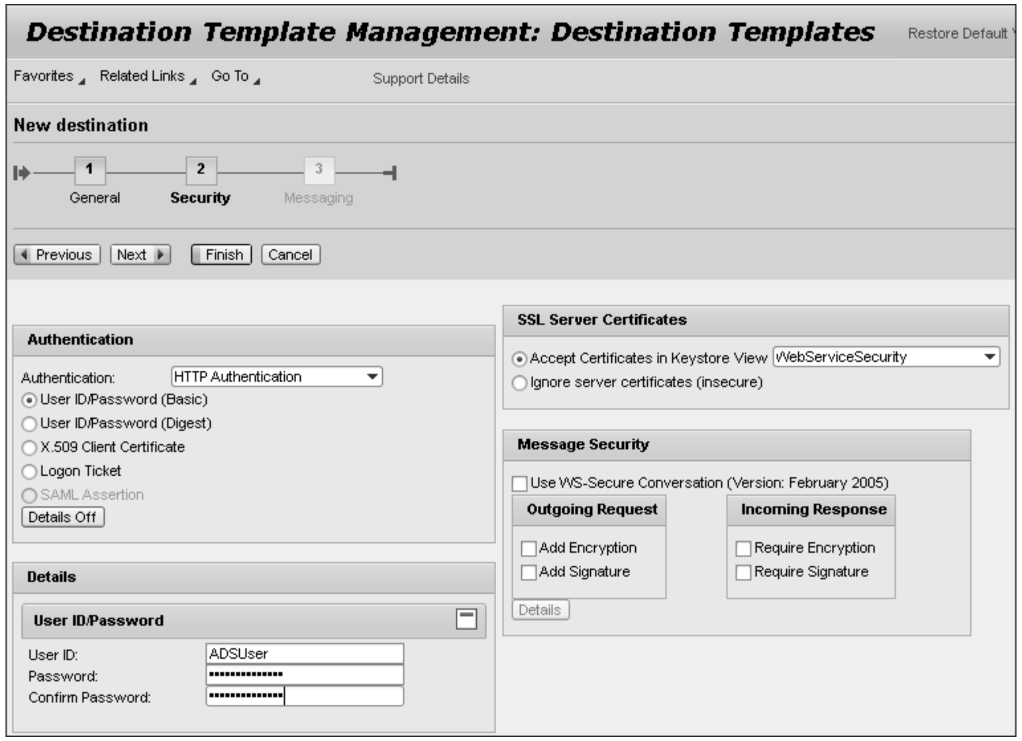
Figure 3.22 SOA Tab


Destination Template Management: Destination Templates
Favorites | Related Links | Go To | Support Details
New destination
1 General
2 Security
3 Messaging
Previous | Next | Finish | Cancel

Enter a name for the logical port

Destination Type: WSIL
Destination Name: * ConfigPort_Document1
URL: * http://localhost:50000/inspection.wsil
Socket Timeout (in Milliseconds): 60,000
System: ● Java ○ ABAP
System Name:
Host Name:

Figure 3.23 Create a Web Service Proxy


Destination Template Management: Destination Templates
Restore Default
Favorites | Related Links | Go To | Support Details
New destination
1 General
2 Security
3 Messaging
Previous | Next | Finish | Cancel
Details Off
Details

Authentication

Authentication: HTTP Authentication
● User ID/Password (Basic)
○ User ID/Password (Digest)
○ X.509 Client Certificate
○ Logon Ticket
○ SAML Assertion

Details

User ID/Password
User ID: ADSUser
Password: ••••••••••••••
Confirm Password: ••••••••••••••

SSL Server Certificates
● Accept Certificates in Keystore View WebServiceSecurity
○ Ignore server certificates (insecure)

Message Security
□ Use WS-Secure Conversation (Version: February 2005)

Outgoing Request
□ Add Encryption
□ Add Signature

Incoming Response
□ Require Encryption
□ Require Signature

Figure 3.24 Entering Credentials for the Web Service Proxy

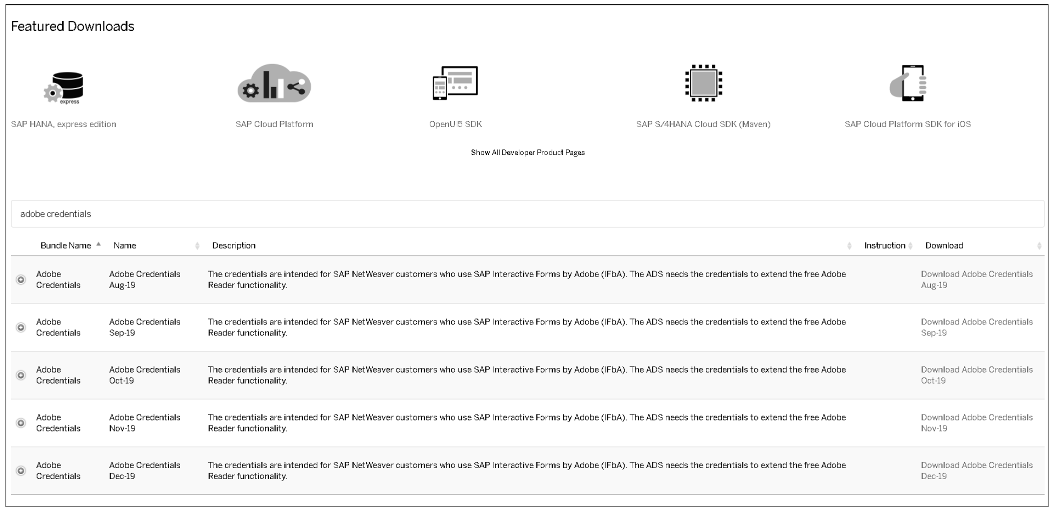
8. Click the **Finish** button to complete the configuration and save the changes. Then click the **Back** button to return to the SAP NetWeaver Administrator initial screen.

To test the web service proxy, select **Operations • Systems • Start & Stop • Java Applications**, and start the `tc~wd~pdfobject` application. This configuration also allows the Web Dynpro Java runtime to access ADS.

3.3.4 Configuring the ReaderRights Credential

To create interactive forms, you need a *ReaderRights credential*, which is a right to use SAP Interactive Forms by Adobe. To obtain a ReaderRights credential for production use, see SAP Note 736902. You can also download a ReaderRights credential valid for a limited time from the SAP download page for developer and trial versions of the SAP software at *http://s-prs.co/v224207*. To configure the credential, follow these steps:

1. Go to the website just mentioned, and then search for the keyword “credentials”. You may need to log in to this page with your S-User. A list of available credentials for SAP Interactive Forms by Adobe is displayed (see Figure 3.25).
2. Click the link in the **Name** column for the credential that is relevant to you.
3. Confirm the license agreements, and start the download via the **Submit** button.
4. In the following step, you can check again if the correct S-User is specified. Then click on **Click here to start your trial!**


Featured Downloads
SAP HANA, express edition
express
SAP Cloud Platform
OpenUI5 SDK
SAP S/4HANA Cloud SDK (Maven)
SAP Cloud Platform SDK for iOS
Show All Developer Product Pages

adobe credentials

Bundle Name	Name	Description	Instruction	Download
Adobe Credentials	Adobe Credentials Aug-19	The credentials are intended for SAP NetWeaver customers who use SAP Interactive Forms by Adobe (IFbA). The ADS needs the credentials to extend the free Adobe Reader functionality.		Download Adobe Credentials Aug-19
Adobe Credentials	Adobe Credentials Sep-19	The credentials are intended for SAP NetWeaver customers who use SAP Interactive Forms by Adobe (IFbA). The ADS needs the credentials to extend the free Adobe Reader functionality.		Download Adobe Credentials Sep-19
Adobe Credentials	Adobe Credentials Oct-19	The credentials are intended for SAP NetWeaver customers who use SAP Interactive Forms by Adobe (IFbA). The ADS needs the credentials to extend the free Adobe Reader functionality.		Download Adobe Credentials Oct-19
Adobe Credentials	Adobe Credentials Nov-19	The credentials are intended for SAP NetWeaver customers who use SAP Interactive Forms by Adobe (IFbA). The ADS needs the credentials to extend the free Adobe Reader functionality.		Download Adobe Credentials Nov-19
Adobe Credentials	Adobe Credentials Dec-19	The credentials are intended for SAP NetWeaver customers who use SAP Interactive Forms by Adobe (IFbA). The ADS needs the credentials to extend the free Adobe Reader functionality.		Download Adobe Credentials Dec-19

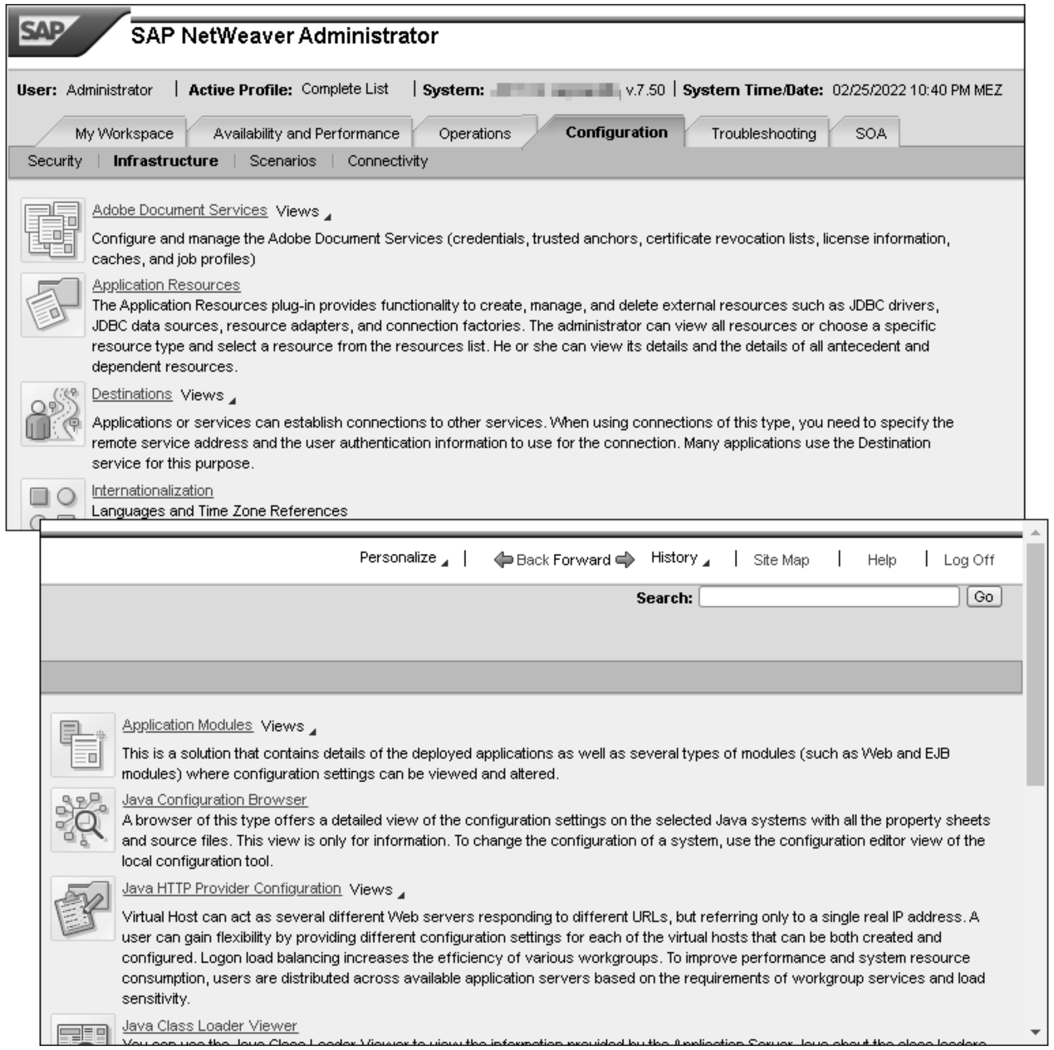
Figure 3.25 Available Credentials for SAP Interactive Forms by Adobe

The download consists of a ZIP archive file (*.zip*) that contains a password-protected private key (credential). The format of the credential file conforms to the Public Key Cryptography Standards (PKCS#12) specification. The file ends in *.p12* or *.pfx*. In addition to the key, there is a text file (*.txt*) in the ZIP archive that stores the associated password.

To create interactive forms with usage rights, you must configure the ReaderRights credential. *Usage rights* are used to enable advanced functionality in Adobe Acrobat Reader. You configure the key within SAP NetWeaver Administrator. In addition to the private keys (credentials) for ADS, you can also manage *certificates* and *certificate revocation lists* (CRLs) there.

First, open SAP NetWeaver Administrator to configure the ReaderRights credential as follows:

1. Select the **Configuration** tab, and then click **Infrastructure** (see Figure 3.26). There, click the **Adobe Document Services** link.



User: Administrator | Active Profile: Complete List | System: [redacted] v.7.50 | System Time/Date: 02/25/2022 10:40 PM MEZ

Adobe Document Services Views
Configure and manage the Adobe Document Services (credentials, trusted anchors, certificate revocation lists, license information, caches, and job profiles)

Application Resources
The Application Resources plug-in provides functionality to create, manage, and delete external resources such as JDBC drivers, JDBC data sources, resource adapters, and connection factories. The administrator can view all resources or choose a specific resource type and select a resource from the resources list. He or she can view its details and the details of all antecedent and dependent resources.

Destinations Views
Applications or services can establish connections to other services. When using connections of this type, you need to specify the remote service address and the user authentication information to use for the connection. Many applications use the Destination service for this purpose.

Internationalization
Languages and Time Zone References

Application Modules Views
This is a solution that contains details of the deployed applications as well as several types of modules (such as Web and EJB modules) where configuration settings can be viewed and altered.

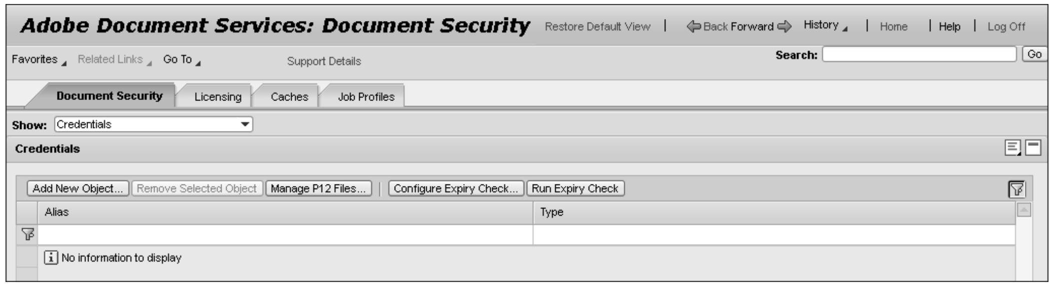
Java Configuration Browser
A browser of this type offers a detailed view of the configuration settings on the selected Java systems with all the property sheets and source files. This view is only for information. To change the configuration of a system, use the configuration editor view of the local configuration tool.

Java HTTP Provider Configuration Views
Virtual Host can act as several different Web servers responding to different URLs, but referring only to a single real IP address. A user can gain flexibility by providing different configuration settings for each of the virtual hosts that can be both created and configured. Logon load balancing increases the efficiency of various workgroups. To improve performance and system resource consumption, users are distributed across available application servers based on the requirements of workgroup services and load sensitivity.

Java Class Loader Viewer

Figure 3.26 Configuration Tab

2. Next, you'll see the **Document Security** tab. Here you manage credentials, certificates, and revoked certificates, or you can monitor the license status. The application also allows you to get started managing job profiles (see Figure 3.27). Select **Credentials** in the **Show** field, and click the **Manage P12 Files** button.


Adobe Document Services: Document Security
Restore Default View | Back Forward | History | Home | Help | Log Off
Favorites | Related Links | Go To | Support Details
Search: | Go
Document Security | Licensing | Caches | Job Profiles
Add New Object... | Remove Selected Object | Manage P12 Files... | Configure Expiry Check... | Run Expiry Check
Alias | Type

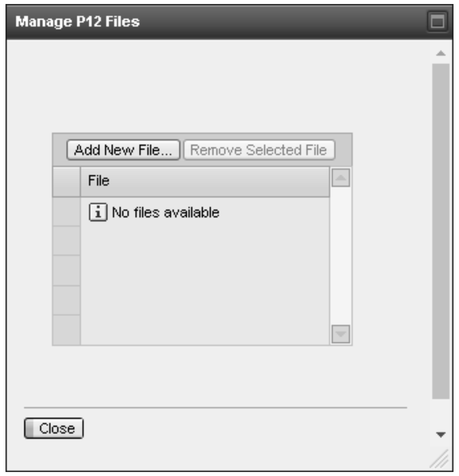
Show: Credentials

Credentials

No information to display

Figure 3.27 Access to Credentials Management

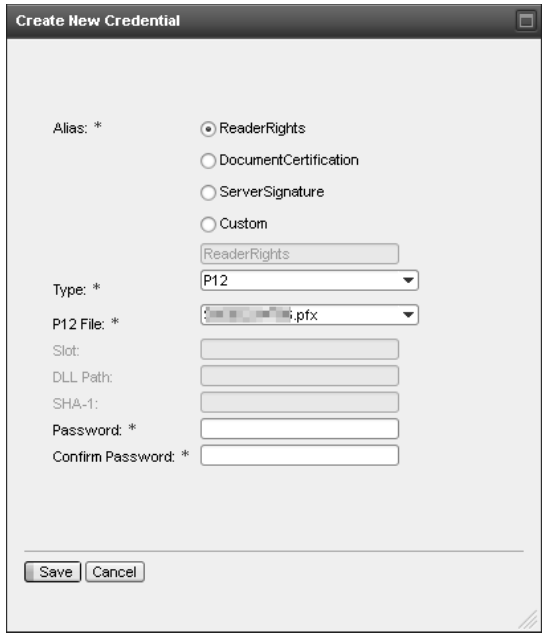
3. Use the following popup window to load the credential into your system by clicking the **Add New File** button (see Figure 3.28).
4. In the dialog that opens, use the **Browse** button to select the PFX file, and then click **Upload**. Close the dialog from Figure 3.28 via the **Close** button.



No files available

Figure 3.28 Credentials Management

5. Now you can configure the credential by clicking the **Add New Object** button from Figure 3.27.
6. The dialog shown in Figure 3.29 opens. Select the **ReaderRights** option, and set the type in the **Type** field to **P12**.
7. In the **P12 File** dropdown, select your credential.



Alias: * ● ReaderRights ○ DocumentCertification ○ ServerSignature ○ Custom

ReaderRights

Type: * P12

P12 File: * [redacted].pfx

Slot:

DLL Path:

SHA-1:

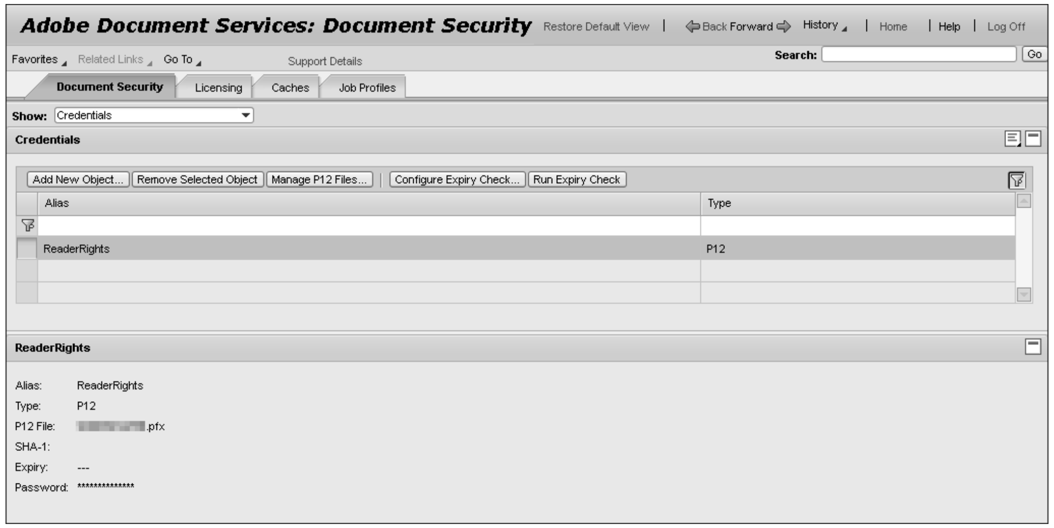
Password: *

Confirm Password: *

Figure 3.29 Selecting the ReaderRights Credential

8. Under **Password** and **Confirm Password**, enter the password you received as part of the credential (text file in the ZIP file).
9. Close the dialog via the **Save** button.

Your ADS web services are now configured, and you can use the ReaderRights credential. Note the validity expiration (**Expiry**) if this is a test credential (see Figure 3.30).


Adobe Document Services: Document Security
Restore Default View | Back Forward | History | Home | Help | Log Off
Favorites | Related Links | Go To | Support Details
Search: | Go
Document Security | Licensing | Caches | Job Profiles
Add New Object... | Remove Selected Object | Manage P12 Files... | Configure Expiry Check... | Run Expiry Check

Show: Credentials

Credentials

Alias	Type
ReaderRights	P12

ReaderRights

Alias: ReaderRights
Type: P12
P12 File: [redacted].pfx
SHA-1:
Expiry: ---
Password: *************

Figure 3.30 Ready Configured Credential: ReaderRights

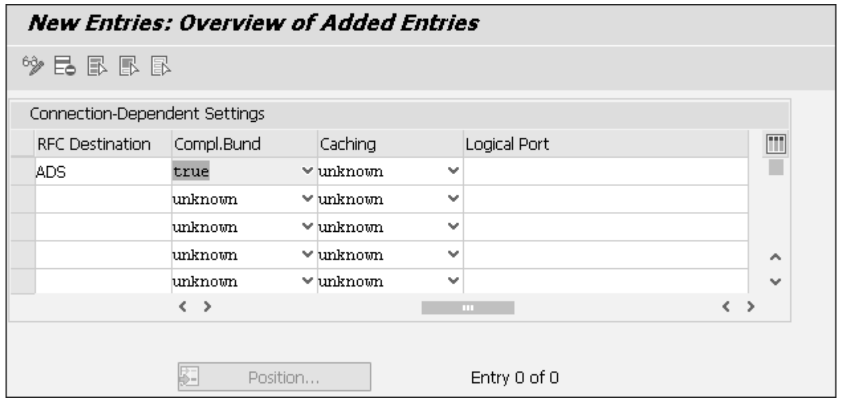
To complete the configuration, you must restart the *Document Service Trust Manager* service on the Java application server; otherwise, the changes won't take effect. To do this, select the **Operations** tab in SAP NetWeaver Administrator. Under it, select the **Systems** area and then the **Start & Stop** link. Then select **Java Services**, enter "adobe" under **Service Component Name**, and press `Enter`. In the list that appears, you can now select the service and restart it.

You can check the correct installation of the ReaderRights credential using program FP_TEST_03. Call this test program in Transaction SE38, and select the option **X (Interactive Form with Additional Usage Rights)** for the parameter **Form** - **Interactive**. If you then display the print preview of the form, Adobe Acrobat Reader DC reports that it's a fillable form.

3.3.5 Additional Settings for the Remote Function Call Connection

In table `FPCONNECT`, you can make additional settings for the RFC connection you've configured (Section 3.3.1), for example, to optimize performance. To do this, call Transaction SM30 (Table View Maintenance). Enter "FPCONNECT" in the **Table/View** field, and click the **Maintain** button. You can skip the warning message **Attention, the table is client-independent!** that appears.

To add settings for an RFC connection, click the **New Entries** button. Then enter the name of the connection in the **RFC Destination** column (e.g., "ADS", as shown in Figure 3.31). Under certain circumstances, the RFC connection will also already be displayed to you, such as when you've already called up a form with the same connection. The system adjusts table `FPCONNECT` each time a form is called. In this case, you don't need to create a new entry.


New Entries: Overview of Added Entries

Connection-Dependent Settings

RFC Destination	Compl.Bund	Caching	Logical Port
ADS	true	unknown	
	unknown	unknown	
	unknown	unknown	
	unknown	unknown	
	unknown	unknown	

Entry 0 of 0

Figure 3.31 Maintain Table FPCONNECT

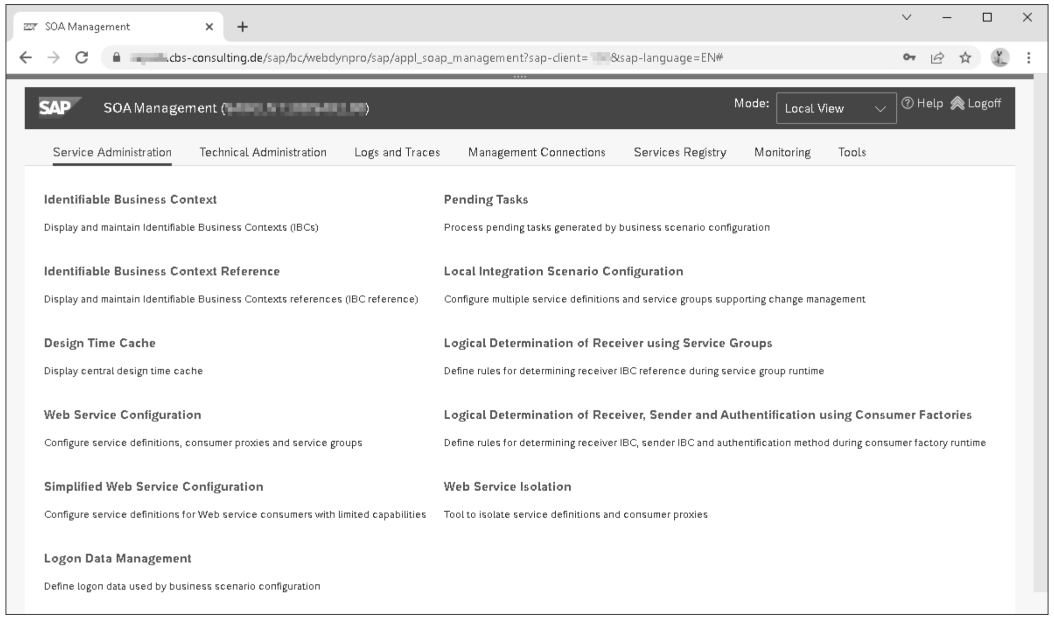
The **Compl.Bund** column enables system-wide control of form bundling. To switch this on, select the value **true.** This significantly reduces the runtime of print jobs with many small- or medium-sized forms, provided the print program outputs these forms in one print job (see Chapter 6, Section 6.7.1). This is a system-wide setting. Alternatively, ABAP print programs can control form bundling via a parameter.

You can use the **Caching** column to specify whether the ADS form template should be cached in the *destination cache*. This improves performance because the ADS web services don't have to request the template from the ABAP server each time a form is generated. The system automatically recognizes changes to the form template. In this case, the updated template is requested and then stored in the destination cache. Note that caching is turned on by default, so you must set it to **false** if you want to turn it off.

To save your changes, click the **Save** button. For the other columns in table `FPCONNECT`, it's recommended to use default values.

3.4 Parallelization Configuration

For the parallelization of print jobs (see Chapter 6, Section 6.8), two web services must be configured. One service (`FpPrintRequestService`) is responsible for the actual parallelization, while another service (`FpQueueAdminService`) is required for administration and monitoring. The configuration of the web services is done via SOA Manager. To do this, start Transaction SOAMANAGER , which you can also start directly in the browser using the following link structure: *http://<server>:<port>/sap/bc/webdynpro/sap/appl_soap_management?saplanguage=EN*. Select the **Service Administration** tab, and then click the **Web Service Configuration** link (see Figure 3.32).



Identifiable Business Context
Display and maintain Identifiable Business Contexts (IBCs)

Identifiable Business Context Reference
Display and maintain Identifiable Business Contexts references (IBC reference)

Design Time Cache
Display central design time cache

Web Service Configuration
Configure service definitions, consumer proxies and service groups

Simplified Web Service Configuration
Configure service definitions for Web service consumers with limited capabilities

Logon Data Management
Define logon data used by business scenario configuration

Pending Tasks
Process pending tasks generated by business scenario configuration

Local Integration Scenario Configuration
Configure multiple service definitions and service groups supporting change management

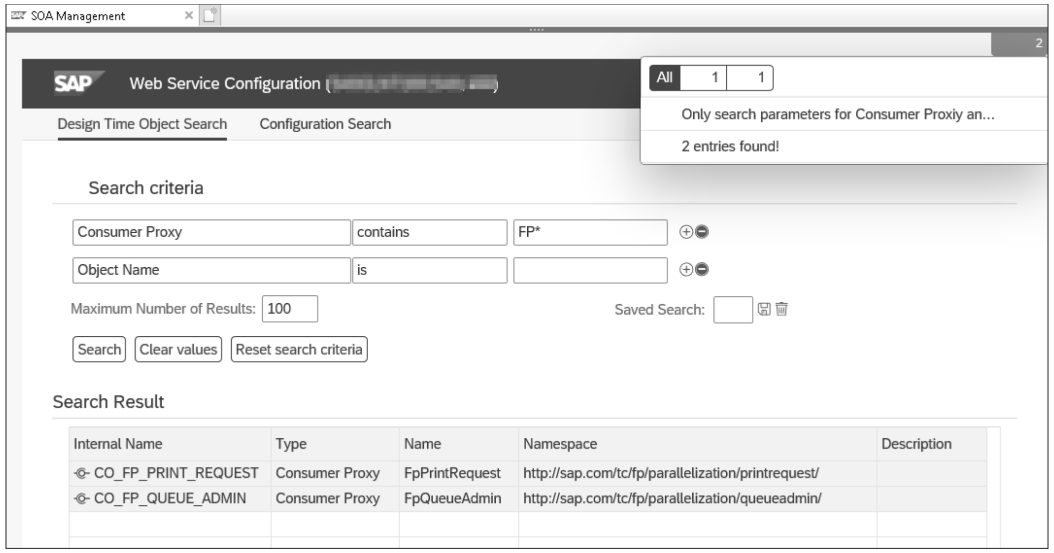
Logical Determination of Receiver using Service Groups
Define rules for determining receiver IBC reference during service group runtime

Logical Determination of Receiver, Sender and Authentification using Consumer Factories
Define rules for determining receiver IBC, sender IBC and authentification method during consumer factory runtime

Web Service Isolation
Tool to isolate service definitions and consumer proxies

Figure 3.32 Transaction SOAMANAGER: Service Administration Tab

In the following dialog, set **Consumer Proxy** in the **Search criteria** area, and enter “FP*” as the search expression. Click the **Search** button to display the search results at the bottom of the screen (see Figure 3.33).


SOA Management

SAP Web Service Configuration ([redacted])

All | 1 | 1

Only search parameters for Consumer Proxy an...

2 entries found!

Design Time Object Search | Configuration Search

Search criteria

Consumer Proxy | contains | FP*

Object Name | is |

Maximum Number of Results: 100

Saved Search:

Search | Clear values | Reset search criteria

Search Result

Internal Name	Type	Name	Namespace	Description
CO_FP_PRINT_REQUEST	Consumer Proxy	FpPrintRequest	http://sap.com/tc/fp/parallelization/printrequest/	
CO_FP_QUEUE_ADMIN	Consumer Proxy	FpQueueAdmin	http://sap.com/tc/fp/parallelization/queueadmin/	

Figure 3.33 List of Relevant Consumer Proxies

This section will dive further into configuring web services, including testing the connection between web services and using RFCs to connect to ADS.

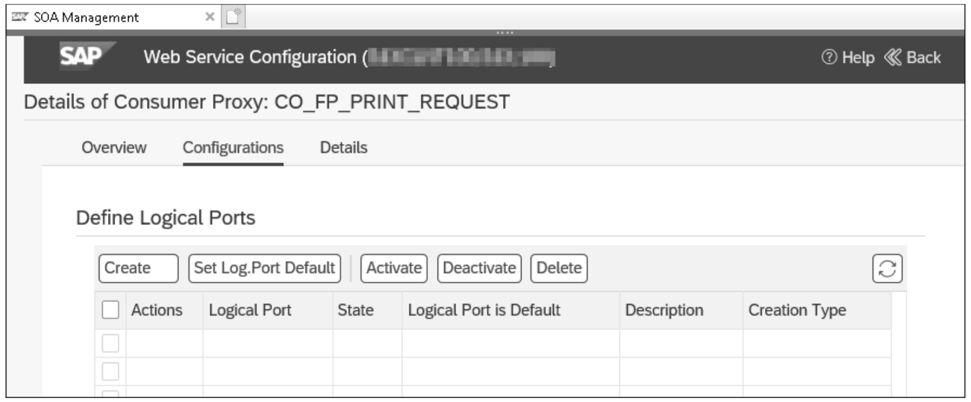
Client Dependency of the Web Service Connection

Note that the web service connections are *client dependent*. You must configure the connection from each client from which parallelization is to be used. As a reminder, the direct connection to ADS via Transaction SM59 was *client independent*.

3.4.1 Configure Web Services

We’ll first configure the `CO_FP_PRINT_REQUEST` web service, which is responsible for the actual parallelization, with the following steps:

1. Select the `CO_FP_PRINT_REQUEST` entry from Figure 3.33, shown earlier. The logical ports configuration appears (see Figure 3.34). Click the **Create** button, and select **WSDL-Based Configuration** from the drop-down list.


SOA Management

SAP Web Service Configuration ([redacted])

Help | Back

Details of Consumer Proxy: CO_FP_PRINT_REQUEST

Overview | Configurations | Details

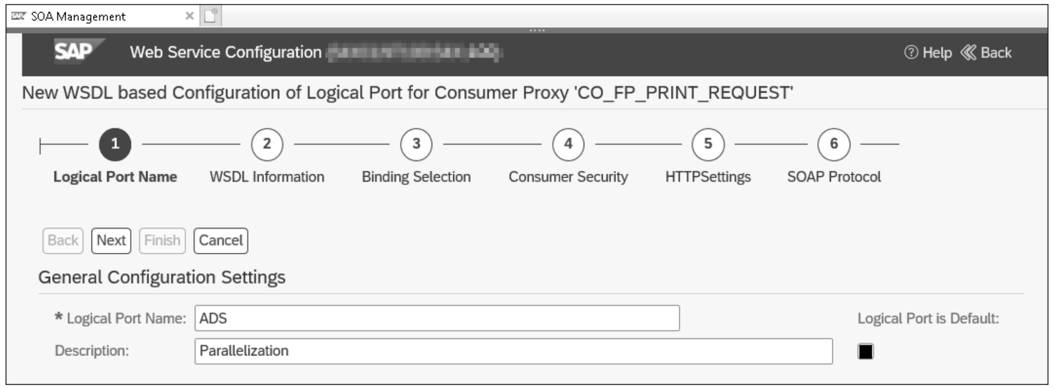
Define Logical Ports

Create | Set Log.Port Default | Activate | Deactivate | Delete

☐ Actions	Logical Port	State	Logical Port is Default	Description	Creation Type
☐					
☐					

Figure 3.34 Logical Ports Configuration

2. In step **1**, you can freely assign a name in the **Logical Port Name** field, for example, “ADS”. Also enter a **Description**, and check the **Logical Port is Default** checkbox (see Figure 3.35). Then click the **Next** button.


SOA Management

SAP Web Service Configuration [redacted]

Help | Back

New WSDL based Configuration of Logical Port for Consumer Proxy 'CO_FP_PRINT_REQUEST'

1 Logical Port Name — 2 WSDL Information — 3 Binding Selection — 4 Consumer Security — 5 HTTPSettings — 6 SOAP Protocol

Back | Next | Finish | Cancel

General Configuration Settings

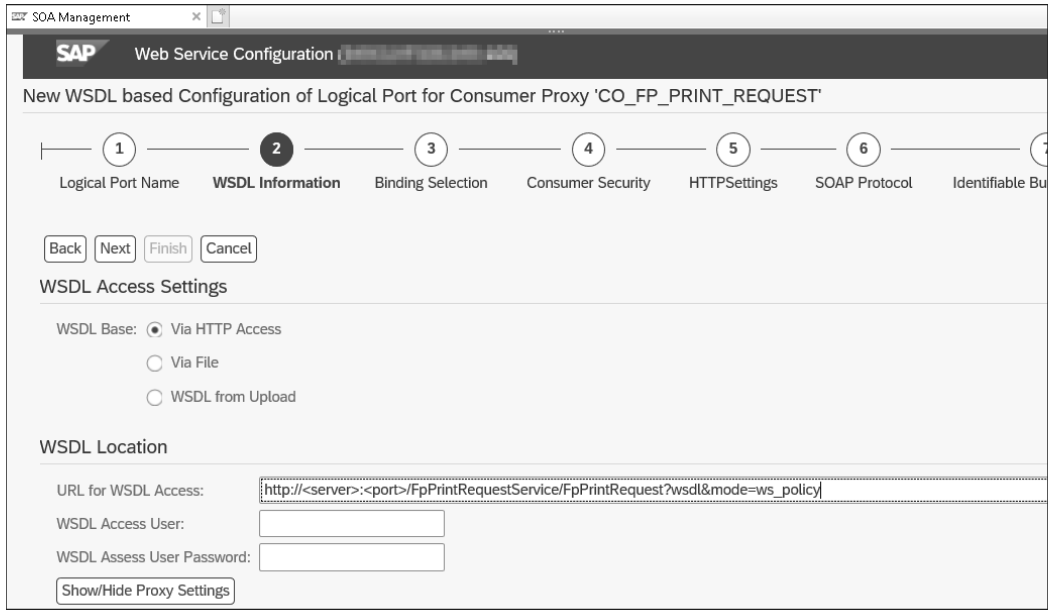
* Logical Port Name: ADS

Description: Parallelization

Logical Port is Default: ■

Figure 3.35 General Settings of the Logical Port

3. In step **2**, leave the **WSDL Access Settings** at **Via HTTP Access** (see Figure 3.36). In the **WSDL Location** section, specify “http://<server>:<port>/FpPrintRequestService/FpPrintRequest?wsdl&mode=ws_policy” in the **URL for WSDL access** field. Here, you must replace <server> and <port> with the parameters of your Java stack. You can also find these parameters in the RFC destination to ADS (Section 3.3.1). Continue with **Next**.
4. This takes you to step **3**, **Binding Selection** (see Figure 3.37). Click the **Next** button again here.



Web Service Configuration [redacted]

New WSDL based Configuration of Logical Port for Consumer Proxy 'CO_FP_PRINT_REQUEST'

WSDL Access Settings

WSDL Base: ● Via HTTP Access
○ Via File
○ WSDL from Upload

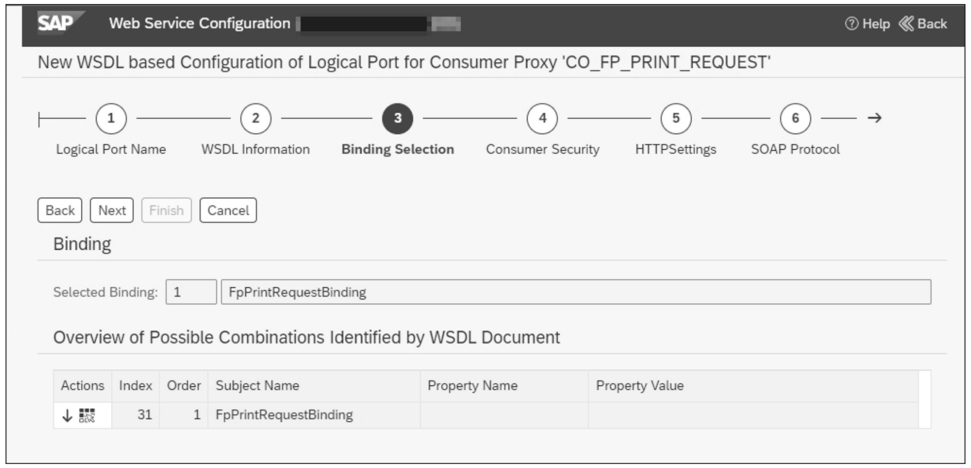
WSDL Location

URL for WSDL Access: http://<server>:<port>/FpPrintRequestService/FpPrintRequest?wsdl&mode=ws_policy

WSDL Access User:

WSDL Assess User Password:

Figure 3.36 Logical Port Settings (Parallelization)



Web Service Configuration [redacted]

New WSDL based Configuration of Logical Port for Consumer Proxy 'CO_FP_PRINT_REQUEST'

Binding

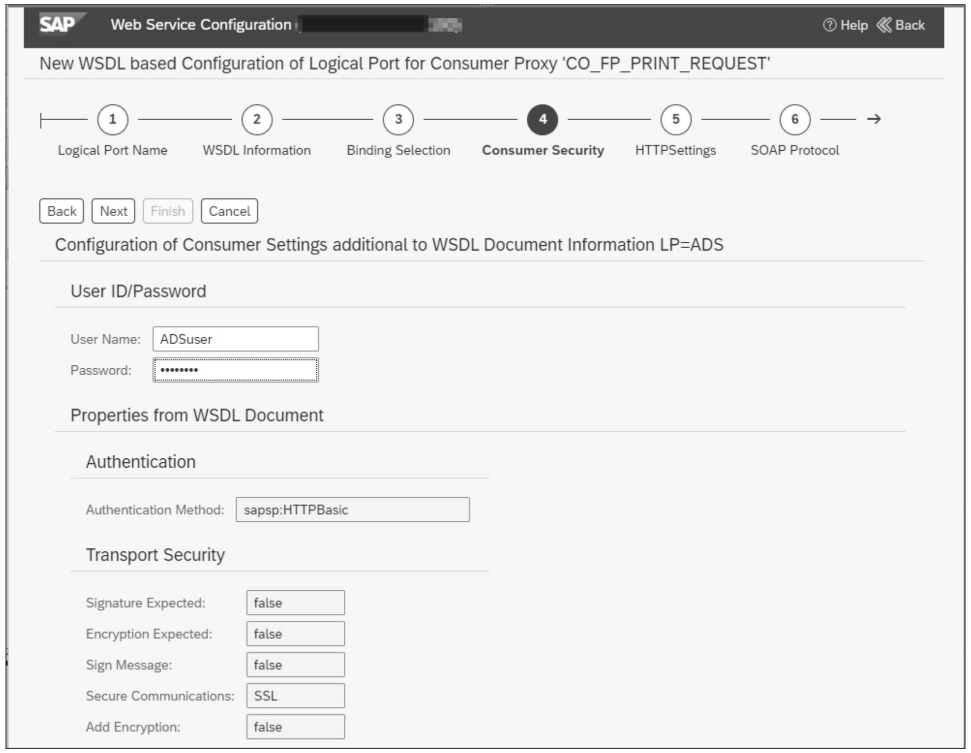
Selected Binding: 1 FpPrintRequestBinding

Overview of Possible Combinations Identified by WSDL Document

Actions	Index	Order	Subject Name	Property Name	Property Value
↓	31	1	FpPrintRequestBinding		

Figure 3.37 Binding Selection for the WSDL Document

5. In step **4**, you make the security settings (see Figure 3.38). In our case, the `sapsp:HTTPBasic` authentication method is used by default, where securing via user and password is sufficient. Enter the **User Name** for ADS ("ADSUser") and the set password, and click the **Next** button.



Web Service Configuration [redacted]

New WSDL based Configuration of Logical Port for Consumer Proxy 'CO_FP_PRINT_REQUEST'

Configuration of Consumer Settings additional to WSDL Document Information LP=ADS

User ID/Password

User Name: ADSuser

Password: ••••••••

Properties from WSDL Document

Authentication

Authentication Method: sapsp:HTTPBasic

Transport Security

Signature Expected: false

Encryption Expected: false

Sign Message: false

Secure Communications: SSL

Add Encryption: false

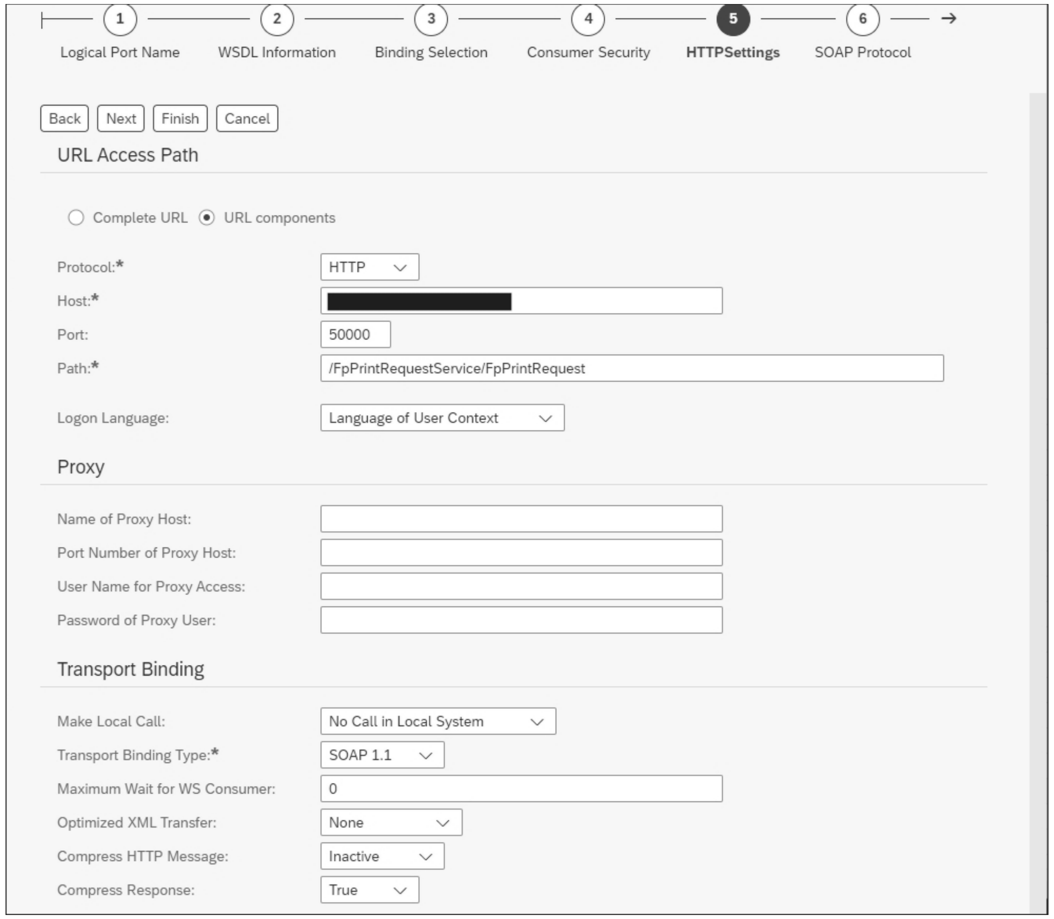
Figure 3.38 Logical Port Security Settings

6. In step **5**, **HTTPSettings**, finish the configuration using the **Done** button (see Figure 3.39).

You're returned to the logical port **Configurations** view, where you can see the **ADS** logical port you just created (see Figure 3.40).

Now that you've configured the web service for parallelization, let's turn to the second web service from Figure 3.33, shown earlier. The `FpQueueAdminService` service is needed to monitor and manage the jobs. Select the **CO_FP_QUEUE_ADMIN** entry in the SOA Manager and perform the same steps for this web service as described in the previous section. It's important that you give the logical port the same name, in our case, "ADS" again.

The crucial difference is in the **URL for WSDL Access** input field in the **WSDL Location** section. The URL for access is for the monitoring service *http://<server>:<port>/FpQueueAdminService/FpQueueAdmin?wsdl&mode=ws_policy*. Maintain the security settings as shown earlier in Figure 3.38, and save your entries. You'll return to the **Web Service Configuration** dialog. There you can now see the two logical ports for parallelization and monitoring.



URL Access Path

○ Complete URL ● URL components

Protocol:* HTTP

Host:* [redacted]

Port: 50000

Path:* /FpPrintRequestService/FpPrintRequest

Logon Language: Language of User Context

Proxy

Name of Proxy Host:

Port Number of Proxy Host:

User Name for Proxy Access:

Password of Proxy User:

Transport Binding

Make Local Call: No Call in Local System

Transport Binding Type:* SOAP 1.1

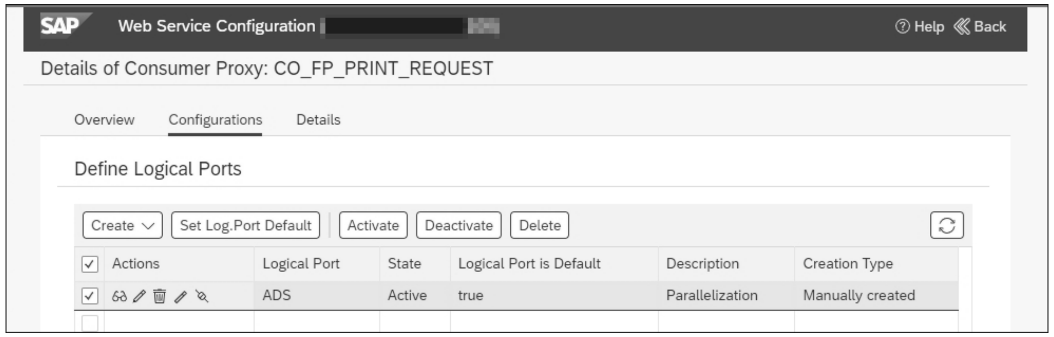
Maximum Wait for WS Consumer: 0

Optimized XML Transfer: None

Compress HTTP Message: Inactive

Compress Response: True

Figure 3.39 HTTP Logical Port Settings



Web Service Configuration [redacted]

Details of Consumer Proxy: CO_FP_PRINT_REQUEST

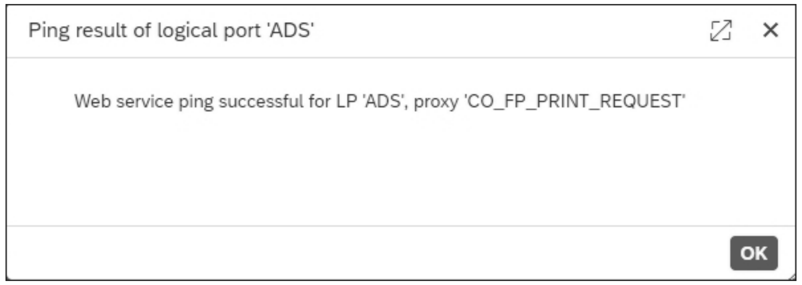
Define Logical Ports

☑	Actions	Logical Port	State	Logical Port is Default	Description	Creation Type
☑		ADS	Active	true	Parallelization	Manually created
□						

Figure 3.40 Logical Port of Parallelization

3.4.2 Testing the Connection between Two Web Services

You can use a simple connection test to check whether the web services for parallelization can be addressed from the ABAP stack. To do this, select a consumer proxy from Figure 3.33, shown earlier. You return to the configuration overview. Here, select the logical port (**ADS**) you just created, and click the **Ping to Web Service** button . This automatically checks whether the technical connection settings are correct, the services are available on the Java stack, and whether you've entered the security settings (e.g., user and password) correctly. A success message appears, as shown in Figure 3.41.



Ping result of logical port 'ADS'

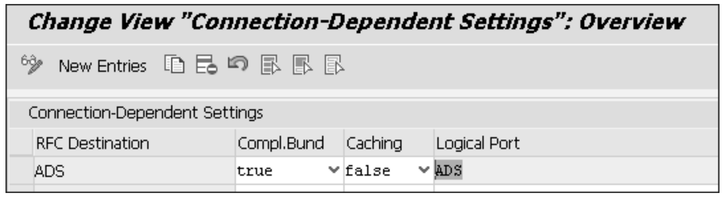
Web service ping successful for LP 'ADS', proxy 'CO_FP_PRINT_REQUEST'

Figure 3.41 Successfully Tested Configuration

3.4.3 Assigning the Logical Port to the Remote Function Call Connection ADS

The connection to ADS is made in the ABAP program via an RFC connection, which usually has the name `ADS`. If you choose a different name, you must specify it explicitly in each application program. To use parallelization, you must define a mapping between this RFC connection and the logical ports created. You define this mapping in table `FPCONNECT`, via which you also make the settings for bundling (Section 3.3.5).

Call this up via the table maintenance in Transaction SM30, click on **Maintain**, and confirm the information regarding client independence with Enter. Select the connection entry **ADS** in the table, and scroll to the right until you see the **Logical Port** field. Enter here the name of the logical port you selected in Section 3.4.1 (see Figure 3.42).



Change View "Connection-Dependent Settings": Overview

Connection-Dependent Settings

RFC Destination	Compl.Bund	Caching	Logical Port
ADS	true	false	ADS

Figure 3.42 Assign Logical Port

Parallelization is basically possible with and without bundling the forms. However, by bundling, you also reduce the number of calls to ADS, as in the case of sequential processing.

3.5 Optional Configuration

In this section, we'll cover other possible ADS settings, including licenses, document sercurity, service characteristics, parallel print jobs, fonts, and XML Forms Architecture (XFA) Device Configuration (XDC) files.

3.5.1 Licensing

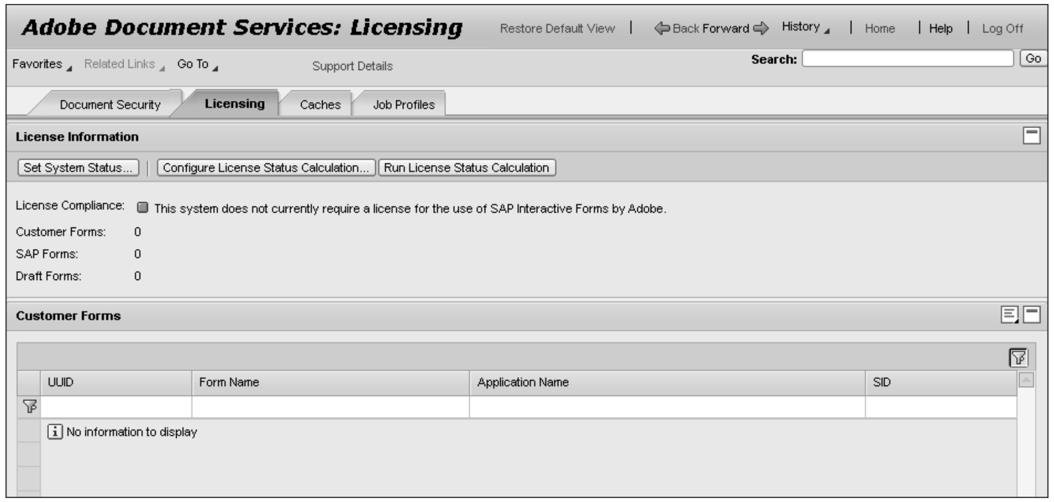
An additional license is required to create custom interactive forms (see SAP Note 750784). ADS has a component that monitors license-required usage and allows you to change the license status. To view the corresponding usage data as well as make changes to the status, start SAP NetWeaver Administrator and the Adobe Document Services app, as described in the ReaderRights Credentials configuration (Section 3.3.4). You'll return to the screen shown earlier in Figure 3.27, and click on the **Licensing** tab there.

In Figure 3.43 you can see the number of forms processed on this system, divided into the following three categories:

- **Number of SAP forms**
  This number indicates how many forms delivered by SAP (not modified or only minimally modified) have been processed. No additional license is required for these forms.
- **Number of customer forms**
  Customer forms are self-developed interactive forms or SAP forms that have been fundamentally changed. A fundamental change is considered to be, among other things, an adjustment of the data binding, which becomes necessary, for example, when adding new data fields.
- **Number of draft forms**
  Forms that are generated from nonproduction systems or clients aren't taken into account when determining the license status.

As soon as customer forms requiring a license have been processed, the status (**License Fulfillment**) changes from green to red. If you've purchased a license, you can configure this in the system by clicking the **Set System Status** button and checking the appropriate box (**This system is licensed to use SAP Interactive Forms by Adobe**) in the dialog that appears. This will change the status from red to green again. In addition to displaying the license status in the administration interface, ADS writes a log entry if a license is required but the status hasn't been configured.

The **Customer Forms** table at the bottom of the user interface (UI) displays all customer forms that require a license, as well as additional information such as the SAP system ID and client from which ADS was called, and the name of the form and calling application.



License Compliance: ▪ This system does not currently require a license for the use of SAP Interactive Forms by Adobe.

Customer Forms: 0

SAP Forms: 0

Draft Forms: 0

UUID	Form Name	Application Name	SID

No information to display

Figure 3.43 Licensing Tab

3.5.2 Document Security: Credentials, Trusted Anchors, and Certificate Revocation Lists

ADS can also be used to generate and verify digital signatures. Here, a distinction must first be made between certification and signature:

- **Certification**
  When a PDF document is certified, it's provided with an invisible signature that ensures the integrity of the form; that is, as soon as the PDF document has been tampered with, the certification is recognized as invalid—both in Adobe Reader and by ADS.
- **Signature**
  In addition, any number of signatures can be added to a document, which can be set in business processes, for example, in an approval step.

Private keys (credentials), trusted certificates (trusted anchors), and CRLs are required to use digital signatures, as discussed in the following sections.

Credentials

The PDF object (ABAP and Java) provides various methods both for certification and for applying signatures. Among other things, these objects allow the key to be specified by means of a configured alias. However, if these optional parameters aren't supplied, ADS uses the following default values:

- `DocumentCertification` as alias for certification
- `ServerSignature` as alias for the digital signature

The procedure for installing and configuring these and other keys is identical to that for the ReaderRights credential (Section 3.3.4). Only the correct spelling of the alias and the entry of a valid password must be observed. As already indicated in the description of the ReaderRights credential, the Document Services Trust Manager service must also be restarted for additional keys.

Under **Credentials,** all registered keys are displayed in a table. ADS checks their expiration dates daily and writes log entries as soon as the current date approaches or exceeds the expiration date. Consequently, there is no need for the system administrator to regularly check the expiration dates of the keys via this UI; instead, monitoring the logs is sufficient.

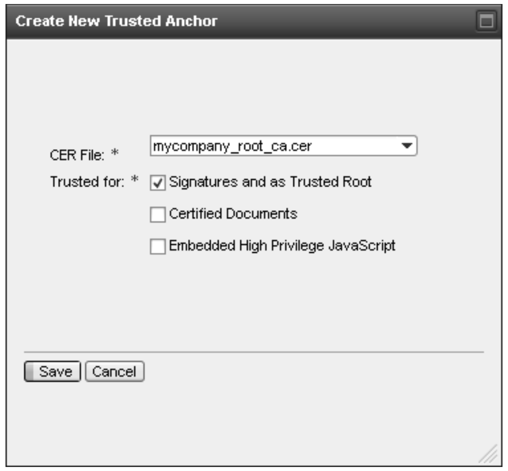
In the **Configure Sequence Check** area, settings can be made regarding the log entries. For example, the time of the daily executed verification process can be adjusted. It's also possible to specify how many days before a key expires that warnings should be written to the log file.

Trusted Anchors

ADS enables you to verify signatures in PDF forms. This requires the configuration of trusted certificates, which are either the public key of a private key used for the signature or the public key of the issuing authority—a *root certificate*. These public keys are managed via the **Trusted Anchors** field.

You define a list of trusted certificates in your system as follows:

1. In the **Adobe Document Services** screen, on the **Document Security** tab (refer to Figure 3.27), select **Trusted Anchors** instead of **Credentials** in the **Show** dropdown**.** You can use the **Manage CER Files** button to load the *.cer* file for the trusted anchors into the system in the same way as for the ReaderRights credential.
2. After that, you can select this file using the **Add New Object** button.
3. Use the indicators next to **Trusted For** to determine what the certificate should be used for (see Figure 3.44):
   - **Signatures and as a Trusted Entry Point**
     Use this option to control whether the certificate should be used to validate signatures.
   - **Certified Documents**
     If the public key is to be used to verify certified PDF forms, this flag must be enabled.
   - **Embedded High Privilege JavaScript**
     Interactive forms may contain JavaScript instructions. Use this flag to specify whether such PDF forms are also trusted if they are certified. You can select this option only if you've also enabled the **Certified Documents** flag.



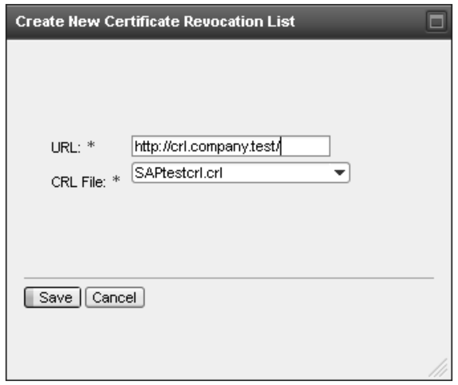
CER File: * mycompany_root_ca.cer
Trusted for: * ☑ Signatures and as Trusted Root
□ Certified Documents
□ Embedded High Privilege JavaScript

Figure 3.44 Managing Trusted Anchors

4. By clicking the **Save** button you finish the registration of the public key. Afterwards, it will appear in the **Trusted Anchors** table.
5. Then restart the Document Services Trust Manager service. Proceed in the same way as described for the ReaderRights credential (Section 3.3.4). After that, the list of trusted certificates is active in your system.

List of Revoked Certificates (Certificate Revocation Lists)

With the help of revoked certificate lists, keys that were originally classified as trusted can be subsequently marked as invalid. You also manage these lists on the **Document Security** tab, but select **Revoked Certificate List** to specify revoked certificates. Analogous to the ReaderRights credential and trusted anchors, you must load the CRL into the system via the **Manage CRL Files** button. Use the **Add New Object** button to access the selection of available CRLs (see Figure 3.45).



URL: * http://crl.company.test/
CRL File: * SAPtestcrl.crl

Figure 3.45 List of Revoked Certificates

In addition, fill in the **URL** field. This is a unique identifier contained in keys (`crldp` field) that can be retracted via this list. Typically, the list of revoked certificates can also be retrieved via this URL.

To complete the registration, click the **Save** button. The list now appears in the **List of Revoked Certificates** table. After configuring revoked certificate lists, the Document Services Trust Manager service must be restarted.

Secure Sockets Layer Connection Configuration

To work with certifications and signatures, the ADS web service must be accessed via a Secure Sockets Layer (SSL) connection. Detailed instructions for performing the necessary configuration steps can also be found in the SAP Help Portal. There, navigate to the **Adobe Document Services Configuration Guide** (*http://s-prs.co/v224282*). Choose **Configuration of Communication Security • Configuration of the Web Service SSL Connection** for a description of the SSL configuration.

3.5.3 Service Characteristics

Other important ADS properties need to be set as well: the size of the destination cache for buffering forms, the number of processes that can process documents in parallel, and the number of parallel jobs that can be used to parallelize print jobs. We'll discuss these in the following sections.

Destination Cache

As mentioned in Section 3.3.2, the ADS web services have a local cache where, among other things, form templates are stored. To modify the size of this cache, call up **Adobe Document Services** again in SAP NetWeaver Administrator under **Configuration • Infrastructure**. Select the **Caches** tab. The current size of the cache is displayed there in megabytes. The default value is **4 MB**.

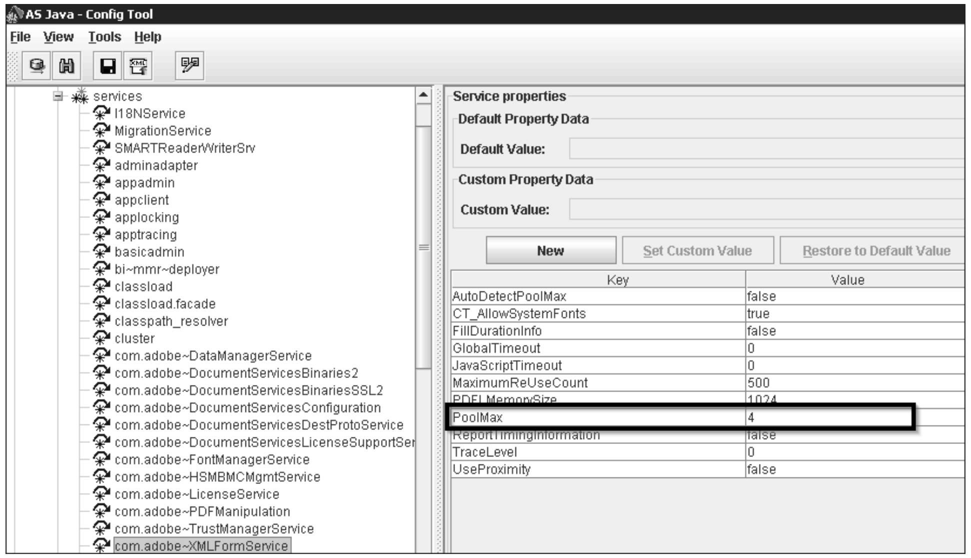
You can change the value via the **Configure Destination Cache** button. If you're using very large form templates or forms in many different languages, we recommend increasing this value so that the cache can be used optimally. The other two buttons, **Reset Destination Cache** and **Reset PDF Cache**, can be useful if you ever need to reset the cache. This should normally not be necessary. In the event of an error, for example, if a form change doesn't appear in the finished document, you can clear the buffer manually here.

Adjust PoolMax Value for Parallel Processing

There is a PoolMax property for each *XML Form Module* service. This property specifies how many documents can be processed in parallel. The XML Form Module service generates PDF documents and print forms and has a default value of "4". If you send many parallel requests to ADS and have a PC with multiple processors, you can increase this value to achieve optimum utilization of resources.

You can't make the setting via SAP NetWeaver Administrator. Instead, start the Config Tool on the operating system level under the path */usr/sap/<SID>/J<instance>/j2ee/configtool/configtool.bat* (or *sh*), and follow these steps:

1. Open the tree structure under **services**. You'll find the **com.adobe~XMLFormService** service there (see Figure 3.46).



Key	Value
AutoDetectPoolMax	false
CT_AllowSystemFonts	true
FillDurationInfo	false
GlobalTimeout	0
JavaScriptTimeout	0
MaximumReUseCount	500
PDFLMemorySize	1024
PoolMax	4
ReportTimingInformation	false
TraceLevel	0
UseProximity	false

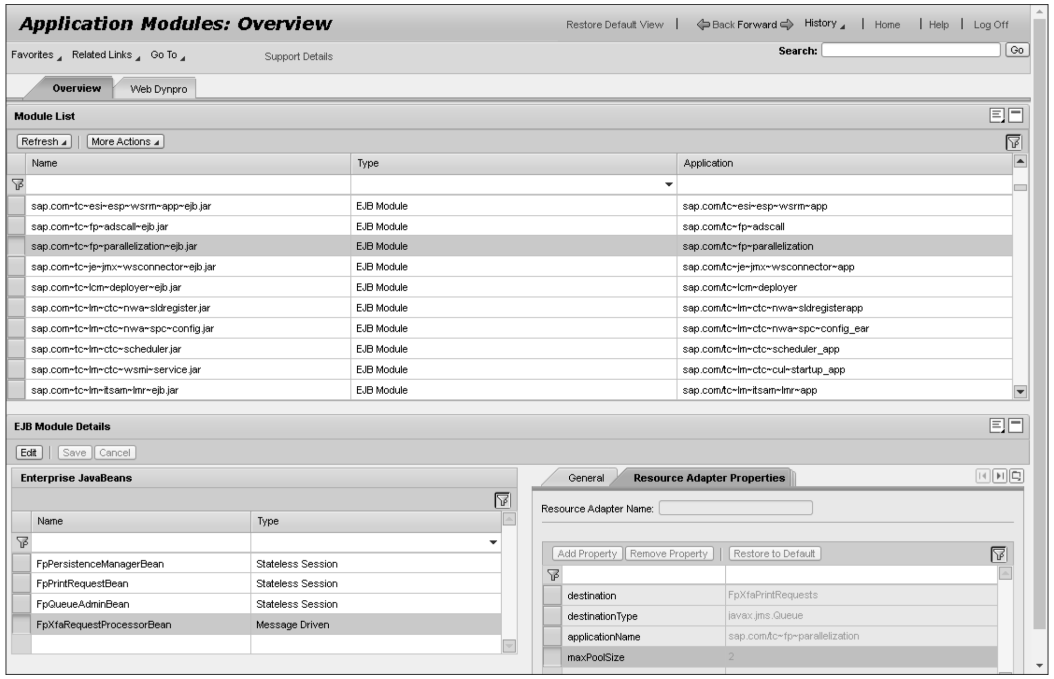
Figure 3.46 Config Tool: PoolMax Setting

2. In the **PoolMax** setting for this entry on the right, you can set the number of processes.
3. To change an entry, click **New**. Then specify the parameter name and the new value, and confirm with **OK.**
4. Select the menu path **File • Apply Changes**.
5. Restart the cluster, for example, via SAP Management Console.

3.5.4 Number of Parallel Print Jobs

When parallelizing print jobs, the individual requests are distributed to parallel processes. You can configure the number of jobs to be started simultaneously in SAP NetWeaver Administrator under **Configuration • Infrastructure**. Then follow these steps:

1. Select the **Application Modules** link.
2. Select the **sap.com~tc~fp~parallelization~ejb.jar** module (see Figure 3.47). To find the module faster, specify "*fp*" as the filter.
3. In the **Details EJB Module** area, select the **Enterprise JavaBean FpXfaRequestProcessorBean**, and click the **Edit** button.
4. Select the **Properties of the Resource Adapter** tab at the bottom right. Change the **maxPoolSize** value to the number of jobs to be processed simultaneously by the parallelization framework.
5. Save the settings.



Name	Type	Application
sap.com~tc~esi~esp~wsrm~app~ejb.jar	EJB Module	sap.com/tc~esi~esp~wsrm~app
sap.com~tc~fp~adscall~ejb.jar	EJB Module	sap.com/tc~fp~adscall
sap.com~tc~fp~parallelization~ejb.jar	EJB Module	sap.com/tc~fp~parallelization
sap.com~tc~je~jmx~wsconnector~ejb.jar	EJB Module	sap.com/tc~je~jmx~wsconnector~app
sap.com~tc~lcm~deployer~ejb.jar	EJB Module	sap.com/tc~lcm~deployer
sap.com~tc~lm~ctc~nwa~sldregister.jar	EJB Module	sap.com/tc~lm~ctc~nwa~sldregisterapp
sap.com~tc~lm~ctc~nwa~spc~config.jar	EJB Module	sap.com/tc~lm~ctc~nwa~spc~config_ear
sap.com~tc~lm~ctc~scheduler.jar	EJB Module	sap.com/tc~lm~ctc~scheduler_app
sap.com~tc~lm~ctc~wsmi~service.jar	EJB Module	sap.com/tc~lm~ctc~cul~startup_app
sap.com~tc~lm~itsam~lmr~ejb.jar	EJB Module	sap.com/tc~lm~itsam~lmr~app

Name	Type
FpPersistenceManagerBean	Stateless Session
FpPrintRequestBean	Stateless Session
FpQueueAdminBean	Stateless Session
FpXfaRequestProcessorBean	Message Driven

Resource Adapter Name:

destination	FpXfaPrintRequests
destinationType	javax.jms.Queue
applicationName	sap.com/tc~fp~parallelization
maxPoolSize	2

Figure 3.47 Number of Parallel Print Jobs

3.5.5 Fonts

The ADS installation includes a set of fonts located in the following directory: */usr/sap/<SID>/J2E/J<instance>/j2ee/os_libs/adssap/FontManagerService/fonts/adobe*. To use additional fonts in the form output, you need to make them available to the Document Services Font Manager. To do this, copy the fonts to the */usr/sap/<SID>/SYS/global/AdobeDocumentServices/FontManagerService/fonts/customer* directory.

The following types are supported:

- OpenType (*.otf*)
- TrueType (*.ttf*)
- PostScript Type 1 (*.pfb*, *.pfm*)

Before the newly installed fonts can be used, two components of ADS must be restarted. To do this, call up SAP NetWeaver Administrator, and select **Operations • Systems • Start & Stop • Java Services**. Start the **Document Services Font Manager** service. Then select the **Java Applications** tab, and click on the **Adobe Document Services** application.

3.5.6 Providing XDC Files for Print Output

To generate form outputs, ADS requires XML Forms Architecture (XFA) Device Configuration (XDC) files. These are XML files that describe the printer or the output to it. The XDC files are located in the */usr/sap/<SID>/SYS/global/AdobeDocumentServices/lib* directory. You can add modified or additional XDC files by copying the files to the *XDC/Customer* subdirectory. The delivered XDC files should remain basically unchanged.

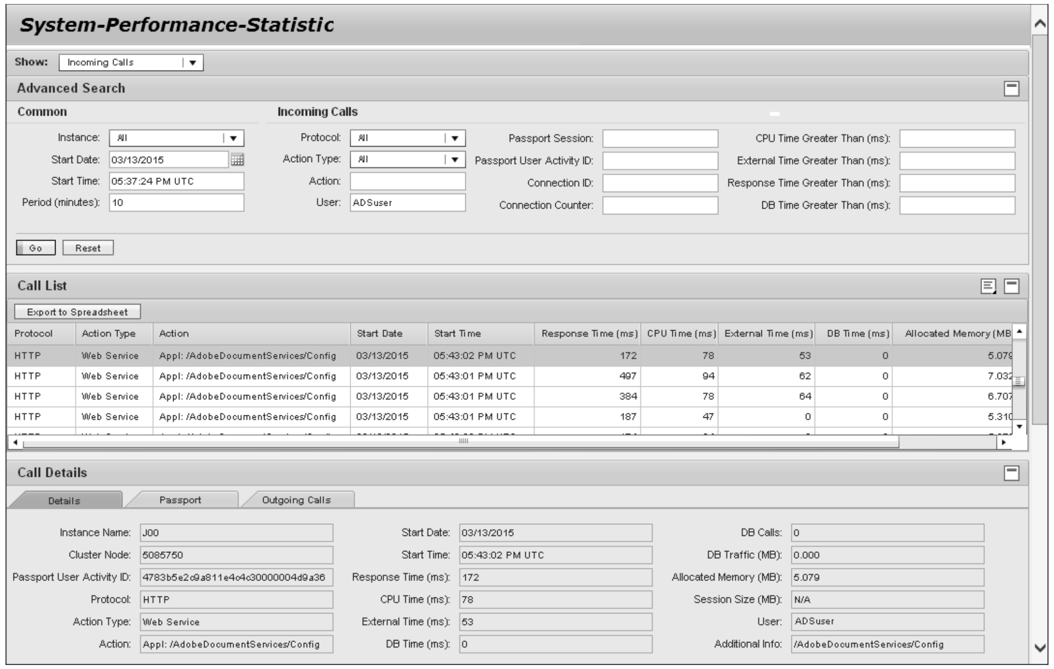
For more information on XDC files and the necessary assignment to device types on the ABAP application server, see Chapter 6, Section 6.3. The report for XDC administration (program RSPO0022) is also described there, with which you can conveniently manage XDC files without having to copy them to the operating system level.

3.6 Operating Adobe Document Services

This section covers important aspects for the operation of ADS, including monitoring individual components and settings, as well as detecting and analyzing errors (logging, tracing).

3.6.1 Performance Monitoring and Log Configuration

You can find information about the performance behavior of the Java system in the SAP NetWeaver Administrator **Availability and Performance** tab. Select the **Resource Monitoring** area, and then **System Performance Statistics**. You can filter ADS calls, for example, by entering "ADSUser" in the **User** field (see Figure 3.48). In the **Response Time (ms)** column, you can see the response times. You'll also find information about the memory consumption (**Allocated Memory (MB)** column).



Show: Incoming Calls

Instance: All
Start Date: 03/13/2015
Start Time: 05:37:24 PM UTC
Period (minutes): 10

Protocol: All
Action Type: All
Action:
User: ADSuser

Passport Session:
Passport User Activity ID:
Connection ID:
Connection Counter:

CPU Time Greater Than (ms):
External Time Greater Than (ms):
Response Time Greater Than (ms):
DB Time Greater Than (ms):

Protocol	Action Type	Action	Start Date	Start Time	Response Time (ms)	CPU Time (ms)	External Time (ms)	DB Time (ms)	Allocated Memory (MB
HTTP	Web Service	Appl: /AdobeDocumentServices/Config	03/13/2015	05:43:02 PM UTC	172	78	53	0	5.079
HTTP	Web Service	Appl: /AdobeDocumentServices/Config	03/13/2015	05:43:01 PM UTC	497	94	62	0	7.032
HTTP	Web Service	Appl: /AdobeDocumentServices/Config	03/13/2015	05:43:01 PM UTC	384	78	64	0	6.707
HTTP	Web Service	Appl: /AdobeDocumentServices/Config	03/13/2015	05:43:01 PM UTC	187	47	0	0	5.310

Instance Name: J00
Cluster Node: 5085750
Passport User Activity ID: 4783b5e2c9a811e4c4c30000004d9a36
Protocol: HTTP
Action Type: Web Service
Action: Appl: /AdobeDocumentServices/Config

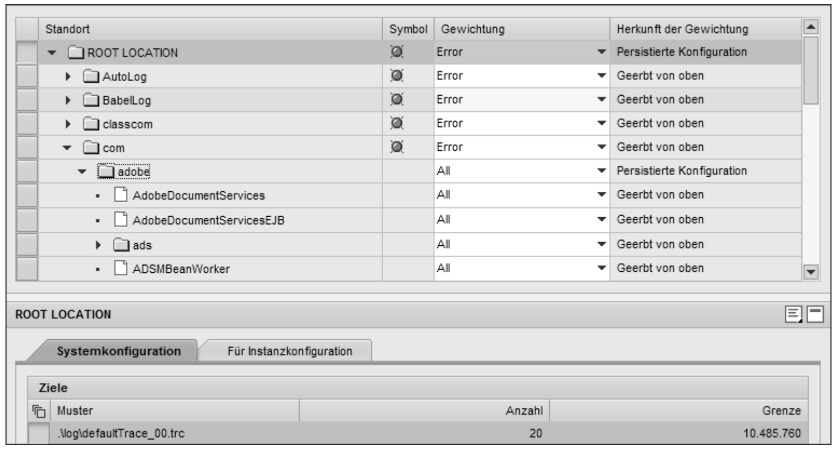
Start Date: 03/13/2015
Start Time: 05:43:02 PM UTC
Response Time (ms): 172
CPU Time (ms): 78
External Time (ms): 53
DB Time (ms): 0

DB Calls: 0
DB Traffic (MB): 0.000
Allocated Memory (MB): 5.079
Session Size (MB): N/A
User: ADSuser
Additional Info: /AdobeDocumentServices/Config

Figure 3.48 Performance Tracing

Using the *Log Configurator* available in SAP NetWeaver Administrator, you can configure which components should write how many logs or traces. To obtain precise processing information from ADS, proceed as follows:

1. Select the **Error Analysis** tab, and then **Logs and Traces**. Then click the **Log Configuration** link.
2. In the **Display** field, select **Trace Locations**, and navigate to the **ROOT LOCATION • com • adobe** node (see Figure 3.49).



Standort	Symbol	Gewichtung	Herkunft der Gewichtung
ROOT LOCATION		Error	Persistierte Konfiguration
AutoLog		Error	Geerbt von oben
BabelLog		Error	Geerbt von oben
classcom		Error	Geerbt von oben
com		Error	Geerbt von oben
adobe		All	Persistierte Konfiguration
AdobeDocumentServices		All	Geerbt von oben
AdobeDocumentServicesEJB		All	Geerbt von oben
ads		All	Geerbt von oben
ADSMBeanWorker		All	Geerbt von oben

Muster	Anzahl	Grenze
.\log\defaultTrace_00.trc	20	10.485.760

Figure 3.49 Adobe Document Services Tracing

3. In the **Weight** column, select the **All** value, and then click the **Copy to Subtree** and **Save Configuration** buttons.

This results in all subcomponents of ADS writing the maximum number of traces. Traces can be helpful in troubleshooting and are often required by support. To return to the default setting, click the **Default Configuration** button.

You can view log and trace files using the *Log Viewer* in SAP NetWeaver Administrator:

1. Select the **Error Analysis** tab, and then select **Logs and Traces**. Click the **Log Viewer** link.
2. Depending on the default settings in your system, the default trace is displayed next. If this isn't the case, select **View • Open View • System Logs and Traces**. Filter the results by entering "com.adobe" in the **Location** column.

3.6.2 Error Analysis

In this section, we discuss typical error messages and possible causes for these problem situations. Using the overview in Table 3.1 you can quickly and easily locate and correct common configuration errors.

Error When Calling ADS from the ABAP Application Server	Possible Cause
A dialog box specifying the login data appears.	`ADSUser` is either locked (refer to Section 3.2.1), or an incorrect password has been entered in the RFC connection data (Section 3.3.1).
The message **HTTP receive failed with exception communication_failure** appears (possibly as part of a longer error message).	An incorrect PC name or HTTP port was entered in the RFC connection data (refer to Section 3.3.1).
The message **HTTP send returned with status code** is displayed (possibly as part of a longer error message).	An incorrect path was specified in the RFC connection data (refer to Section 3.3.1).
The error message contains **com.adobe.ProcessingException: Problem accessing data from Destination....**	The `ADS_AGENT` user is locked or doesn't have the required permissions. The ICF services haven't been activated. An incorrect PC name or HTTP port has been entered in the destination, or the specified password is incorrect (refer to Section 3.3.2).

Table 3.1 Typical Error Messages When Calling Adobe Document Services and Possible Causes

Error When Calling ADS from Java Application Server	Possible Cause
The message **com.sap.tc.webdynpro-pdf-object.core.PDFObjectRuntime-Exception: Service call exception** is displayed (possibly as part of a longer error message).	ADSUser is locked (refer to Section 3.2.1), or incorrect information has been entered in the web service proxy (password, PC name, HTTP port, or path). The Java PDF object wasn't restarted after configuration changes were made (refer to Section 3.3.3). Typically, the stack trace containing the aforementioned exception includes additional information from which you can deduce the cause of the error.
General Errors When Calling ADS	**Possible Cause**
The message **User ADSUser doesn't have access to method rpData** is displayed (possibly as part of a longer error message).	The ADSUser isn't linked to the ADSCaller role (refer to Figure 3.6).
The message **com.adobe.ProcessingException: Could not retrieve a password for credential: ReaderRights** appears (possibly as part of a longer error message).	The ReaderRights credential hasn't been configured or has been configured under an incorrect alias (refer to Section 3.3.4).
The message **com.adobe.ProcessingException: Not allowed by credential error while applying usage rights to PDF** is displayed (possibly as part of a longer error message).	The ReaderRights credential has expired (refer to Section 3.3.4).
The error message contains **com.adobe.ProcessingException: Credential login error while applying usage rights to PDF**.	The Document Services Trust Manager service wasn't restarted after changes were made to the configuration (refer to Section 3.3.4).

Table 3.1 Typical Error Messages When Calling Adobe Document Services and Possible Causes (Cont.)

If you can't find a description of your problem in Table 3.1, SAP Note 944221 provides a procedure on how to locate the problem using test reports.

3.7 Hub Concept

Because ADS runs in a Java-based environment, the installation of an SAP NetWeaver Java stack is necessary. The application systems that require ADS can be based on either ABAP or Java. Most applications (and indeed all those discussed in this book) are developed in ABAP. Therefore, the installation of the Java stack is always an additional application server. This additional effort makes it possible to connect several ABAP systems to ADS. In addition to the sizing, which must be considered separately, the question of release dependency between the two stacks basically arises.

SAP positions ADS web services as a *hub*, which means that they are interoperable with other systems as a standalone system (as part of SAP NetWeaver AS Java). They also have their own patch cycle, for example, compared to an ERP system that calls ADS. Version reconciliation is performed via SAP Solution Manager to ensure that there are no version problems. However, new functions of ADS that are only available in later releases can't be supported, of course. For all application programs, you must therefore bear in mind that you can't automatically assume that all ADS functions exist. For this purpose, SAP provides an availability query every ABAP program should call before using a new function via the form runtime environment (see Chapter 6) or the ABAP PDF object (see Chapter 9). Only if the availability query is successful may a function be used.

This could be done automatically in the background, but it's not possible to decide in general whether the desired feature is necessary for the application scenario or only optional. In the first case, either an error message would be issued or a workaround would have to be programmed; in the latter case, the function would simply be dispensed with. Therefore, the decision must be made in the respective application program. If the release of your ADS is always greater than or equal to the ABAP release, you can do without this query.

The availability query is implemented using ABAP class CL_FP_FEATURE_TEST, which provides static method IS_AVAILABLE. The return is a Boolean value that provides information about whether the requested feature can be used. The feature itself is passed to the IS_AVAILABLE method. The possible values depend on the release of the ABAP stack and are listed as attributes of class CP_FP_FEATURE_TEST. Table 3.2 shows the values from the SAP NetWeaver AS ABAP 7.5 release. You probably won't use any of these features directly in your programs. Rather, they are functions that are queried internally, for example, in Web Dynpro ABAP.

Feature	Description
GC_JOB_PROFILES	Job profiles
GC_CLEAR_TEMPLATE_CACHE	Clear the form buffer
GC_SELF_TEST	Self-test
GC_ZCI_SCRIPT_INJECTION	Zero Client Installation (ZCI) script injection

Table 3.2 Features That Can Be Used for Querying

Feature	Description
GC_HIGHLIGHT_FIELDS_ON_ERROR	Highlight fields in case of error
GC_ASSEMBLE_STITCH	Merge into one PDF document, even for very large PDFs
GC_NO_DROP_DOWN_TO_ZCI	Don’t send dropdown lists to ZCI
GC_XDC_API	XDC API
GC_ZCI_GET_SET_FORM_DATA	ZCI: High-performance data transfer to the Web Dynpro framework
GC_LAYOUT_MIRRORING	Template mirroring
GC_RENDER_PDFA1	PDF/A-1
GC_EXT_PRN_OPTIONS	Additional printing options
GC_PDFA_EXT	PDF/A-1b, 2b, 3b, Converter, ZUGFeRD
GC_CAB	CAB output
GC_DEF_LOCALE	Fallback default locale
GC_FONT_OP	Font operations

Table 3.2 Features That Can Be Used for Querying (Cont.)

You may be interested in the PDF/A-1 (GC_RENDER_PDFA1) and job profiles (GC_JOB_PROFILES) features. We explain job profiles in Chapter 6, Section 6.1.3. However, because they have also been supported by the ADS hub concept since the first version (Release 7.0, SP13), you can dispense with the query. For more information about PDF/A-1, a format which can be used for archiving PDFs, see Chapter 6, Section 6.5.1.

The basic invocation of the feature test is shown in Listing 3.1.

```
DATA lv_flag TYPE abap_bool.
DATA lx_exception TYPE REF TO cx_fp_runtime.
TRY.
lv_flag = cl_fp_feature_test=>is_available(
          lv_feature = cl_fp_feature_test=>gc_job_profiles )
CATCH cx_fp_runtime INTO lx_exception.
* Error during ADS communication.
ENDTRY.
IF lv_flag = abap_true.
* OK…
.
```

Listing 3.1 Availability Query of an ADS Feature

The feature test itself accesses a cached version number of ADS and performs a remote call only in exceptional cases. Therefore, the query isn’t performance critical.

Further Information on the Hub Concept

The ADS hub concept is also described in the “ADS Hub – Central Installation of Adobe Document Service” blog post in the SAP Community at *http://s-prs.co/v224283*.

3.8 Summary

In this chapter, we’ve shown you how to configure the ABAP and Java systems to run ADS. This is necessary to try out the examples in this book. Furthermore, you’ve gained an overview of the individual components of ADS and how they interact. You’ve gained knowledge about important aspects of the operation of ADS and can independently locate and correct typical configuration errors.

In the next chapter, we’ll discuss the structure, interface, and context of a form.

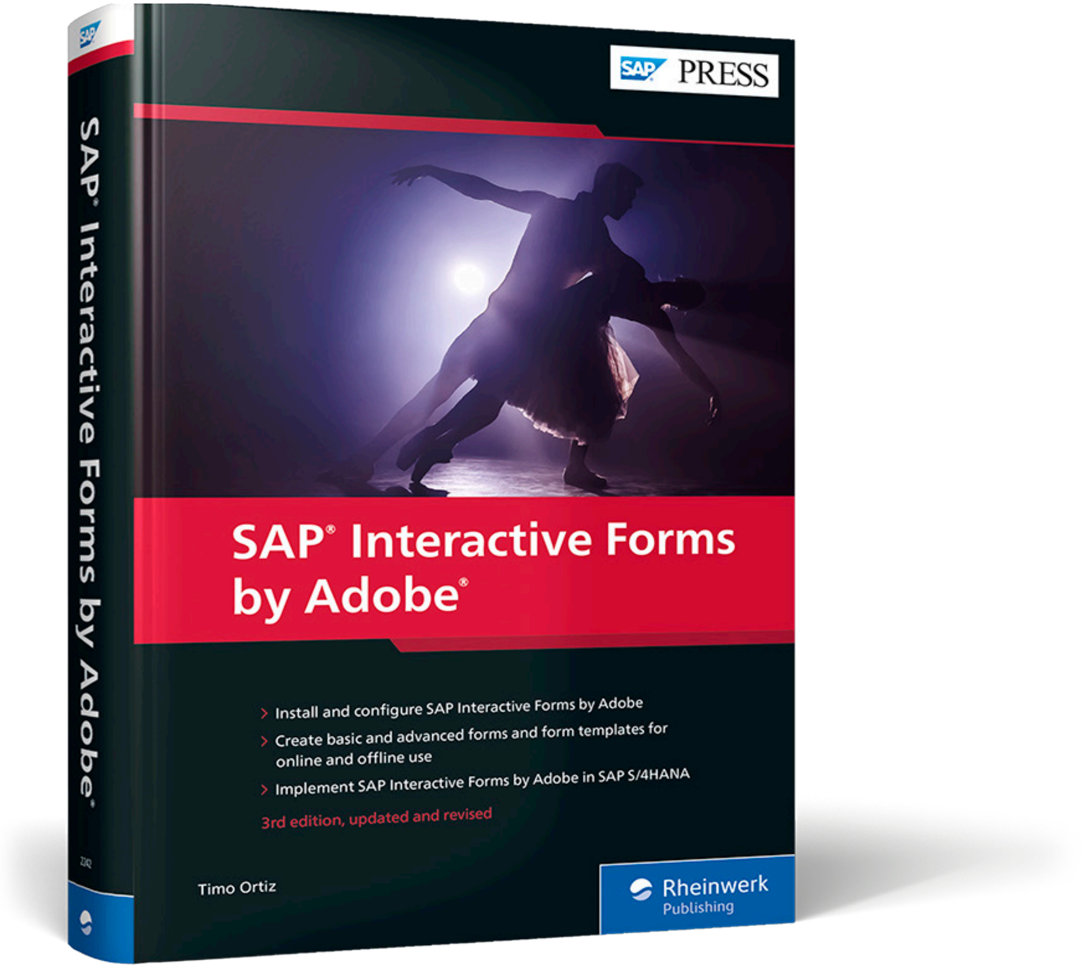
Contents

Appendices 667

Index

SAP PRESS First-hand knowledge.

Rheinwerk Publishing


SAP PRESS
SAP® Interactive Forms by Adobe®
› Install and configure SAP Interactive Forms by Adobe
› Create basic and advanced forms and form templates for online and offline use
› Implement SAP Interactive Forms by Adobe in SAP S/4HANA
3rd edition, updated and revised
Timo Ortiz
Rheinwerk Publishing
SAP® Interactive Forms by Adobe®

Timo Ortiz has worked in the field of forms development and output management for more than 20 years. He has worked for the consulting group Corporate Business Solutions Unternehmensberatung GmbH (cbs) in Heidelberg as a solution architect since 2008. He is responsible for output management and setting up comprehensive print processes and output scenarios. This includes form design, control and customizing options, as well as language-independent developments. As a project member and co-project manager, he implements cross-module projects within the scope of support activities up to global rollouts of template developments with corresponding country-specific adaptations and characteristics.

Timo Ortiz

SAP Interactive Forms by Adobe

724 pages, 3rd, updated and revised edition 2022, $99.95
ISBN 978-1-4932-2242-1

www.sap-press.com/5494