Oracle® Agile Engineering Data Management

Synchronous SAP Connector for Enterprise Integration Platform 2.2.1

Part No. E15825-02

March 2010

Copyright and Trademarks

CONTENTS

Preface

The Oracle documentation set includes Adobe® Acrobat™ PDF files. The Oracle Technology Network (OTN) Web site (http://www.oracle.com/technology/documentation/agile.html) contains the latest versions of the Oracle Agile EDM PDF files. You can view or download these manuals from the Web site, or you can ask your Agile administrator if there is an Oracle Documentation folder available on your network from which you can access the documentation (PDF) files.

Note	To read the PDF files, you must use the free Adobe Acrobat Reader™ version 7.0 or later. This program can be downloaded from the Adobe Web site (http://www.adobe.com).

Note	Before calling Agile Support about a problem with an Oracle Agile EDM manual, please have the full part number, which is located on the title page.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Readme

Any last-minute information about Oracle Agile EDM can be found in the Release Notes file on the Oracle Technology Network (OTN) Web site (http://www.oracle.com/technology/documentation/agile.html)

Agile Training Aids

Go to the Oracle University Web page (http://www.oracle.com/education/chooser/selectcountry_new.html) for more information on Agile Training offerings.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Chapter 1

Overview

The SAP Source Connector is a connector provided as an additional component of the Enterprise Integration Platform. In general, it allows connecting to the Enterprise Integration Platform from inside SAP R/3. The connection between R/3 and the SAP Source Connector is established in a "synchronous" mode i.e. the calling R/3 system waits until a response comes back from the SAP Connector inside the Enterprise Integration Platform.

Purpose of this document is to provide you with the information how to configure and use the Synchronous SAP Source Connector within the Enterprise Integration Platform.

The configuration of the SAP Source Connector consists of following setup steps:

Chapter 2: provide the SAP Source Connector specific configuration parameters in the configuration file eai_ini.xml e.g. how to register in SAP (gateway service, registration name etc.)

Chapter 3: create a new RFC destination for the SAP Source Connector inside R/3

Chapter 4: create or use an existing Remote Function Module (BAPI, RFC enabled functions) for the remote call (e.g. parameters, behavior, what to do etc.) inside R/3

Chapter 5: develop an ABAP program which wraps the remote call of the Remote Function Module

Chapter 6: define where the ABAP wrapper should be called from inside R/3 e.g. "Display File" in Document Info Record (separate transaction, customizing, user exit etc.)

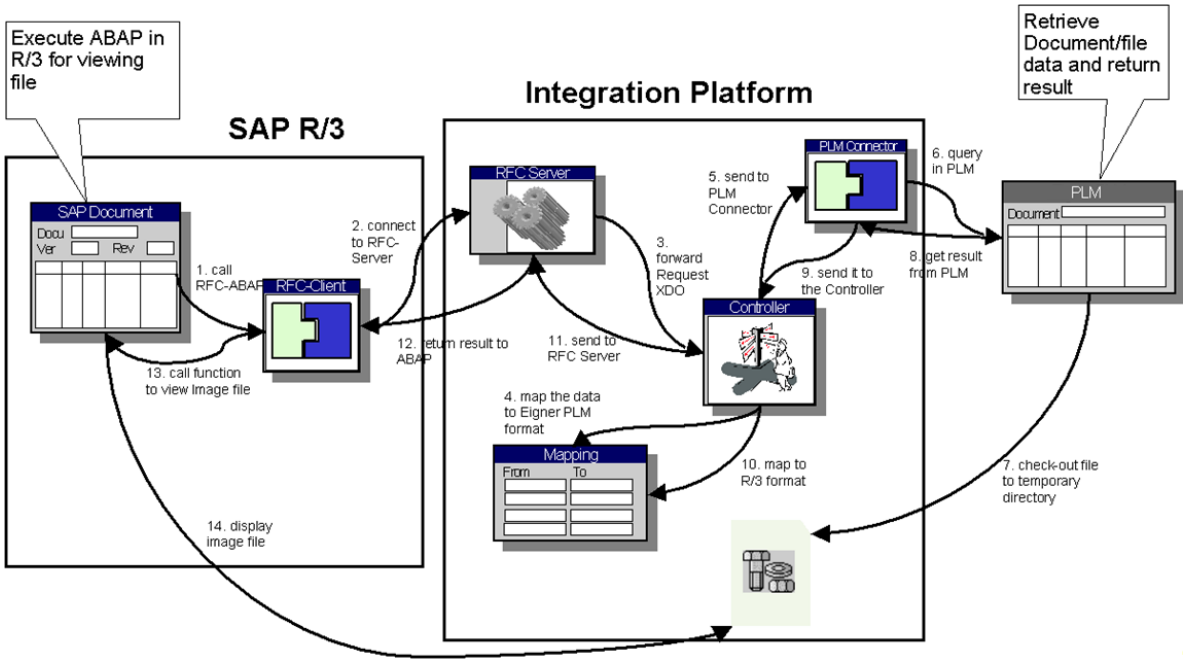
Chapter 7: design the mapping files e.g. for mapping the XML message going from R/3 to Agile EDM and back

Chapter 8: test-run the integration from R/3 and check the result e.g. in the log file

Chapter 9: troubleshooting

In addition to the SAP Source Connector, the Integration Platform provides a solution for viewing Agile EDM files from inside SAP R/3. This solution is based on the SAP Source Connector and comes with pre-configured configuration and code samples (eai_ini.xml , ABAP code, XSL mapping files).

An overview of the technical transfer steps as part of this File Viewing solution is provided below.



Transfer Steps:

1. A special ABAP procedure is called from the R/3 application e.g. from the “Display File” button in the Document Info Record.
2. The ABAP procedure connects to the SAP Source connector via Remote Function Call (RFC) protocol and provides the respective parameters.
3. The SAP Source connector converts those parameters into XML format and sends it to the EIP Controller.
4. The EIP Controller converts the data into the format required by the Target System, e.g. Agile EDM.
5. The XML message is sent to the Agile EDM Connector.
6. The Agile EDM Connector queries for the respective document and files inside Agile EDM.
7. The request by the Agile EDM Connector triggers the Agile EDM FileServer to check-out the files into a predefined shared directory.
8. The result and status of this operation is returned to the Agile EDM Connector.
9. The result and status is converted to XML and sent to the EIP Controller.
10. The XML message is mapped into a format understood by the SAP Source Connector.
11. The XML message is sent to the SAP Source Connector.
12. The SAP Source Connector returns the result to the ABAP procedure in R/3, which it was originally called from.
13. The ABAP routine performs some other operations e.g. opens the R/3 viewer for displaying the files provided by Agile EDM.
14. The files are displayed in the R/3 viewer.

Chapter 2

Configuration

The Synchronous SAP Source connector is technically based on an RFC server. The respective section in the eai_ini.xml file for the SAP Source connector is described below. It basically describes the server connection parameters for registration of the RFC server in the SAP system, the client connection parameters for getting data dictionary information for the incoming Remote Function Module and the supported Business Objects and Actions. The attributes of *registration* describe how the RFC server should register itself in the SAP system. The *connection* section and the *bor* tag are described in the AdminManual.doc. The *bor* section itself is described below.

```
<synchronous name="sap-r3-sync" version="2.2.0" active="false"
class="com.eigner.eai.connector.sap.SyncR3Connector">
     <registration name="registration name" gwhost="gateway_server"
gwservice="gateway_service" unicode=" false"/>
     …
</synchronous>
```

Details of the XML tags:

Tag	Description
registration	Contains the parameter for the RFC-Server (EIP) registration in SAP R/3

Details of the XML tag *registration*:

Attribute	Description
name	name with which the RFC server registers itself in the SAP R/3 system (default value is: JCOSERVER01)
gwhost	name of the gateway server
gwservice	name of the gateway service number e.g. sapgw00
unicode	SAP-System is Unicode or not; allowed values: true; false (default)

Note If the SAP-System is a Unicode system, the SAP Server has to be registered as a Unicode Server inside SAP. (SM59 -> RFC-Destination -> TCP/IP Connection -> <RFC_SERVER>/change-> Folder “Special Options” -> Character Width in Target System: activate “Unicode”).

Note You can also activate tracing for the RFC-Destination inside SAP. (SM59 -> RFC-Destination -> TCP/IP Connection -> <RFC_SERVER>/change-> Folder “Special Options” -> Special Flags: activate “Trace”).

Next is an overview of the supported Business Objects (e.g. DOCUMENT) and Actions (e.g. QUERY), which are invoked by the incoming Remote Function Module (e.g.Z_REMOTE_DOC_DETAIL). The parameters in each section explain how the connector can handle normal errors and which data should be passed to the requested business object.

```
<bor version="2.1.0">
      <bo name="DOCUMENT" verb="QUERY" function="Z_REMOTE_DOC_DETAIL"
error type="parameter" parameter="RETURN">
            <data>
                  <para>DOKNR</para>
                  <para>DOKAR</para>
                  <para>DOKTL</para>
                  <para>DOKVR</para>
            </data>
      </bo>
      <bo name="DOCUMENT-FILE" verb="CHECKOUT"
function="Z REMOTE DOC_FILE_CHECKOUT" error_type="parameter"
parameter="RETURN">
            <data>
                  <para>DOKNR</para>
                  <para>DOKAR</para>
                  <para>DOKTL</para>
                  <para>DOKVR</para>
                  <para>DOC_FILE</para>
            </data>
      </bo>
</bor>
```

Details of the XML tags:

Tag	Description
bo	business object
data	data of the Remote Function Module call
para	name of the data parameter, substructure of <data>; represents the Import Parameters of the RFC module

Details of the XML tag *bo*:

Attribute	Description	Values
name	name of the business object	
verb	action of the business object	
function	name of the incoming Remote Function Module (must be unique) ; must match the one as defined in chapter 4 e.g. Z_REMOTE_DOC_FILE_CHECKOUT	
error_type	type of the error which should be “raised” if a normal error occurred e.g. DOCUMENT QUERY --> document does not exist (depending on the Remote Function Module)	exception, parameter
exception	identifier of the exception (required if error_type is exception)	
parameter	name of the parameter (must be based on SAP structure BAPIRET2) (required if error_type is parameter)	

Chapter 3

RFC Destination

An RFC destination (transaction SM59) has to be created for the SAP Source Connector (RFC server) with the connection type T (TCP/IP), the activation type “Registration” and program ID with which the RFC server registers itself in the SAP system.

Please use following parameters for the standard solution to work correctly:

RFC Destination: RFC_SERVER

Connection Type: T (TCP/IP connection)

Activation Type: Registration

Program ID: JCOSERVER01

The connection can be tested for the RFC destination.

First, the RFC server (i.e. EIP) has to be started. When the EIP logger info "*(SyncR3Connector) - Server JCOSERVER01 changed state from [STARTED] to [STARTED LISTENING]*" appears, the RFC server is ready to work. Then the connection test can be executed by clicking on the button “Test Connection”.

Chapter 4

Remote Function Module

The Remote Function Module implements the functionality of the interface, behavior, what has to be done etc. for the remote program execution. You create the Remote Function Module with the function builder (transaction SE37) or search for an existing Remote Function Module via BAPI Explorer respectively via F4-Help of the function builder.

For a detailed example, please see the RFC function Z_REMOTE_DOC_FILE_CHECKOUT.

Chapter 5

Wrapper ABAP

The Wrapper ABAP provides the input data, calls the Remote Function Module remotely and analyzes the result data.

Example: DMS: the wrapping ABAP program determines the file, calls remote the template (see chapter 4) and shows the checked out file with the Internet Explorer.

You can change this behavior by modifying this ABAP or developing your own one!

```
IF applikationsnummer = 1.
  view_file  = draw-filep.
  view_dappl = draw-dappl.
ELSEIF applikationsnummer = 2.
  view_file  = draw-filep1.
  view_dappl = draw-dappl1.
ELSE.
  EXIT.
ENDIF.

doc_file-wsapplication = view_dappl.
doc_file-docfile       = view_file.

CALL FUNCTION 'Z_REMOTE_DOC_FILE_CHECKOUT'
     DESTINATION 'RFC_SERVER'
     EXPORTING
       doknr            = draw-doknr
       dokar            = draw-dokar
       doktl            = draw-doktl
       dokvr            = draw-dokvr
       doc_file         = doc_file
     IMPORTING
       checked_out_file = co_doc_file
       return           = doc_return
     EXCEPTIONS
       system           = 1
       OTHERS           = 2.
```

- remote call
- name of the RFC destination

Note When calling the respective function, e.g. Z_FILE_VIEW_VIA_SERVER, make sure a Destination (in this example RFC_SERVER) is added.

The following function modules are part of the standard EIP Source Connector delivery:

- Z_FILE_VIEW_VIA_SERVER_IE:

  get file from PLM into shared directory and display it in the Internet Explorer

- Z_FILE_VIEW_VIA_SERVER_INTERN:

  get file from PLM into shared directory and display it in the R/3 internal viewer

- Z_GET_FILE_VIA_SERVER_FOR_VIEW:

  get file from PLM, make it available in R/3 file vault and call program for respective Workstation Application

Chapter 6

Integration

Integrate the wrapper ABAP program in SAP R/3 (separate transaction, customizing, user exit, adaptation of SAP standard source code, etc.)

Example in DMS:

1. The wrapper ABAP program is integrated in the customizing.
2. Run “Define workstation application” under *Implementation Guide / Cross-Application Components / Document Management System / General Data*.
3. Then run “Define workstation application in network” for the chosen workstation application.

   Here you assign the ABAP wrapper to the used data carrier type and the application type 1 (view).

   Now you have integrated the ABAP wrapper.
4. For using the functionality of the ABAP wrapper, assign the special workstation application to an original of the document management record.
5. Select the application and press the “Display” button and the ABAP wrapper will be executed.

Chapter 7

Mapping

In the standard delivery, mapping files are already provided, which map the request data from R/3 to PLM format and vice-versa. Please adapt them if you want to support different use cases and objects.

Mapping Example SAP R/3 --> Agile EDM: sync_r3_axa_request.xsl

```
<!-- DOCUMENT TEMPLATES -->
<xsl:template match="record[@type='DOCUMENT' and @verb='QUERY']/data">
    <!-- Z_REMOTE_DOC_DETAIL -->
    <xsl:variable name="sapDocNumber" select="DOKNR"/>
    <xsl:variable name="sapSelectDocNumber" select="EignerExtension:addPercentRight(string($sapDocNumber),25)"/>
    <operation object="XML-DOC" name="query">
        <xsl:element name="where">
            <xsl:attribute name="DOCUMENT_ID"><xsl:value-of select="$sapSelectDocNumber"/></xsl:attribute>
        </xsl:element>
    </operation>
</xsl:template>

<!-- DOCUMENT-FILE TEMPLATES -->
<xsl:template match="record[@type='DOCUMENT-FILE' and @verb='CHECKOUT']/data">
    <!-- Z_REMOTE_DOC_FILE_CHECKOUT -->
    <xsl:variable name="sapDocNumber" select="DOKNR"/>
    <xsl:variable name="sapSelectDocNumber" select="EignerExtension:addPercentRight(string($sapDocNumber),25)"/>
    <operation object="XML-DOC" name="query">
        <xsl:element name="where">
            <xsl:attribute name="DOCUMENT_ID"><xsl:value-of select="$sapSelectDocNumber"/></xsl:attribute>
        </xsl:element>
        <xsl:apply-templates select="DOC_FILE"/>
    </operation>
</xsl:template>
<xsl:template match="record[@type='DOCUMENT-FILE' and @verb='CHECKOUT']/data/DOC_FILE">
    <select>
        <operation object="XML-DOC-FILE" name="checkout" ckopath="c:/temp/" ckoflag="one">
            <xsl:element name="where">
                <xsl:attribute name="T_FILE_DAT.ORG_NAME"><xsl:value-of select="DOCFILE"/></xsl:attribute>
            </xsl:element>
        </operation>
    </select>
</xsl:template>
```

Mapping Example Agile EDM --> SAP R/3: sync_r3_axa_response.xsl

```
<xsl:template match="record[@type='DOCUMENT' and @verb='QUERY']/result/XML-DOC">
    <!-- Z_REMOTE_DOC_DETAIL -->
    <DOC_DATA>
        <DOCUMENTNUMBER><xsl:value-of select="T_DOC_DAT.DOCUMENT_ID"/></DOCUMENTNUMBER>
        <DESCRIPTION><xsl:value-of select="T_DOC_DAT.DOC_NAME_GER"/></DESCRIPTION>
    </DOC_DATA>
</xsl:template>

<!-- DOCUMENT-FILE TEMPLATES -->
<xsl:template match="record[@type='DOCUMENT-FILE' and @verb='CHECKOUT']/result/XML-DOC-FILE">
    <!-- Z_REMOTE_DOC_FILE_CHECKOUT -->
    <CHECKED_OUT_FILE>
        <DOCFILE><xsl:value-of select="T_FILE_DAT.ORG_NAME"/></DOCFILE>
        <DOCPATH><xsl:value-of select="../../data/operation/select/operation/@ckopath"/></DOCPATH>
    </CHECKED_OUT_FILE>

</xsl:template>
```

Chapter 8

Logging Information

These are sample logging information from the RFC server (excerpts shown in the example below).

... Logging Information ...

```
INFO  (Controller) - Initializing synchronous connector 'sap-r3' ...
DEBUG (SyncR3Connector) - Entering method init
INFO  (SyncR3Connector) - Initializing connection parameter
DEBUG (SyncR3Connector) - Leaving method init
DEBUG (SyncR3Connector) - Entering method init
INFO  (SyncR3Connector) - Initializing server connection parameter
INFO  (SyncR3Connector) - Reading Configuration (SAP)
DEBUG (SyncR3Connector) - Reading configuration for SERVER / TEST_EX / Z_STFC_CONNECTION
DEBUG (SyncR3Connector) - Reading configuration for DOCUMENT / QUERY / Z_REMOTE_DOC_DETAIL
DEBUG (SyncR3Connector) - Reading configuration for DOCUMENT-FILE / CHECKOUT / Z_REMOTE_DOC_FILE_CHECKO
DEBUG (SyncR3Connector) - Leaving method init
```

Initialization of the SAP Source Connector

... Logging Information ...

```
INFO  (Controller) - Synchronous Connectors starting ...
INFO  (Controller) - Starting synchronous connector: sap-r3
DEBUG (SyncR3Connector) - Entering method start
INFO  (SyncR3Connector) - Starting connection to SAP
DEBUG (SyncR3Connector) - Leaving method start
DEBUG (SyncR3Connector) - Entering method start
INFO  (SyncR3Connector) - Creating function templates (SAP)
DEBUG (SyncR3Connector) - Creating function template for : Z_REMOTE_DOC_FILE_CHECKOUT
DEBUG (SyncR3Connector) - Checking data para for function : Z_REMOTE_DOC_FILE_CHECKOUT
DEBUG (SyncR3Connector) - Creating function template for : Z_STFC_CONNECTION
DEBUG (SyncR3Connector) - Checking data para for function : Z_STFC_CONNECTION
DEBUG (SyncR3Connector) - Creating function template for : Z_REMOTE_DOC_DETAIL
DEBUG (SyncR3Connector) - Checking data para for function : Z_REMOTE_DOC_DETAIL
INFO  (SyncR3Connector) - Server JCOSERVER01 changed state from [ STOPPED ] to [ STARTED ]
DEBUG (SyncR3Connector) - Leaving method start
INFO  (Controller) - Controller threads starting ...
INFO  (SyncR3Connector) - Server JCOSERVER01 changed state from [ STARTED ] to [ STARTED  LISTENING ]
```

Starting up the SAP Source Connector

... Logging Information ...

```
INFO  (SyncR3Connector) - Server JCOSERVER01 changed state from [ STARTED  LISTENING ] to [ STARTED  LI
INFO  (SyncR3Connector) - Entering method handleRequest for function Z_REMOTE_DOC_FILE_CHECKOUT
DEBUG (SyncR3Connector) - Entering method process
DEBUG (SyncR3Connector) - method createControlArea
DEBUG (SyncR3Connector) - Entering method createRecordArea
DEBUG (SyncR3Connector) - Import-Parameter : DOKVR
DEBUG (SyncR3Connector) - Import-Parameter : DOC_FILE
DEBUG (SyncR3Connector) - Import-Parameter : DOKTL
DEBUG (SyncR3Connector) - Import-Parameter : DOKAR
DEBUG (SyncR3Connector) - Import-Parameter : DOKNR
DEBUG (SyncR3Connector) - Leaving method createRecordArea
INFO  (SyncR3Connector) - Sending SAP R/3 data to Controller
INFO  (Controller) - Processing object: com.eigner.eai.businessobject.BusinessObject@7f11fb: guid = 9dd
INFO  (Controller) - Transforming from source 'sap-r3' to target 'axalant' via pipe 'r3-axa-request' ..
DEBUG (AxalantConnector) - Writing records(s) to axalant.
DEBUG (AxalantConnector) - Reading data for Business Object/Verb:DOCUMENT-FILE CHECKOUT REQUEST/RECEIVE
DEBUG (AxalantConnector) - Business Object Index:28
INFO  (PollingWatcher) - PollingWatcher running ...
INFO  (QueueWatcher) - QueueWatcher running (0 entries) ...
DEBUG (AxalantConnector) - Leaving writeRecords.
INFO  (Controller) - Processing object: com.eigner.eai.businessobject.BusinessObject@78aae1: guid = 9dd
INFO  (Controller) - Transforming from source 'axalant' to target 'sap-r3' via pipe 'r3-axa-response' .
DEBUG (SyncR3Connector) - Leaving method process
DEBUG (SyncR3Connector) - Entering method setOutputParameter
DEBUG (SyncR3Connector) - Function Z_REMOTE_DOC_FILE_CHECKOUT : setting parameter CHECKED_OUT_FILE
DEBUG (SyncR3Connector) - Leaving method setOutputParameter
INFO  (SyncR3Connector) - Server JCOSERVER01 changed state from [ STARTED  LISTENING  BUSY ] to [ START
```

Called function to check out file from Agile EDM

Chapter 9

Important Notes

The business objects and actions are identified by the incoming Remote Function Module. Therefore, the Remote Function Modules have to be unique.

The remotely called Remote Function Module is the incoming function for the RFC server.

The RFC server will raise the ABAP exception “SYSTEM” if an unexpected error occurs (e.g. conversion error, communication etc.). Therefore, the ABAP wrapper must catch this exception.

If the ABAP Wrapper is integrated by customizing “Define workstation application in network” in the IMG, the suffix “%SAP-FUNCTION%” will be required.