Agile
Version e6.1

ORACLE

Oracle® Agile Engineering Data Management

Manual for SAP Link on Enterprise Integration Platform 2.2.2

Part No. E22053-01

March 2011

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_eseries.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

Installation

Requirements

The SAP Link supports the same platforms as the corresponding Enterprise Integration Platform (see EIP Installation Manual).

As the EIP is currently certified on 32-bit runtime environments only (also on 64-bit operating system), please check the platform support for the SAP JCo 2.1.x components on SAP Support Portal (http://service.sap.com/connectors) before choosing the hardware.

Minimum Network Connectivity and Bandwidth

Note 100 MB/s LAN connections based on TCP/IP between the Integration Platform, Agile EDM Server and SAP R/3 Server.

Minimum Disk Space

Note The basic server installation requires 120 MB which also includes administration client installation.

Note If running the SAP Link, additional space is required for the SAP libraries (approx. between 8.5 and 26 MB depending on the platform).

SAP R/3

Minimum version of SAP R/3 required by the SAP Link is 4.6C.For a complete list and the maximum SAP system version please check on the SAP JCo support site.

The following privileges are required for the R/3 user(s), which is used by the R/3 Connector:

- Z_TRANS_RFC
- S_A.SCON
- All object-related privileges, e.g. for creating a material

Checklist

Please see at the end of this document.

Steps

The SAP Link gets installed on top of an existing Enterprise Integration Platform.

Please extract the SAP Link ZIP file into the root installation directory of the EIP.

Then all files will be automatically placed in the proper folders, e.g. the file *bin\eip-sap.jar* in the SAP Link ZIP would go into the directory *<eai.home>\bin\eip-sap.jar*.

Chapter 2

Asynchronous SAP Connector

Overview

The SAP Connector is part of the SAP link solution, which is based on the Enterprise Integration Platform, but provides predefined configuration and mapping in addition. It provides access to following SAP R/3 objects out-of-the-box: Materials Management, Bill of Materials, Document Management, Change Management and Classification. In addition, it is possible to extend the SAP link solution by calling any BAPI or RFC-Function in R/3.

Configuration of the SAP R/3 Connector

Below is a description of the SAP R/3 connector section in the eai_ini.xml file. It describes the connection parameters and the supported business objects and actions. The client, user, password, language and connection parameters describe how the connector should connect to SAP R/3.

```
      <connector name="sap-r3" version="2.2.0" active="true"
class="com.eigner.eai.connector.sap.R3Connector">
            …
            <connection name="default" active="true">
                  <server host="sap_server" sysno="00"/>
                  <client>100</client>
                  <user>sap_user</user>
                  <password>sap_pwd</password>
                  <language>EN</language>
                  <rfc-trace>false</rfc-trace>
                  <jco-trace-level>OFF</jco-trace-level>
                  <in-schema>${eai.conf}/sap_in_schema.xsd</in-schema>
                  <out-schema>${eai.conf}/sap_out_schema.xsd</out-schema>
            </connection>
            <connection name="load-balance" active="false">
                  <load-balance msghost="sap_server"
r3name="sap_system_name" group="group_server"/>
                  <client>100</client>
                  <user>sap_user</user>
                  <password>sap_pwd</password>
                  <language>EN</language>
                  <rfc-trace>false</rfc-trace>
                  <jco-trace-level>OFF</jco-trace-level>
                  <in-schema>${eai.conf}/sap_in_schema.xsd</in-schema>
                  <out-schema>${eai.conf}/sap_out_schema.xsd</out-schema>
            </connection>
            …
      </connector>
```

Details of the XML tags for a standard connection:

Attribute	Description	Values
server	host: SAP server host sysno: system number	
client	SAP logon client	
user	SAP logon user	
password	encrypted SAP logon password	
language	SAP logon language	
rfc-trace	enable SAP RFC trace logging (will be written into the ${eai.log} directory)	true false (default)
jco-trace-level	enable SAP JCo trace logging (will be written into the ${eai.log} directory)	OFF: off (default) JAVA: JCo Java API JNI: JCo JNI API ERROR: error diagnostic
in_schema	name of the SAP input schema file for all BOR entries (will be created only in case of test)	
out_schema	name of the SAP output schema file for all BOR entries (will be created only in case of test)	

Details of the XML tags for a load-balanced connection:

Attribute	Description	Values
load_balance	msghost: Message host r3name: Name of SAP system group: Group server	
client	SAP logon client	
user	SAP logon user	
password	encrypted SAP logon password	
language	SAP logon language	
rfc-trace	enable SAP RFC trace logging (will be written into the $eai-home directory)	true false (default)
jco-trace-level	enable SAP JCo trace logging (will be written into the $eai-home/tmp directory)	OFF: off (default) JAVA: JCo Java API JNI: JCo JNI API ERROR: error diagnostic
in_schema	name of the SAP input schema file for all BOR entries (will be created only in case of test)	
out_schema	name of the SAP output schema file for all BOR entries (will be created only in case of test)	

Next is an overview of the supported business objects (e.g. MATERIAL) and actions (e.g. CREATE). The parameters in each section explain how the connector can get access to the data in R/3, e.g. which BAPIs should be used for reading and writing the data.

```
<bor version="2.1.0">
      <bo name="ITEM">
            <verb name="CREATE" direction="RECEIVE">
                  <function>BAPI_MATERIAL_SAVEDATA</function>
                  <return code="RETURN-TYPE" message="RETURN-MESSAGE"
commit="true" logging="false"/>
                  <result>
                        <para>RETURNMESSAGES</para>
                  </result>
            </verb>
            <verb name="QUERY" direction="RECEIVE">
                  <function>/EIGNER/MATERIAL_DETAILS</function>
                  <return code="RETURN-TYPE" message="RETURN-MESSAGE"
commit="false" logging="false"/>
                  <result>
                        <para>EXP_MARA</para>
                        <para>TAB_MARC</para>
                        <para>TAB_MPOP</para>
                        <para>TAB_MPGD</para>
                        <para>TAB_MARD</para>
                        <para>TAB_MBEW</para>
                        <para>TAB_MLGN</para>
                        <para>TAB_MVKE</para>
                        <para>TAB_MLGT</para>
                        <para>TAB_MAKT</para>
                        <para>TAB_MARM</para>
                        <para>TAB_MEAN</para>
                        <para>TAB_MLTX</para>
                        <para>TAB_MLAN</para>
                        <para>TAB_MFHM</para>
                  </result>
            </verb>
            ...
      </bo>
      ...
</bor>
```

Details of the XML tags:

Tag	Description
function	Name of the BAPI or RFC function, which should be called for this business object/verb combination, e.g. ITEM/CREATE.
return	Defines which parameter (-field) has to be checked for the determination if the transaction was successful or not.
result	Defines which export structures and tables should be returned as a result of the BAPI or RFC

	function call.
para	Name of the result parameter and tables, substructure of <result>.

Details of the XML tag return:

Attribute	Description	Values
code	Name of the parameter/parameter field which indicates the return code of the function or empty if the function uses EXCEPTIONS.	
message	Name of the parameter/parameter field which indicates the return message of the function or empty if the function uses EXCEPTIONS.	
commit	Defines whether external transaction commit is required (BAPI_TRANSACTION_COMMIT / BAPI_TRANSACTION_ROLLBACK) or not (For further information please read the documentation of the respective BAPI).	true false (=empty)
logging	Defines whether the function uses the old APIs for logging (CALO_INIT_API / CALO_LOG_READ_MESSAGES) or not.	true false (=empty)

Configuration inside SAP R/3

Overview

In general, it is not necessary to configure anything inside SAP R/3 if the standard BAPIs/RFCs or the ones shipped with the SAP Link are used. In some cases, it might be helpful though to call other BAPIs/RFCs or develop your own RFCs for satisfying special requirements. The chapters below will explain how to find and utilize existing RFCs and how to use the additional RFCs, which have been provided by Agile.

Additional Remote Function Calls (RFCs)

Below list provides an overview of the RFCs, which were shipped with the SAP link solution. Please see the Installation Manual for further information on how to load them into SAP R/3.

Following RFC functions are provided by Agile:

/EIGNER/BOM_ITEM_EFFECTIVITY

This RFC was developed in order to transfer a Bill of Materials to R/3, providing additional effectivity information on BOM position level.

First, the BOM effectivity (valid-from date) is determined. Then the item effectivity (valid-from date, valid-to date) is found out, both by the defined the priority.

If an old BOM exists, the old BOM is read at the first effectivity. In case of CAD indication, the old items are identified, which have to be deleted. For every item, the time schedule is detected by merging and comparing the old and new effectivity data. Starting with the first effectivity, the BOM is changed with the respective items at all effectivities via CSAP_MAT_BOM_MAINTAIN.

If an old BOM does not exist, the time schedule of every item is detected directly.

At the first effectivity, the BOM is created with the respective items via CSAP_MAT_BOM_CREATE. At the other effectivities, the BOM is changed with the respective items via CSAP_MAT_BOM_MAINTAIN.

Priority of BOM effectivity:

```
CHANGE_NO
REVISION_LEVEL,
VALID_FROM
```

Priority of item effectivity from:

```
T_STPO-CHANGE_NO
T_STOP-VALID_FROM
CHANGE_NO
REVISION_LEVEL
VALID_FROM
```

Priority of item effectivity to:

```
T_STPO-CHG_NO_TO
T_STOP-VALID_TO
Infinitive
```

For every BOM item the time schedule is determined by the found valid-from date and the found valid-to date

```
Parameter RETURNMESSAGES: return messages;
Parameter EFFECTIVITY_LOG: logging about every effectivity;
```

The other parameters are corresponding with CSAP_MAT_BOM_MAINTAIN.

For further information see CSAP_MAT_BOM_MAINTAIN.

/EIGNER/BOM_MULTI_EXPL

This RFC was provided in order to export a multi-level Bill of Materials from R/3.

Returns the multi-level BOM explosion (determination via CS_BOM_EXPL_MAT_V2) otherwise an error message is returned.

Parameter RETURN: error message;
the other parameters are corresponding with CS_BOM_EXPL_MAT_V2. For further information see CS_BOM_EXPL_MAT_V2.

/EIGNER/COMPLETE_BOM_CHANGE

This RFC was provided in order to transfer a Bill of Materials to R/3, providing additional effectivity information on BOM position level.

If the parameter FL_CAD is set, the old BOM is determined and the old BOM items with CAD indicator are compared with the new BOM items that are provided as parameters. The new BOM items are identified by the following fields of the table parameter T_STPO: ID_ITM_CTG, ID_ITEM_NO, ID_COMP_ID, ID_DOC, ID_DOC_TYP, ID_ODC_PRT, ID_DOC_VRS, ID_CLASS, ID_CLD_TYPE and ID_SORT. If an old item is not in the new BOM, then the old item will be added with deletion indicator to the new BOM. Afterwards the changes are done by CSAP_MAT_BOM_MAINTAIN. If the parameter FL_CAD is not set, CSAP_MAT_BOM_MAINTAIN is called directly.

Parameter RETURNMESSAGES: return messages;
the other parameters are corresponding with CSAP_MAT_BOM_MAINTAIN. For further information see CSAP_MAT_BOM_MAINTAIN.

/EIGNER/MATERIAL_CHANGE

This RFC was provided in order to only change a material in R/3. Creation of a new material cannot be done with this RFC.

First checks the existence of the material. If it exists, changes the material via BAPI_MATERIAL_SAVEDATA otherwise returns an error.

Parameters are corresponding with BAPI_MATERIAL_SAVEDATA. For further information see BAPI_MATERIAL_SAVEDATA.

/EIGNER/MATERIAL_CREATE

This RFC was provided in order to only create materials in R/3. Changing an existing material cannot be done with this RFC.

First checks the existence of the material. If it does not exist, it creates the material via BAPI_MATERIAL_SAVEDATA, otherwise returns an error.

Parameters are corresponding with BAPI_MATERIAL_SAVEDATA. For further information see BAPI_MATERIAL_SAVEDATA.

/EIGNER/MATERIAL_DETAILS

This RFC was provided in order to read the material details from R/3.

Returns the required material data, otherwise an error message is returned.

Import parameter:

- MATERIAL: material number;
- GET_MARA: determination of general material data;
- GET_MARC: determination of plant data for material;
- GET_MPOP: determination of forecast parameters;
- GET_MPGD: determination of change document structure for material/product group;
- GET_MARD: determination of storage location data for material;
- GET_MBEW: determination of material valuation;
- GET_MLGN: determination of material data for each warehouse number;
- GET_MVKE: determination of sales data for material;
- GET_MLGT : determination of material data for each storage type;
- GET_MAKT : determination of material descriptions;
- GET_MARM : determination of units of measure for material;
- GET_MEAN : determination of international article numbers (EANs) for material; GET_MLTX : determination of long texts;

- GET_MLAN : determination of tax classification for material;
- GET_MFHM : determination of production resource tool (PRT) fields in the material

Export parameter:

- RETURN : return message;
- EXP_MARA : general material data

Table parameter :

- TAB_MARC : plant data for material (export);
- TAB_MPOP : forecast parameters (export);
- TAB_MPGD : change document structure for material/product group (export);
- TAB_MARD : storage location data for material (export);
- TAB_MBEW : material valuation (export);
- TAB_MLGN : material data for each warehouse number (export);
- TAB_MVKE : sales data for material (export);
- TAB_MLGT : material data for each storage type (export);
- TAB_MAKT : material descriptions (export);
- TAB_MARM : units of measure for material (export);
- TAB_MEAN : international article numbers (EANs) for material (export);
- TAB_MLTX : long texts (export);
- TAB_MLAN : tax classification for material (export);
- TAB_MFHM : production resource tool (PRT) fields in the material (export)

Since SAP R/3 Version 4.7, the BAPI “BAPI_MATERIAL_GETALL” can be used for retrieving the material data.

/EIGNER/MAT_DOC_LINKS

This RFC was provided in order to update the link between an R/3 material and one or many document info records in one operation.

Coming from the material, the existing material-document-links are determined in SAP R/3 and the material-document-links are updated based on the document list provided as parameters. Old object links are deleted, which are not in this list. New object links are created in case of nonexistence. The determination of the existing material-document-links can be limited by a selection parameter for document types. The link document-plant material will be updated if a plant is given.

Import parameter:

- MATERIAL: material number;
- PLANT: plant;
- CHANGENUMBER: change number

Export parameter:

- RETURN: return message

Table parameter:

- DOCUMENTS: document list (document keys, import);
- SEL_DOKAR: selection parameter for document types (SELECT-OPTIONS, import)

/EIGNER/REV_LEVEL_MAINTAIN

This RFC was provided in order to create a new revision of a material in R/3.

This RPC ill be removed with the next release, since the wrapped CCAP_REV_LEVEL_MAINTAIN is a ‘Remote-enabled-module’ (see SAP-Note 523737).

/EIGNER/REV_LEVEL_SELECT

This RFC was provided in order to retrieve material revision information from R/3.

Returns the revision level of a material or document (selection via REVISION_LEVEL_SELECT), otherwise an error message is returned.

Parameter RETURN: error message;
the other parameters are corresponding with REVISION_LEVEL_SELECT. For further information see REVISION_LEVEL_SELECT.

/EIGNER/WHERE_USED_MAT

This RFC was provided in order to retrieve the single-level where-used list of a material in R/3.

Returns the where-used list of a material (determination via CS_WHERE_USED_MAT), otherwise an error message is returned.

Parameter RETURN: error message;
the other parameters are corresponding with CS_WHERE_USED_MAT. For further information see CS_WHERE_USED_MAT.

BAPI Explorer

The BAPI Explorer is a tool inside R/3 in order to query and display BAPIs (Business Application Programming Interface). In order to open the BAPI Explorer, just call the following transaction BAPI or open it via SAP menu -> Tools -> Business Framework -> BAPI Explorer.

In the BAPI Explorer, the tree browser on the left side shows the different areas for which the BAPIs are available. Just click on the tree nodes of your interest.

Once you have found the method/functionality which you are looking for, just click on the tree node in order to get the Detail information on the right side. The field Function module refers to the BAPI name, which you should call from the R/3 connector.

Function Builder

Once you have found the name of the BAPI (Function Module, as described in the previous chapter) or the RFC-Function, which you would like to use, you could use the Function Builder in order to look at the details of the BAPI/RFC-Function.

The Function Builder exposes all parameters and the source code of the respective function. You can call the Function Builder directly via transaction code SE37 or via SAP Menu -> Tools -> ABAP Workbench -> Development -> Function Builder.

In the Function Builder mask, in order to see the details of that respective function module, just click on the button Display.

- The tab Import shows the available import parameters of that function module.
- The tabs Export and Tables show the parameters, which can be returned to the R/3 connector:

By clicking on the reference type of a certain parameter, you get more detail information on that.

ABAP Dictionary

Once you have found the name of the reference type, you could use the ABAP Dictionary in order to look at the details of the reference type and the database table.

The ABAP Dictionary exposes all dictionary objects and their technical details. You can call the ABAP dictionary directly via transaction code SE11 or via SAP Menu -> Tools -> ABAP Workbench -> Development -> ABAP Dictionary.

In order to see the technical details of that respective structure, just click on the button Display.

In addition, create a second session and go to the ABAP dictionary (transaction code SE11).

In order to see the technical details of the corresponding Database table, click on the button Display.

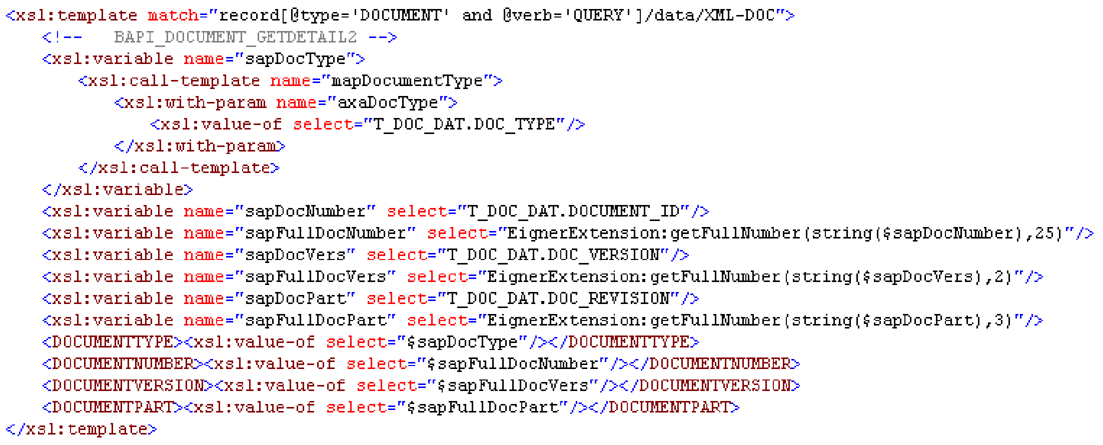
The component type of the structure component has to correspond to the field type of the transparent table field.

AND / OR

The short text of the structure component has to correspond to the short text of the transparent table field, e.g. the component ECNNUMBER of the structure BAPI_DOC_DRAW2 corresponds to the field AENNR of the transparent table DRAW.

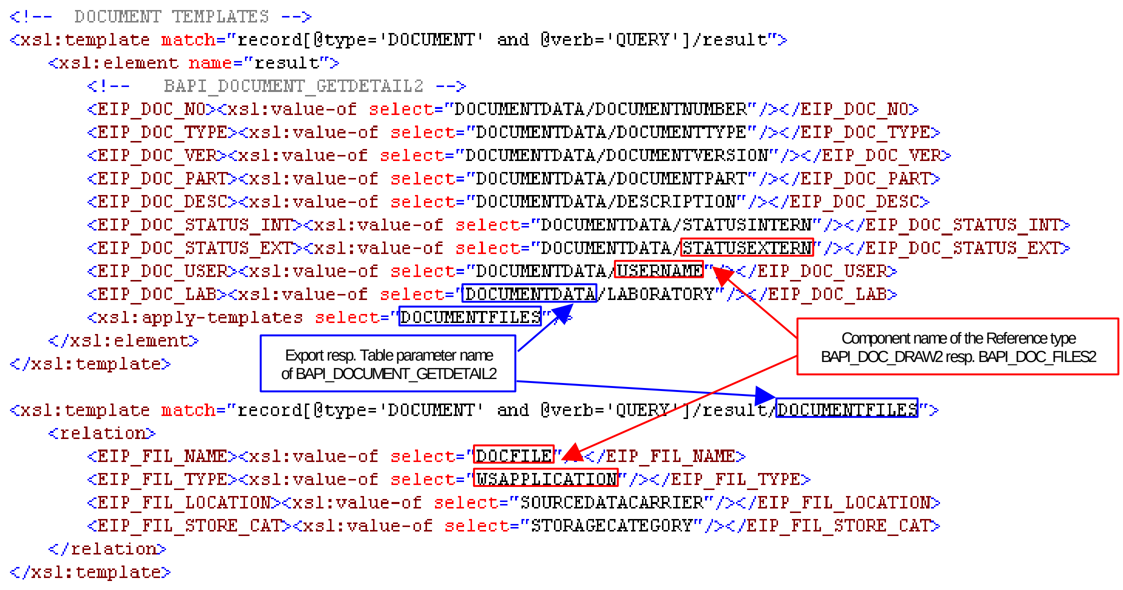
Mapping for BAPI_DOCUMENT_GETDETAIL2

Example 1: Input data from Agile EDM to SAP R/3



```
<xsl:template match="record[@type='DOCUMENT' and @verb='QUERY']/data/XML-DOC">
   <!--   BAPI_DOCUMENT_GETDETAIL2 -->
   <xsl:variable name="sapDocType">
      <xsl:call-template name="mapDocumentType">
         <xsl:with-param name="axaDocType">
            <xsl:value-of select="T_DOC_DAT.DOC_TYPE"/>
         </xsl:with-param>
      </xsl:call-template>
   </xsl:variable>
   <xsl:variable name="sapDocNumber" select="T_DOC_DAT.DOCUMENT_ID"/>
   <xsl:variable name="sapFullDocNumber" select="EignerExtension:getFullNumber(string($sapDocNumber),25)"/>
   <xsl:variable name="sapDocVers" select="T_DOC_DAT.DOC_VERSION"/>
   <xsl:variable name="sapFullDocVers" select="EignerExtension:getFullNumber(string($sapDocVers),2)"/>
   <xsl:variable name="sapDocPart" select="T_DOC_DAT.DOC_REVISION"/>
   <xsl:variable name="sapFullDocPart" select="EignerExtension:getFullNumber(string($sapDocPart),3)"/>
   <DOCUMENTTYPE><xsl:value-of select="$sapDocType"/></DOCUMENTTYPE>
   <DOCUMENTNUMBER><xsl:value-of select="$sapFullDocNumber"/></DOCUMENTNUMBER>
   <DOCUMENTVERSION><xsl:value-of select="$sapFullDocVers"/></DOCUMENTVERSION>
   <DOCUMENTPART><xsl:value-of select="$sapFullDocPart"/></DOCUMENTPART>
</xsl:template>
```

Example 2: Output data from SAP R/3 to Agile EDM



```
<!--  DOCUMENT TEMPLATES -->
<xsl:template match="record[@type='DOCUMENT' and @verb='QUERY']/result">
   <xsl:element name="result">
      <!--   BAPI_DOCUMENT_GETDETAIL2 -->
      <EIP_DOC_NO><xsl:value-of select="DOCUMENTDATA/DOCUMENTNUMBER"/></EIP_DOC_NO>
      <EIP_DOC_TYPE><xsl:value-of select="DOCUMENTDATA/DOCUMENTTYPE"/></EIP_DOC_TYPE>
      <EIP_DOC_VER><xsl:value-of select="DOCUMENTDATA/DOCUMENTVERSION"/></EIP_DOC_VER>
      <EIP_DOC_PART><xsl:value-of select="DOCUMENTDATA/DOCUMENTPART"/></EIP_DOC_PART>
      <EIP_DOC_DESC><xsl:value-of select="DOCUMENTDATA/DESCRIPTION"/></EIP_DOC_DESC>
      <EIP_DOC_STATUS_INT><xsl:value-of select="DOCUMENTDATA/STATUSINTERN"/></EIP_DOC_STATUS_INT>
      <EIP_DOC_STATUS_EXT><xsl:value-of select="DOCUMENTDATA/STATUSEXTERN"/></EIP_DOC_STATUS_EXT>
      <EIP_DOC_USER><xsl:value-of select="DOCUMENTDATA/USERNAME"/></EIP_DOC_USER>
      <EIP_DOC_LAB><xsl:value-of select="DOCUMENTDATA/LABORATORY"/></EIP_DOC_LAB>
      <xsl:apply-templates select="DOCUMENTFILES"/>
   </xsl:element>
</xsl:template>

<xsl:template match="record[@type='DOCUMENT' and @verb='QUERY']/result/DOCUMENTFILES">
   <relation>
      <EIP_FIL_NAME><xsl:value-of select="DOCFILE"/></EIP_FIL_NAME>
      <EIP_FIL_TYPE><xsl:value-of select="WSAPPLICATION"/></EIP_FIL_TYPE>
      <EIP_FIL_LOCATION><xsl:value-of select="SOURCEDATACARRIER"/></EIP_FIL_LOCATION>
      <EIP_FIL_STORE_CAT><xsl:value-of select="STORAGECATEGORY"/></EIP_FIL_STORE_CAT>
   </relation>
</xsl:template>
```

Export resp. Table parameter name
of BAPI_DOCUMENT_GETDETAIL2
Component name of the Reference type
BAPI_DOC_DRAW2 resp. BAPI_DOC_FILES2

Within <data> or <result> the first level is the name of the Import-, Export-, Table-Parameter and the second level is the name of the component or field of the reference type (structure or table).

Note If a parameter, a component or a field starts with a digit (0-9) respective contains slash '/') a replacement has to be done to be XML compliant. The SAP Connectors replace these signs in case of creating the XML tags or reset the replacements in case of reading the XML tags. The slash is replaced by the string '_-_' and the digit is by the string '_--XX', whereas the XX is the hexadecimal code in the ASCII table. Examples: 2STEP_PICK ↔ _--32STEP_PICK; /KKK/NNN ↔ _-_KKK_-_NNN.

Chapter 3

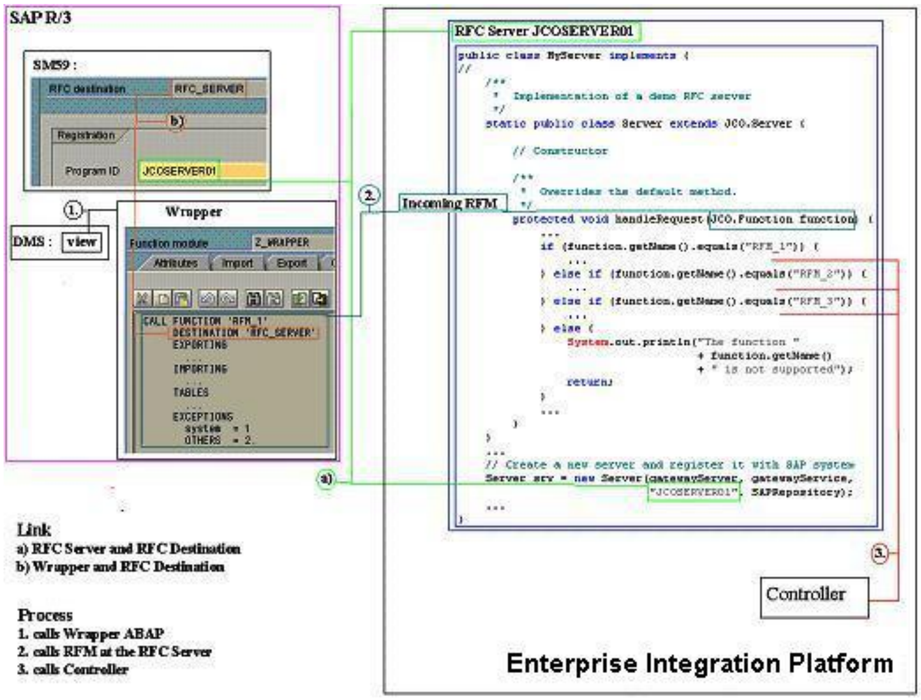
Synchronous SAP Connector

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.

Below is an excerpt of the communication process between SAP R/3 (RFC client!) and the EIP SAP Source Connector (acts as an RFC server!):



```
CALL FUNCTION 'RFM_1'
     DESTINATION 'RFC_SERVER'
     EXPORTING
       ...
     IMPORTING
       ...
     TABLES
       ...
     EXCEPTIONS
       system  = 1
       OTHERS  = 2.
```

```
public class MyServer implements {
//
    /**
     *  Implementation of a demo RFC server
     */
    static public class Server extends JCO.Server {

        // Constructor

        /**
         *  Overrides the default method.
         */
        protected void handleRequest(JCO.Function function) {
            ...
            if (function.getName().equals("RFM_1")) {
                ...
            } else if (function.getName().equals("RFM_2")) {
                ...
            } else if (function.getName().equals("RFM_3")) {
                ...
            } else {
                System.out.println("The function "
                                   + function.getName()
                                   + " is not supported");
                return;
            }
            ...
        }
    }
    ...
    // Create a new server and register it with SAP system
    Server srv = new Server(gatewayServer, gatewayService,
                            "JCOSERVER01", SAPRepository);
    ...
}
```

Link
a) RFC Server and RFC Destination
b) Wrapper and RFC Destination

Process
1. calls Wrapper ABAP
2. calls RFM at the RFC Server
3. calls Controller

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.)

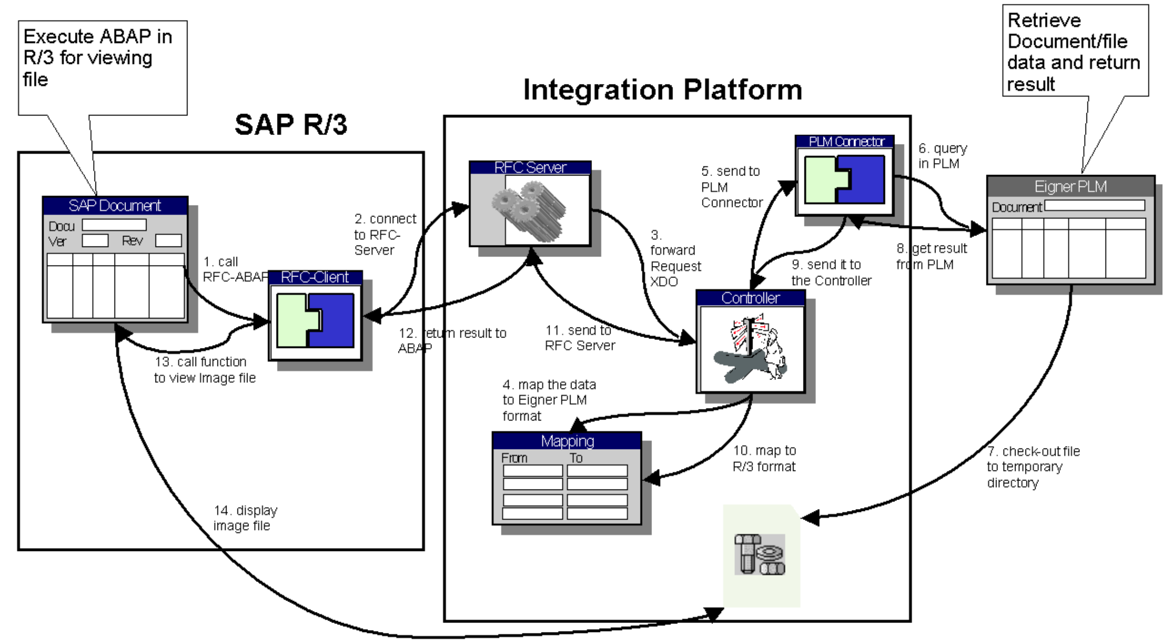
Chapter 7: design the mapping files e.g. for mapping the XML message going from R/3 to Agile EDM and back

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.

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)	

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".

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.

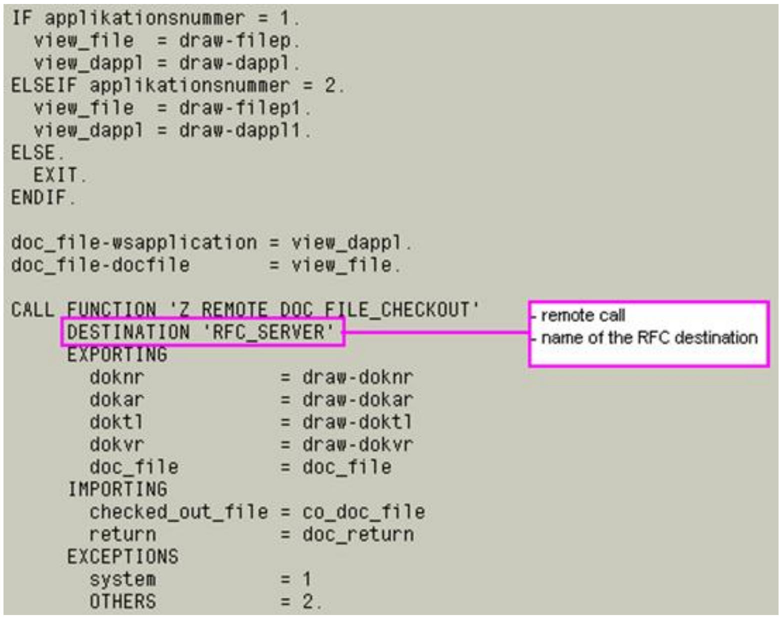
Example: DMS: the Remote Function Module checks out the required file and returns the name of the checked out file.

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 IExplorer.

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

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.

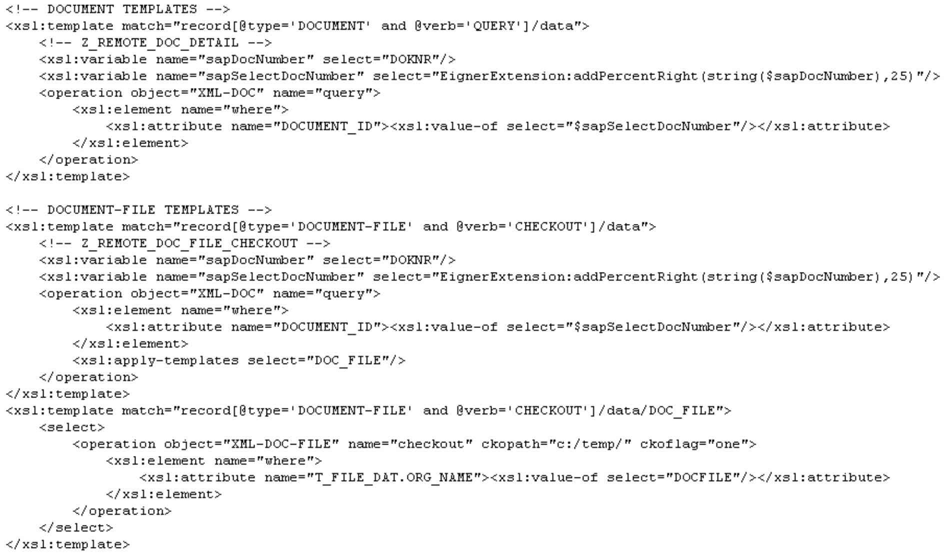
   Here you assign the ABAP wrapper to the used data carrier type and the application type 1 (view).

   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.

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.

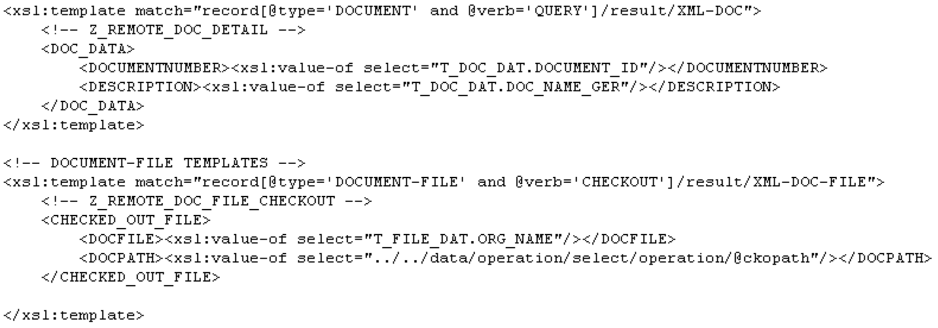
Mapping Example SAP R/3 --> Agile EDM: sync_r3_axa_request.xsl



```
<!-- 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



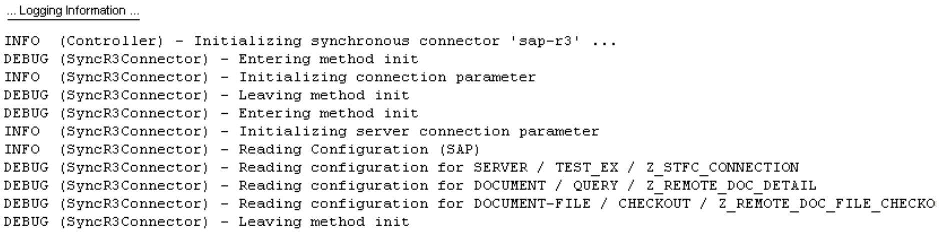
```
<xsl:template match="record[@type='DOCUMENT' and @verb='QUERY']/result/XML-DOC">
    <!-- Z_REMOTE_DOC_DETAIL -->
    <DOC_DATA>
        <DOCUMENTNUMBER><xsl:value-of select="T_DOC_DAT.DOCUMENT_ID"/></DOCUMENTNUMBER>
        <DESCRIPTION><xsl:value-of select="T_DOC_DAT.DOC_NAME_GER"/></DESCRIPTION>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

Loggin Information

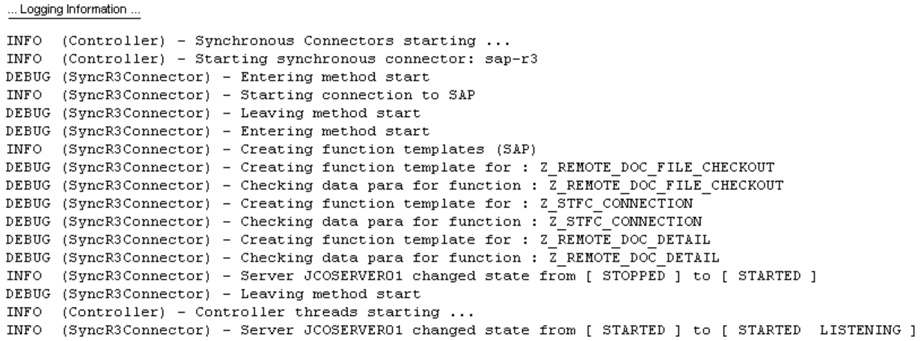
These are sample logging information from the RFC server (excerpts shown in the example below).



... 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
```

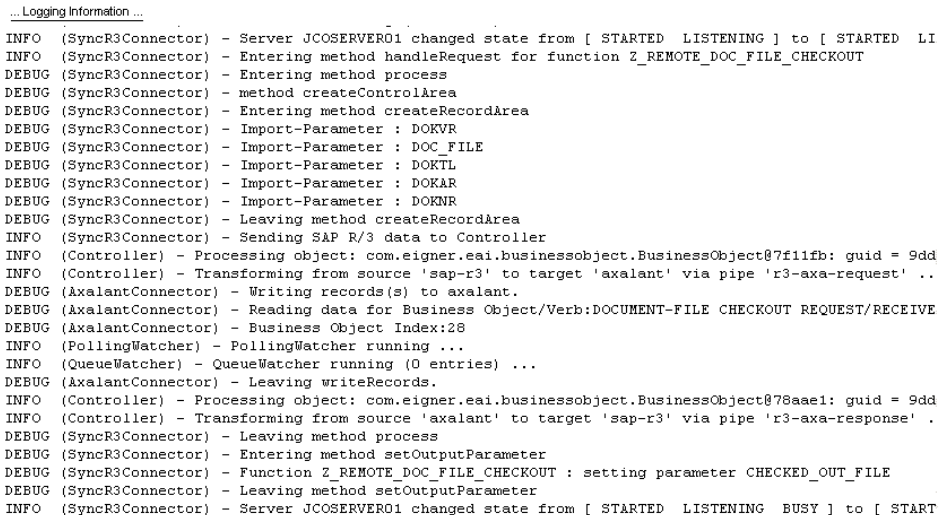
Initialization of the SAP Source Connector



... 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

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.

Chapter 4

Quick Start on SAP Link

Installation and Configuration

Below is an overview of the steps, which you may have to go through, in order to install and configure the SAP link for initial use. The scenario is based on the assumption, that Agile EDM is the source system and SAP R/3 is the target system of the data transfer. Implementation of other scenarios (e.g. using other connectors) would look similar, though.

Install the software as described in the Installation Manual

6. Setup the controller and the Admin Tools
   - Set up the <controller> area in eai_ini.xml (e.g. polling rate and trace level)
   - Set up the <admin> area in eai_ini.xml (e.g. queue port and locale)
   - Use the encryption tool (crypt.cmd/.sh, encrypt.cmd/sh) to generate encrypted passwords for the Agile EDM and R/3 connectors
   - Set up the connection parameters to the queue database in <controller/queue> and <admin/queue>and run the dbmaint.cmd/.sh tool
7. Setup Workflow
   - Activate the workflow Agile EDM to R/3 in eai_ini.xml (<workflow name=”plm-sap” active=”true”>)
8. Setup connectors
   - Agile EDM connector section in eai_ini.xml

     Set up connection parameters, e.g. host, application, user and encrypted password

     Set up allowed business objects, verbs and used schemas (IEF / EXI) in bor_plm.xml
   - R/3 connector section in eai_ini.xml

     Set up connection parameters for connecting to the R/3 system in eai_ini.xml

     Set up allowed business objects, verbs, used BAPIs/RFCs in bor_r3.xml
   - Agile EDM synchronous connector section in eai_ini.xml:

     Set up connection parameters (synchronous PLM connector host and port) inside Agile EDM

     Set up business objects, verbs and used entities/masks in bor_plm_sync.xml
   - Test run EIP with test.cmd/.sh to check the connections
9. Setup inside SAP R/3
   - Set up special interface user and respective privileges (if required)
   - Query for required BAPIs and RFC functions in the BAPI Explorer and check parameters,

if out-of-the-box solution is not sufficient
- Test run the BAPIs/RFCs with R/3 Function Builder (Transaction se37)

10. Setup inside Agile EDM
    - Configure target sites (if used) for pointing to the target connector
    - Configure transfer selections (LGV calls) in masks menus and adapt the parameters, e.g. Create Item in Item Navigator
    - Develop and integrate OnSuccess/OnError LogiView procedures, which will be called from the transfer queue after the transfer operation
    - Set up the Display masks for synchronous communication (e.g. SAP Material mask) in order to show the result data from R/3
    - Set up the XML Interface
    - Set up the transfer schemas in the IEF setup screen
    - Set up the XML interface objects for complex queries, predefined queries and file check-in/check-out
11. Mapping Configuration
    - Configure Agile EDM to R/3 mapping file axa_r3.xsl (e.g. attribute and structure mapping) for transformation of the messages going from Agile EDM to SAP R/3.
    - Configure R/3 to Agile EDM mapping file r3_axa.xsl (e.g. result and message mapping) for transformation of the messages going from SAP R/3 to Agile EDM.
12. Test-run the complete scenario, e.g. “Create Item”
    - Start up the manager via the “manager.cmd/.sh” script
    - Kick off the transfer in Agile EDM, i.e. select “Replication -> Item -> Create Item” in Item Navigator
    - Check the content of the Agile EDM transfer queue with the Admin GUI
    - Check the logging information in the file <eai.home>/log/eai.log
    - Check the mapping and result in the debug files in /tmp directory, e.g. bo_request.xml or bo_response.xml (only if <trace-level> DEBUG was activated in eai_ini.xml)
    - Check if the new material was created inside R/3
    - Check the return code and message in the Agile EDM transfer queue

There may be - of course - more steps required before you can use the Integration Platform in a production environment, e.g. load balancing the Integration Platform, setting the reconnect option in the respective connector and e-mail notification. However, these steps may change from customer to customer based on the specific requirements.

Release Agile EDM Work Set to SAP

One of the new features in Agile EDM is the Enhanced Change Management. This allows easier planning and execution of changes against various objects in EDM like items, BOMs and documents. One vehicle to keep track of all these planned and executed changes is the “Work Set”. The Work Set is a container for putting all changes together in one bucket. Upon release of the Work Set, the planned change operations will be activated.

The SAP link comes with a set of predefined processes and configurations in order to transfer such released Work Sets including affected objects to SAP R/3.

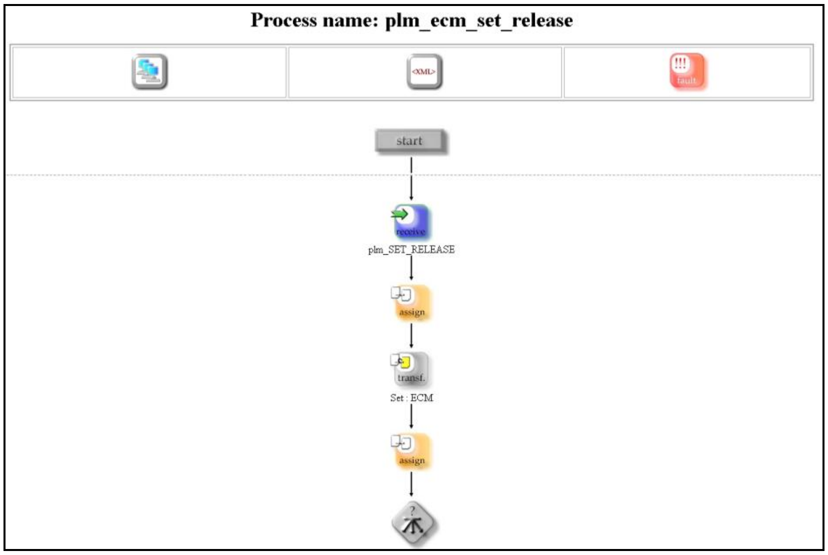
Below is an overview of the steps, which you may have to go through in order to activate the process and configuration for exporting Agile EDM Work Sets to SAP R/3. The scenario is based on the assumption, that Agile EDM (standard configuration) is the source system and SAP R/3 is the target system of the data transfer.

Load the binary loader files *e6_eip_exp_chg_set.bld* from the directory *<eai_home>/install/plm6/loader*. This loader contains additional masks, IEF definitions, XML interface object definition and menu selections for retrieving and exporting a Work Set.

1. Load the binary loader files e6_eip_bpm_history.bld from the directory <eai_home>/install/plm6/loader. This loader contains the EIP history table.
2. Log off from Agile EDM and log on (log in?) again. Then create the table T_EIP_HIS.
3. Activate the following connectors in your configuration file eai_ini.xml:

Agile EDM Connector:	plm
SAP R/3 Connector:	sap-r3
BPM Connector:	bpm
Mail Connector:	mail

4. Activate the following workflows in your configuration file:
    - <workflow name=”plm-bpm” active=”true”>)
    - <workflow name=”bpm-mail” active=”true”>)
    - <workflow name=”bpm-sap” active=”true”>)
5. Check the BPM process
    - The BPM process being used for transferring the Work Set from Agile EDM to SAP R/3 is called bpel_plm_ecm_set_release.xml. This process can be found in the directory <eai_home>/conf/bpm (along with the necessary mapping files).
    - Please adapt the process to your needs.
    - Alternatively you can check the process in the Web-Browser after converting it to HTML with the BPM-Converter:



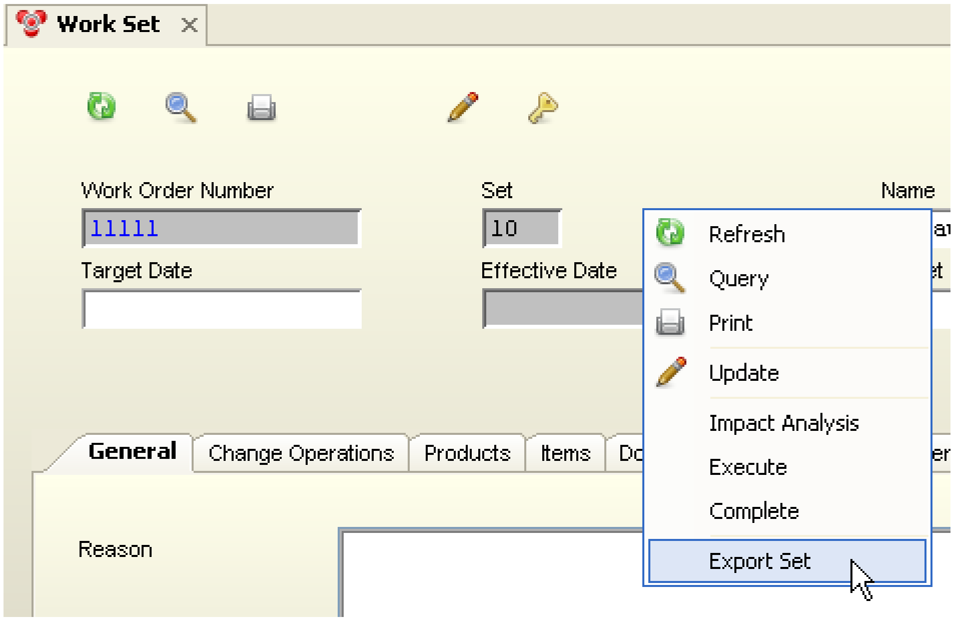
Process name: plm_ecm_set_release

- The process roughly executes following steps:
  1. Checks the existence of (a/the) change master in R/3 and if not, creates the change master.
  2. Gets the affected objects and checks which operation has to be performed in the R/3 system.
  3. Loops through all items, documents, item-document links, BOMs and raw materials and decides whether to create, update or inactivate them in R/3 and then perform the operation in R/3.
  4. Updates the EIP transfer history in Agile EDM with the outcome of the transactions above.

     Returns the overall status back to Agile EDM, which is written to the EIP transfer queue.

6. Starts up the Enterprise Integration Platform / SAP link and checks the trace output whether the process bpel_plm_ecm_set_release.xml was read in correctly by the BPM engine.
7. Kicks off the transfer from Agile EDM

- Creates your Work Set in Agile EDM. Following affected objects are supported by the BPM process:
  Items (T_MASTER_DAT)
  BOMs (T_MASTER_STR)
  Item-Document relations (T_MASTER_DOC)
  Documents (T_DOC_DAT)
  Change Master (T_WRK_SET_DAT)
- Starts the export of the Work Set by selecting the following menu entry on the form of the Work Set:



Work Order Number: 11111
Set: 10

- The following new entry is then created in the EIP transfer queue:

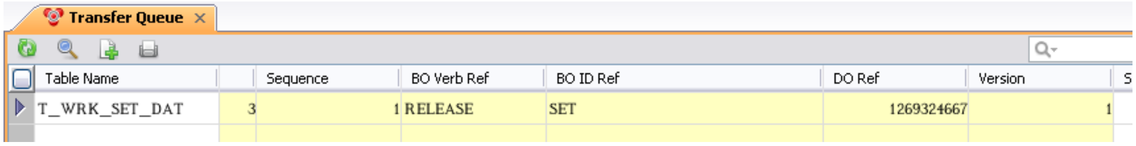
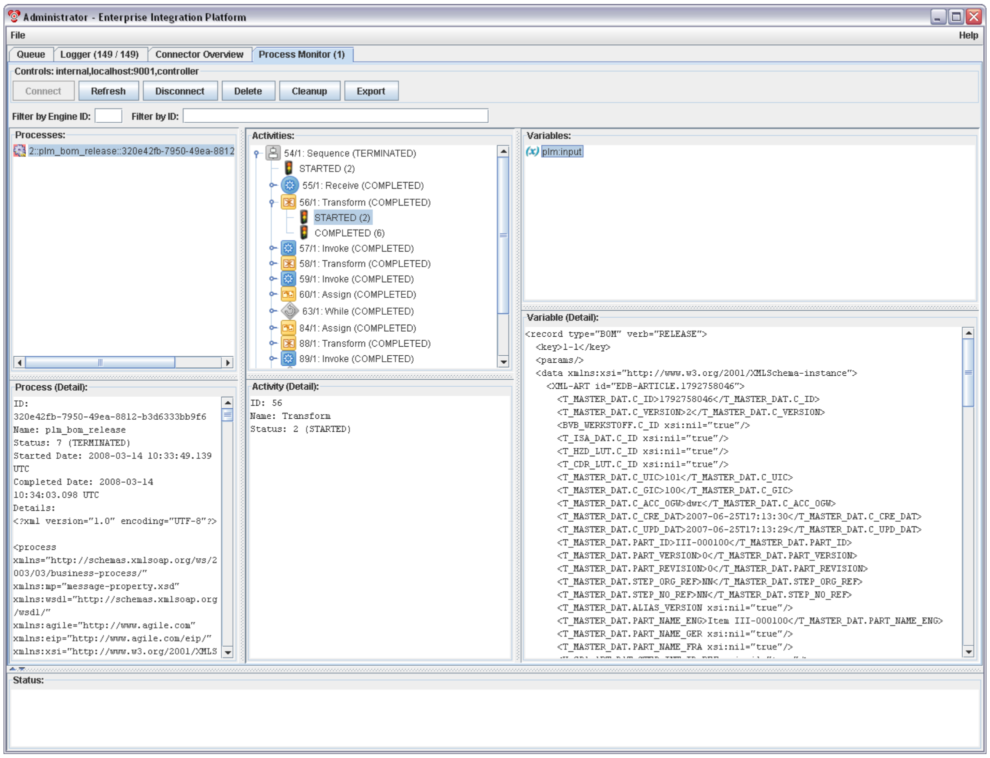


Table Name		Sequence	BO Verb Ref	BO ID Ref	DO Ref	Version	S
T_WRK_SET_DAT	3	1	RELEASE	SET	1269324667	1	

8. Checks the process in the process monitor.
   - The process monitor as part of the EIP Admin GUI keeps track of all processes which have been run through. Please check whether your process was run correctly.



Controls: internal,localhost:9001,controller

Processes: 2::plm_bom_release::320e42fb-7950-49ea-8812

Activities:

```
54/1: Sequence (TERMINATED)
  STARTED (2)
  55/1: Receive (COMPLETED)
  56/1: Transform (COMPLETED)
    STARTED (2)
    COMPLETED (6)
  57/1: Invoke (COMPLETED)
  58/1: Transform (COMPLETED)
  59/1: Invoke (COMPLETED)
  60/1: Assign (COMPLETED)
  63/1: While (COMPLETED)
  84/1: Assign (COMPLETED)
  88/1: Transform (COMPLETED)
  89/1: Invoke (COMPLETED)
```

Variables: (x) plm:input

Process (Detail):

```
ID:
320e42fb-7950-49ea-8812-b3d6333bb9f6
Name: plm_bom_release
Status: 7 (TERMINATED)
Started Date: 2008-03-14 10:33:49.139
UTC
Completed Date: 2008-03-14
10:34:03.098 UTC
Details:
<?xml version="1.0" encoding="UTF-8"?>

<process
xmlns="http://schemas.xmlsoap.org/ws/2
003/03/business-process/"
xmlns:mp="message-property.xsd"
xmlns:wsdl="http://schemas.xmlsoap.org
/wsdl/"
xmlns:agile="http://www.agile.com"
xmlns:eip="http://www.agile.com/eip/"
xmlns:xsi="http://www.w3.org/2001/XMLS
```

Activity (Detail):

```
ID: 56
Name: Transform
Status: 2 (STARTED)
```

Variable (Detail):

```
<record type="BOM" verb="RELEASE">
  <key>1-1</key>
  <params/>
  <data xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <XML-ART id="EDB-ARTICLE.1792758046">
      <T_MASTER_DAT.C_ID>1792758046</T_MASTER_DAT.C_ID>
      <T_MASTER_DAT.C_VERSION>2</T_MASTER_DAT.C_VERSION>
      <BVB_WERKSTOFF.C_ID xsi:nil="true"/>
      <T_ISA_DAT.C_ID xsi:nil="true"/>
      <T_HZD_LUT.C_ID xsi:nil="true"/>
      <T_CDR_LUT.C_ID xsi:nil="true"/>
      <T_MASTER_DAT.C_UIC>101</T_MASTER_DAT.C_UIC>
      <T_MASTER_DAT.C_GIC>100</T_MASTER_DAT.C_GIC>
      <T_MASTER_DAT.C_ACC_OGW>dwr</T_MASTER_DAT.C_ACC_OGW>
      <T_MASTER_DAT.C_CRE_DAT>2007-06-25T17:13:30</T_MASTER_DAT.C_CRE_DAT>
      <T_MASTER_DAT.C_UPD_DAT>2007-06-25T17:13:29</T_MASTER_DAT.C_UPD_DAT>
      <T_MASTER_DAT.PART_ID>III-000100</T_MASTER_DAT.PART_ID>
      <T_MASTER_DAT.PART_VERSION>0</T_MASTER_DAT.PART_VERSION>
      <T_MASTER_DAT.PART_REVISION>0</T_MASTER_DAT.PART_REVISION>
      <T_MASTER_DAT.STEP_ORG_REF>NN</T_MASTER_DAT.STEP_ORG_REF>
      <T_MASTER_DAT.STEP_NO_REF>NN</T_MASTER_DAT.STEP_NO_REF>
      <T_MASTER_DAT.ALIAS_VERSION xsi:nil="true"/>
      <T_MASTER_DAT.PART_NAME_ENG>Item III-000100</T_MASTER_DAT.PART_NAME_ENG>
      <T_MASTER_DAT.PART_NAME_GER xsi:nil="true"/>
      <T_MASTER_DAT.PART_NAME_FRA xsi:nil="true"/>
```

9. Checks the process history inside Agile EDM
   - Independent of success or failure of the BPM process, the process history will be written back to Agile EDM. Please open the Process History mask via Manager -> Integration Platform -> Transfer History and query for the transfer ID, which was used initially for exporting the Work Set.

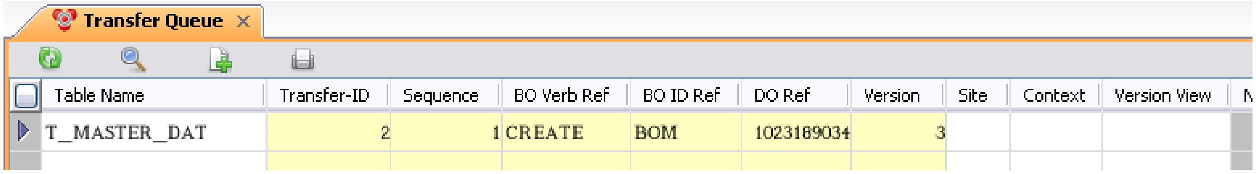


Table Name	Transfer-ID	Sequence	BO Verb Ref	BO ID Ref	DO Ref	Version	Site	Context	Version View	N
T_MASTER_DAT	2	1	CREATE	BOM	1023189034	3				

Chapter 5

Appendix

Installation Checklist

SAP R/3 administration requirements

Please check:

- SAP R/3 server can be contacted from Agile EDM server machine via ping command.
- You have a development license for the SAP R/3 system.
- The access to the SAP system via SAP GUI is available and the SAP GUI and Agile EDM GUI can be used from the same display.
- During the installation of the Integration Platform, it must be possible to create test data in the SAP system (material, bill of material, documents etc.).
- The SAP administrator has the following knowledge/privileges:
- Create a new (dialog) user and assign certain privileges as described in chapter 3.
- Create a new user with ABAP/4 development and customizing rights. These rights are necessary for the installation of the additional RFC-Functions provided by Agile.

Please list SAP R/3 system name, system version and the server node name, which should be used with the Integration Platform:

- R/3 System Name: ____________________________________
- R/3 System Version: ____________________________________
- Server Node Name: ____________________________________

Checklist filled out by:	
Name:	____________________
Phone:	____________________
Fax:	____________________
Email:	____________________