Agile
Enterprise Integration Platform

ORACLE

Upgrade Manual

Enterprise Integration Platform 2.1.2
SAP-Link 4.1.2

Part No. E11177-01

Make sure you check for updates to this manual at the
Oracle Technology Network Website

Copyrights and Trademarks

December 03, 2007

REVISIONS

Revision	Date	Pages Affected	Description
A	2005-07-04	All	Release for EIP 2.1.0
B	2007-02-16	All	Release for EIP 2.1.1
C	2007-12-03	All	Release for EIP 2.1.2

PREFACE

The Agile documentation set includes Adobe® Acrobat™ PDF files. The Oracle Technology Network (OTN) Web site (http://www.oracle.com/technology/documentation/index.html) contains the latest versions of the Oracle|Agile PLM 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|Agile Documentation folder available on your network from which you can access the Oracle| Agile 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).

The Oracle Technology Network (OTN) Web site (http://www.oracle.com/technology/documentation/index.html) can be accessed through **Help** > **Manuals** in both the Agile Web Client and the Agile Java Client. If applicable, earlier versions of Oracle|Agile PLM documentation can be found on the Agile Customer Support Web site (http://www.agile.com/support).

If you need additional assistance or information, please contact support@agile.com or phone (408) 284-3900 for assistance.

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

Readme

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

Agile Training Aids

Go to the Agile Training Web page (http://training.agile.com) for more information on Agile Training offerings.

Chapter 1
Overview

This document should guide through an upgrade from previous versions of the Enterprise Integration Platform.

Its chapters describe the upgrade from one major version to the next one, e.g. from 2.0 to 2.1.

For changes in minor versions, please refer to the respective Release Notes documents.

Chapter 2
Upgrade Tool

The Upgrade tool allows to upgrade previous versions of the eai_ini.xml file to the current EIP version.

The tool can be started with the script upgrade.cmd (Windows) and upgrade.sh (UNIX) in the *bin* directory.

The following startup options are available (you will get this by adding the --help option to the startup script):

```
Usage: Enterprise Integration Platform Upgrade [-c <conf-dir> ] [-h] -i <in> -o <out> [-p <props-
file> ] [-x <xsl> ]

Options:
   -c | --conf-dir     Specifies the configuration directory
   -h | --help         Shows this help
   -i | --in           Input file (REQUIRED)
   -o | --out          Output file (REQUIRED)
   -p | --props-file   Specifies the properties file
   -x | --xsl          XSL file (default: upgrade.xsl)
```

This is a example on how the tools might be called on Windows:

```
bin\upgrade.cmd –i C:\eigner\eip-old\conf\eai_ini.xml –o C:\eigner\eip-new\conf\eai_ini.xml
```

The Upgrade tool may provide the following output:

```
[<date>] FORCE (Upgrade) - Input file (2.1.1) : C:\eigner\eip-old\conf\eai_ini.xml
[<date>] FORCE (Upgrade) - Transformation file: C:\eigner\eip-new\conf\upgrade.xsl
[<date>] FORCE (Upgrade) - Output file (2.1.2): C:\eigner\eip-new\conf\eai_ini.xml
[<date>] FORCE (Upgrade) - Transformation done in 0 h 00 min 00 s 297 ms
```

Chapter 3
EIP 2.0 to 2.1

General information

This document should help you with the installation of the Enterprise Integration Platform Version 2.1 on top of the older version 2.0. It should replace neither the EIP 2.1 installation guide nor the EIP 2.1 Release Notes.

Please keep always in mind, that additional customizing of the Integration Platform (e.g. XSL Mapping) and Agile e6 (e.g. additional Query Forms) will not be upgraded automatically.

Installation

Due to the fact that quite some libraries and configuration files changed between the versions 2.0 and 2.1 we recommend installing EIP 2.1 in a directory separate from EIP 2.0. Additional mapping files and configuration should be incorporated into the EIP 2.1 installation one by one.

Configuration and Customizing

Configuration Settings

The structure of the configuration file eai_ini.xml has changed. Therefore, please copy your 2.0 configuration settings carefully one by one to the 2.1 eai_ini.xml file.

> **Note:** The usage of the Upgrade tool is highly recommended although some manual work might be required.

Following portions of the eai_ini.xml file have changed:

- New sections have been added for following connectors
  - SOAP Connector
  - Web-Service Connector
  - HTTP Connector
  - XML-RPC Connector
- New <transformation> elements have been added to following connectors:
  - Mail Connector
  - ftp Connector
  - Data File Connector
- New <id> elements have been added to the <queue> sub-sections for the EIP database connector (in <controller> and <admin> sections)
- New <engine-id> element has been added to the BPM connector sections for storing the processes in the EIP database.
- The axalant and PLM connectors have a new element <queue-mask> to define which mask should be used for retrieving transfer queue entries

- The PLM connector has a new element <snapshot> in order to activate or deactivate the snapshot functionality.
- The SAP R/3 connector has the new elements <rfc-trace> and <jco-trace-level> for defining the trace levels, which should be logged in external files.
- The SAP R/3 source connector (synchronous) has a new attribute whether the R/3 system supports Unicode or not (@unicode="true")
- The <controller> section has following changes:
  - <webserver>: added to define the port how to connect to the webserver embedded in EIP
  - <tasks>: allows to define certain tasks for the controller e.g. <cleanup-task>
  - <feature>: allows to define the new feature (@persist-synchronous) whether XML messages of synchronous calls should be stored in the EIP database
- the element <log/host> has been removed
- The admin section has following new entries:
  - <admin-logger-host>: defines where to retrieve the new log entries from the EIP server

Agile e5 and e6 Data Model

Due to some functional enhancements, the content of the Agile e5 and e6 Loader Files (separate loader files) have been changed in many ways. We cannot provide a general recommendation, whether the new loader files should be loaded in Insert mode or Overwrite or even loaded in a separate environment in order to perform the changes manually. This mainly depends on the number of modifications, which were done since the initial installation of EIP 2.0. Below list should provide you some hints what needs to be considered:

- New fields have been added to the Transfer Queue table (e5 and e6) with regards to supporting Preliminary Records and Phase Filters:
  - T_EER_SEN.PRELIMINARY_FLAG
  - T_EER_SEN.PHASE_FLAG
  - T_EER_SEN. PHASE_OPERATOR
  - T_EER_SEN. PHASE_ID
- Above new fields and some more have been added to the Transfer Queue form (EDB-EIP-SEN-SLI):
  - T_EER_SEN.PRELIMINARY_FLAG
  - T_EER_SEN.PHASE_FLAG
  - T_EER_SEN. PHASE_OPERATOR
  - T_EER_SEN. PHASE_ID
  - T_PHA_OPR_LUT.EDB_NAME
  - T_PHA_LUT.EDB_NAME
- The join definition of the Transfer Queue form (EDB-EIP-SEN-SLI) has been changed: *T_EER_SEN.VIEW_VER=T_EIP_VIEW.VIEW_ID(+)&T_EER_SEN.PHASE_ID=T_PHA_LUT.EDB_ID(+)& T_EER_SEN.PHASE_OPERATOR=T_PHA_OPR_LUT.EDB_OPER(+)*
- A snapshot table T_EIP_SNAPSHO has been replaced with following objects:
  - Entity: EDB-EIP-SNAPSHOT (table name: T_EIP_SNAPSHT)
  - Entity Relation between transfer queue (EDB-EER-SEND) and Snapshot entity (EDB-EIP-SNAPSHOT) with table T_EIP_SNP_STR.

Two loader files have been provided for Agile e6 only, which could be loaded optionally:

- e6_eip_bpm_history.bld: BPM History Option for storing back the results of BPM processes
- e6_eip_exp_chg_set.bld: Work Set Export Option for transferring complete Works Sets (part of new Change Management in Agile e6) to R/3

They lead to following additional objects in Agile e6 (on top of the changes already described above):

BPM History Option:

- New entity EDB-EIP-HISTORY (table T_EIP_HIS)
- New mask EDB-EIP-HIS-SLI
- New fields have been added to the Transfer Queue table (e5 and e6) with regards to supporting Preliminary Records and Phase Filters:
  - T_EER_SEN.PRELIMINARY_FLAG
  - T_EER_SEN.PHASE_FLAG
  - T_EER_SEN. PHASE_OPERATOR
  - T_EER_SEN. PHASE_ID

Work Set Export Option:

- New IEF Definitions for exporting Work Sets including affected objects e.g. Item, BOM, etc.
- New EXI Interface Object EXI-CHG-SET
- New mask and link definitions required for exporting the Work Set data

Agile e5 and e6 Connector

The libraries for connecting to the Synchronous PLM Connector have been changed. Therefore, please replace the old eipsync shared libraries with the new one in the directory axalant/bin/<operating-system>.

EIP Message Queue

Some tables of the EIP message queue have changed due to additional features supported by the EIP. We recommend using a new (separate) queue schema for EIP 2.1 and rerunning the dbmaint tool.

Business Process Engine

The Business Process Engine comes with additional and changed predefined processes:

- <eai_home>/conf/bpm/bpel_plm_ecm_set_release.xml: for transferring Work Sets from Agile e6 to SAP R/3.
- <eai_home>/conf/bpm/bpel_plm_bom_release.xml: optimized for transferring Agile e6 BOMs to SAP R/3.

Business Process Viewer

A tool has been provided which allows to converted BPEL based process definitions into HTML. These HTML files could be viewed in the Internet Explorer in order to have the process displayed in a graphical way. A new script bpmconvert.cmd/ bpmconvert.sh has been provided in the <eai_home>/bin directory therefore.

EIP Embedded Web Server

In order to save as a basis for the SOAP Connector, Web-Service Connector and HTTP Connector, a Web-Server has been provided along with the EIP Installation. In the directory <eai_home>/data following new sub-directories have been added:

- html: contains the image files relevant for the BPM Viewer
- webapps: contains the web services provided through the AXIS framework

Chapter 4
EIP 1.2 to 2.0

General information

This document should help you with the installation of the Enterprise Integration Platform Version 2.0 on top of the older version 1.2. It should neither replace the EIP 2.0 installation guide nor the EIP 2.0 Release Notes.

Please keep always in mind, that additional customizing of the Integration Platform (e.g. XSL Mapping) and Eigner PLM (e.g. additional Query Forms) will not be upgraded automatically.

Installation

Due to the fact, that a lot of libraries and configuration files changed between the versions 1.2 and 2.0 , we recommend to install EIP 2.0 in a directory separate from EIP 1.2. Additional mapping files and configuration should be incorporated into the EIP 2.0 installation one by one.

Note: The usage of the Upgrade tool is highly recommended although some manual work might be required.

Configuration and Customizing

Configuration Settings

The structure of the configuration file eai_ini.xml has changed. Therefore please copy your 1.2 configuration settings carefully one by one to the 2.0 eai_ini.xml file.

Following portions of the eai_ini.xml file have changed:

- new sections have been added for following Connector
  - BPM Connector (Business Process Manager)
  - Mail Connector
  - JDBC (Database) Connector
  - FTP Connector
- the <bor> sections of the PLM and SAP Connector have been extracted into separate files
- new <feature> elements have been added for the PLM and SAP Connector to configure whether to connect to the systems on-demand.
- new <connection> elements have been added to several connectors in order to more easily activate or deactivate connection parameters.
- new <queue> elements have been added to more easily activate or deactivate the queue connection parameters
- a new <encoding> element has been added to the controller section in order to define the encoding for newly generated XML messages in EIP

Eigner PLM Data Model

Due to some functional enhancements, the content of the Eigner PLM Loader Files have been changed in many ways. We cannot provide a general recommendation, whether the new loader files should be loaded in Insert mode

or Overwrite or even loaded in a separate environment in order to perform the changes manually. This mainly depends on the number of modifications which were done since the initial installation of EIP 1.2. Below list should provide you some hints what needs to be considered:

- A new field hase been added to the Transfer Queue table (T_EER_SEN) and form (EDB-EIP-SEN-SLI) for Snapshot Functionality: T_EER_SEN.SNAPSHOT_FLAG
- Also the sequence of the fields in T_EER_SEN has been changed and the index EER_SEN_IND has been optimized.
- The view (V_EIP_VIE) has been removed and a new table (T_EIP_VIEW) has been provided instead for the version view support.
- The mask of the Transfer Queue EDB-EIP-SEN-SLI has been modified to support above changes:
  - new field T_EIP_VIEW.VIEW_NAME instead of V_EIP_VIE.C_NAME
  - changed JOIN definition on the mask: T_EER_SEN.VIEW_VER=T_EIP_VIEW.VIEW_ID(+)
- A new table (T_EIP_SNAPSHO) has been provided for storing snapshots. This table is related to the table T_EER_SEN via a REFINE definition in DataView.
- Additional LGV procedures have been provided:
  - EP_REPLICATION/RPL_CreateSnapshot
  - EP_REP_QUEUE/ShowSnapshot

Eigner PLM Connector

The libraries for connecting to the Synchronous PLM Connector have been changed. Therefore please replace the old eipsync dll´s or libraries with the new one in the directory axalant/bin/<operating-system>.

EIP Message Queue

Some tables of the EIP message queue have changed due to improvements in performance. We recommend using a new (separate) queue schema for EIP 2.0 and rerunning the dbmaint tool.

Business Process Engine

A new Business Process Engine has been provided with this release which introduces several changes:

- a new directory has been provided for storing the process definitions: <eai_home>/conf/bpm
- the new Business Process Connector was added to the configuration file eai_inl.xml as new <connector> section, which is also used in the <workflow> definitions.

SAP Connector

4 new RFCs have been provided for SAP R/3.

- /EIGNER/MATERIAL_CHANGE
- /EIGNER/MATERIAL_CREATE
- /EIGNER/MATERIAL_DOC_LINKS
- /EIGNER/BOM_ITEM_EFFECTIVITY

The JCo (SAP Java Connector) libraries have been removed from the installation package and need to be installed now by you as described in the installation manual.

SAP Source Connector

A new SAP Source Connector has been provided, which allows to connect to the EIP from inside SAP R/3. Installation of this will be described in a separate document.

Chapter 5
EIP 1.1 to 2.0

General information

This document should help you with the installation of the Eigner Integration Platform Version 1.2 on top of the older version 1.1. It should neither replace the EIP 1.2 installation guide nor the EIP 1.2 Release Notes.

Please keep always in mind, that additional customizing of the Integration Platform (e.g. XSL Mapping) and Eigner PLM (e.g. additional Query Forms) will not be upgraded automatically.

Installation

Due to the fact, that a lot of libraries and configuration files changed between the versions 1.1 and 1.2 , we recommend to install EIP 1.2 in a directory separate from EIP 1.1. Additional mapping files and configuration should be incorporated into the EIP 1.2 installation one by one.

Note: The usage of the Upgrade tool is highly recommended although some manual work might be required.

Configuration and Customizing

Configuration Settings

The structure of the configuration file eai_ini.xml has changed. Therefore please copy your 1.1 configuration settings carefully one by one to the 1.2 eai_ini.xml file.

Following portions of the eai_ini.xml file have changed:

- the <reconnect> option has been added to the connectors
- the <notification> option has been added to the controller
- files can be renamed during check-out from Eigner PLM; therefore a new “rename” option has been provided in the <bor> section of the PLM and axalant connector
- the PLM Connector can now check out multiple files (instead of only a single file) at once. The “ckoflag” option does now allow the value “all”
- The EIP –internal queue can now be stored in Oracle and SQL-Server database systems instead of only the internal one. If you want to activate this option, configuration changes are necessary in the <controller> and <admin> section in order to point to the new database.
- The SAP Connector does now support the SAP Load Balancing Server option. Therefore the logon information of the SAP-Connector changed in the configuration file.
- Due to the new Archive functionality in the Admin GUI (described below) an additional element <archive-dir> was added to the <controller> section.
- A new Data File Connector has been provided, which allows to perform XSL mapping before writing into a file. Therfore a new connector section is provided.
- The library for encrypting the passwords for connecting to the external systems and database has been exchanged. Therefore ALL passwords have to be encrypted newly and put into the configuration file.

Database Support

As mentioned above, you can now use Oracle and SQL-Server database systems for storing the queue. Therefore the “dbmaint” script has been provided to generate the necessary tables and entries in the respective database system.

EIP running as Background Process (Service)

This is a new feature, which needs to be configured and activated first. Please refer to the separate installation guide about the EIP Daemon.

eipping tool

The new eipping tool allows you to automatically check the status of the EIP process and its connectors.

SAP RFC installation

Some of the R/3 RFCs provided in a transport file have changed. Unless you changed some of the RFCs, provided by Eigner, we recommend that you load the new RFCs and retest the SAP Connector.

Eigner PLM - SAP R/3 Mapping Files

The mapping files axa_sap.xsl and sap_axa.xsl have been optimized. Longtext is now better supported. Please consult Eigner representatives for further information or directly compare the old and new mapping files in order to extract the differences.

New Archive directory

The new Administration GUI does now allow to archive and cleanup “old” XDOs in the queue. Therefore a new directory <archive> has been provided as part of the installation, where each XDO (with its different versions) gets stored in a separate XML archive file.

Eigner PLM Loader Files

Due to some functional enhancements, the content of the Eigner PLM and axalant Loader Files have been changed in many ways. We cannot provide a general recommendation, whether the new loader files should be loaded in Insert mode or Overwrite or even loaded in a separate environment in order to perform the changes manually. This mainly depends on the number of modifications which were done since the initial installation of EIP 1.1. Below list should provide you some hints what needs to be considered:

- New fields have been added to the Transfer Queue table (T_EER_SEN) and form for the version view support.
- A new view (V_EIP_VIE) has been provided of the version view support.
- The forms for displaying the query results from R/3 have been changed. Mainly additional USX triggers have been added. This was necessary in order to support the PLM Web-Client and Java Client.
- Due to the new feature that in every success or error case, the response to the calling Eigner PLM system is guaranteed, the Callback Triggers in the Transfer Queue might have to be adapted to the new behavior.

Eigner PLM Library

The synchronous communication between Eigner PLM and the Integration Platform has been enhanced and therefore the new eipsync library file needs to replace the older one in your Eigner PLM installation e.g. <ep_root>\axalant\bin\<OperatingSystem>

Chapter 6
EIP 1.0 to 1.1

General information

This document should help you with the installation of the Eigner Integration Platform Version 1.1 on top of the older version 1.0. It should neither replace the EIP 1.1 installation guide nor the EIP 1.1 Release Notes.

Please keep always in mind, that additional customizing of the Integration Platform (e.g. XSL Mapping) and Eigner PLM (e.g. additional Query Forms) will not be upgraded automatically.

Installation

There had not been any major changes, so running the Upgrade tool should do the job.

Note: The usage of the Upgrade tool is highly recommended although some manual work might be required.

Chapter 7
ESI to SAP Link

General information

This chapter should explain how to upgrade from the old point-to-point integration between Eigner PLM called Eigner SAP Interface (ESI) to the new SAP Link (EIP with SAP Connector).

User-defined Functions

Below table provides a comparison of the currently available User Functions (Agile ABAP) for the Agile e6 module "SAP Link 2.0" and the names of the BAPIs or RFC-Functions, which provide the same functionality. This comparison may help to decide whether the BAPIs and RFC-Functions used by the SAP-Link Version 4 are sufficient to provide certain functionality, which had to be developed in addition for the SAP Link Version 2 (ESI).

Those objects/functions in the table, which don't show a BAPI or RFC-Function name, are not covered by any BAPI or RFC-Function and still require special programming.

This table is based on the functionality in an SAP R/3 System Version 4.6C. This table might differ in the case of older or newer versions of SAP R/3.

Basic

Object	Description	Function	Comment
SAPscript	creates/changes Standard Text	zepbc01b, zepbc01c, zepbc01d	see OSS note 449848

Material

Object	Description	Function	BAPI
basic data text	creates/changes the basic data text in the given language	zepmm01e zepmm01j zepmm26	BAPI_MATERIAL_SAVEDATA
basic data text	creates/changes the basic data text in the given language via QM view	zepmm08d zepmm08e	BAPI_MATERIAL_SAVEDATA

Object	Description	Function	BAPI
internal comment	creates/changes the internal comment in the given language	zepmm07e zepmm07h zepmm28	BAPI_MATERIAL_SAVEDATA
purchase order text	creates/changes the purchase order text in the given language; purchase view must exist, depending on plant	zepmm04e zepmm04h	BAPI_MATERIAL_SAVEDATA
purchase order text	creates/changes the purchase order text in the given language; purchase view can exist, depending on plant	zepmm17 zepmm18	BAPI_MATERIAL_SAVEDATA
purchase order text	creates/changes the purchase order text in the given language; purchase view can exist, independent of plant	zepmm19 zepmm20	BAPI_MATERIAL_SAVEDATA
descriptions	creates/changes the descriptions in the given languages	zepmm05z zepmm11	BAPI_MATERIAL_SAVEDATA
descriptions	reads the descriptions for the given languages; one step	zepmm09 zepmm10	/EIGNER/MATERIAL_DETAILS
descriptions	reads the descriptions; two steps	zepmm30	/EIGNER/MATERIAL_DETAILS
alternative units of measure	creates/changes alternative units of measure with conversion factors	zepmm13	BAPI_MATERIAL_SAVEDATA
alternative units of measure, unit of issue	creates/changes alternative units of measure with conversion factors and unit of issue	zepmm14	BAPI_MATERIAL_SAVEDATA
alternative units of measure, manufacturer part number, unit of issue	creates/changes alternative units of measure with conversion factors, manufacturer part number and unit of issue	zepmm12b	BAPI_MATERIAL_SAVEDATA
alternative units of measure, manufacturer part number, external material group, unit of issue	creates/changes alternative units of measure with conversion factors, manufacturer part number, external material group and unit of issue	zepmm12c	BAPI_MATERIAL_SAVEDATA
purchasing info record	reads purchasing info records for material and plant	zepmm15	
purchasing info record	ME11 : create purchasing info record	zepmm41	
purchasing info record	ME12 : change purchasing info record	zepmm42	
object links	reads for a material the object links referring to the documents	zepmm16	

Object	Description	Function	BAPI
stock overview	determines for a material the stocks of the individual levels	zepmm21	
stock overview	determines for a material and plant the stocks of the individual levels	zepmm21b	
MM12	schedule changing of material, depending on plant	zepmm22c	
CAD indicator	material: setting CAD indicator	zepmm31	BAPI_MATERIAL_SAVEDATA
basic material	material: create/change basic material	zepmm32	BAPI_MATERIAL_SAVEDATA
general item category group	material: create/change general item category group	zepmm34	BAPI_MATERIAL_SAVEDATA
EAN	base units of measure, alternative units of measure, main EANS, additional EANs	zepmm35	BAPI_MATERIAL_SAVEDATA
total shelf life	general plant data / storage: total shelf life, min. remaining shelf life	zepmm36	BAPI_MATERIAL_SAVEDATA
IS Automotive	IS Automotive: create material (basic data)	zepmm40_ automotive	BAPI_STDMATERIAL_GETINTNUMBER
QM view	MM02 : change QM view (inspection setup)	zepmm43	

BOM

Object	Description	Function	BAPI
BOM header	creates/changes long text in the logon language	zepbo01c zepbo01d	
BOM item	creates/changes long text in the logon language	zepbo02k zepbo02l	
BOM item	N* creates/changes long text in the logon language	zepbo08b zepbo08c zepbo08d	
BOM item	creates/changes sub-items (installation point)	zepbo05 zepbo05b	
BOM item	N* creates/changes sub-items (installation point)	zepbo07b zepbo07c	
BOM item	creates/changes customer fields; variant 1	zepbo04b	CSAP_MAT_BOM_MAINTAIN

Object	Description	Function	BAPI
BOM item	creates/changes customer fields; variant 2	zepbo04c zepbo04d	CSAP_MAT_BOM_MAINTAIN
BOM item	N* creates/changes customer fields; variant 2	zepbo11	CSAP_MAT_BOM_MAINTAIN
BOM item	sets CAD indicator for all items	zepbo10	CSAP_MAT_BOM_MAINTAIN

Engineering Change Management

Object	Description	Function	BAPI
header	creates/changes long text in the logon language	zepch01b	CCAP_ECN_HEADER_CHANGE
objects	creates/changes the change index text in the logon language for the object management record	zepch02b zepch02c zepch02d	CCAP_ECN_MAINTAIN (TEXTLINES : TEXTKEY)
objects	reads object management records	zepch03	
objects	deletes object management record	zepch04 zepch04b	
change master	creates change master (special)	zepch05	CCAP_ECN_CREATE
change master	creates change master with serial number	zepch07	CCAP_ECN_CREATE
change master	changes change master with serial number	zepch08	CCAP_ECN_MAINTAIN
change master	reads change master with serial number	zepch09	

Document

Object	Description	Function	BAPI
long text	creates/changes long text in the given language	zepdo03 zepdo04 zepdo20	BAPI_DOCUMENT_CHANGE2
long text	reads long text in the given language	zepdo07 zepdo08	BAPI_DOCUMENT_GETDETAIL2
descriptions	creates/changes descriptions in the given languages	zepdo05 zepdo06 zepdo22	BAPI_DOCUMENT_CHANGE2
descriptions	reads descriptions in the given languages; one step	zepdo01 zepdo02	BAPI_DOCUMENT_GETDETAIL2

Object	Description	Function	BAPI
descriptions	reads descriptions; two steps	zepdo18	BAPI_DOCUMENT_GETDETAIL2
document structure	creates document structure with one BOM item	zepdo13 zepdo13b	CSAP_DOC_BOM_CREATE
document structure	creates/changes BOM items	zepdo09 zepdo09	CSAP_DOC_BOM_MAINTAIN
document structure	deletes BOM item	zepdo15 zepdo15b	CSAP_DOC_BOM_MAINTAIN
document structure	reads document structure	zepdo11 zepdo12	CSAP_DOC_BOM_READ
SAP ArchiveLink:	determines document id of an original of a document info record	zepdo17	
object link	deletes all object links of a document info record	zepdo31	

Others

Object	Description	Function	BAPI
classification	reads class descriptions	zepcl05 zepcl06	BAPI_CLASS_GETDETAIL
project system	CN33 : interface project system - BOM	zepps01	
project system	reads work breakdown structure and element with customer fields	zepps0t2	
QM	determines q-notes and their states for the given material and notification item short text	zepqm01	
QM	first determines the material for the given description, then determines q-notes and their states for material and notification item short text	zepqm01b	
QM	QI01 : create q-info record	zepqm03	
QM	QI02 : change q-info record	zepqm04	
QM	QP02 : create inspection plan	zepqm05	
QM	QP01 : change inspection plan	zepqm06	

Converting User Functions to RFC Functions

Goal

The user function is a subroutine, which is dynamically called from the SAP CAD interface so that they can only be used in this area. However, RFC functions can be used anywhere and can be called directly from outside SAP.

RFC functions can easily handle a lot of data and tables; and the communication is done in one step. User functions have to manage data and tables in a more complex way, and the communication has to be split up into several user functions in the same report, especially for a large amount of data and tables.

Overview

The following chapter gives an overview of the steps necessary to convert an SAP user function to an RFC function.

User functions have a “General Information” section, which describes the parameter (INPUT, OUTPUT). The user function parameters have to be mapped to RFC function parameters as listed below:

- INPUT -> IMPORTING
- INPUT + multiple entries -> TABLES
- OUTPUT -> EXPORTING
- OUTPUT + multiple entries -> TABLES

RFC functions require an additional export parameter, which needs to be added.

The EXPORTING parameter is named RETURN (message etc.) and is based on the BAPIRET2 structure. This parameter can be filled and modified with the function BALW_BAPIRETURN_GET2.

In general, parameters for RFC functions are based on the SAPData Dictionary (structure, table) and have to be marked as “Call by value”.

Everything of the CAD Dialog Interface (COMMON Data Area) is no longer required and everything specific to the CAD Dialog Interface needs to be replaced during the conversion of an existing user function into an RFC function.

The following statements are obsolete and have to be deleted:

- INCLUDE rccadcom.
- variables based on tablescadin, cadout
- FIELD-SYMBOLS: <data_string>.
- Initialization of variables from the include rccadcom
- Error message, if no parameter is given
- Internal table CADIN … copy data to data_string
- Split the data string … value of field TCIS-DELIM
- Confirm and send returnstring
- Write the results … send it to external system
- Reset interface

Replace (delete + insert)

```
- IF … ENDIF.
   CHECK error_flag IS INITIAL.
   with
   IF … EXIT. ENDIF.
```

The other subroutines are put in a separate INCLUDE file of the function group.

The following adaptation has to be made for the message handling:

The “built_message” subroutine has to get an additional CHANGING parameter based on the BAPIRET2 structure.

Example

Call :

```
PERFORM built_messageusing 'ZX' '189' 'E' mat_part_1 mat_part_2 '' ''
                   CHANGING return.
```

Declaration :

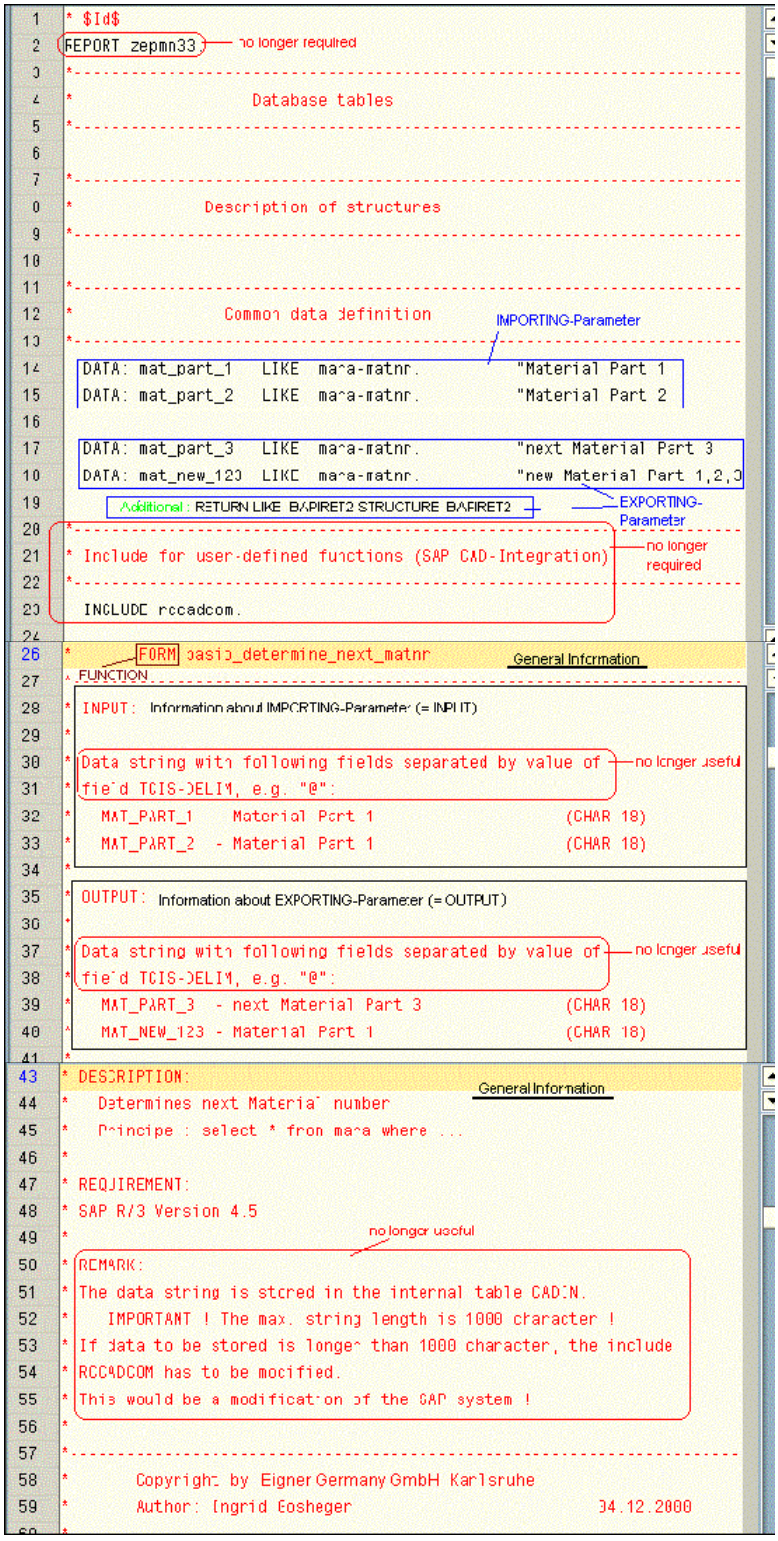
```
FORM built_message USING msgid msgno msgty msgv1 msgv2 msgv3 msgv4
                   CHANGING bapi_ret2 LIKE bapiret2.
```

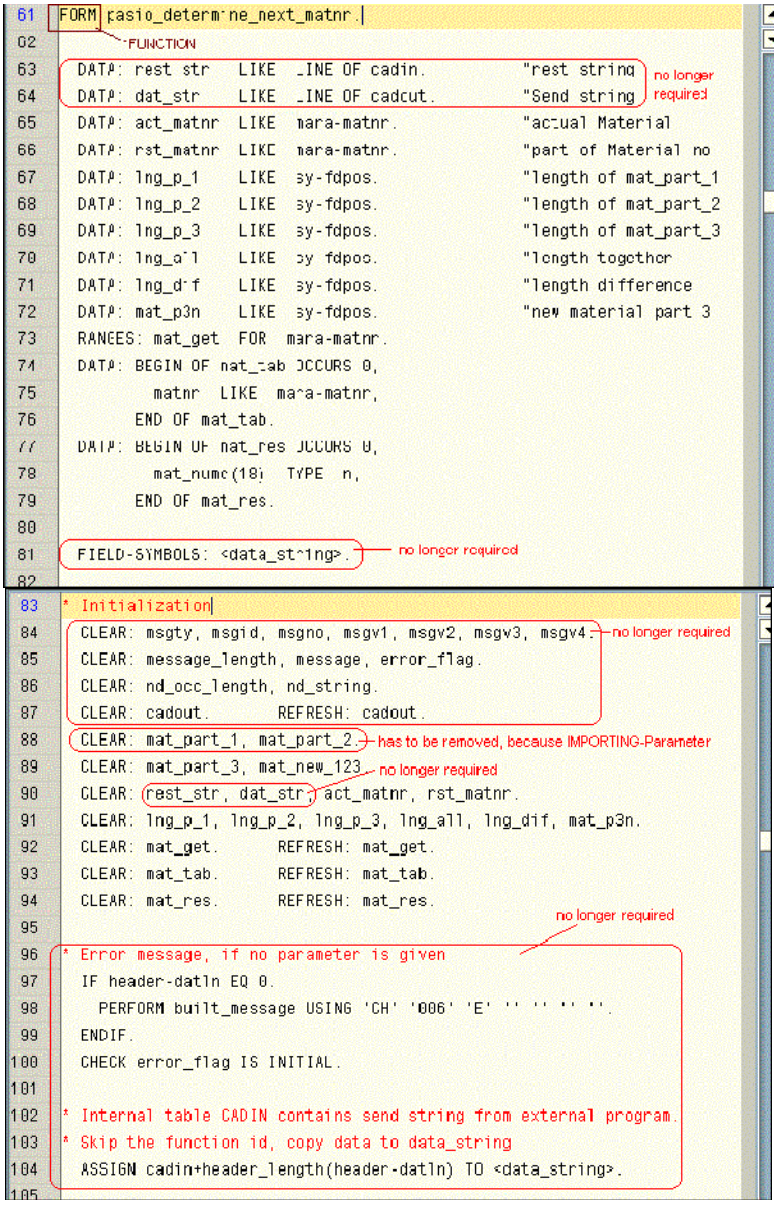
Example

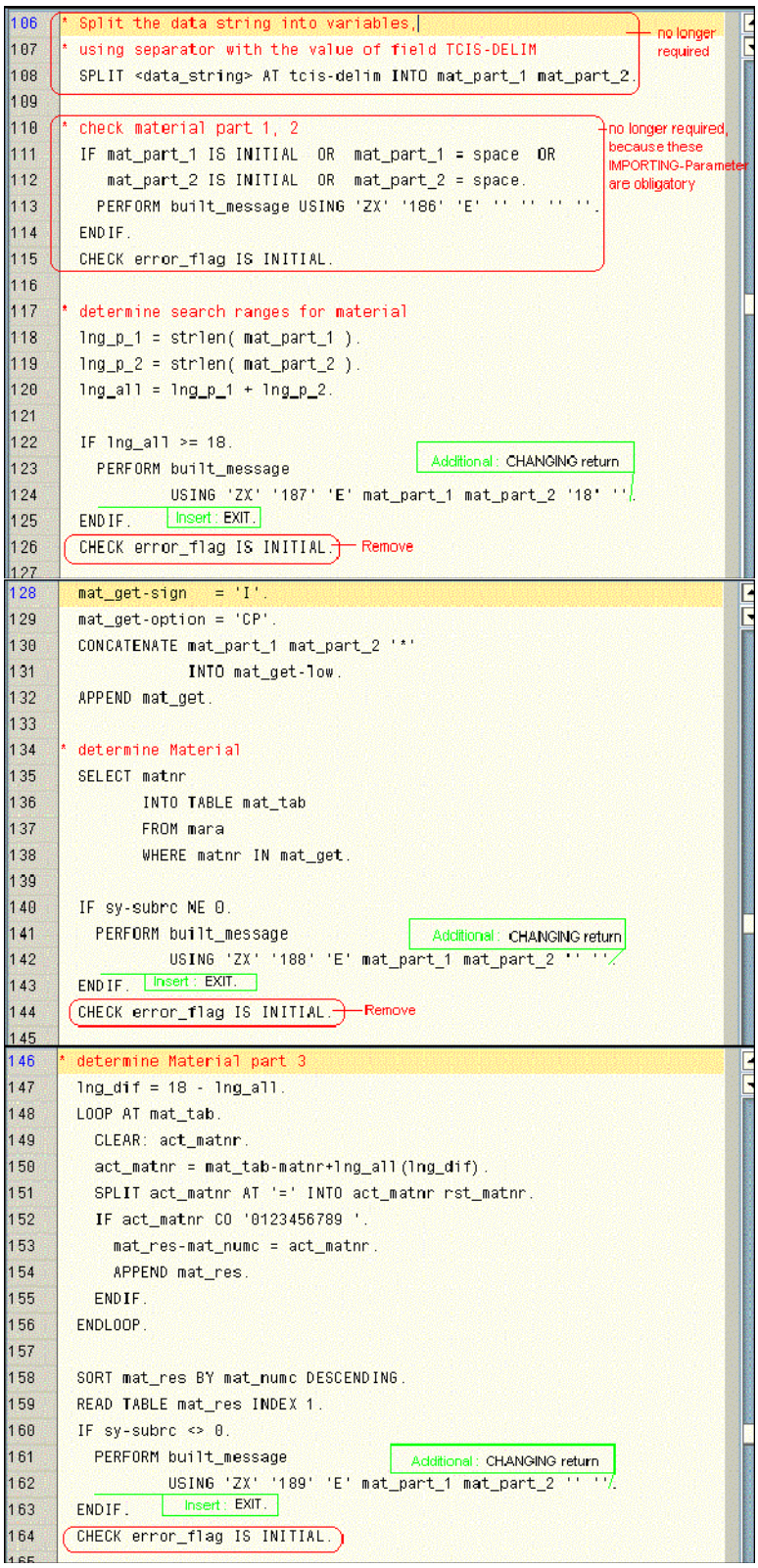
The example describes how to convert a user function to an RFC function.

Legend

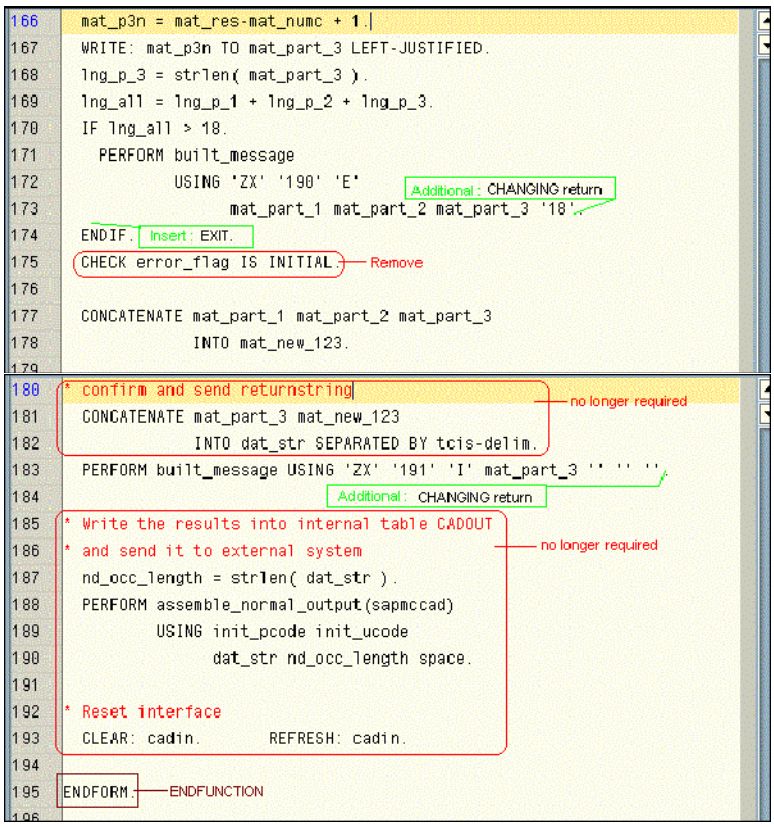
- red : no longer required
- blue : parameter
- green : additional or insert
- brown : automatically inserted or replaced


no longer required
Database tables
Description of structures
Common data definition
IMPORTING-Parameter
DATA: mat_part_1 LIKE mara-matnr. "Material Part 1
DATA: mat_part_2 LIKE mara-matnr. "Material Part 2
DATA: mat_part_3 LIKE mara-matnr. "next Material Part 3
DATA: mat_new_123 LIKE mara-matnr. "new Material Part 1,2,3
Additional : RETURN LIKE BAPIRET2 STRUCTURE BAPIRET2
EXPORTING-Parameter
* Include for user-defined functions (SAP CAD-Integration)
INCLUDE rccadcom.
no longer required
FORM basis_determine_next_matnr
FUNCTION
General Information
* INPUT: Information about IMPORTING-Parameter (= INPUT)
Data string with following fields separated by value of field TCIS-DELIM, e.g. "@":
MAT_PART_1 Material Part 1 (CHAR 18)
MAT_PART_2 - Material Part 1 (CHAR 18)
no longer useful
* OUTPUT: Information about EXPORTING-Parameter (= OUTPUT)
Data string with following fields separated by value of field TCIS-DELIM, e.g. "@":
MAT_PART_3 - next Material Part 3 (CHAR 18)
MAT_NEW_123 - Material Part 1 (CHAR 18)
no longer useful
* DESCRIPTION:
General Information
Determines next Material number
Principe : select * from mara where ...
* REQUIREMENT:
SAP R/3 Version 4.5
no longer useful
* REMARK:
The data string is stored in the internal table CADIN.
IMPORTANT ! The max. string length is 1000 character !
If data to be stored is longer than 1000 character, the include
RCCADCOM has to be modified.
This would be a modification of the CAD system !
Copyright by Eigner Germany GmbH Karlsruhe
Author: Ingrid Eosheger 04.12.2000

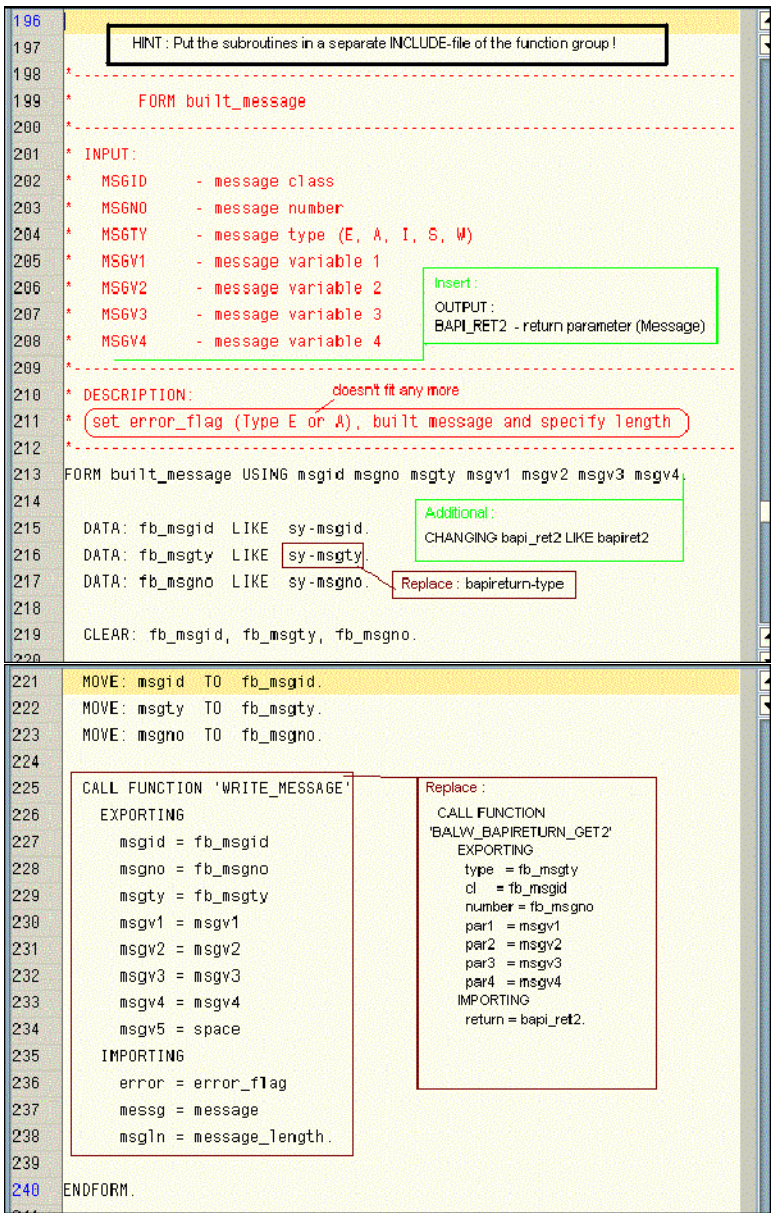

FORM casio_determine_next_matnr.
FUNCTION
DATA: rest_str LIKE LINE OF cadin. "rest string
DATA: dat_str LIKE LINE OF cadcut. "Send string
no longer required
DATA: act_matnr LIKE mara-matnr. "actual Material
DATA: rst_matnr LIKE mara-matnr. "part of Material no
DATA: lng_p_1 LIKE sy-fdpos. "length of mat_part_1
DATA: lng_p_2 LIKE sy-fdpos. "length of mat_part_2
DATA: lng_p_3 LIKE sy-fdpos. "length of mat_part_3
DATA: lng_all LIKE sy-fdpos. "length together
DATA: lng_dif LIKE sy-fdpos. "length difference
DATA: mat_p3n LIKE sy-fdpos. "new material part 3
RANGES: mat_get FOR mara-matnr.
DATA: BEGIN OF mat_tab OCCURS 0,
matnr LIKE mara-matnr,
END OF mat_tab.
DATA: BEGIN OF mat_res OCCURS 0,
mat_numc(18) TYPE n,
END OF mat_res.
FIELD-SYMBOLS: <data_string>.
no longer required
* Initialization
CLEAR: msgty, msgid, msgno, msgv1, msgv2, msgv3, msgv4.
no longer required
CLEAR: message_length, message, error_flag.
CLEAR: nd_occ_length, nd_string.
CLEAR: cadout. REFRESH: cadout.
CLEAR: mat_part_1, mat_part_2.
has to be removed, because IMPORTING-Parameter
CLEAR: mat_part_3, mat_new_123.
no longer required
CLEAR: rest_str, dat_str, act_matnr, rst_matnr.
CLEAR: lng_p_1, lng_p_2, lng_p_3, lng_all, lng_dif, mat_p3n.
CLEAR: mat_get. REFRESH: mat_get.
CLEAR: mat_tab. REFRESH: mat_tab.
CLEAR: mat_res. REFRESH: mat_res.
no longer required
* Error message, if no parameter is given
IF header-datln EQ 0.
PERFORM built_message USING 'CH' '006' 'E' '' '' '' ''.
ENDIF.
CHECK error_flag IS INITIAL.
* Internal table CADIN contains send string from external program.
* Skip the function id, copy data to data_string
ASSIGN cadin+header_length(header-datln) TO <data_string>.




```
106 * Split the data string into variables,
107 * using separator with the value of field TCIS-DELIM
108   SPLIT <data_string> AT tcis-delim INTO mat_part_1 mat_part_2.
109
110 * check material part 1, 2
111   IF mat_part_1 IS INITIAL  OR  mat_part_1 = space  OR
112      mat_part_2 IS INITIAL  OR  mat_part_2 = space.
113     PERFORM built_message USING 'ZX' '186' 'E' '' '' '' ''.
114   ENDIF.
115   CHECK error_flag IS INITIAL.
116
117 * determine search ranges for material
118   lng_p_1 = strlen( mat_part_1 ).
119   lng_p_2 = strlen( mat_part_2 ).
120   lng_all = lng_p_1 + lng_p_2.
121
122   IF lng_all >= 18.
123     PERFORM built_message
124             USING 'ZX' '187' 'E' mat_part_1 mat_part_2 '18' ''.
125   ENDIF.
126   CHECK error_flag IS INITIAL.
127
128   mat_get-sign   = 'I'.
129   mat_get-option = 'CP'.
130   CONCATENATE mat_part_1 mat_part_2 '*'
131               INTO mat_get-low.
132   APPEND mat_get.
133
134 * determine Material
135   SELECT matnr
136          INTO TABLE mat_tab
137          FROM mara
138          WHERE matnr IN mat_get.
139
140   IF sy-subrc NE 0.
141     PERFORM built_message
142             USING 'ZX' '188' 'E' mat_part_1 mat_part_2 '' ''.
143   ENDIF.
144   CHECK error_flag IS INITIAL.
145
146 * determine Material part 3
147   lng_dif = 18 - lng_all.
148   LOOP AT mat_tab.
149     CLEAR: act_matnr.
150     act_matnr = mat_tab-matnr+lng_all(lng_dif).
151     SPLIT act_matnr AT '=' INTO act_matnr rst_matnr.
152     IF act_matnr CO '0123456789 '.
153       mat_res-mat_numc = act_matnr.
154       APPEND mat_res.
155     ENDIF.
156   ENDLOOP.
157
158   SORT mat_res BY mat_numc DESCENDING.
159   READ TABLE mat_res INDEX 1.
160   IF sy-subrc <> 0.
161     PERFORM built_message
162             USING 'ZX' '189' 'E' mat_part_1 mat_part_2 '' ''.
163   ENDIF.
164   CHECK error_flag IS INITIAL.
```



The "built_message" subroutine is part of the report and should be put in a separate INCLUDE file of the function group. Then the subroutine can be shared by several RFC functions.



User Function

```
          REPORT zepmm33.
*---------------------------------------------------------------------- ---------
*                   Database tables
*---------------------------------------------------------------------- ---------


*---------------------------------------------------------------------- ---------
*              Description       of structures
*---------------------------------------------------------------------- ---------


*---------------------------------------------------------------------- ---------
*                       Common data definition
*---------------------------------------------------------------------- ---------
  DATA: mat_part_1   LIKE  mara-matnr.          "Material Part 1
```

```
  DATA: mat_part_2   LIKE  mara-matnr.          "Material Part 2

  DATA: mat_part_3   LIKE  mara-matnr.          "next Material Part 3
  DATA: mat_new_123  LIKE  mara-matnr.          "new Material Part 1,2,3

*---------------------------------------------------------------------- ---------
* Include for user-defined functions (SAP CAD-Integration)

*---------------------------------------------------------------------- ---------
  INCLUDE rccadcom.

*---------------------------------------------------------------------

*       FORM pasio_determine_next_matnr
*---------------------------------------------------------------------



* INPUT:

*

* Data string with following fields separated by value of

* field TCIS-DELIM, e.g. "@":

*   MAT_PART_1        - Material Part 1                   (CHAR 18)

*   MAT_PART_2        - Material Part 1                   (CHAR 18)

*

* OUTPUT:

*

* Data string with following fields separated by value of
* field TCIS-DELIM, e.g. "@":
*   MAT_PART_3        - next Material Part 3              (CHAR 18)
*   MAT_NEW_123       - Material Part 1                   (CHAR 18)
*

*---------------------------------------------------------------------
* DESCRIPTION:
*   Determines next Material number
*   Principe : select * from mara where ...

*

* REQUIREMENT:
* SAP R/3 Version 4.5
*

* REMARK:
* The data string is stored in the internal table CADIN.
*    IMPORTANT ! The max. string length is 1000 character !
* If data to be stored is longer than 1000 character, the include
* RCCADCOM has to be modified.
* This would be a modification of the SAP system !

*

FORM pasio_determine_next_matnr.

  DATA: rest_str   LIKE  LINE OF cadin.          "rest string
  DATA: dat_str    LIKE  LINE OF cadout.         "Send string
```

```
  DATA: act_matnr  LIKE  mara-matnr.            "actual Material
  DATA: rst_matnr  LIKE  mara-matnr.            "part of Material no
  DATA: lng_p_1    LIKE  sy-fdpos.              "length of mat_part_1
  DATA: lng_p_2    LIKE  sy-fdpos.              "length of mat_part_2
  DATA: lng_p_3    LIKE  sy-fdpos.              "length of mat_part_3
  DATA: lng_all    LIKE  sy-fdpos.              "length together
  DATA: lng_dif    LIKE  sy-fdpos.              "length difference
  DATA: mat_p3n    LIKE  sy-fdpos.              "new material part 3
  RANGES: mat_get  FOR  mara-matnr.
  DATA: BEGIN OF mat_tab OCCURS 0,
          matnr  LIKE  mara-matnr,
        END OF mat_tab.
  DATA: BEGIN OF mat_res OCCURS 0,
          mat_numc(18)  TYPE  n,
        END OF mat_res.

  FIELD-SYMBOLS: <data_string>.

* Initialization
  CLEAR: msgty, msgid, msgno, msgv1, msgv2, msgv3, msgv4.
  CLEAR: message_length, message, error_flag.
  CLEAR: nd_occ_length, nd_string.
  CLEAR: cadout.       REFRESH: cadout.
  CLEAR: mat_part_1, mat_part_2.
  CLEAR: mat_part_3, mat_new_123.
  CLEAR: rest_str, dat_str, act_matnr, rst_matnr.
  CLEAR: lng_p_1, lng_p_2, lng_p_3, lng_all, lng_dif, mat_p3n.
  CLEAR: mat_get.      REFRESH: mat_get.
  CLEAR: mat_tab.      REFRESH: mat_tab.
  CLEAR: mat_res.      REFRESH: mat_res.

* Error message, if no parameter is given
  IF header-datln EQ 0.
    PERFORM built_message USING 'CH' '006' 'E' '' '' '' ''.
  ENDIF.
  CHECK error_flag IS INITIAL.

* Internal table CADIN contains send string from external program.
* Skip the function id, copy data to data_string
  ASSIGN cadin+header_length(header-datln) TO <data_string>.

* Split the data string into variables,
* using separator with the value of field TCIS-DELIM
  SPLIT <data_string> AT tcis-delim INTO mat_part_1 mat_part_2.

* check material part 1, 2
  IF mat_part_1 IS INITIAL  OR  mat_part_1 = space  OR
     mat_part_2 IS INITIAL  OR  mat_part_2 = space.
    PERFORM built_message USING 'ZX' '186' 'E' '' '' '' ''.
  ENDIF.
  CHECK error_flag IS INITIAL.

* determine search ranges for material
  lng_p_1 = strlen( mat_part_1 ).
  lng_p_2 = strlen( mat_part_2 ).
  lng_all = lng_p_1 + lng_p_2.

  IF lng_all >= 18.
    PERFORM built_message
            USING 'ZX' '187' 'E' mat_part_1 mat_part_2 '18' ''.
  ENDIF.
  CHECK error_flag IS INITIAL.

  mat_get-sign   = 'I'.
  mat_get-option = 'CP'.
  CONCATENATE mat_part_1 mat_part_2 '*'
              INTO mat_get-low.
  APPEND mat_get.

* determine Material
  SELECT matnr
         INTO TABLE mat_tab
         FROM mara
```

```
          WHERE matnr IN mat_get.

  IF sy-subrc NE 0.
    PERFORM built_message
            USING 'ZX' '188' 'E' mat_part_1 mat_part_2 '' ''.
  ENDIF.
  CHECK error_flag IS INITIAL.

* determine Material part 3
  lng_dif = 18 - lng_all.
  LOOP AT mat_tab.
    CLEAR: act_matnr.
    act_matnr = mat_tab-matnr+lng_all(lng_dif).
    SPLIT act_matnr AT '=' INTO act_matnr rst_matnr.
    IF act_matnr CO '0123456789 '.
      mat_res-mat_numc = act_matnr.
      APPEND mat_res.
    ENDIF.
  ENDLOOP.

  SORT mat_res BY mat_numc DESCENDING.
  READ TABLE mat_res INDEX 1.
  IF sy-subrc <> 0.
    PERFORM built_message
            USING 'ZX' '189' 'E' mat_part_1 mat_part_2 '' ''.
  ENDIF.
  CHECK error_flag IS INITIAL.

  mat_p3n = mat_res-mat_numc + 1.
  WRITE: mat_p3n TO mat_part_3 LEFT-JUSTIFIED.
  lng_p_3 = strlen( mat_part_3 ).
  lng_all = lng_p_1 + lng_p_2 + lng_p_3.
  IF lng_all > 18.
    PERFORM built_message
            USING 'ZX' '190' 'E'
                  mat_part_1 mat_part_2 mat_part_3 '18'.
  ENDIF.
  CHECK error_flag IS INITIAL.

  CONCATENATE mat_part_1 mat_part_2 mat_part_3
              INTO mat_new_123.

* confirm and send returnstring
  CONCATENATE mat_part_3 mat_new_123
              INTO dat_str SEPARATED BY tcis-delim.
  PERFORM built_message USING 'ZX' '191' 'I' mat_part_3 '' '' ''.

* Write the results into internal table CADOUT
* and send it to external system
  nd_occ_length = strlen( dat_str ).
  PERFORM assemble_normal_output(sapmccad)
          USING init_pcode init_ucode
                dat_str nd_occ_length space.

* Reset interface
  CLEAR: cadin.       REFRESH: cadin.

ENDFORM.



*---------------------------------------------------------------------
*       FORM built_message
*---------------------------------------------------------------------
* INPUT:
*   MSGID     - message class
*   MSGNO     - message number
*   MSGTY     - message type (E, A, I, S, W)
*   MSGV1     - message variable 1
*   MSGV2     - message variable 2
*   MSGV3     - message variable 3
*   MSGV4     - message variable 4
```

```
*---------------------------------------------------------------------
* DESCRIPTION:
*  set error_flag (Type E or A), built message and specify length
*---------------------------------------------------------------------
FORM built_message USING msgid msgno msgty msgv1 msgv2 msgv3 msgv4.

  DATA: fb_msgid  LIKE  sy-msgid.
  DATA: fb_msgty  LIKE  sy-msgty.
  DATA: fb_msgno  LIKE  sy-msgno.

  CLEAR: fb_msgid, fb_msgty, fb_msgno.

  MOVE: msgid  TO  fb_msgid.
  MOVE: msgty  TO  fb_msgty.
  MOVE: msgno  TO  fb_msgno.

  CALL FUNCTION 'WRITE_MESSAGE'
    EXPORTING
      msgid = fb_msgid
      msgno = fb_msgno
      msgty = fb_msgty
      msgv1 = msgv1
      msgv2 = msgv2
      msgv3 = msgv3
      msgv4 = msgv4


      msgv5 = space
    IMPORTING
      error = error_flag
      messg = message
      msgln = message_length.

ENDFORM.
```

RFC Function

```
FUNCTION /eigner/determine_next_matnr.

*"----------------------------------------------------------------------
*"*"LocalLokaleInterfaceSchnittstelle:
*"  IMPORTING
*"     VALUE(MAT_PART_1) LIKE  MARA-MATNR
*"     VALUE(MAT_PART_2) LIKE  MARA-MATNR
*"  EXPORTING
*"     VALUE(MAT_PART_3) LIKE  MARA-MATNR
*"     VALUE(MAT_NEW_123) LIKE  MARA-MATNR
*"     VALUE(RETURN) LIKE  BAPIRET2 STRUCTURE  BAPIRET2
*"----------------------------------------------------------------------



  DATA: act_matnr  LIKE  mara-matnr.            "actual Material



  DATA: rst_matnr  LIKE  mara-matnr.            "part of Material no



  DATA: lng_p_1    LIKE  sy-fdpos.              "length of mat_part_1



  DATA: lng_p_2    LIKE  sy-fdpos.              "length of mat_part_2
  DATA: lng_p_3    LIKE  sy-fdpos.              "length of mat_part_3
  DATA: lng_all    LIKE  sy-fdpos.              "length together
  DATA: lng_dif    LIKE  sy-fdpos.              "length difference
  DATA: mat_p3n    LIKE  sy-fdpos.              "new material part 3
  RANGES: mat_get  FOR  mara-matnr.
```

```
  DATA: BEGIN OF mat_tab OCCURS 0,
          matnr  LIKE  mara-matnr,
        END OF mat_tab.
  DATA: BEGIN OF mat_res OCCURS 0,
          mat_numc(18)  TYPE  n,
        END OF mat_res.



* Initialization
  CLEAR: mat_part_3, mat_new_123.
  CLEAR: act_matnr, rst_matnr.
  CLEAR: lng_p_1, lng_p_2, lng_p_3, lng_all, lng_dif, mat_p3n.
  CLEAR: mat_get.      REFRESH: mat_get.
  CLEAR: mat_tab.      REFRESH: mat_tab.
  CLEAR: mat_res.      REFRESH: mat_res.



* determine search ranges for material
  lng_p_1 = strlen( mat_part_1 ).
  lng_p_2 = strlen( mat_part_2 ).
  lng_all = lng_p_1 + lng_p_2.



  IF lng_all >= 18.
    PERFORM built_message
            USING 'ZX' '187' 'E' mat_part_1 mat_part_2 '18' ''
            CHANGING return.
    EXIT.
  ENDIF.



  mat_get-sign   = 'I'.
  mat_get-option = 'CP'.
  CONCATENATE mat_part_1 mat_part_2 '*'
              INTO mat_get-low.
  APPEND mat_get.
* determine Material
  SELECT matnr
         INTO TABLE mat_tab
         FROM mara
         WHERE matnr IN mat_get.



  IF sy-subrc NE 0.
    PERFORM built_message
            USING 'ZX' '188' 'E' mat_part_1 mat_part_2 '' ''
            CHANGING return.
    EXIT.
  ENDIF.



* determine Material part 3
  lng_dif = 18 - lng_all.
  LOOP AT mat_tab.
    CLEAR: act_matnr.
    act_matnr = mat_tab-matnr+lng_all(lng_dif).
    SPLIT act_matnr AT '=' INTO act_matnr rst_matnr.
    IF act_matnr CO '0123456789 '.
      mat_res-mat_numc = act_matnr.
      APPEND mat_res.
    ENDIF.
  ENDLOOP.
```

```
  SORT mat_res BY mat_numc DESCENDING.
  READ TABLE mat_res INDEX 1.
  IF sy-subrc <> 0.
    PERFORM built_message
            USING 'ZX' '189' 'E' mat_part_1 mat_part_2 '' ''
            CHANGING return.
    EXIT.
  ENDIF.


  mat_p3n = mat_res-mat_numc + 1.
  WRITE: mat_p3n TO mat_part_3 LEFT-JUSTIFIED.
  lng_p_3 = strlen( mat_part_3 ).
  lng_all = lng_p_1 + lng_p_2 + lng_p_3.
  IF lng_all > 18.
    PERFORM built_message
            USING 'ZX' '190' 'E'
                  mat_part_1 mat_part_2 mat_part_3 '18'
            CHANGING return.
    EXIT.
  ENDIF.



  CONCATENATE mat_part_1 mat_part_2 mat_part_3
              INTO mat_new_123.
  PERFORM built_message USING 'ZX' '191' 'I' mat_part_3 '' '' ''
                       CHANGING return.


ENDFUNCTION.
```

Include of Function Group

```
*---------------------------------------------------------------------- ---------*
***INCLUDE /EIGNER/LUSERFCTF01 .
*---------------------------------------------------------------------- ---------*


*---------------------------------------------------------------------- ---------*
*       FORM built_message
*---------------------------------------------------------------------- ---------*
* INPUT:
*   MSGID     - message class
*   MSGNO     - message number
*   MSGTY     - message type (E, A, I, S, W)
*   MSGV1     - message variable 1
*   MSGV2     - message variable 2
*   MSGV3     - message variable 3
*   MSGV4     - message variable 4
* OUTPUT:
*   BAPI_RET2 - return parameter
*---------------------------------------------------------------------- ---------*
* DESCRIPTION:
*---------------------------------------------------------------------- ---------*
FORM built_message USING msgid msgno msgty msgv1 msgv2 msgv3 msgv4
                  CHANGING bapi_ret2 LIKE bapiret2.



  DATA: fb_msgid  LIKE  sy-msgid.
  DATA: fb_msgty  LIKE  bapireturn-type.
  DATA: fb_msgno  LIKE  sy-msgno.



  CLEAR: fb_msgid, fb_msgty, fb_msgno.
```

```
  MOVE: msgid  TO  fb_msgid.
  MOVE: msgty  TO  fb_msgty.
  MOVE: msgno  TO  fb_msgno.



  CALL FUNCTION 'BALW_BAPIRETURN_GET2'
       EXPORTING
         type   = fb_msgty
         cl     = fb_msgid
         number = fb_msgno
         par1   = msgv1
         par2   = msgv2
         par3   = msgv3
         par4   = msgv4
       IMPORTING
         return = bapi_ret2.

ENDFORM
```