

Oracle Agile Engineering Data Management

System Customization Management

Release e6.2.1.0

E77560-02

October 2017

Copyright © 2010, 2017, Oracle and/or its affiliates. All rights reserved.

Primary Author: Oracle

Contributing Author:

Contributor:

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	vii
Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	vii
 1 Introduction	
What is System Customization Management (SCM)?	1-1
SCM Supports Best Practices for System Customization	1-1
Overview of SCM Capabilities	1-1
Usage of SCM	1-1
Overview of SCM Objects and Features	1-2
Bundles	1-2
Patches/Modules	1-3
Data Analysis Tools	1-3
Check Structure	1-3
Import	1-3
All Modifications	1-3
Loader Imports	1-3
Configuration	1-3
 2 SCM Toolbars, Menus, and Tabs	
SCM Toolbar	2-1
SCM from the Manager Menu	2-2
Standard Menus	2-2
Non-select and Select Standard Menus	2-2
SCM Context Menus	2-2
 3 SCM Settings	
Defaults	3-1
LGV Constants	3-1
Number Variant	3-4

4 How SCM Works

Setup and Configure Source and Target Environments	4-1
Overview of Source Application Setup	4-1
Overview of Target Application Setup	4-1
Select / Identify Modifications	4-1
CTRL-A	4-2
Export Modifications from the Source Environment	4-2
Import Modifications into the Target Environment	4-2

5 Configuring the Source Environment and the Target Environment

Configuring the Source Environment	5-1
Configuring the Target Environment	5-2
Configuring Modification Types for SCM	5-3
Define and Add Customization Modification Objects	5-3

6 Bundle Objects

Bundle Form	6-2
General Information Tab, Bundle Form	6-2
Modifications Tab	6-3
Modifications List	6-4
SCM Modification Types	6-4
In Patches Tab	6-7
Loader Tab	6-7
Documents Tab, Bundle Form	6-7
Exp/Imp Check Tab	6-7
History Tab	6-8
Bundle List	6-8
Managing Bundles	6-9
Collect Modifications	6-9
Methods to Record Modifications	6-9
Release Procedure for Bundles	6-9
View All	6-10

7 Managing Patches and Modules

Patches	7-1
Patch/Module Management Tab Menus	7-1
Patch/Module Management Form	7-2
Locking and Reserving Objects (Ctrl-B)	7-3

8 Transfer Content from the Source Environment to the Target Environment

Creating Customization Files	8-1
Example: Transfer Content by Releasing the Bundle	8-2
Importing Customization Files into the Target Environment	8-2
Identify Modifications with Post Action UserExit	8-2

9 Export Modifications from the Source Environment

Export Procedure 9-1

10 Import Modifications into the Target Environment

Import Procedure..... 10-1

Preface

Agile PLM is a comprehensive enterprise PLM solution for managing your product value chain.

Audience

This document is intended for administrators and users who do customization of the Agile EDM product.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

Oracle's Agile PLM documentation set includes Adobe® Acrobat PDF files. The Oracle Technology Network (OTN) Web site

<http://www.oracle.com/technetwork/documentation/agile-085940.html> contains the latest versions of the 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 Agile PLM Documentation folder available on your network from which you can access the Agile PLM documentation (PDF) files.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.

Convention	Meaning
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Introduction

System Customization Management (SCM) is a tool for Agile e6 administrators that provides a way to manage customization code in Agile e6.

What is System Customization Management (SCM)?

System Customization Management (SCM) is a tool for Agile e6 administrators and consultants that provides a sophisticated method to manage customization code in Agile e6. The SCM tool provides a controlled method for transferring customization code from one source environment to another target environment.

SCM Supports Best Practices for System Customization

Best practice for creating custom enhancements and modifications is to:

1. Develop the custom code in a separate development environment of Agile e6.
2. In the development environment, thoroughly test and validate the custom code.
3. Transfer the development custom code to an Agile e6 test environment where key users can verify that all features are fully functional.
4. Transfer the now validated custom code from the Agile e6 test environment to the production Agile e6 environment.

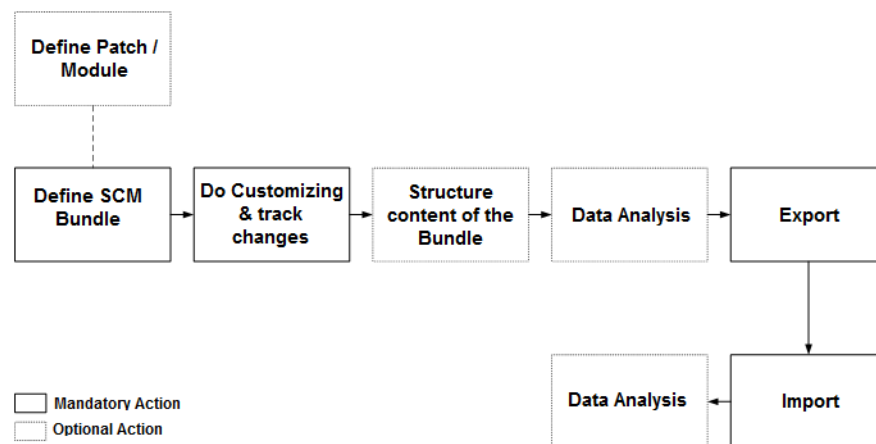
Overview of SCM Capabilities

In the system customization process, you must identify the modifications in one environment and transport them to a different environment. SCM provides capabilities that enable you to:

- Select/identify modifications
- Assign modifications to so-called change packages or bundles
- Manage the existing bundles within an environment
- Export the bundles into a set of files
- Import the set of files and activate the modifications in another environment

Usage of SCM

The following figure illustrates the general flow of actions, when working with SCM tool. This is an iterative process where the user will repeat the process as many times as needed to thoroughly define, export, and test the system customization.

Figure 1–1 Flow of Action when working with SCM Tool

In order to follow best practices, you will use two different Agile e6 environments:

- A separate test environment, which includes a test source application and a test target application.
Develop, test, analyze, and verify the customization code in the test environment.
 - The production environment.
Only after the customization code has been fully tested and verified in the test environment, then transfer the customization code to the production environment.
1. Configure the test environment Source and Target application as described in the Chapter “Configuring the Source Environment and the Target Environment.
 2. Develop the custom code in the separate development test environment of Agile e6. Define the SCM bundle and prepare to transfer it to the test target application.
 3. Transfer the custom code to the test development target application. In the test development environment, thoroughly test and validate the custom code.
 4. Transfer the development custom code to an Agile e6 test environment where key users can verify that all features are fully functional.
 5. Repeat steps 2 through 4 as many times as needed until the customized code is ready to be deployed in the Agile e6 production environment.
 6. Transfer the now validated custom code from the Agile e6 test environment to the production Agile e6 environment.

Overview of SCM Objects and Features

Bundles

Bundles are the basic “envelope” to organize and manage modifications.
For more information, see Chapter “Bundle Objects”.

Patches/Modules

In addition to bundles, SCM provides the concepts of Patches and Modules. Patches and Modules are a set of bundles. For example, you might create a patch for a rollout which includes several independent bundles from different application areas such as document management, item management, or project management.

For more information, see Chapter "Managing Patches and Modules".

Data Analysis Tools

SCM also provides data analysis tools. The "where-used" for a specific object, shows whether the object is part of any other bundle. Consultants and administrators use this feature to assure that the correct, consolidated modification makes it into the bundle. The "Check Structure" feature reviews the structure of the current object and adds the subordinate objects to the current bundle.

In addition to adding objects to a bundle, you can enhance the bundle with objects of type INF. INF objects can be used to structure the content of the bundle or they can be used to include further instructions for handling the bundle.

Check Structure

The "Check Structure" feature reviews the structure of the current object and adds the subordinate objects to the current bundle. Check Structure will consider, for example:

- Menus
- (Sub-) Forms
- Tables
- Logiview procedures
- Messages

Import

Opens a dialog to specify the loader file name and the path to import the loader file

All Modifications

Opens a list of all modifications which allows you to run where-used queries across all bundles in the current environment.

Loader Imports

Opens the list of Loader Imports and shows which packages have been imported in the current environment.

Configuration

Opens the SCM configuration list.

SCM Toolbars, Menus, and Tabs

This chapter describe the SCM toolbars, menus, and tabs.

SCM Toolbar

The SCM Tool includes a toolbar which provides access to the most important objects/features.

Figure 2–1 *SCM Toolbar*



After you select a bundle from the drop-down list, or you click the **Set Active** button on the bundles form, the SCM toolbar drop-down list displays the selected bundle, as shown below:

Figure 2–2 *SCM Toolbar, bundle selected*



Table 2–1 *SCM Toolbar functions*


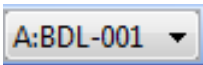

SCM toolbar function	Description
Patches	Opens the Patches/Modules form
Patch L	Opens the Patches/Modules list
Bundles	Opens the Bundle form
Bundle L	Opens the Bundles list
 (pencil icon)	Opens the modification list of the current bundle
 (toolbar drop-down list)	Shows the currently selected bundle and allows to switch to another bundle

Table 2–1 (Cont.) SCM Toolbar functions

SCM toolbar function	Description
 (loader icon)	Opens the Import dialog

SCM from the Manager Menu

You can access SCM functionality in the Manager Menu.

Manager > Tools > SCM - System Customization Management >...

Table 2–2 Manager Menu SCM functions

Function	Description
Bundle	Opens the Bundles form.
Patches	Opens the Patches/Modules form.
Check Tables	Used to detect and correct deviations between the table definition in the repository and the table definition in the database.
Get Objects	Used to search for object and then add them to the current bundle.
Import	Opens a dialog to specify the loader file name and the path to import the loader file.
All Modifications	Opens a list of all modifications.
Loader Imports	Opens the list of Loader Imports and shows which packages have been imported in the current environment.
Configuration	Opens the SCM configuration list.

Standard Menus

The following topics describe the menu functions in SCM.

Non-select and Select Standard Menus

Standard menu functions are available in icon menus in SCM.

Non-select menu functions are available when there is no object selected.

Select menu functions are available when a table row is selected.

SCM Context Menus

Beside the standard menu functions e.g. **Refresh**, **Search** etc. the context menu includes the following SCM-specific functions:

Table 2–3 Context menu functions

Context menu function	Description
List	Opens the bundle List view.
Change Status	Changes the status of the current bundle.
All Modifications	Opens the list to search for bundles which include a specified object.
Create Parent Entries	Create parent entries (object type FAT) in the modification list for the modification records in the bundle.
Check Import	Verifies that the import was completed successfully.
C_ID Check	After importing a bundle into current environment, this function updates the C_IDs in the SCM change log with current C_IDs.
Set Active	Defines the selected bundle as the current package. SCM modifications are then added into this bundle.
Import	Opens file selection dialog to select a SCM control file for import.
Export	Initiates the export of the current bundle.

SCM Settings

Defaults

Table 3–1 Default Settings

Name	T	Value	Description
EDB-SCM-CHG-LCK-CHGS	L	TRUE or FALSE	Defines if Ctrl-B key combination will set reservation for the current record. For user EDB-SCM default is set to TRUE. For GLOBAL default is set to FALSE
EDB-SCM-EXC-SQL-CMD	L	TRUE or FALSE	Defines if the SQL scripts are executed automatically during import process. By default (FALSE) the user has to start executing the SQL statements manually.
EDB-SCM-IMP-ERR	S	CONTINUE or EXIT	If parameter is set to "CONTINUE", the import process will continue even if errors occurred. If set to "EXIT" the import will be cancelled after an error.
EDB-SCM-STO-MTY-AUTO	S	TERA	Parameter defines the Object types for which the system will track changes in SCM automatically.

LGV Constants

Table 3–2 LFV Constants

Name	T	Value	Description
EDB_SCM_C_EIC_FLG	L	FALSE TRUE	Execute Export/Import Check.
EDB_SCM_C_FAT_FLG	L	TRUE FALSE	Write parent information to bundle.
EDB_SCM_CHG_DOC_AREA	S	"TEST"	Vault to use for SCM loader files during bundle release.

Table 3–2 (Cont.) LFV Constants

Name	T	Value	Description
EDB_SCM_CHG_DOC_PFX	S	"SCM-BDL-"	Prefix for document id, when value starts with # then the number server is used.
EDB_SCM_CHG_DOC_TYPE	S	"TEXTFILE"	"doctype[maskname]" Document type used for document record with loader files.
EDB_SCM_CHG_GRP_ONLK	L	TRUE FALSE	Change group Id with user Id when locking records
EDB_SCM_CHK_VERSION	S	"FALSE"	Check object versions: FALSE: no check TRUE: warnings when false ERROR: abort when false
EDB_SCM_CRE_DELETE	L	TRUE FALSE	Create SQL-delete commands when creating SQL-script
EDB_SCM_CRE_SQL_FILE	L	TRUE FALSE	Create SQL file
EDB_SCM_DBI_LOA_TAB	L	TRUE FALSE	Flag whether dbi_loa_tab is supported or not
EDB_SCM_DSC_MAS	L	FALSE TRUE	Create mask description (userexits, menus) for compatibility check
EDB_SCM_DSC_TAB	L	FALSE TRUE	Create table description (fields) for compatibility check
EDB_SCM_EXP_DSC	L	TRUE FALSE	Export description records
EDB_SCM_EXP_LOG	S	"/FILE /MASK"	Export log is written to /FILE: file /MASK: Message window
EDB_SCM_FIL_NAM_DEF	S	"@SCM-ID"	Default value for loader file name. @EXP-ID, @SCM-ID, text
EDB_SCM_IMP_DEL_ENT	L	FALSE TRUE	Delete entity record on import (e.g. T_MASK, T_MENU). This should be set to FALSE. Entity record will be modified by loader import and needs not be deleted prior to import. The deletion of relation records is not affected by this switch.

Table 3–2 (Cont.) LFV Constants

Name	T	Value	Description
EDB_SCM_ IMP_EXEC_ FLG	S	"ASK"	ASK: Ask for Executing EXEC commands EXEC: Execute without asking Other: Don't You can control the behavior for execution of LogiView commands which are stored in the bundle as type "EXEC".
EDB_SCM_ IMP_LOG	S	"/FILE /MASK"	Import log is written to /FILE: file /MASK: Message window
EDB_SCM_ LGV_HIE_ SON	S	"LV_NULL LV_DT_ MET-CRO"	Open parameter for opening LGV hierarchy
EDB_SCM_ LGV_SYS_ INI	L	TRUE FALSE	Initialize LGV system variables when first called (don't when FALSE). If system variables are not initialized, every time CTRL-A is pressed, the administration data of this tool must be loaded from database. This will last some seconds. So if possible, this value should be set to TRUE.
EDB_SCM_ LGV_USE_ CRO	L	TRUE FALSE	Use LV_DT_CRO table for cross reference
EDB_SCM_ LOG_VIR	L	TRUE FALSE	First print log messages to virtual widget
EDB_SCM_ MAS_OPN_ USX	S	"iwf_lis_ent, iwf_frm_ent, iwf_lis_ref, iwf_lis_agg, bvb_usx_opn_ent, bvb_usx_opn_typ"	Userexits for opening masks
EDB_SCM_ MAX_ADM	I	200	Maximum number of entries in SCM configuration list which is supported. (Manager > Tools > SCM > Configuration)
EDB_SCM_ MAX_OBJ	I	9999	Maximum allowed number of modification records handled by SCM.
EDB_SCM_ NODEL_ CHGTYP	S	"LCK"	Change record types that cannot be deleted by user
EDB_SCM_ NOUPD_ CHGTYP	S	"LCK"	Change record types that cannot be modified by user
EDB_SCM_ REC_USX_ NAM	S	"EDB_DEV/RecClsMas"	Userexit to add to userexits when creating save mask
EDB_SCM_ REC_USX_ TYP	S	"AM"	Type of userexit to add to userexits when creating save mask

Table 3–2 (Cont.) LFV Constants

Name	T	Value	Description
EDB_SCM_STO_AUTO	S	A	Store Change: A (Auto), M (Manual), AF (Auto Forced), C (Changed), D (Deleted), CD (Change Deleted), AD (Auto Deleted), AFD (Auto Forced Deleted) Store Change: A - automatic, when SCM post mask userexit is installed M - manual, only with CTRL-A AF - (Auto Forced) when no bundle is set, you will be asked for one to store the modifications
EDB_SCM_STOCHG_USX	S	"EP_SCM/ScmStoChgPoa"	PostAction Userexit for store of modifications
EDB_SCM_TEST	L	FALSE TRUE	Write test output to message window
EDB_SCM_TRACE_ALL	L	FALSE TRUE	Trace all procedures
EDB_SCM_XPRG_FLG	L	TRUE FALSE	Use Progress reporter (TRUE)
EDB_SCM_XPRG_THEME	S	"Patch/Change Management"	

Number Variant

Manager > Tools > Number Server

Number variant SCM-EXP-ID

When you transfer the customization from one environment to another, then the number variant prefix must be different in the different environments. This is checked during import of the SCM loader files.

So you have to adapt/replace prefix "SCM-DEV-" when you have more than one environment. Suggestion: Use "SCM-TST-" for your testing environment, "SCM-PRD-" for your production environment.

How SCM Works

This chapter provides an overview of how SCM works.

Setup and Configure Source and Target Environments

The “source” application is where you do the customization. The “target” application is where you want to apply the customizing.

Before you can execute Export of Import, you must first setup or configure both the source application and the target application.

Overview of Source Application Setup

1. Define a vault where the loader files are checked in.
2. Set the LogiView constant EDB_SCM_CHG_DOC_AREA
3. Define the directory where the loader files are stored (EDB-FMS-ALLOWED-PATHS)
4. Update the number variant for SCM. For more information see

For more information, see “Configuring the Source Environment” on page 5-1.

Overview of Target Application Setup

1. Update the number variant for SCM
2. Set the LogiView constant EDB_SCM_CHG_DOC_AREA
3. Define the directory where the loader files are stored (Refine the number server, number variant. EDB-FMS-ALLOWED-PATHS)

For more information, see “Configuring the Target Environment” on page 5-2.

Select / Identify Modifications

There are several methods to assign an object to a bundle.

- Automatically by triggers

SCM comes with triggers which automatically assign an object to the current bundle if the object has been changed. These triggers must be included as post action triggers in the corresponding (definition) forms.
- Manually by the user

You can assign the current object to the current bundle by pressing Ctrl-A

- Automatically by searching for changes authored by a certain user and created or modified since a specified date and time.

The difference between CTRL-A and Post-Action Userexit is:

- CTRL-A

If no bundle is active you are asked for the active/current bundle.

The binary file has a variant number in the file name.

- Post-Action Userexit

If no bundle is active, no changes recorded.

There is no variant number in the file name.

CTRL-A

The basic CTRL-A process is:

1. Define a bundle where you book all modifications.
2. Set the bundle to active.
3. Perform customization tasks and track changes.
4. Use CTRL-A to book modifications to the bundle.

Export Modifications from the Source Environment

See Chapter 9, "Export Modifications from the Source Environment."

Import Modifications into the Target Environment

See Chapter 10, "Import Modifications into the Target Environment."

Configuring the Source Environment and the Target Environment

Before you can use SCM to transfer content from the source environment to the target environment, you must first configure both the source and target environments.

Note: In order to work with Agile e6, the user EDB-SCM must be activated. For security reasons, the user EDB-SCM is not activated in a new Agile e6 application. The EDB-SCM user must be activated by the Administrator by either setting a password or by configuring the LDAP login for the EDB-SCM user.

For more information about how to activate a user, see the *Administrator Guide for Agile e6*, "Setting and Changing Initial Passwords in a New Agile e6.2.0.0 Application."

Configuring the Source Environment

Configure the source application where you define the customization code.

Note: A valid vault must exist before you begin this procedure.

1. Define which vault is used by SCM to store created files.

Configure the LogView Constant EDB_SCM_CHG_DOC_AREA to a valid vault name.

For more information about how to create a valid vault, refer to the *Administrator Guide for Agile e6*.

Choose **Manager > Macro > LogiView > Declarations > Constants**.

The Constants tab is displayed.

Search for the EDB_SCM_CHG_DOC_AREA constant. Change the value from "TEST" to the valid vault name you have defined, for example, "SCMFiles".

2. Register the directory to store SCM files as allowed directories for server-side checkout/checkin.

For more information see *Administrator Guide for Agile e6*, section/chapter "File Server Administration."

In the context of SCM, normally the application dump directory is used (%ep_root%\<app_name>\dmp,)

You must define the directory where the loader files are stored as an allowed directory.

Choose **System > Other Parameters**.

The Configuration form is displayed.

Create a new parameter and enter a name in the Rubric/Parameter field, for example, EDB-FMS-ALLOWED-PATHS.

On the Configuration Parameter tab, add a new directory, for example, EDB-FMS-UNIX-DMP-DIR. In the Configuration dialog, enter the appropriate path for your installation.

Select the row you just added in the Configuration dialog and choose **Take Entry** in the context menu to assign the parameter to the rubric.

3. Update the number variant for SCM.

Choose **Manager > Tools > Number Server**.

Search for SCM, and then create a new number variant by editing all occurrences of the name of SCM-DEV to SCM-SRC. Save it.

Configuring the Target Environment

Configuring the target environment is very similar to configuring the source environment.

For more information about how to create a valid vault, refer to *Oracle Agile Engineering Data Management Administration Guide for Agile e6.2.1.0*.

1. Define which vault is used by SCM to store files from the source environment.

Configure the LogView Constant EDB_SCM_CHG_DOC_AREA to a valid vault name.

For more information about how to create a valid vault, refer to the *Oracle Agile Engineering Data Management Administration Guide for Agile e6.2.1.0*.

Choose **Manager > Macro > LogiView > Declarations > Constants**.

The Constants tab is displayed.

Search for the EDB_SCM_CHG_DOC_AREA constant. Change the value from "TEST" to the valid vault name you have defined, for example, "SCMTARGETFILES".

2. You must define the directory where the loader files are stored as an allowed directory.

Choose **System > Other Parameters**.

The Configuration form is displayed.

Create a new parameter and enter a name in the Rubric/Parameter field, for example, EDB-FMS-ALLOWED-PATHS.

On the Configuration Parameter tab, add a new directory, for example, EDB-FMS-UNIX-DMP-DIR. In the Configuration dialog, enter the appropriate path for your installation.

Select the row you just added in the Configuration dialog and select **Take Entry** in the context menu to assign the parameter to the rubric.

3. Update the number variant for SCM.

Choose **Manager > Tools > Number Server**.

Search for "SCM-EXP-ID", and then update the number variant by editing all occurrences of the name of SCM-DEV to SCM-TRG. Save it.

Note: For the Target environment, you must use a different prefix value for number server "SCM-EXP-ID" than you defined for the Source environment.

Configuring Modification Types for SCM

In the source system, you define the types of customization objects than can be assigned to a bundle.

Agile e6 SCM is delivered with a preconfigured list of customization object types (Agile e6 Masks).

You can review the list of customization object types:

Choose **Manager > Tools > SCM - System Customization Management > Configuration**

Define and Add Customization Modification Objects

The customization objects recognized by SCM are defined in the SCM Configuration.

You can define your own customization type objects and then add those objects to the Configuration list.

1. Open the Configuration list:

Choose **Manager > Tools > SCM - System Customization Management > Configuration**.

2. In the SCM Configuration list, you define the mask names.

Refresh the tab to see the names supplied with the standard delivery.

The masks are grouped by the rubrics and define the data models associated with each rubric.

Bundle Objects

Bundles are the basic "envelope" to organize and manage modifications. Bundles are used to bundle customizing modifications. A bundle might hold several application objects, for example, a mask object (EDB-ART-SLI). These objects are exported to a loader file at the time the bundle is exported. Before you can begin tracking any modifications with SCM, a bundle must exist and needs to be set "current".

A Bundle consists of a list of changed objects, which have been grouped together. Each Bundle is defined by:

- Bundle Number
- Bundle Description
- Short Description
- Creator
- Creation Date
- Release Date

Table 6–1 *Bundle general information attributes*

Bundle definition attributes	Description
Bundle Number	Unique identifier
Short Description	Description of the bundle
General Description (Request)	Description that explains why the customization is needed
Detailed Description (Solution)	Description of how the customization request has been solved and resolved
Creator	The user who created the bundle
Start Date	When the bundle was created
Release Date	The release date of the bundle
PI (Progress Indicator)	The release state of the bundle. The available states are defined in the lifecycle EDB-SCM
Lifecycle	Lifecycle name
Valid From, Valid Until, Current	Lifecylce fields that define the validity period

Bundle Form

To open the Bundle form, perform one of the following task:

- In the Manager Menu, choose **Tools > SCM > Bundle**.
- Click the **L** button beside the **Bundle** button on the SCM Toolbar to display the bundle list. See "SCM Toolbar" on page 2-1.

Figure 6–1 Bundle Form

Table 6–2 Bundle Form Select menu functions

Menu function	Description
List	Opens the bundle list.
All Modifications	Opens list of all modifications for all bundles.
Create Parent Entries	Create parent entries (object type FAT) in the modification list for the modification records in the bundle.
Check Import	Verifies if the import completed successfully. This feature compares the “as-is” update date and/or the number of objects for the imported objects with the “to be” update date and number of objects documented in the export file. Be aware, the corresponding information has to be logged in export file.
Check C_ID	After importing a bundle into current environment, this function updates the C_IDs in the SCM modification with current C_IDs. This is a prerequisite to re-export the bundle from current environment and to open the object with “Show” command in the current environment.
Set Active	Defines the selected bundle as the current active bundle, so that all SCM modifications will be booked onto this bundle.
Import	Opens the dialog to select SCM control file (.loa.bin) for import.
Export	Starts exporting the selected bundle.

General Information Tab, Bundle Form

The General Information tab defines attributes for the bundle.

Table 6–3 Bundle Form General Information tab

Tab Attribute Name	Description
General Description (Request)	General description.
Detailed Description (Solution)	Detailed description.
Memo	Additional notes.

Modifications Tab

A modification record represents the changed records which will be exported and other meta information for the changed records.

Figure 6–2 Modifications tab

Position	Type	Object Type	Object Name	Object CID	Object Version	Modification Description	Object Key1	Object Key2	Object Key3	Object Key4	Object Key5	Object Key6	Object Key7	Object Key8	Object Key9	Subobject	Old Value	New Value
30 LOA	MES		MY-NEW-MESSAGE	1015559335	1		MY-NEW-MESSAGE											
30 LOA	TAB		MY-ENTITY	1001380670	1		MY-ENTITY											
30 LOA	ENT-REL	EDB-ARTICLE STR		1000000000	3		EDB-ARTICLE	STR										
40 FAT	ENT-REL	EDB-ARTICLE		1000000001	5		EDB-ARTICLE									EDB-ARTICLE STR	1000000000	
50 LOA	MAS-FLD	BVB-ARTIKELARTTYP 50		1525584618	2		BVB-ARTIKELARTTYP 50											
60 FAT	MAS-FLD	BVB-ARTIKEL SLI		1948677244	18		BVB-ARTIKEL SLI										BVB-ARTIKELARTTYP 50 1525584618	

Modifications include the following fields.

Table 6–4 Modifications tab fields

Modification field	Description
Position	Unique identifier number.
Type (modification type)	For a list of modification types, refer to "SCM Modification Types" on page 6-4.
Object Type	Key for object as defined in column "Object Type" in the SCM configuration list.
Object Name	Concatenated values from key fields as defined in columns "Field 1" to "Field 9" in the SCM configuration list (max. 80 characters).
Object CID	C_ID value of changed/new record
Object Version	C_VERSION value of changed/new record
Modification Description	Text to describe modification
Object Key1 to 9	Values of key fields of changed record or new record. In the SCM configuration list for each entry the key fields are defined in columns "Field 1" to "Field 9". In each modification record the corresponding values for these fields are stored in columns "Object Key1" to "Object Key9".
Subobject	For records of object type "FAT", this field contains the concatenated object keys for the corresponding relation record.
Old Value	Old value
New Value	New value

Modifications List

Table 6–5 Modifications list Select menu functions

Menu function	Description
Usage	Show the bundles that include the current object.
Show	<p>Opens the corresponding form to show the object details. Right now this userexit is only supported for records with object type.</p> <ul style="list-style-type: none"> ■ TAB ■ TAB-FLD ■ MAS ■ MAS-FLD ■ MEN ■ LV-DT-PRC ■ LV_DT
Set Type to "LOA"	Converts an entry of type INF into an entry of type LOA and fills the mandatory fields accordingly.
Check C_ID	After importing a bundle into current environment, this function updates the C_IDs in the SCM modification with current C_IDs. This is a prerequisite to re-export the bundle from current environment and to open the object with "Show" command in the current environment.
Check Table	Checks for inconsistencies between table definition in the repository and database.
Check Structure	Checks the structure of modification records.
LGV duplicate Line No.	Search for duplicate position numbers in LogiView.

SCM Modification Types

When you add a modification to a bundle number, several modification entries are inserted, depending on LGV constants or default value settings.

The different type of modifications (T_SCM_CHG_OBJ) assigned to a bundle number are

Table 6–6 SCM Modification types

Modification Type	Object Type	Remark
DEL	As defined in the SCM configuration list e.g. MAS, MEN, MAS-FLD	LGV constant EDB_SCM_STO_AUTO = D or CD or AD or AFD

Table 6–6 (Cont.) SCM Modification types

Modification Type	Object Type	Remark
DSC	MAS.UBM	LGV constant EDB_SCM_STO_AUTO = A (Auto)
	MAS.UAM	If LGV constant EDB_SCM_DSC_MAS is set to TRUE
	MAS.USA	For object type "MAS"
	MAS.UEA	--
	MAS.UBA	Store each premask userexit/parameter for compatibility check (UBM). OBJECT_KEY1: Premask Userexit OBJECT_KEY2: Userexit Parameter
	MAS.LAY	--
	MAS.SIG	Store each postmask userexit/parameter for compatibility check (UAM). OBJECT_KEY1: Postmask Userexit OBJECT_KEY2: Userexit Parameter
	MAS.MEN	--
	MAS.BUT	Store each select action userexit/parameter for compatibility check (USA). OBJECT_KEY1: Select Action Userexit OBJECT_KEY2: Userexit Parameter
		--
		Store each edit action userexit/parameter for compatibility check (UEA). OBJECT_KEY1: Edit Action Userexit OBJECT_KEY2: Userexit Parameter
		--
		Store each pre action userexit/parameter for compatibility check (UBA). OBJECT_KEY1: Pre Action Userexit OBJECT_KEY2: Userexit Parameter
		--
		Store each post action userexit/parameter for compatibility check (UAA). OBJECT_KEY1: Posmask Userexit OBJECT_KEY2: Userexit Parameter
		--
		Store mask layout part1 for compatibility check (LAY). OBJECT_KEY1: T_MASK.C_ROW (row) OBJECT_KEY2: T_MASK.C_COL (column) OBJECT_KEY3: T_MASK.C_LEN (length) OBJECT_KEY4: T_MASK.C_WID (width)
		--
		Store mask layout part2 for compatibility check (SIG). OBJECT_KEY1: T_MASK.C_SIGNUM (significant number of columns) OBJECT_KEY2: T_MASK.C_MAXREC (maximum records in mask) OBJECT_KEY3: T_MASK.C_HOR_BAR (horizontal scrollbar) OBJECT_KEY4: T_MASK.C_VER_BAR (vertical scrollbar)
		--
		Store mask menus for compatibility check (MEN). OBJECT_KEY1: Select Menu OBJECT_KEY2: Non Select Menu OBJECT_KEY3: Edit Menu
		--
		Store mask buttons for compatibility check (BUT). OBJECT_KEY1: Select Button OBJECT_KEY2: Non Select Button OBJECT_KEY3: Edit Button

Table 6–6 (Cont.) SCM Modification types

Modification Type	Object Type	Remark
DSC	TAB.FLD	<p>LGV constant EDB_SCM_STO_AUTO = A (Auto)</p> <p>If LGV constant EDB_SCM_DSC_TAB is set to TRUE</p> <p>For object type "TAB" (You get object type "TAB" only when you call it from Dataclasses widget)</p> <p>--</p> <p>Store field definition for compatibility check (TAB.FLD).</p> <p>For each field of the table a TAB.FLD record is created:</p> <p>OBJECT_KEY1: Field Name</p> <p>OBJECT_KEY2: Field Type (in DB format) + NOT NULL (if NN flag set)</p> <p>OBJECT_KEY3: Index Type + Index Name (not working correctly when more than one index defined like in T_CFG_DAT)</p> <p>OBJECT_KEY4: Sequence</p>
EXEC		<p>LGV constant EDB_SCM_IMP_EXEC_FLG = "ASK" or "EXEC"</p> <p>Execute EXEC commands of bundle listed in column Object Name [T_SCM_CHG_OBJ.OBJECT_NAME] (Can be called from bundle context menu: Execute EXEC)</p>
FAT	The object type for the relation e.g. MAS-FLD	<p>If LGV constant EDB_SCM_C_FAT_FLG is set to TRUE:</p> <p>For refine or aggregate object relations, store parent record information to the loader file to resolve the exported relations in the target environment.</p> <p>OBJECT_KEY1: Key value for first key field of parent entity</p> <p>OBJECT_KEY2: Key value for second key field of parent entity (if defined)</p> <p>OBJECT_KEY3: ...</p>
INF		<p>LGV constant EDB_SCM_STO_AUTO = A (Auto)</p> <p>You can add an INFO record manually and put a text to column "Modification Description" to provide hints for users who use your SCM loader files.</p>
INF	Mask name	<p>LGV constant EDB_SCM_STO_AUTO = A (Auto)</p> <p>If CTRL-A is called in list or form that is not configured in the SCM configuration list then an info record is added to the modification list:</p> <p>Object Name: "MASK: maskname CID: c_id of record"</p>
INS	As defined in the SCM configuration list e.g. MAS, MEN, MAS-FLD	<p>LGV constant EDB_SCM_STO_AUTO = C</p> <p>Insert record in list where SCM postaction EP_SCM/ScmStoChgPoa(...) has been added.</p> <p>Entries are not exported during Export. You have to change to type LOA before.</p>

Table 6–6 (Cont.) SCM Modification types

Modification Type	Object Type	Remark
LOA	As defined in the SCM configuration list e.g. MAS, MEN, MAS-FLD.	<p>LGV constant EDB_SCM_STO_AUTO = A or AF</p> <p>The object record with Object CID will be exported to the loader file during export.</p> <p>OBJECT_KEY1: Key value for first key field of the record</p> <p>OBJECT_KEY2: Key value for second key field of the record (if defined)</p> <p>OBJECT_KEY3: ...</p>
STR	MEN	Inserted by "Check Structure".
	TAB	Inserted by "Check Structure".
	LGV-DT	Inserted by "Check Structure".
UPD	As defined in the SCM configuration list e.g. MAS, MEN, MAS-FLD	<p>LGV constant EDB_SCM_STO_AUTO = C or CD</p> <p>Update record in list where SCM postaction EP_SCM/ScmStoChgPoa(...) has been added</p> <p>Entries are not exported during Export. You have to change to type LOA before.</p>

In Patches Tab

Displays the patches to which the bundle is assigned.

Table 6–7 In Patches Tab Select menu functions

Menu function	Description
Form	Opens the Patch form and shows the patch data.
Bundles	Opens the list of bundles assigned to patch.
Export	Starts exporting the selected patch.

Loader Tab

The Loader tab displays information about when the current bundle was imported and exported. The types of information recorded include:

Documents Tab, Bundle Form

The Documents tab lists the documents associated with the bundle form. You can search for documents and add documents on this tab.

Exp/Imp Check Tab

When LGV constants EDB_SCM_C_EIC_FLG and EDB_SCM_DSC_MAS and EDB_SCM_DSC_TAB are set to TRUE in the source environment then Export/Import Check

(EIC) records are created. This allows you to check in the target environment for inconsistencies after import (update date for record is stored).

History Tab

The History tab records standard history information.

Bundle List

To open the Bundle list, perform one of the following tasks:

- Choose **List** in the Bundle form context menu.
- Click the **Bundle** button on the SCM Toolbar. See "SCM Toolbar" on page 2-1.

Table 6–8 Bundle List Non-select menu functions

Menu function	Description
Form	Opens the Bundle form.
All Modifications	Opens the list of all modifications for all bundles.
Get Objects	Get all modifications done by user or/and by creation / update date.
Import	Opens the dialog that allows you to select the SCM control file for import.

Table 6–9 Bundle List Select menu functions

Menu function	Description
Form	Opens the bundle form for the selected record.
Modifications	Opens the modification list for bundle.
In Patches	Opens the list with patches where bundle is assigned to.
Exp/Imp Check	When LGV constant EDB_SCM_C_EIC_FLG is set to TRUE in the source environment then EIC records are created to be able to check in the target environment for inconsistencies after import (update date for record is stored and number of refine records).
In Other Bundles	Lists other bundles which also contain the modified object.
Create Parent Entries	Create parent entries (object type FAT) in the modification list for the modification records in the bundle.
Check Import	Verifies if the import completed successfully. This feature compares the "as-is" update date and/or the number of objects for the imported objects with the "to be" update date and number of objects documented in the export file. Be aware, the corresponding information has to be logged in export file.
Check C_ID	After importing a bundle into current environment, this function updates the C_IDs in the SCM modification with current C_IDs. This is a pre-requisite to re-export the bundle from current environment and to open the object with "Show" command in the current environment.

Table 6–9 (Cont.) Bundle List Select menu functions

Menu function	Description
Execute EXEC	Execute EXEC commands (LogiView procedures) of bundle listed in column "Object Name"
Set Active	Defines the selected bundle as the current active bundle, so that all SCM modifications will be booked onto this bundle.
Export	Starts exporting the selected bundle.

Managing Bundles

Collect Modifications

Modifications are listed in the Modification list of a bundle. The entries in the list can be different modification types.

There are several methods to assign an object to a bundle.

- Automatically by triggers
SCM comes with triggers which automatically assign an object to the current bundle if the object has been changed. These triggers must be included as post action triggers in the corresponding (definition) forms.
- Manually by the user
You can assign the current object to the current bundle by pressing Ctrl-A
- Automatically by searching for changes authored by a certain user and created or modified since a specified date and time.

Methods to Record Modifications

There are three methods to record modifications:

- Select the object row and press Ctrl-A.
- Post-Action Userexit (see "Identify Modifications with Post Action UserExit" on page 8-2.)
- Get Objects
Determines which objects have changed and adds them to the bundle. Choose **Manager > Tools > SCM > Get Objects**.
Get Objects is limited to 9999 objects per table.

Release Procedure for Bundles

Bundle objects can have several states:

- 110 In Work -> 230 Released
- 230 Released -> 240 In Change
- 240 In Change -> 230 Released

When creating a bundle definition, it begins in the state In Work. In this state, objects can be added and data can be modified.

When a Bundle is released, the bundle definition cannot be modified. All reservations on this bundle are reset and the original owner of the objects is restored.

Also a document is created in the document management of Agile e6 which is linked to this Bundle number. (Document type and document id are customizable).

A loader file is generated automatically and this loader file is checked-in at the created document (used vault is definable).

A record is written to history folder which indicates the transition from In Work to Released.

All locks for objects assigned to the released bundle number are also released.

When changing the state to In Change, objects of the bundle definition can be modified or new objects can be added. This transition also is also documented in history.

View All

The **All Modifications** view is a collection of all objects belonging to any bundle. Use this view to find a specific object and to determine in which bundle it is contained.

Also, you can use this view to analyze and determine whether an object is included in different bundles, which may cause concurrency issues.

Table 6–10 *History tab, Bundle Form menu functions*

Menu function	Description
Form	Opens the bundle form for selected record.
Modifications	Opens the modification list of the bundle for selected record.
In Patches	Opens the patch list of where the bundle of the selected record is assigned to.
Exp/Imp Check	Export/Import Check See also "Exp/Imp Check Tab" on page 6-7.
Export	Begins exporting the bundle of the selected record.

Managing Patches and Modules

Patch/Module Management is used to collect bundles and then export them or import them as a whole set of information.

Use **Patch/Module Management** to combine several bundles and their modifications in one loader file. This function can manage patches, modules or working packages.

- Patches are used to package a set of modifications (bug fixes and new features) for roll-out.
- Modules contain all customization required to deliver a module with functionality.
- A Working Package is a bundle which is configured by an administrator to capture the modifications related to a specific work package with respect to the overall project.

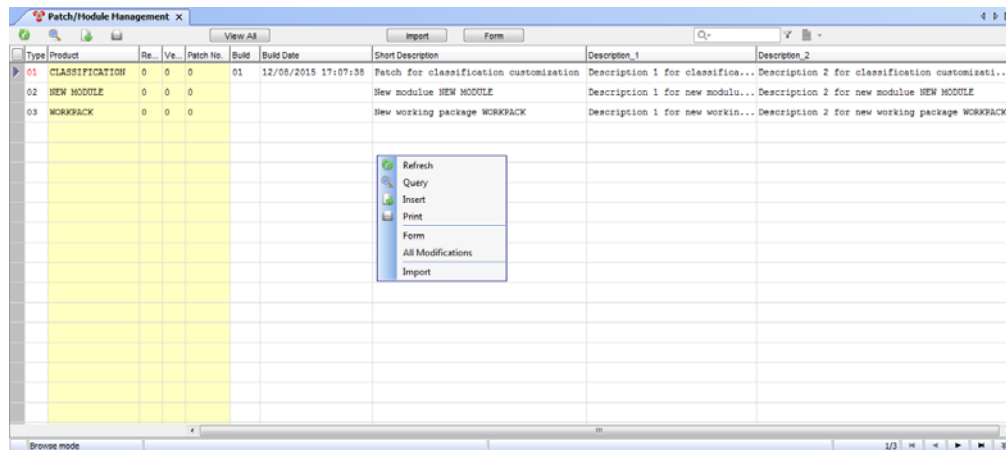
Patches

A patch is defined by:

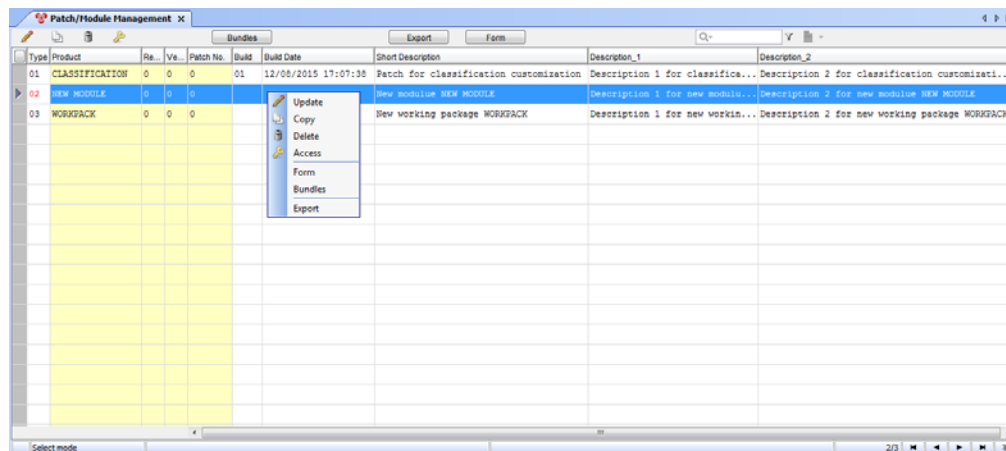
Table 7–1 Patch definition attributes

Patch attribute	Description
Type	01 - Patch type 02 - Module type 03 - Working Package type
Product	Name for patch
Revision	Revision number
Version	Version number
Patch No	Patch number
Build	Build number
Build Date	Build date when patch is exported
Short Description	A short description for the patch
Description 1	Additional description for the patch
Description 2	Additional description for the patch

Patch/Module Management Tab Menus

Figure 7–1 Patch/Module Management tab, no row selected**Table 7–2 Patch/Module Management tab non-select menu**

Menu function	Description
Form	Opens form for patches.
All Modifications	Opens list of all modifications for all bundles.
Import	Opens the dialog to select SCM control file (.loa.bin) for import.

Figure 7–2 Patch/Module Management tab, row selected**Table 7–3 Patch/Module Management select menu**

Menu function	Description
Form	Opens form for patches for the selected record. See "Patch/Module Management Form" on page 7-2.
Bundles	Opens list of all assigned bundles.
Export	Begins exporting the selected patch.

Patch/Module Management Form

The Patch/Module Management form is described below.

Figure 7–3 Patch/Module Management form
Table 7–4 Patch/Module Management form context menu

Menu function	Description
Update	Update selected row.
Copy	Copy selected row.
Delete	Delete selected row.
Access	
List	Opens the list of patches
All Modifications	Opens a list of all modification for all bundles
Export	Begins exporting the selected patch.

Locking and Reserving Objects (Ctrl-B)

To reserve an object for modification, select the object row and press Ctrl-B.

Although the parent object (for example, a mask) is locked by one user, related objects (for example, fields) can be inserted by users who have not locked the object.

The owner of the object is then set to the user who reserved the object.

When the bundle is released, all reservations are removed.

Transfer Content from the Source Environment to the Target Environment

The chapter illustrates an example of transferring content from the source environment to the target environment

Creating Customization Files

1. Log in to the Agile e6 source application.
2. For this example, choose Message Data.

Define a Bundle to contain all the modification. Choose Manager > Tools > SCM - System Customization Management > Bundles.

The customization files are created and are checked into the file vault (name is defined by LGV constant EDB_SCM_CHG_DOC_AREA).
3. In the Bundles form, create a new bundle. Enter a bundle number and a short description. Click **Set Active**.
4. Click the Message Data tab and refresh. Select the row you want to modify and press Ctrl-B to lock the record. Modify the message and click **Save**.
5. To add the message to the current active bundle, select the row and press Ctrl-A. On the Modifications tab, you can view the modifications you can view the modifications that you made.
6. To create a new message, clear the active bundle in the SCM toolbar. Insert a new row, create the message and save it.
7. Select the new message and press Ctrl-A. Because there is no active bundle selected in the SCM toolbar, a dialog appears that prompts you to select a bundle, and click Continue.
8. To export, click the **Export** button. The Entity Relations for Export dialog appears. Click **Continue**. Click **Yes**.
9. The File Names dialog appears. It displays the loader file name and the path. Click **Continue**.

An export report message is displayed.

Example: Transfer Content by Releasing the Bundle

1. On the Bundle form, change the status of the bundle from In Work (110) to Released (230).
Click the change status icon (Change status of the current object).
2. The "Please choose an option" dialog appears. Released is selected. Click **OK**.
3. In the next dialog, you can select the validity range: a Valid From date (defaults to the current date) and a Valid Until date. Click **OK**. The files are checked in.

Importing Customization Files into the Target Environment

1. Prerequisite: Copy the SCM files created by the "Export" from the dump directory of the source environment to the dump directory of the target environment.
2. Log in to the target environment.
3. Choose **Manager > Tools > SCM > Import**. The File Names dialog appears.
4. Enter the file name you want to import in the Loader field. Click **Continue**. Click **Yes**, then click **Yes** again. The file is imported, and an import report is displayed.
5. Refresh the Message Data tab to view the imported customizations.

In the target environment, after import, you can check for inconsistencies by using the bundle list/form context menu function **Check Import**.

Identify Modifications with Post Action UserExit

An active bundle must be set to use Post Action UserExit.

1. Log in to the source environment.
2. Create a new bundle. Set the bundle to Active.
3. Determine modification you wish to include.
4. Choose **Manager > Tools > SCM > Configuration**.

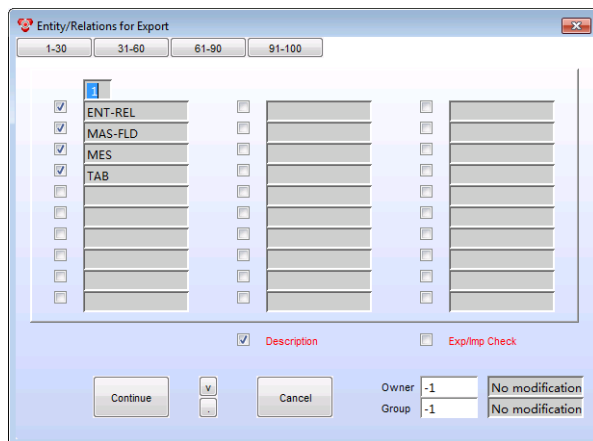
Export Modifications from the Source Environment

This chapter describes the export procedure, which exports the selected modifications or bundles to a loader file.

Export Procedure

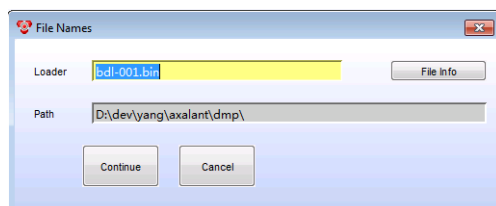
1. Begin exporting by choosing Export in the select menu or context menu.
2. Select modifications assigned to bundles or bundles assigned to patches to export.

Figure 9–1 Select entities to export



3. Click **Yes** when asked "Continue?"
4. Enter the loader file name and click **Continue**.

Figure 9–2 Enter Loader file name



A report is opened after a successful export.

Note: The export of a bundle can be executed directly by using the context menu Export (as described "Export Procedure" on page 9-1).

In addition, when you release a bundle, the release process includes the export of the bundle.

See "Release Procedure for Bundles" on page 6-9.

Import Modifications into the Target Environment

This chapter describes the import procedure, which imports the loader file into the Target application.

Import Procedure

If the bundle contains database table modification (for example, adding a new field to a table), then the import process provides the necessary DB SQL statements, which can be executed from within the SCM user interface (**Select > Execute SQL Commands**) or the SQL statements are executed automatically (default EDB-SCM-EXC-SQL-CMD = TRUE).

1. Prerequisite: Copy the SCM files created by the "Export" from the dump directory of the source environment to the dump directory of the target environment.
2. Choose **Manager > Tools > SCM - System Customization Management > Import**. The File Names dialog appears.
3. Enter the file name you want to import in the Loader field. Click **Continue**. Click **Yes**, then click **Yes** again. The file is imported, and an import report is displayed.
4. Refresh the Message Data tab to view the imported customizations.

Note: If you attempt to import the same loader file again, you will see the error message:

"Prefixes of numbers in source and target system must be different."
and the import is canceled.

You must delete the loader import records (**Manager > Tools > SCM - System Customization Management > Loader Imports**) first before you are able to import the file into the same environment again.
