Oracle® Agile

Engineering Data Management

Upgrade Tool for Agile e6.1.2.2

Part No. E27829-01

April 2012

Copyright and Trademarks

Copyright © 1995, 2012, Oracle and/or its affiliates. All rights reserved.

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

U.S. GOVERNMENT RIGHTS
Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on 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. 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.

CONTENTS

Copyright and Trademarks iii
Preface vii

Introduction 1

Overview 1
Prerequisites 1
How it Works 2

Installation and Configuration 5

Database Settings 5
- Oracle Parameters 5
- SQL Net Configuration 6
- Global Database Name 6
- Service Name 6
- listener.ora 7
- sqlnet.ora 7

Character Scan 7
Pre-Activities on the Original Production Environment 9
Import Dumps 10
- Import Source Reference Dump 10
- Import Customer Dump 11
- Create Statistics for All Involved Database Schemas 11

Run Upgrade Tool 12
Configuring the Upgrade Tool 12
- Configuring New Database Environment Connections 12
- Configuring Production Dump Connection 14
- Setting Upgrade Tool Parameters 15
- Save Configuration 17

Perform the Upgrade 19

Perform the Upgrade in Interactive Mode 19
- Step 1: Run Pre-Action-Scripts 19
- Step 2: DTV-Upgrade 20
- Step 3: Run Before-Sync-Scripts 20
- Step 4: Synchronize_Repository 21
- Step 5: Run After-Sync-Script 24

Step 6: Run Before-Common-Scripts 25
Step 7: EDB-Upgrade (Configuration) 25
Step 8: BRW-Upgrade (Browser) 25
Step 9: DODE-Upgrade (Print Studio) 25
Step 10: LGV-Upgrade (LogiView) 26
Step 11: WFL-Upgrade (Workflow) 26
Step 12: CHG-Upgrade (Change Management) 27
Step 13: GTM-Upgrade (Classification) 27
Step 14: GDM-Upgrade (Office Suite) 28
Step 16: RMT-Upgrade (Requirement Management Traceability) 28
Step 17: ASE-Upgrade (Advanced Structure Editor) 28
Step 18: CMG-Upgrade (Configuration Management) 29
Step 19: Run After-Common-Scripts 29
Step 20: Upgrading Classification (optional) 29
Step 21: Classification Attribute Inheritance 30
Step 22: Special Replace 30
Specific Dump Changes 30
Repeat Upgrade in Batch Mode 31
Perform Upgrade in Batch Mode 31
Perform Upgrade in Compact Batch Mode 32
Convert XML Files to HTML 33
Takeover Production Data 33
Define Reference Tables 34
Edit Production Table List 34
Perform Transfer 36
Taking Over Number Server Values 36
Post-Action Scripts 36
Classification Stage 2 (optional) 37
Manual Dump Changes 37
Additional Information 39
Browser Upgrade 39
Cleanup BVB_ARTIKEL 39
STA_LUT 39
Project References in Processes 40
BVB_ARTMEH Upgrade 40
Favorites Upgrade 40
Classification 40
Workflow Takeover 43

Appendix A: CMD Scripts 45

Appendix B: Configuration Files in /conf/ 49

Appendix C: Contents of the Folder "upgrade/conf/template" 51

Appendix D: SQL Scripts 53

Appendix E: Directories 57

Appendix F: Migration Rules 59

Standard Rules for Delete 59
Standard Rules for Update 59
Standard Rules for Insert 59
Special Rules 59

Appendix G: Repeatable Tasks 61

Appendix H: Error Messages 63

Preface

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

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

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

TTY Access to Oracle Support Services

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

Readme

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

Agile Training Aids

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Chapter 1

Introduction

Overview

This guide is intended as a manual for upgrading the Agile e-series products to Agile e6.1.2.2 with the Upgrade Tool.

The Upgrade Tool also allows a direct upgrade to Agile e6.1.2.2 from one of the following product versions. Please contact Oracle support, if you intend to do such a legacy version upgrade.

- axalant 2000 SPx
- Eigner PLM 5.x

Prerequisites

- Do not use the software in any situation where significant damage to property or business could occur from a software error. In no event will Oracle or any other party who has been involved in the creation, production or delivery of the software be liable for special, direct, indirect, incidental, punitive or consequential damages, including loss of profits, revenue, business, goodwill, data or computer programs, or inability to use the software, however, caused and regardless of the theory of liability even if Oracle or such other party has been advised of the possibility of such damages.
- The Upgrade Tool addresses experienced project engineers and PLM administrators with customizing and database experience. Do not use the tool without the necessary knowledge. Read the complete manual in order to get all necessary information.
- Do not attach to, or even change the production system. Always work on a copy of the production database dump. Avoid working on production computers to exclude any influence on the system. Never insert database connections of production database users in any configuration file or script except for exporting the dump or a source for copying tables (Production Database).
- You should always be able to restart an older PLM version (Agile e6.0.x or e6.1.x, axalant, or Eigner PLM 5.x) as a fallback strategy.
- The best performance is reached when installing the Upgrade Tool on your database server. It is also possible to work on any machine in the LAN.
- Dump upgrade takes up to 8 hours runtime per each environment - please be patience.

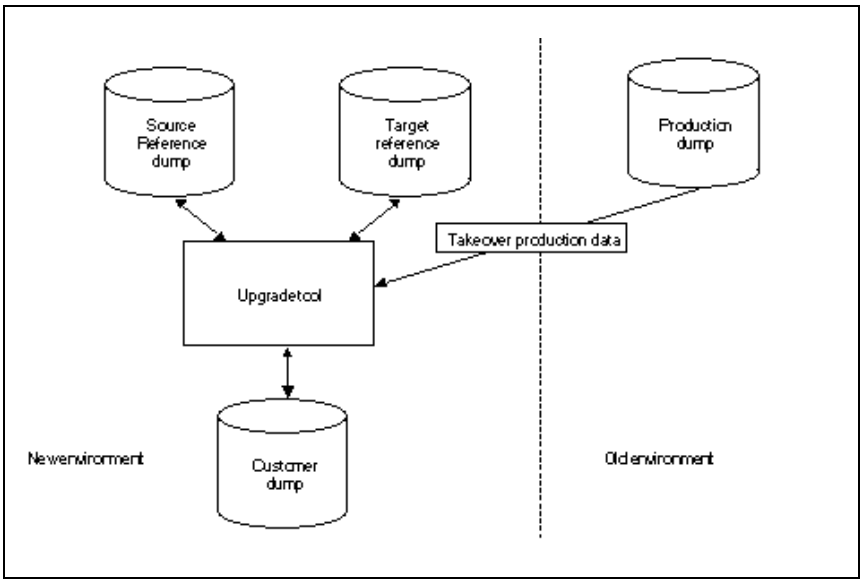
- The Upgrade Tool is running on Windows and UNIX platforms officially supported for Agile e6.1.2.2.
- Java 1.6 JRE, latest version, should be installed on the machine and configured in the environment (JAVA_HOME must be set – this can be done manually in upg_env.cmd resp. upg_env.sh script).
- Database version supported in this upgrade tool is Oracle 11.2.0.3 (English/German).
- ORACLE_HOME variable must be set for “system V” operating system – Unix/Linux etc. This can be done in upg_env.cmd (upg_env.sh) script.
- SQL*PLUS must be installed on the machine.
- At least 250 MB hard disk space must be available for the software and generated log and data files on the local hard disk where the tool is installed.
- At least 2 GB free RAM must be available to run a dump upgrade.
- Sufficient disc space must be available to store copies of your production database and reference dumps on the machine / within the database.
- All database users have to be assigned to the role AGILE_E_ROLE. The default table space must be EDB. Additionally, the users must have quota permissions on the following table spaces: EDB, EDB_IDX, EDB_LOB, EDB_TMP, EDB_TMPIDX.

  AGILE_E_ROLE role and plm schema can be created by executing the sql script:

  ```
  sqlplus ora/sql/cre_plm_usr.sql <username> <password>
  ```

How it Works

The Upgrade Tool is implemented in Java. The tool accesses the databases directly using a JDBC connection. The configuration of all upgrade steps is stored in a set of xml control files. In addition SQL scripts are used for special steps.



The upgrade process is organized in following phases:

- Pre-actions on the original production environment
- Customizing Upgrade

  In this step the customization and configuration stored in the database is updated to Agile e6.1.2.2. The minimum passing time will be 4-5 hours (depending on the system. Main parameter is memory!). Make a copy of your production database dump. Do not attach your production system. Always work on a copy of your data.

- Customizing / Test phase

  The upgrade tool will create a dump on which you can run Agile e6.1.2.2. All functionalities must be tested to run correctly, including the whole customer-specific functionality. This dump is not error free. You have to check all functionalities and clear out the errors caused by setting up the upgrade tool.

- Take over production data

  All tables containing production data, like document and item master data, are copied from the production system to your new Agile e6.1.2.2 dump. They will be adapted so that you can work on this data within Agile e6.1.2.2.

Log files are created and stored in the log/ directory. After the dump upgrade is done, these files have to be reviewed, especially the errors.log file, to make sure there are no errors, otherwise, manual dump changes have to be made. Further “synchronize step” log files placed in data/sync have to be reviewed. Please control log files after each upgrade step and correct occurred errors manually before proceeding with upgrade.

Chapter 2

Installation and Configuration

Database Settings

The Upgrade Tool needs a well-configured database to provide a good performance. The Oracle standard database settings are not sufficient to run the program within the estimated time.

Oracle Parameters

Check the Oracle parameters and verify that at least the following minimum values are set in your database instance:

- memory_target >= 1 GB
- log_buffer >= 163,840 (3*64 Kbytes)

If the database memory consumption is too small, adapt the values.

If you use the server parameter file spfile (like in the Agile e6 standard installation), execute the following commands to change the values of the initialization parameters.

- Login into Sql*Plus as user sys

```
C:\> Sqlplus /nolog
SQL> CONNECT <sys>@<db_service> as sysdba
SQL>ALTER SYSTEM SET <parameter name>=<Value>  SCOPE=BOTH
```

Note Do not change the values of production systems. Make a copy of the initialization file and adapt the values.

- Also read the Oracle online manuals and the Oracle 11.2 installation manual.

Oracle needs physical memory. If the system starts swapping or paging, the Oracle performance degrades or causes errors. Examine your free physical memory and prevent the OS from swapping.

Some Unix systems have maximum values for shared memory. Refer to the installation instructions before changing any value.

SQL Net Configuration

The network domain is part of different Oracle settings. Please check if the domain is consistently used for the following settings:

- Global Database
- Service name
- Listener.ora
- Default domain name

Global Database Name

1. Login into Sql*Plus as user sys and check the global database name.

```
The name should contain the network domain. Here is an example:
sqlplus  <system>/<db_password>@<db_service
SWL>select * from global name;
GLOBAL_NAME -----------------
PLM.WORLD
```

   The example uses the default network domain in world. Also possible are values like example.com.

2. Change the global database name login to Sql*Plus and execute the following commands:

   SQL>alter database rename global_name to <db_name>.<domain_name>

   Example:

```
SQL>alter database rename global_name to plm.agile.oracle.com
```

Service Name

The service name in the SQL net configuration file tnsnames.ora in the directory $ORACLE_HOME/network/admin must include the network domain.

1. Change to the directory $ORACLE_HOME/network/admin
2. Open the file tnsnames.ora and check if the service name is fully defined.

   That means the name contains the same network domain as the global database name.

```
PLM.WORLD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1524))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = PLM.WORLD)
```

```
    )
  )
```

listener.ora

1. Check if the global database name in the section SID_List of the listener configuration file contains also the same fully qualified global database name.

```
SID_LIST_LISTENER_PLM =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = PLM.WORLD)
      (SID_NAME = plm)
    )
  )
LISTENER_PLM =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
  )
```

sqlnet.ora

The default setting for the network domain in the sqlnet.ora file should be the same. Change to the directory $ORACLE_HOME/network/admin and open the file sqlnet.ora and check the default domain settings.

```
names.default_domain = world
```

Character Scan

With Agile e6.1, a new UTF-8 database server-side encoding was introduced, which must be executed only when you are upgrading from Agile e6.0.x or older. Therefore, a character scan has to be performed prior to the upgrade procedure. This should guarantee that all characters within the dump can be converted to the target encoding and no invalid characters are used.

This check has to be performed in the current production database instance, or in an identical database installation. It has to be executed prior to any dump imports for upgrade purposes.

Note In case of a check of an identical database, no charset conversion should be performed prior to this check. This means, NLS_LANG client settings and database character set should have the same settings as defined in the current production environment.

Note The entries found during this check have to be corrected manually in the upgraded environment after the take-over phase of the upgrade.

The following is a description of how to proceed with this check.

1. Install CSSCAN on the database server using user oracle and check the csminst.log for errors.

   ```
   cd $ORACLE_HOME/rdbms/admin
   set oracle_sid=<your SID>
   sqlplus sys@<your SID> as sysdba
   SQL>spool csminst.log
   SQL>@csminst.sql
   ```

2. Create a separate directory from where csscan will be run and output files are placed - */opt/oracle/tmp/charcheck*

3. Find out, which character set is used in your production database. Please make sure you are running this statement in your current production environment.

4. sqlplus system@PLM

   ```
   > SELECT VALUE AS ORIGINAL_CHARACTER_SET from v$nls_parameters
   > WHERE PARAMETER='NLS_CHARACTERSET';
   ```

5. Switch to the directory created in step 2 and perform a database character health check. To perform a DB character health check, run the csscan utility.

   This could take a few hours if huge tables are scanned.

   ```
   csscan <dba_user> USER=<username> FROMCHAR=<original_character_set>
   TOCHAR=<target_character_set> ARRAY=1024000 LOG=charcheck CAPTURE=Y
   PROCESS=4
   ```

   - <original_character_set>: The character set of your productive database (as determined in Step 3)
   - <target_character_set>: The character set of your upgrade database (UTF8)
   - <username>: The schema to be scanned.

   Example:

   ```
   csscan system@PLM61 USER=PLM FROMCHAR=WE8MSWIN1252 TOCHAR=UTF8
   ARRAY=1024000 LOG=charcheck CAPTURE=Y PROCESS=4
   ```

   In this example, the csscan utility will create 3 files -

   - charcheck.out - logging of the output of csscan
   - charcheck.txt - Database Scan Summary Report
   - charcheck.err - log file that normally should contain the row ids of the rows of the tables reported in charcheck.txt

6. Logon to the database as user SYSTEM and create a directory CSSCAN_DIR

   ```
   sqlplus system@<your SID>
   SQL>create directory CSSCAN_DIR as '/opt/oracle/tmp/charcheck';
   ```

```
SQL>grant read, write on directory CSSCAN_DIR to <username>;
```

7. Copy the file ora/sql/char_check.sql to /opt/oracle/tmp/charcheck.

8. Open sqlplus session and connect as PLM database user <username>. Then execute the script char_check.sql.

   ```
   SQL>@char_check.sql
   ```

   The script will create the file charcheck.rep. Values to be corrected are listed by table, column and C_ID. These values have to be fixed manually.

   ```
   ========================
   = Table : BVB_ARTIKEL =
   ========================
   C_ID  : 1951227764
   COLUMN: ARTID
   Value : GEH<80>USE
   C_ID  : 1951224513
   COLUMN: ARTID
   Value : DREHSTABGEST<80>NGE
   C_ID  : 1951227764
   COLUMN: ARTID
   Value : GEH<80>USE
   ========================
   = Table : BVB_ARTIKEL =
   ========================
   ```

Pre-Activities on the Original Production Environment

Note Only for Agile 6.1.1 or earlier versions!

Since Eigner PLM 5.0 UIC <= 1000 and GIC <= 1000 are reserved for the standard development, existing users or groups using such C_IC / C_GIC must be migrated to a higher value.

This action must be executed only on a production dump with PLM version older than Eigner PLM 5.0. This migration must be executed before you start any other upgrade activity.

Note It can be very time consuming.

In a big customer dump it takes 1h/6 users. To solve the time conflict, break down the update into different subsets and try to run it in parallel.

To do this, adapt the following statement in the SQL script:

```
INSERT INTO TEMP_U (OLD_U) SELECT C_IC FROM T_USER a
```

```
WHERE
C_IC > 200 AND C_IC < 1000 AND C_NAME NOT LIKE 'EDB%' AND
C_NAME NOT LIKE 'DEMOEP%';
```

To execute the UIC/GIC Migration run the following commands:

```
sqlplus <user>/<password>@PLM
SQL> @update_customer_UIC.sql
```

Import Dumps

Three dumps need to be imported into one new database environment:

- Source reference dump
- Target reference dump
- Customer dump

Note Schemas for source reference dump, target reference dump, and customer dump have to be in one database. Otherwise, the Upgrade Tool will return an error.

For importing the dumps, do not change the table space names because the created table statements on tables containing a blob clause will fail if the original table spaces like EDB, EDB_IDX and EDB_LOB do not exist

To import dumps into the database you have to be familiar with your database environment and Oracle import utilities as well. Alternatively, a batch script called imp_dmp.cmd can be used for automated imports (Oracle only). This script uses dumps located in the directory upgrade/dumps.

Import Source Reference Dump

1. Download appropriate source reference dump from the Oracle Software Delivery Cloud (http://edelivery.oracle.com), and import it into the new database environment.

   Example: import plm604upgref.dmp into a user named PLM604UPGREF

2. Import target reference dump.

3. Download the latest target reference dump from the Oracle Software Delivery Cloud (http://edelivery.oracle.com) and import it into the new database environment.

   Example: import plm6122upgref.dmp into a user named PLM61UPGREF.

Import Customer Dump

1. Export the current production environment into a dump file and import it in the new environment.

   Note Always work on this customer dump. Do not attach production dump during the upgrade process except for the takeover step described below.

   Note If you are about to import item data into T_MASTER_DAT by using the binary loader, please keep in mind that you have to set the C_ID value of the EDB-NULL-ITEM lower than 2,000,000,000. Otherwise, an error will occur during the import.

   This dump will be the new production dump after the upgrade is completed.

2. Give the new environment an expressive name.

   Note Agile e6.1.2.2 database uses an UTF8 Database encoding, only if old production is Agile e6.0.x or older. Therefore, a simple import (using IMP tool) will fail. The following commands must be executed to import an older (non-UTF8) PLM dump into the new database:

The following variables are used in this example:

- sysuser = system user name
- syspwd = system user password
- dbname = database service name
- fromuser = user name the dump was exported from
- impdbuser = schema name where the dump should be imported
- impfile = dump file name

To import the dump execute the following commands in the shell:

```
imp <sysuser>/<syspwd>@<dbvname>fromuser=<fromuser>touser=<impdbuser>
FILE=../dumps/<impfile>.dmp BUFFER=132000 feedback=50000
statistics=none rows=n log=imp_customer1.log
sqlplus <impdbuser>/<impdbuser>@<dbname>
@../ora/sql/convert_to_utf8.sql > imp_customer2.log
imp <sysuser>/<syspwd>@<srvname> fromuser=<fromuser> touser=<impdbuser>
FILE=../dumps/<impfile>.dmp BUFFER=132000 feedback=50000
statistics=none ignore=y log=imp_customer3.log
```

Create Statistics for All Involved Database Schemas

1. Check Language settings:

The setting for the environment variable NLS_LANG must be AMERICAN_AMERICA.WE8MSWIN1252, otherwise the statistics will not be computed correctly.

2. Login in Sql*Plus as user sys with sysdba privilege and perform analyzing.

```
SQL> EXECUTE DBMS_STATS.GATHER_SCHEMA_STATS(<DBUSER>,CASCADE =>true);
```

Run Upgrade Tool

1. Set a DISPLAY variable on UNIX.

2. Control the NLS_LANG setting in upg_env.sh (upg_env.cmd on windows) which specifies the client character set. “AMERICAN_AMERICA.WE8MSWIN1252” is a default setting for upgrade reference dumps.

3. Adapt the following environment definitions in upg_env.sh on UNIX:

   - JAVA_HOME: at least the JRE used for Agile e6.1.2.2 installation is required by the Upgrade Tool.

   - ORACLE_HOME: make sure that the Oracle 11.2.0.3 environment is set before proceeding with the upgrade. To check the environment for UNIX, execute the following commands:

```
> env|grep.NLS
NLS_LANG=AMERICAN_AMERICA.WE8MSWIN1252
ORA_NLS10=/opt/oracle/product/11.2/nls/data

> env|grep ORA
ORACLE_BASE=/opt/oracle
ORACLE_HOME=/opt/oracle/product/11.2
ORACLE_TERM=xterm
ORA_NLS10=/opt/oracle/product/11.2/nls/data
```

Configuring the Upgrade Tool

Configuring New Database Environment Connections

1. Run start_upg.cmd (start_upg.sh on UNIX) and enter the following information for dumps in your new database environment.

   Each database connection can be tested with a “TEST” button on an appropriate tab.

Note Press return after every change in a field. The color turns back from red to black. Otherwise the changes will be lost.

Parameter	Description
Host	Host name of database server
Port	Port number of Oracle listener (default 1521)
SID	Oracle_SID (uppercase). For RAC databases use the SID of the Instance where the upgrade tool should run.
User	Database user name
Password	Password of database user
Connection String	Service name which is used to run SQL*PLUS commands on the machine the Upgrade Tool is installed on. Use fully qualified name including the network domain, i.e. plm60.agile.agilesoft.com **Note** The service name cannot be tested in the current version of the Upgrade Tool.

Adapt the tablespace names for each database connection since they are used, e.g. for creation of new database objects or running SQL scripts:

Table	Default EDB
Index	Default EDB_IDX
LOB	Default EDB_LOB

Temporary table	Default EDB_TMP
Temporary index	EDB_TMPIDX

Configuring Production Dump Connection

1. Enter a database connection for the current production dump.

   This connection will be used at the end of the upgrade process in the “Take over” phase. The definition of this connection is different, because it is implemented as a database link, which is temporarily created in your new customer dump. Creation of the database link can be tested with the “TEST” button.

   This connection will not be used until the takeover phase of the upgrade process.

2. Press return after every change in a field. The color turns back from red to black. Otherwise, the changes will be lost.

Parameter	Description
Service Name	Oracle service name including network domain, i.e. AGILE.AGILESOFT.COM. Service name must be defined in tnsnames.ora.
SID	Oracle_SID (uppercase)
User	Database user name.
Password	Password of database user.

Setting Upgrade Tool Parameters

Review and correct the entries if necessary and check the following table for valid entries. The correct values can be determined with the Compute button. Always check the computed values.

Note Press return after every change in a field. The color turns back from red to black. Otherwise, the changes will be lost.

PLM-Version	The customer dump version (before Agile e6 upgrade process). Following values are valid: 02 = AXALANT SP1 03 = AXALANT SP2 04 = AXALANT SP3 05 = PLM 5.0 06 = e6.0 LA 07 = e6.0 GA 08 = e6.0.1 09 = e6.0.2 10 = e6.0.3 11 = e6.0.4 20 = e6.1.0.0 20 = e6.1.0.0004 21 = e6.1.1

<table>
<tr><td></td><td>22 = e6.1.2.0
23 = e6.1.2.2</td></tr>
<tr><td>LogiView Timestamp</td><td>All LogiView items with a change date after this time point will be deleted. You can adapt this value manually. Following values are possible:
<table>
<tr><td>axalant2000 SP1</td><td>20001109140557</td></tr>
<tr><td>axalant2000 SP2</td><td>20010723102350</td></tr>
<tr><td>axalant2000 SP3</td><td>20011113092600</td></tr>
<tr><td>axa2000 SP3 PA1</td><td>20020808110309</td></tr>
<tr><td>Eigner PLM 5.0.1</td><td>20020830153411</td></tr>
<tr><td>Agile e6.0 LA</td><td>20050414160530</td></tr>
<tr><td>Agile e6.0 GA</td><td>20050615170000</td></tr>
<tr><td>Agile e6.0.1</td><td>20051111135800</td></tr>
<tr><td>Agile e6.0.2 LA</td><td>20060630200000</td></tr>
<tr><td>Agile e6.0.2</td><td>20060731220000</td></tr>
<tr><td>Agile e6.0.3</td><td>20070213080000</td></tr>
<tr><td>Agile e6.0.4</td><td>20070704180000</td></tr>
<tr><td>Agile e6.1.0.0</td><td>20080929210000</td></tr>
<tr><td>Agile e6.1.0.0004</td><td>20081107210000</td></tr>
<tr><td>Agile e6.1.1</td><td>20090812210000</td></tr>
<tr><td>Agile e6.1.2.0</td><td>20101105210000</td></tr>
<tr><td>Agile e6.1.2.2</td><td>20111116210000</td></tr>
</table></td></tr>
<tr><td>Classification – Control file</td><td>A file name of the classification control file for the customer dump. Valid entries are:

cla_ctl.xml
cla_ctl_with_multi_lang.xml
cla_ctl_with_multi_lang_repl.xml
cla_ctl_with_repl.xml</td></tr>
<tr><td>Database Language</td><td>Language for the database dump. This influences the migration of the classification date.
Values: German, English
Default: German</td></tr>
<tr><td>Level</td><td>Status that is set during classification upgrade for records in the tables T_CLA_DAT (pool attributes), T_GROUP_DAT (classes)</td></tr>
<tr><td>Replication server</td><td>The valid name of the database server, in case of an implemented database replication to the environment, should be migrated.</td></tr>
</table>

Save Configuration

To save the current configuration, select File / Save.

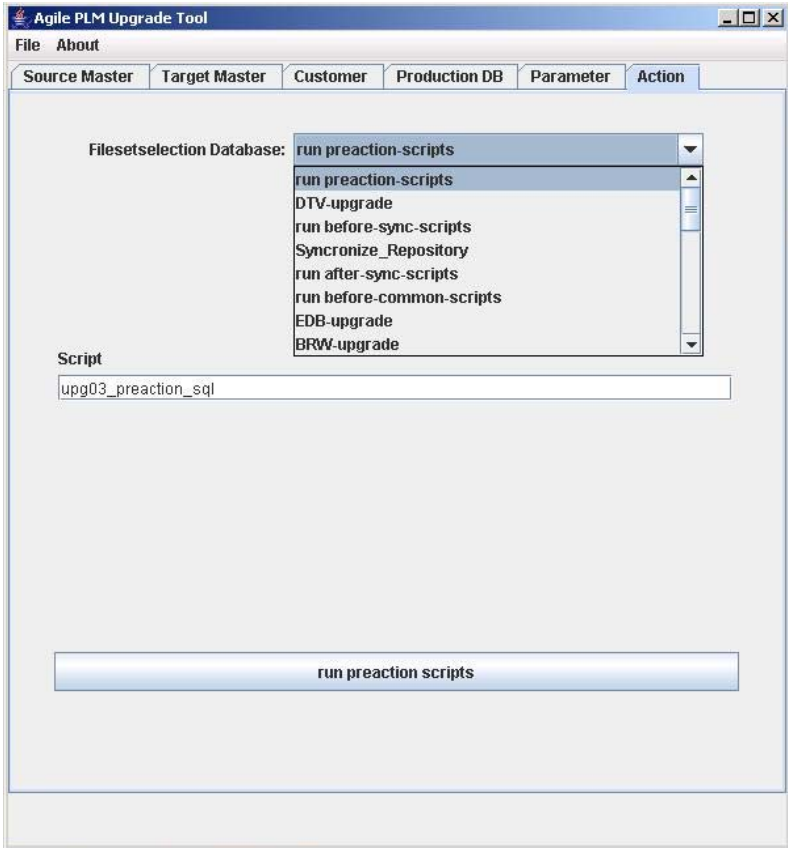
Chapter 3

Perform the Upgrade

Perform the Upgrade in Interactive Mode

This chapter describes how the customizing upgrade is executed interactively. There are different steps available for selection on the ACTION tab, which should be executed in the respective order.

Note Some steps may be repeated without re-importing the customer dump. For further information please refer to Appendix G: Repeatable Tasks



Step 1: Run Pre-Action-Scripts

The command script upg03_preaction_sql.cmd will be executed. This script executes a couple of

SQL scripts (depending on the customer dump version). Log files created in this step in the upgrade/log directory have to be reviewed manually before proceeding with the upgrade.

Additionally, DTV internal repository is taken over from target reference dump into the customer dumps by using sql script grant_dtv.sql and takeover_dtv.sql.

A complete list of SQL / LOG files created in this step can be found in the Appendix.

Step 2: DTV-Upgrade

This step upgrades the DataView repository and is generally split into two steps:

- Compares the data sets from the different dumps and stores the changes in an XML file for the three possible operations: delete, insert and update:

  dtv_del.xml, dtv_ins.xml, dtv_upd.xml

  The Upgrade Tool selects each row from the source and the target reference dumps, compares the data sets from both dumps to identify the differences and checks if the customer has modified these data. The upgrade action (Insert, Update, and Delete) is determined for each record and the information is stored in a set of XML files. The migration rules are listed in the Appendix.

  Note This step takes 10 min to 4 hours.

- Performing operations. The upgrade tool reads the XML files created during the previous operation and performs the corresponding SQL-statements. This step takes about 15 - 40 min. During the performing an error log file data/dtv/dtv_err.xml is created and should be checked for possible errors.

  - A dtv_customizing.log is created in the directory data/dtv/ also. It contains conflicts, which occurred during the DTV-upgrade. This log file must be reviewed.

  Check log files in the directory upgrade\data\dtv. If an error occurs, check the error description in upgrade\log\erorrdetails.log.

  Note The upgrade tool is only able to compare tables with the same table structure. Therefore, the DataView tables in the reference dumps (edb234upgref …) have an Agile e6 structure.

  Note For better readability, it is possible to create HTML files from the XML log files (see Convert XML files to HTML).

Step 3: Run Before-Sync-Scripts

The command script upg07_sync_update.cmd (upg07_sync_update.sh on Unix) is executed. This script executes an SQL script (depending on the customer dump version). Log files

created in this step in the upgrade/log directory have to be reviewed manually before proceeding with the upgrade.

A complete list of SQL / LOG files created in this step can be found in the Appendix.

Step 4: Synchronize_Repository

During this step the table definition in DataView within the customer dump is compared to the physical table structure in the database. SQL statements to create and alter database objects are generated and executed automatically. The adaptation of the physical data structure will consist of the following steps:

- Analyze phase

  In this phase you can see which statements the program will perform. It creates also a control file named conf/special.xml for field values and special tasks. This step takes about 1-6 min.

- Review and edit the control file conf/special.xml

- Synchronize phase

  In this step, database structure is converted according to the new DataView repository. This step takes about 1 min - 4 hours.

The data and log files are placed in the *data**sync* directory. A detailed description of the errors can be found in *log**errordetail.log*.

Note If the program terminates the process and releases the connection because of a server error, drop the special.xml file in conf/ directory and copy the preconfigured template from conf/template into the conf/ directory. Restart the process.

Analyze

In this step, a proposal special.xml file is written to the conf/ directory. Error messages are written in in file <upg_root>\data\sync\ sync_analysis.log. Appendix H contains list of error messages.

Configure special.xml for Synchronizing Repository

The delivery package contains a preconfigured special.xml, which defines standard settings for all expected cases. Very often the customer dump contains inconsistencies so that in the analyze mode the tool will add entries to the special.xml file. In this case you need to review and adapt the following configuration subsets:

Note Do not change or delete the default settings.

- Static default values for columns changed from null to not null.

  In the following example the column T_TRE_DAT.CUR_FLAG is set to ‘n’.

```
<FieldDefault>
    <FieldName>T_TRE_DAT.CUR_FLAG</FieldName>
    <FieldType>S</FieldType>
    <FieldSize>1</FieldSize>
    <DefaultValue>
        <Value>n</Value>
    </DefaultValue>
</FieldDefault>
```

- Dynamic default values: The field values can be computed dynamically based on a Java function / SQL Statement. Preconfigured functions are available to use the number server to set values. Here is an example for an SQL statement and JAVA generated field default:

```
<FieldDefault>
    <FieldName>T_CTX_DAT.EDB_SEQ</FieldName>
    <FieldType>I</FieldType>
    <FieldSize>4</FieldSize>
    <DefaultValue>
        <Select>
            DISTINCT (SELECT COUNT(*) FROM T_CTX_DAT T WHERE T.C_ID
&lt;=  thisRec.C_ID)*10
        </Select>
        <Where>C_ID &gt; 0</Where>
    </DefaultValue>
</FieldDefault>
<FieldDefault>
    <FieldName>T_MASTER_DOC.EDB_ID</FieldName>
    <FieldType>I</FieldType>
    <FieldSize>10</FieldSize>
    <DefaultValue>
        <Function>GetNewEDBID(EDBEDBID)</Function>
    </DefaultValue>
</FieldDefault>
```

- Example to Rename tables:

```
<RenameTable>
    <TableName>T_EER_SIT</TableName>
    <NewTableName>T_DDM_SIT</NewTableName>
</RenameTable>

<RenameTable>
    <TableName>T_EER_SIT_STR</TableName>
    <NewTableName>T_DDM_SIT_STR</NewTableName>
```

```
</RenameTable>
```

- Move fields:

  This option allows moving a column of a table's inclusive stored values to a new location. To move a field you have to specify:

  - Source field (<table_name>.<column_name>)
  - Target field (<table_name>.<column_name>)and
  - Path (join condition between old and new table)

  The following sample configuration files show 3 different possibilities to move field values to a new location.

```
<!-- Example transfer from typetable to entitytable. -->
<MoveField>
   <SourceField>T_DOC_DRW.CRE_USER</SourceField>
   <Path>T_DOC_DRW.C_ID_2</Path>
   <Path>T_DOC_DAT.C_ID</Path>
   <DestField>T_DOC_DAT.CAX_CRE_SYSTEM</DestField> </MoveField>

<!-- Example transfer from entitytable to entitytable via
releationtable. -->
<MoveField>
   <SourceField>T_MASTER_DAT.PART_ID</SourceField>
   <Path>T_MASTER_DAT.C_ID</Path>
   <Path>T_MASTER_DOC.C_ID_1</Path>
   <Path>T_MASTER_DOC.C_ID_2</Path>
   <Path>T_DOC_DAT.C_ID</Path>
   <DestField>T_DOC_DAT.CAX_CRE_SYSTEM</DestField>
</MoveField>

<!-- Example transfer in table. -->
<MoveField>
   <SourceField>T_MASTER_DAT.PART_ID</SourceField>
   <DestField>T_MASTER_DAT.EDB_ICON</DestField>
</MoveField>
```

An example configuration file special_move.xml containing a definition of moved fields is stored in the template directory conf/template/

- Change data type of a field

The upgrade tool allows changing the type definition of columns like integer to string. If the value of a column for all records is null then also incompatible data type changes can be executed (e.g. string to integer). Cutting a string field is only possible if no record contains a longer value. Please check max length of stored values directly within Sql*Plus.

You have to replace “false” by “true” to confirm such critical changes. The type definition “oldType” comes from the database; “newType” is the DataView definition (stored in T_FIELD. C_FORMAT).

```
<FieldChange>
 <FieldName>T_DOC_DAT.FOO</FieldName>
 <ConfirmChange oldType="S80.0"
newType="S40.0">false</ConfirmChange>
</FieldChange>
```

Synchronize

In this step, dump structures are adapted to DataView repository using the special.xml entries to fill NOT NULL fields.

Step 5: Run After-Sync-Script

The command script upg08_postactions is executed. This script executes a set of SQL scripts. Check the results in the log files named log/08_*.log. For a complete list of log files please refer to the Appendix.

During this upgrade step a valid reference to T_STA_LUT is established with values in EDB_STA_REF for each entry in T_CHK_STA. Additionally, EDB_LEVEL within the table T_STA_LUT is filled.

Note The values of the table T_STA_LUT have to be reviewed after the customizing upgrade.

Additionally, BVB_ARTIKEL clean up step is executed for PLM 5.x and older versions. For more information please refer to 0 Cleanup BVB_ARTIKEL.

Note PLM5 and older favorites are migrated in this upgrade step as well. For more information please refer to 14.5 Favorites upgrade.

Note The favorite migration can be performed after the first takeover execution because it needs standard browser entries, which are inserted later within the step BRW-upgrade.

And finally, the T_PRC_DAT.EDB_PRO_REF column is filled during this upgrade step. For more information please refer to the chapter "Project references in processes".

Step 6: Run Before-Common-Scripts

The command script upg09_common_get is executed. This script starts the SQL script to save and delete standard LogiView models. Check log files named log\09%.log

Step 7: EDB-Upgrade (Configuration)

During this step the content of the common Oracle configuration tables are upgraded. The appropriate xml files are stored in data/edb/ directory.

1. Click create files.

   XML files edb_ins.xml, edb_del.xml, and edb_upd.xml are created. They can be reviewed before performing these actions on the database dump.

2. Click Perform delete, insert, update.

   XML files created in the previous step are read and executed. Errors which occur during these actions are stored in edb_err.xml – please control this file.

Note If an error occurs, check the error description in upgrade\log\erorrdetails.log.

Step 8: BRW-Upgrade (Browser)

During this step the content of the browser configuration tables is upgraded. The appropriate xml files are stored in the data/brw/ directory.

1. Click create files.

   XML files brw_ins.xml, brw_del.xml, and brw_upd.xml are created. They can be reviewed before performing these actions on the database dump.

2. Click Perform delete, insert, update.

   XML files created in the previous step are read and executed. Errors which occur during these actions are stored in brw_err.xml – please control this file.

Note If an error occurs, check the error description in upgrade\log\erorrdetails.log.

Step 9: DODE-Upgrade (Print Studio)

During this step the content of the Print Studio configuration tables will be upgraded. The appropriate xml files are stored in data/dode/ directory.

1. Click create files.

XML files dode_ins.xml, dode_del.xml, and dode_upd.xml are created. They can be reviewed before performing these actions on the database dump.

2. Click Perform delete, insert, update.

   XML files created in the previous step are read and executed. Errors which occur during these actions are stored in dode_err.xml – please control this file.

Note If an error occurs, check the error description in upgrade\log\erorrdetails.log.

Step 10: LGV-Upgrade (LogiView)

Several standard LogiView procedures are changed in every new Agile e-series software release. All LogiView logic models are deleted – if they are identical to new ones respectively saved if they were changed. Next, all new logical models are re-inserted. After the upgrade all LogiView changes made by customer must be made again.

Additionall standard LogiView variables, constants, and system variables are upgraded in the usual manner.

The appropriate xml files are stored in data/lgv/ directory.

1. Click create files.

   XML files lgv_ins.xml, lgv_del.xml, and lgv_upd.xml are created. They can be reviewed before performing these actions on the database dump.

2. Click Perform delete, insert, update.

   XML files created in the previous step are read and executed. Errors occurred during these actions are stored in lgv_err.xml – please control this file.

Note If an error occurs, check the error description in upgrade\log\erorrdetails.log.

Step 11: WFL-Upgrade (Workflow)

During this step the content of the workflow configuration tables is upgraded. The appropriate xml files are stored in the data/wfl/ directory.

1. Click create files.

   XML files wfl_ins.xml, wfl_del.xml, and wfl_upd.xml are created. They can be reviewed before performing these actions on the database dump.

2. Click Perform delete, insert, update.

XML files created in the previous step are read and executed. Errors which occur during these actions will be stored in wfl_err.xml – please control this file.

Note If an error occurs, check the error description in upgrade\log\erorrdetails.log.

Note Since Agile version e6.0.4 the field T_ACT_HIS.PV_VALUE has been replaced by two new fields (T_ACT_HIS_PV_SL_VALUE and T_ACT_HIS.PV_ML_VALUE) in the History for Workflow activities. Because the old data cannot be mapped exactly for the new schema, they are still stored in the existing old field. The old field is invisible and the new fields T_ACT_HIS_PV_SL_VALUE (single-language value) and T_ACT_HIS.PV_ML_VALUE (multi-language value) are empty after the upgrade. If requested, the old field can be made visible. Since Agile e6.0.4, resp. Agile e6.1 new data is only filled the new fields.ii

Step 12: CHG-Upgrade (Change Management)

During this step the content of the change management configuration tables is upgraded. The appropriate xml files are stored in the data/chg/ directory.

1. Click create files.

   XML files chg_ins.xml, chg_del.xml, and chg_upd.xml are created. They can be reviewed before performing these actions on the database dump.

2. Click Perform delete, insert, update.

   XML files created in the previous step are read and executed. Errors which occur during these actions are stored in chg_err.xml – please control this file.

Note If an error occurs, check the error description in upgrade\log\erorrdetails.log.

Step 13: GTM-Upgrade (Classification)

During this step the content of the classification configuration tables is upgraded. The appropriate xml files are stored in the data/gtm/ directory.

1. Click create files.

   XML files gtm_ins.xml, gtm_del.xml ,and gtm_upd.xml are created. They can be reviewed before performing these actions on the database dump.

2. Click Perform delete, insert, update.

   XML files created in the previous step are read and executed. Errors which occur during these actions are stored in gtm_err.xml – please control this file.

Note If an error occurs, check the error description in upgrade\log\erorrdetails.log.

Step 14: GDM-Upgrade (Office Suite)

During this step the content of the Office Suite configuration tables is upgraded. The appropriate xml files are stored in the data/gtm/ directory.

1. Click create files.

   XML files gdm_ins.xml, gdm_del.xml, and gdm_upd.xml are created. They can be reviewed before performing these actions on the database dump.

2. Click Perform delete, insert, update.

   XML files created in the previous step are read and executed. Errors which occur during these actions are stored in gdm_err.xml – please control this file.

Note If an error occurs, check the error description in upgrade\log\erorrdetails.log.

Step 16: RMT-Upgrade (Requirement Management Traceability)

During this step the content of the RMT-module configuration tables is upgraded. The appropriate xml files are stored in the data/rmt/ directory.

1. Click create files.

   XML files rmt_ins.xml, rmt_del.xml, and rmt_upd.xml are created. They can be reviewed before performing these actions on the database dump.

2. Click Perform delete, insert, update.

   XML files created in the previous step are read and executed. Errors which occur during these actions are stored in gdm_err.xml – please control this file.

Note If an error occurs, check the error description in upgrade\log\erorrdetails.log.

Step 17: ASE-Upgrade (Advanced Structure Editor)

During this step the configuration tables of the Advanced Structure Editor (ASE) are upgraded. The appropriate xml files are stored in the data/ase/ directory.

1. Click create files.

   XML files ase_ins.xml, ase _del.xml, and ase _upd.xml are created. They can be reviewed before performing these actions on the database dump.

2. Click Perform delete, insert, update.

   XML files created in the previous step are read and executed. Errors which occur during these actions are stored in ase _err.xml – please control this file.

Note If an error occurs, check the error description in upgrade\log\erorrdetails.log.

Step 18: CMG-Upgrade (Configuration Management)

During this step the content of the CMG-module configuration tables is upgraded. The appropriate xml files are stored in the data/cmg/ directory.

1. Click create files.

   XML files cmg_ins.xml, cmg _del.xml, and cmg _upd.xml are created. They can be reviewed before performing these actions on the database dump.

2. Click Perform delete, insert, update.

   XML files created in the previous step are read and executed. Errors which occur during these actions are stored in cmg_err.xml – please control this file.

Note If an error occurs, check the error description in upgrade\log\erorrdetails.log.

Step 19: Run After-Common-Scripts

The command script upg10_common_update is executed. This script executes an SQL script to migrate non-standard browser entries to the new Agile e6 browser based on new database tables having a prefix T_EXP_*. Check the results in the log files named log/10_*.log. For a complete list of log files please refer to the Appendix.

This step can be skipped in GUI mode if you are migrating from Agile e6.0 or newer.

Note Please control the log file full.log for the results of LogiView comparison between upgraded and targeted reference dumps.

Step 20: Upgrading Classification (optional)

In Eigner PLM 5.0 a new concept – Attribute pool was introduced. This step converts attributes within the customer dump to this new concept. This step must be skipped, if you are migrating from PLM5.x version or newer.

1. Click Determine merge condition.

A special xml control file data/cla/cla_merge_ctl.xml is created. It describes how attributes are moved into a global attribute pool that contains so-called merge groups. All attributes mapped onto the same pool attribute are in the same merge group. Please review it and adapt if changes are necessary. This mapping file is used at the end of the takeover phase to migrate current production data to an existing attribute pool.

2. Click Perform mapping.

   This will build the attribute pool and perform mapping to item, to documents, etc.

Note Please check log files located in the data/cla/ directory. For more information please refer to Chapter 4, section Classification.

Step 21: Classification Attribute Inheritance

Since Agile e6.0 a new attribute inheritance concept was introduced. To convert existing class attributes, this upgrade step has to be executed.

Note It is not necessary if you are migrating from Agile e6.0 or newer.

Step 22: Special Replace

This optional upgrade step allows a string replacement within table content.

This configuration file *conf/specialreplace.xml* contains an example definition for replacing a string with another string.

Example: Update the strings ‘T_EER_SIT’ with ‘T_DDM_SIT’ in LogiView procedures

```
<?xml version="1.0" encoding="UTF-8"?>
<special>
     <replace>
          <table>LV_DT_PRC</table>
          <field>MAIN</field>
          <example>T_EER_SIT</example>
          <replacewith>T_DDM_SIT</replacewith>
     </replace>
</special>
```

Specific Dump Changes

Recreate customer specific indexes, views, packages, procedures and triggers, database constraints like field defaults, etc. And additionally recreate adapted standard indexes, views, packages, procedures and triggers, database constraints like field defaults, etc.

Note Since DataView does not support GLOBAL TEMPORARY tables (and even Upgrade Tool), you have to remove table definitions from the DataView repository for global temporary tables. Otherwise, errors will occur during the synchronize phase while creating indexes for global temporary tables. Such error messages must be ignored. The tables have to be adapted manually.

Repeat Upgrade in Batch Mode

After the upgrade is tested completely in the interactive mode, it can be used in batch mode. Batch mode is intended for experienced upgrade users. Many upgrade steps are compressed in few batch scripts, so that they possibly need to be adapted manually.

The batch mode can be used in the following manner:

- Do an upgrade in interactive mode
- Re-import customer dump
- Make a backup copy of your upgrade directory. Clean the log directory by executing upg01_pre_cleanlog.cmd
- Run the following scripts:
  - upg03_preaction_sql.cmd
  - upg05_dtv_update.cmd
  - upg07_sync_update.cmd
  - upg08_postaction.cmd
  - upg10_common_update.cmd
  - upg11_cla.cmd
- Control log files
- Take over production data by executing upg13_prod1_takeover.cmd
- Run post-action scripts by executing upg14_prod2_rep_update.cmd.

Perform Upgrade in Batch Mode

Alternatively, a whole upgrade process may be performed in the batch mode: UNIX scripts for batch mode execution (without interactive actions) are available, too. These scripts have the .sh extension (instead of .cmd for Windows).

- Import all dumps
- Start start_upg.cmd and configure the upgrade tool.
- There is a sample script named batch_upgrade.cmd (batch_upgrade.sh on UNIX), which can be used for a complete dump upgrade. All dumps have to be reimported before doing the batch upgrade
  - upg03_preaction_sql.cmd
  - upg04_dtv_get.cmd
  - upg05_dtv_update.cmd
  - upg06_sync_get.cmd
- Adapt special.xml
- Run the following scripts:
  - upg07_sync_update.cmd
  - upg08_postaction.cmd
  - upg09_common_get.cmd
  - upg10_common_update.cmd
  - upg11_cla.cmd
- Control log files
- Take over production data by executing upg13_prod1_takeover.cmd
- Run post-action scripts by executing upg14_prod2_rep_update.cmd

Perform Upgrade in Compact Batch Mode

Alternatively, a whole upgrade process may be performed in the batch mode: UNIX scripts for batch mode execution (without interactive actions) are available, too. These scripts have the .sh extension (instead of .cmd for Windows).

There is a sample script named batch_upgrade.cmd (batch_upgrade.sh on UNIX), which is a template that can be used for a complete dump upgrade. All dumps have to be re-imported before doing the batch upgrade

- Import all dumps

- Start start_upg.cmd and configure the upgrade tool.
- Run the script *batch_upgrade.cmd*
- Control log files
- Take over production data by executing upg13_prod1_takeover.cmd
- Run post-action scripts by executing upg14_prod2_rep_update.cmd

Convert XML Files to HTML

You can convert XML files to HTML and view them with a browser. The batch file xml2html.bat creates HTML files for insert, update, and delete. This function is available on Windows platforms only.

Execute xml2html.cmd with one of the following parameter values, which specify the HTML file to be created for modules:

- All – HTML files for all modules are created
- Module name - HTML files only for specified modules are created, possible values are:

  dtv edb brw dode lgv wfl chg gdm rmt gtm ase cmg

Note You need memory for approximately six times the XML file size!
The batch job will run several hours. The Java Runtime Environment allocates 512MB of memory. You can adjust the memory allocation by editing the file Upgrade\cmd\upg_env.cmd

Takeover Production Data

The next step is to transfer reference data from the production system. This phase is necessary, because during the upgrade and customizing of the new environment new reference data (all tables except customized tables) is available in the old production dump.

Takeover phase is separated in several tasks:

- Generate a relevant table list using a database connection to the current production dump. This database connection is used as source for the tables copied into the new environment. No changes are made in the production database.
- Edit the list of tables, which should be copied during the takeover step.
- Takeover: drop tables in the new dump and copy them from the current production dump.

- Takeover the latest number server values.
- Make the dump up-to-date again by executing the step “Synchronize_repository” and appropriate post-actions.
- Perform a test in the new environment. Test all functionalities, maybe during training of the user. If errors occur, remove them via customizing. Not everything will be done automatically. During the test period a lot of new data is created in the production system. Therefore, the last two steps can be repeated till the new environment is error-free.
- If testing does not raise any errors you can switch the production database to the new one. Takeover your data from the production system (and the files!) again and the new environment is your production system. Shut down your old system.

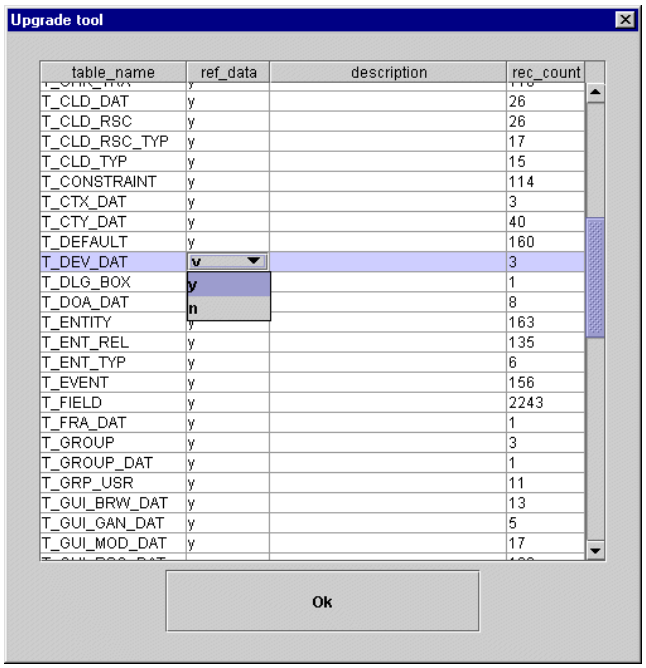
Define Reference Tables

Select the folder Takeover in the upgrade tool and press the button Create Ref File. The upgrade tool will connect to the production database, identify all tables and synchronize the information with the predefined list (ref_tables.xml). Only tables with data will be written to the file.

Edit Production Table List

To adapt the list, press Edit ref. File. For each table you have to define if it is a reference table. (ref_data = y.) Such reference tables will be dropped to the customer dump and copied from the production system using the connection to the production DB.

Select OK to save the XML file (conf/ref_tables.xml).



Note Because of a special BVB_ARTMEH upgrade functionality BVB_ARTIKEL, BVB_ARTMEH, and BVB_ARTMEHUFK tables have to be specified as production tables. Other BVB tables are treated as configuration tables. Otherwise, an error can occur during execution of SQL scripts artmeh_1.sql / artmeh_2.sql.

If you have created new users and / or groups since the date when the dump was exported from the production system the following DataView tables must be migrated.

Note New users like EDB-WFL, EDB-DFM, EDB-DDM, EDB-GDM, EDB-EER, DODEKERNEL will get lost. Export these users first with the binary loader and reload them after the upgrade.

If doing so, the table T_USER has to be eventually upgraded manually, which means you need to proceed with all T_USER changes that happen in SQL scripts called ora/sql/dtvxxx-xxx.sql. Otherwise, you might not be able to login into the application because of missed columns. The list of involved scripts can be found in log/testupgrade_03_preaction_sql.log

- T_USER
- T_GROUP
- T_GRP_USR
- T_PROFILE
- Related tables of the PLM – person management

Note Table T_DEFAULT should be migrated by the loader (import/overload) otherwise, new defaults will be missing.

Perform Transfer

Backup your customer Agile e6 dump and press the Takeover button. The tables containing non-repository information are dropped in your customers dump. The tables will be copied from the defined production environment into your customer dump.

Note Check the log file log/errordetails.log

After copying production database tables, a special upgrade step will be automatically executed – Takeover Workflow Masks. This step is described later in this document.

Taking Over Number Server Values

Since number server is used during the step “Synchronize_repository”, the newest values have to be transferred to the new environment. This is done by executing the step “AFTER TAKEOVER: takeover number server values”. Please control log files 14_01_get_numvalue.log and 14_02_set_numvalue.log.

Note If the production database is Oracle 8i, this step will fail and the corresponding SQL scripts have to be executed manually.afterwards as described below:

- Execute the script ..\ora\sql\get_numvalue.sql in the production environment.
- It will create a new sql script called set_numvalue.sql which should be executed in the upgraded environment.
- Contact Oracle Support for Assistance.

Post-Action Scripts

Tables just copied from the current production environment have an old table structure and must be upgraded. This is done with the following steps:

- Step “AFTER TAKEOVER: run before-sync-scripts”

  The command script upg07_sync_update.cmd will be executed.

  Check log files upg07%.log in the upgrade\log directory.

- Step “AFTER TAKEOVER: Synchronize_repository”

  Execute “Synchronize” and control log/errors.log, control log files in data\sync.

- Step “AFTER TAKEOVER > Run after-sync scripts”

The command script upg15_prod3_postaction.cmd will be executed.

Check log files log/upg15%.log. The complete list of SQL scripts is described in the Appendix.

Note Since this step executes a post-action script, which is also executed after the synchronize step, log files log/upg08_*.log created in this step will be overwritten. Please make a backup copy if you need them

Classification Stage 2 (optional)

In Eigner PLM 5.0 a new concept – Attribute pool was introduced. This step converts attributes within the customer dump to this new concept. This step must be skipped, if you are migrating from PLM5.x version or newer.

It is required that the attributes are already migrated and only the classification lists is updated. Click Perform mapping.

Note Please check log files located in the data/cla/ directory. For more information please refer to Chapter 4, section Classification.

Note If the customer has created new classes and attributes in the production system, after the customization upgrade classes, attributes, and domain values must also be copied and migrated.

Manual Dump Changes

LogiView

After the upgrade all LogiView changes made by customers in the standard must be made again. The previous changed standard procedures can be found within the S<timestamp> LogiView models.

STEP

After the upgrade or productive takeover you have to check the defaults "EDB-STP-DEF-NO-REF" and "EDB-STP-DEF-ORG-REF". During the upgrade to Agile e6 the values of these defaults were changed from "EP" to "NN". If the values of these defaults in your productive environment are already set "NN", nothing needs to be done.

These defaults are used as field defaults in several fields (<TABLE>.STEP_NO_REF, <TABLE>.STEP_ORG_REF).

This upgrade tool provides a default script (ora\sql\upg_org_ref_default.sql), which changes all values of these fields in the standard tables from "EP" to "NN".

Please review the script according to your customizing. If you have not done any changes to your customizing regarding STEP_ORG_REF,STEP_NO_REF you can run the scripts without changes.

If you do not run the script it is possible to duplicate PART_ID's in T_MASTER_DAT, because these two fields are used in the unique key constraint of T_MASTER_DAT.

Apply Customer Specific Dump Changes

If you have made customer specific dump changes to standard db objects (like views, packages, procedures and triggers, database constraints, field defaults, etc.) and these dump changes are not included in the DTV Repository, you must apply these changes again on your new dump. Please keep in mind that the standard db objects you made the changes for may have been changed.

Note Since DataView does not support GLOBAL TEMPORARY tables (and even Upgrade Tool), you have to remove any index information from the repository for such tables if they are defined in the DataView repository. Otherwise, errors will occur during the synchronize phase while creating indexes for global temporary tables. Such error messages must be ignored. The tables have to be adapted manually.

Chapter 4

Additional Information

Following special handling modules are integrated into the Upgrade Tool. Therefore, no additional actions are required to migrate the involved content. The steps described below are already integrated in the upgrade process and will be executed automatically.

Browser Upgrade

With Agile e6.0 release a new browser was introduced. It is based on new database tables having a prefix T_EXP_*.

During the upgrade, standard content of these new tables is inserted. In addition, non-standard entries from the “old” browser are migrated.

Cleanup BVB_ARTIKEL

Some inconsistencies in the existing item data structure are eliminated during this specific upgrade step. Here a list of executed activities:

- Redundant BVB_ARTIKEL entries are dropped.
- All BVB_ARTIKEL records without corresponding T_MASTER_DAT entries are deleted.
- Missing BVB_ARTIKEL records are inserted.
- Missing values of T_MASTER_DAT.UNIT are inserted into BVB_MASSEH.

STA_LUT

For each entry in T_CHK_STA a valid reference to T_STA_LUT is established with values in T_CHK_STA.EDB_STA_REF. Additionally, EDB_LEVEL is filled within the table T_STA_LUT.

Note The values of the table T_STA_LUT must be reviewed after the customizing upgrade.

Project References in Processes

There is a known problem within the projects workflow in Agile e6.0. In case of creation of a relation between process and project, which has several versions, several entries are shown in the list because the PROJ_ID (kind of project name shortcut) is not unique over the stored versions. For this reason a new DB field is added: T_PRC_DAT.EDB_PRO_REF.

If necessary this field is filled with values during the upgrade step upg08_postaction.

BVB_ARTMEH Upgrade

During this step, an upgrade of BVB_ARTMEH* tables is performed. SQL script artmeh_1.sql and artmeh_2.sql is executed within the upg08_postaction step.

Favorites Upgrade

The Upgrade Tool inserts several data items into browser tables to make favorites and stored queries visible in the Agile e6 client browser window.

Classification

Overview attribute concept (old):

- Attributes are defined class specific in the ATT concept.
- Domain values for an attribute are defined in static menus.
- No release procedures and status management for classes and attributes.

Overview new pool concept:

- Attributes can be defined class independent.
- Pool attributes can be assigned to more than one class.
- Domain values for a pool attribute can be stored in special domain tables.
- For every class it can be specified which domain value is valid.

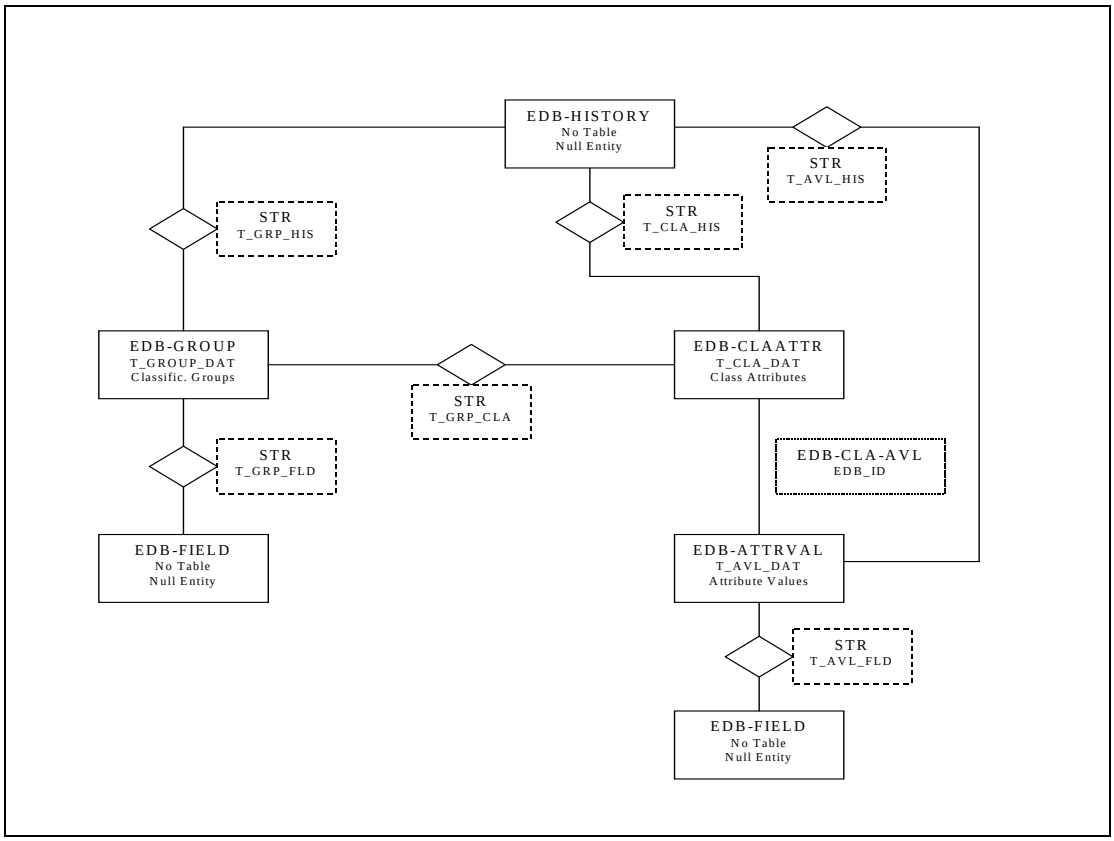
The migration includes:

- Merge attribute definitions. Attributes are considered as identical if the following values are identical:

- C_LETTER
- C_TITLE
- C_TYPE

If you have defined C_LETTER and C_TITLE as multi-lingual fields (standard since axalant 2000), then you have to define with the Parameter “DB language” which language is used as standard for the merge.

- Initial load of the attribute value pool including the activation of the attributes for special classes.
- Update classification lists.
- Update used field name.
- Set of attribute ATT_VAL_REF in the classification lists.
- Update field and mask definition (if you have defined your own forms for classes).



Note If possible, do not create or modify basic definitions of classes and attributes between customization upgrade and takeover data from production system. This influences the migration steps which must be executed after the takeover process.

- No new classes and attributes are created. Only classification list tables must be defined as reference tables.
  - T_GRP_ART
  - T_GRP_DOC
  - T_GRP_ORD
  - T_GRP_PRO
- Customers have created new classes and attributes in the production system after the customization upgrade. In addition to the classification list table classes, attributes, and domain values must be copied and migrated.
  - T_GROUP_DAT
  - T_GROUP_STR

- T_GRP_FLD

Workflow Takeover

Note For the customers using an Upgrade Tool version older than Agile e6.0.2, special generic workflow masks are copied during the takeover phase.

After taking over of the common tables, workflow masks are deleted, which are currently presented in the customer dump. Generic workflow masks are copied from the production database. The following tables are involved in this procedure:

- T_MASK
- T_MAS_FLD
- T_FIELD
- T_MENU
- T_MEN_SEL

Workflow masks have a special naming convention: EDB-WFL-<CID_OF_ACTIVITY>

Entries within T_FIELD follow naming rule EDB-DEC<CID_OF_ACTIVITY>

- T_MEN_SEL entries contain names like EDB-SEL-<CID> and EDB-NSEL-<CID>
- T_MENU entries of C_BUT_EDT / C_BUT_SEL / C_BUT_NOS are considered only for this action.

During the takeover phase, all statements will be generated and executed within the Upgrade Tool and additionally written to the full.log file.

Chapter 5

Appendix A: CMD Scripts

Here is a short description of all shell scripts included in the upgrade download package.

Shell script	Description	Called SQL scripts
batch_upgrade.cmd	Performs a complete upgrade in the batch mode	
convert_it.cmd	Convert XML data file for a single table to HTML files. This script is called from xml2html.cmd.	--
exp_dmp.cmd	This script helps you to export oracle database to a dump file.	--
imp_dmp.cmd	This script helps you to create database users and import oracle dump files.	--
preaction_template.cmd	This file is for upgrade internal use only! It is used as a template for creating the file preaction.cmd, which is needed if all upgrade steps are executed in batch mode.	--
start_upg.cmd	Start upgrade user interface.	--
upg01_pre_cleanlog.cmd	Cleanup all log files within current upgrade project.	--
upg03_preaction_sql.cmd	Run several SQL scripts depending on source, target and customer product versions. Called SQL scripts are saved in the file 03_preaction_sql.log	source: CRE_REP_EDB.SQL > 03_01_CRE_REP_EDB.LOG source: TRUNC_LVTABS.SQL > 03_TRUNC_LVTABS.LOG source: GRANT_SELECT_T_CONSTRAINT.SQL target: CRE_REP_EDB.SQL > 03_15_CRE_REP_EDB.LOG CLEANUP_C_ID_NULL.SQL > 03_03_CLEANUP_C_ID_NULL.LOG ANA_LV.SQL > 03_04_ANA_LV.LOG <=AXA SP 1: PST10P2TOP3.SQL > 03_06_PST10P2TOP3.LOG <=AXA SP 1: ORA403-404.SQL >

Shell script	Description	Called SQL scripts
		03_07_ORA403-404.LOG <=AXA SP 1: AXASP1_TO_SP2.SQL > 03_08_AXASP1_TO_SP2.LOG <=AXA SP 3: DTV405-406.SQL > 03_10_DTV405-406.LOG <=AXA SP 3: UPD_T_SELECTION.SQL > 03_11_UPD_T_SELECTION.LOG <=e6 LA: DTV406-407.SQL > 03_12_DTV406-407.LOG <=e6 GA: DTV407-430.SQL > 03_13_DTV407-430.LOG <= e6.0.1: DTV430-431.SQL > 03_14_DTV430-431.LOG CUSTOMER_DATABASE_TASKS.SQL > 03_17_CUSTOMER_DATABASE_TASKS.LOG
upg04_dtv_get.cmd	Get XML files for the step “DTV-Upgrade” in shell mode.	--
upg05_dtv_update.cmd	Proceed XML files for the step “DTV-Upgrade” in shell mode.	--
upg06_sync_get.cmd	Run the step “Analyze repository” in shell mode.	
upg07_sync_update.cmd	Run the step “Synchronize repository” in shell mode. Called SQL scripts are saved in the file 07_sync_update.log	<= PLM5.x: before_sync.sql > 07_01_before_sync.log
upg08_postaction.cmd	Run some SQL scripts, which are necessary after performing “Synchronize repository” upgrade step. Called SQL scripts are saved in the file 08_postaction.log	cre_rep_edb.sql > 08_01_cre_rep_edb.log cleanup.sql > 08_02_cleanup.log db_defaults.sql > 08_03_db_defaults.log artmeh_1.sql > 08_04_artmeh_1.sql update_defartmehr.sql > 08_05_update_defartmehr.log artmeh_2.sql > 08_06_artmeh_2.sql special602.sql > 08_07_special602.log get_compile_all.sql > compile_all.sql compile_all.sql > 08_08_compile_all.log invalid_objects.sql > 08_09_invalid_objects.log
upg09_common_get.cmd	Generate XML files for several upgrade steps, one for each	del_and_save_lvmodel.sql > 09_01_del_and_save_lvmodel.log

Shell script	Description	Called SQL scripts
	common PLM module. Called SQL scripts are saved in the file 09_common_get.log	
upg10_common_update.cmd	Proceed common PLM modules XML files. Called SQL scripts are saved in the file 10_common_update.log	<=PLM5.x: edb_explorer.sql > 10_02_edb_explorer.log
upg11_cla.cmd	Upgrade PLM Classification.	--
upg13_prod1_takeover.cmd	Start a user interface to proceed with "Takeover production data". Scripts upg14_prod2_rep_update and upg15_prod3_postaction have to be executed after that.	--
upg14_prod2_rep_update.cmd	Synchronize repository (this script includes all necessary pre-action and post-action calls).	SQL Files of upg07_sync_update.cmd and upg08_postaction.cmd are called again get_numvalue.sql > 14_01_get_numvalue.log set_numvalue.sql > 14_02_set_numvalue.log cre_rep_edb.sql > 14_03_cre_rep_edb.log
upg_env.cmd	Common upgrade settings, like Java, JRE, Path, etc.	--
xml2drop.cmd	Generates ora/ ref_data_tab.par and ora/ref_data_tab_drop.sql files for manually import/export of production tables. You have to configure which tables are relevant for the takeover step before.	--
xml2html.cmd	Converts generated XML files to HTML format for a module. Example: "xml2html.cmd dtv" Or "xml2html.cmd all"	Possible module names ('all' for all modules): dtv edb brw dode lgv wfl chg gdm rmt gtm ase cmg
check_edb_crm_null.sql	Reports invalid values in change management columns	EXECUTE WA_BUG7107296;

Chapter 6

Appendix B: Configuration Files in /conf/

- ApplicationParameter.xml - Global application configuration file
- aseDD.xml - Configuration file for upgrade module ASE (Advanced Structure Editor)
- brwDD.xml - Configuration file for upgrade module BRW (Explorer)
- chgDD.xml - Configuration file for upgrade module CHG (Change Management)
- wflDD.xml - Configuration file for upgrade module WFL (Workflow)
- dodeDD.xml - Configuration file for upgrade module DODE (Print Studio)
- dtvDD.xml - Configuration file for upgrade module DTV (DataView Repository)
- edbDD.xml - Configuration file for upgrade module EDB (Agile PLM configuration)
- gdmDD.xml - Configuration file for upgrade module GDM (Office integration)
- gtmDD.xml - Configuration file for upgrade module GTM (Classification)
- lgvDD.xml - Configuration file for upgrade module LGV (LogiView)
- rmtDD.xml - Configuration file for upgrade module RMT (Requirement Management)
- cmgDD.xml - Configuration file for upgrade module CMG (configuration Management)
- special.xml - Configuration file for the step “Synchronize Repository”
- specialreplace.xml - A sample configuration file for special replace cases
- ref_tables.xml - Configuration file for the upgrade step “Takeover production data”
- wfl_ctl.xml - Configuration file for Workflow mapping
- cla_ctl.xml - Configuration file for Classification Upgrade (PLM5.x to Agile e6)
- cla_post_ctl.xml - Configuration file for Classification Upgrade (PLM5.x to Agile e6)
- insert.xsl, delete.xsl, update.xsl, upgrade.xsl - XSL-stylesheet for converting XML control files to HTML, used by xml2html.cmd script

- ref_data_tab_drop.xsl - XSL-stylesheet for generating SQL script, which drops production data tables in the customer dump. This stylesheet is used by xml2drop.cmd
- ref_data_tab_par.xsl - XSL-stylesheet for generating table list clause for oracle EXP command, which can be used alternatively to transfer production data tables from production database. This stylesheet is used by xml2drop.cmd
- cla_stl.xsl - XSL-stylesheet for configuration file for classification upgrade (Axalant 2000 to PLM5.x), which generates a HTML output of performed mapping operations
- dtv_dd.dtd - Document type definition file for module control files

Chapter 7

Appendix C: Contents of the Folder “upgrade/conf/template”

- ApplicationParameterORACLE.xml - Upgrade Tool settings file with standard values for an ORACLE database. Copy this file to upgrade/conf to reset the application settings.
- cla_ctl.xml - Example for the classification control file.
- cla_ctl_with_multi_lang.xml - Example for the classification control file (with multi-lingual definition of the attributes).
- cla_ctl_with_multi_lang_repl.xml - Example for the classification control file (with multi-lingual definition of the attributes and additional fields for database replication).
- cla_ctl_with_repl.xml - Example for the classification control file (with additional fields for database replication).
- ref_tables.xml – This configuration file is used during the takeover phase for read only reasons. Based on settings in this file, a ref_table.xml file is created in the CONF directory during the “Create ref. File” step.
- special_move.xml - Examples for the special case with table field moving.
- special_rename.xml - Examples for the special case with table field renaming.
- Special.xml - Default template.
- specialreplace.xml - Examples for special replace cases.

Chapter 8

Appendix D: SQL Scripts

Here is a short description of SQL scripts delivered with the Agile upgrade tool. All script executions create a log file in the log/ directory which are named like the script itself with a prefix like 08.

SQL Script	Description
ana_lv.sql	Analyze LogiView content in the customer dump. Please control the log file of this script as described in the manual.
artmeh_1.sql	Conversion of BVB_ARTMEH* tables, for PLM version <= Eigner PLM 5.x (Part 1).
artmeh_2.sql	Conversion of BVB_ARTMEH* tables, for PLM version <= Eigner PLM 5.x (Part 2).
axasp1_to_sp2.sql	Upgrade axalant sp1 to axalant sp2.
before_sync.sql	This script has to be executed before running the step “Synchronize Repository”. It is done by default with the standard upgrade configuration. It prepares the table T_STA_LUT and drops triggers, because otherwise it is impossible to insert rows in the involved tables.
char_check.sql	Creates character scan report
cleanup.sql	This script cleans up some dump content and is executed automatically after the step “Synchronize Repository”.
cleanup_c_id_null.sql	This script cleans up some inconsistencies in the customer dump (like rows with negative C_ID values). It must be executed before DTV-upgrade.
compare_lgv.sql	Performes LogiView comparison between two different schemas
convert_to_utf8.sql	Convert (VAR)CHAR2 columns from BYTE to CHAR semantic after a dump import
customer_database_tasks.sql	This script executes some cleanup statements to get rid of common dump inconsistencies.
cre_plm_tbs.sql	Creates missing Oracle tablespaces.
cre_plm_usr.sql	Create a database user. This script needs 2 parameters: username and password.
cre_rep_edb.sql	Create all schema objects (tables, views, indexes, packages, triggers, sequences, etc.) and insert number server rows, most of them already exist, so a lot of errors will be logged after executing this script.
db_defaults.sql	This script overwrites default constraints on the database level, since they are different from DataView default definitions. It is executed automatically after the step “Synchronize_repository”.

SQL Script	Description
del_and_save_lvmodel.sql	Delete standard LogiView content and save customized models with a prefix “SAVE-“.
drop_obsolete_objects.sql	Ths scipt contains delete statements for all desupported database objects. Such objects are not deleted by the upgrade tool
dtv405-406.sql	Pre-action script has to be executed before DTV-upgrade for customer dump version <= axalant SP3.
dtv406-407.sql	Pre-action script has to be executed before DTV-upgrade for customer dump version <= Agile e6.0 LA.
dtv407-430.sql	Pre-action script has to be executed before DTV-upgrade for customer dump version <= Agile e6.0 GA.
Dtv430-431.sql	Pre-action script has to be executed before DTV-upgrade for customer dump version <= Agile e6.0.1.
Dtv431-432.sql	Pre-action script has to be executed before DTV-upgrade for customer dump version <= Agile e6.1.
Dtv432-433.sql	Pre-action script has to be executed before DTV-upgrade for customer dump version <= Agile e6.0.3.
edb_explorer.sql	Converts DTV explorer to Agile e6 EDB-explorer. This step is executed once after common modules upgrade.
fill_edb_in_contexts.sql	Migrates PLM5.x IN_CONTEXT values to EDB_IN_CONTEXTS introduced with e.6.0
getoradrop.sql	Get script “dropall.sql” which cleans up a complete database schema.
get_compile_all.sql	Generates a script to recompile all db objects.
get_numvalue.sql	This script is executed in the production database and generates a file named “ set_numvalue.sql ” after takeover step. This file updates number server values in the customer database.
get_rebuildidx.sql	This script generates a file named “ rebuildidx.sql ” to rebuild all indexes in a right tablespace of a schema. It has 5 parameters for tablespaces: EDB EDB_IDX EDB_LOB EDB_TMP EDB_TMPIDX
grant_dtv.sql	grant SELECT rights to DTV tables within the target reference dump
Invalid_objects.sql	Lists all objects still invalid in the dump.
levind_in_stalut.sql	This script is called automatically after “Synchronize Repository” and converts records in the table T_STA_LUT. It is needed only for upgrades form <= Eigner PLM to >= Agile e6.
ora403-404.sql	Pre-action script has to be executed before DTV-upgrade for customer dump version <= axalant SP1.
Pst10P2ToP3.sql	Pre-action script has to be executed before DTV-upgrade for customer dump version <= axalant SP1.
special602.sql	Special data modifications for upgrade to 6.0.2

SQL Script	Description
special_61.sql	Special data modifications for upgrade to 6.1.0
takeover_dtv.sql	Takeover DataView internal repository from target reference dump
testconnection.sql	Dummy sql script for testing sqlplus connections
trunc_lvtabs.sql	Truncate all LogiView tables. This script is executed on reference dumps only!
update_customers_UIC.sql	This script has to be executed on the customer dump before proceeding with the upgrade.
upd_t_selection.sql	This script is a workaround for incompatible changes for table T_SELECTION in Eigner PLM5.0 This script is already executed on all reference dumps delivered with Agile upgrade tool. It will automatically executed IDs that are necessary in the step “preaction-scripts”.
update_defartmehr.sql	Fills DEFARTMEHR.BVB_ARTIKEL field during the CLEANUP_BVB phase of the step “Postaction”.
upg_org_ref_default.sql	A sample update script for existing STEP_NO_REF and STEP_ORG_REF values.

Chapter 9

Appendix E: Directories

Directory	Description
cmd	Windows shell scripts of the upgrade tool.
conf	Configuration xml files.
conf\template	Some templates of xml configuration files. The upgrade tool does not use these files. The only exception is the file ref_tables.xml. It will be read by the tool to recreate /conf/ref_tables.xml.
data	This directory contains several subdirectories, each for a module – like BRW, EDB, etc. For each module delete, insert, and update xml files are created. After performing of these operations on the customer database, error xml file is written. Additionally, html files generated for a module are saved here. A file customizing.log in this directory contains conflicts caused by customizing of the original dump
data\dtv	DataView upgrade files as described above are stored here. Please read carefully the file customizing.log because it contains user-exit conflicts.
data\sync	Log files of the synchronize repository upgrade step are stored in this directory.
data\cla	Log files of the classification upgrade step are stored in this directory.
doc	Upgrade tool documentation.
dumps	Database dumps can be stored here. Dumps, which are imported / exported by shell scripts imp_dmp.cmd and exp_dmp.cmd, have to be stored in this directory.
img	Upgrade Tool images
lib	Upgrade tool java executables.
log	Log files of all sql scripts and common application log files.
ora\sql	Oracle SQL-scripts.
scripts	Unix / Linux shell scripts of the upgrade tool.
tmp	Temporary files
unsupported	Upgrade Tool files, that are usupported

Chapter 10

Appendix F: Migration Rules

Standard rules are available for insert, update, and delete and these rules are verified during the comparison of the table contents. They can be overwritten by special definitions.

Standard Rules for Delete

Data records deleted in the standard are also deleted in the customer dump.

Customer-specific dump	Source master	Target master	Action
+	+	-	Delete

Standard Rules for Update

Data records existing in the source master dump that were deleted in the customer dump are not re-created. Existing data records are updated. The standard changes overwrite the customer changes. Special rules apply on field level to protect customer-specific changes.

Customer-specific dump	Source master	Target master	Action
+	+	+	Update

Standard Rules for Insert

Data records not existing in source master dump or in the customer dump are added.

Customer-specific dump	Source master	Target master	Action
-	-	+	Insert

Special Rules

Customer changes that will not be overwritten by standard changes are:

- Field defaults and check strings.
- Customizing hints for fields containing userexits.
- Special handling for mask components.

- Replacements of strings (-> specialreplace.xml).

Chapter 11

Appendix G: Repeatable Tasks

The following table gives an overview of steps which may be repeated without re-importing the customer dump.

Step	Repeatable
run pre-action-scripts	YES
DTV-upgrade "Create Files"	YES
DTV-upgrade "Perform Insert,Update,Delete"	NO
run before-sync-scripts	YES
Synchronize_Repository "Analyze"	YES
Synchronize_Repository "Synchronize"	YES
run after-sync-script	NO
run before-common-scripts	NO
EDB-upgrade (Configuration) "Create Files"	YES
EDB-upgrade (Configuration) "Perform Insert,Update,Delete"	NO
BRW-upgrade (Browser) "Create Files"	YES
BRW-upgrade (Browser) "Perform Insert,Update,Delete"	NO
DODE-upgrade (Print Studio) "Create Files"	YES
DODE-upgrade (Print Studio) "Perform Insert,Update,Delete"	NO
LGV-upgrade (LogiView) "Create Files"	YES
LGV-upgrade (LogiView) "Perform Insert,Update,Delete"	NO
WFL-upgrade (Workflow) "Create Files"	YES
WFL-upgrade (Workflow) "Perform Insert,Update,Delete"	NO
CHG-upgrade (Change Management) "Create Files"	YES
CHG-upgrade (Change Management) "Perform Insert,Update,Delete"	NO
GTM-upgrade (Classification) "Create Files"	YES
GTM-upgrade (Classification) "Perform Insert,Update,Delete"	NO
GDM-upgrade (Office Suite) "Create Files"	YES
GDM-upgrade (Office Suite) "Perform Insert,Update,Delete"	NO
RMT-upgrade (Requirement Management Traceability) "Create Files"	YES
RMT-upgrade (Requirement Management Traceability) "Perform Insert,Update,Delete"	NO

ASE-upgrade (Advanced Structure Editor) "Create Files"	YES
ASE-upgrade (Advanced Structure Editor) "Perform Insert,Update,Delete"	NO
CMG-upgrade (Configuration Management) "Create Files"	YES
CMG-upgrade (Configuration Management) "Perform Insert,Update,Delete"	NO
run after-common-scripts	YES
Upgrading Classification (optional)	NO
Classification attribute inheritance	NO
special replace	NO

Chapter 12

Appendix H: Error Messages

File Name: <upg_root>\data\sync\sync_analysis.log

After the step: **Synchronize Repository > Analyze**

#	Error	Action	Description
1.	DTV field T_ACT_CLA.C_LEN_VIS has the type I10 but its data is incomplete.	Do NOT ignore the message.	FIS-ID: 10288 > Loaderfile In T_FIELD, entries for the 3 fields are not correct: ▫ T_ACT_CLA.C_LEN_VIS ▫ T_ACT_CLA.C_LEN_VIR ▫ T_ACT_CLA.C_WID_VIS
2.	DTV Field T_CHK_DSP.C_ID is found twice	Ignore the message	DTV system fields (also in the default dump) as an exception in the DTV repository. Do not make any changes. The following are the DTV system fields: ▫ C_ID ▫ C_VERSION ▫ C_LOCK ▫ C_UIC ▫ C_GIC ▫ C_CRE_DAT ▫ C_UPD_DAT ▫ C_ACC_OGW
3.	DTV Field V_CHG_OPR_FRM.EDB_EWO_EDB_ID has type I10 but its table does not exist	Ignore the message	T_FIELD contains required fields without an entry in T_TABLE. Do not make any changes.
4.	DB field T_PTL_DAT.DESCR_GER has misbegotten sibling T_PTL_DAT.DESCR	Ignore this message from the Release Agile e6.x onwards	Error is in the default dump axalant2000sp3. Here, the fields DESCR, DESCR_GER, DESCR_ENG and DESCR_FRA exist. The table belongs to the module WEB-Portal, which is not part of the default anymore from the Release Agile e6.x onwards. The associated tables are removed during the upgrade

#	Error	Action	Description
			procedure
5.	DB field V_CHG_OPR_ART.EDB_PARENT is of the type - (CHAR,0,null).	Ignore the message	View is defined in the database. View – Fields do not need a TYPE because this is already defined in the master table. FIS-ID:10289 – FAQ entry - no error.
6.	DTV Field C_GRP_USR.C_ACCESS has no type.	Ignore the message	View in the DTV repository. View – Fields do not need a TYPE because this is already defined in the master table.
7.	DB Field SYS_NC00041$ from index EDB_CLA_IND2 does not exist in table T_CLA_DAT.	Ignore the message	EDB_CAL_IND2 is a function based index. This has been added to the default dump for the performance reasons from the Release Agile e6.0 onwards. FIS-ID:10290 - FAQ entry – no error.
8.	DB Field BVB_ARTIKEL.PART_NAME_FRA has been added but this field seems to have already been multi-language before. It is empty now.	Ignore the message	This message is displayed if it is already defined as multi-lingual in the DTV repository, but the fields are not available in the database.
9.	Missing standard value for NOT NULL field T_MASTER_DAT.UNIT. Append a field default entry to the special cases file.	Do NOT ignore the message	This message is displayed if Tabellen.Feld (Table.Field) NOT NULL is created or modified, but no default value is defined in the file <upg_root>/conf/special.xml. Create missing field default in file spezial.xml. Example: <FieldDefault> <FieldName>T_MASTER_DAT.UNIT</FieldName> <FieldType>S</FieldType> <FieldSize>20.0</FieldSize> <DefaultValue> <Value>nix-unit</Value> </DefaultValue> </FieldDefault>
10.	CREATE UNIQUE INDEX EDB_GDM_PRP1 ON T_GDM_PRP (PROPERTY_NAME, PROPERTY_TYPE) TABLESPACE EDB_IDX ORA-01452: cannot CREATE UNIQUE INDEX; duplicate keys found.	Do NOT ignore the message	Error in the PLM5.x dump with Office Suite hotfix4. The Property_Name “PARAGRAPHCOUNT” exist twice. Delete one of the data records. Check for possible usage in: T_GDM_TPL_PRP T_GDM_PRP.C_ID = T_GDM_TPL_PRP.C_ID_2

#	Error	Action	Description
			FIS-ID: 10622 – Dataset can only be deleted via SQL-Plus.
11.	It is not supported to change the type of the field T_DTO_FNC.C_DESCRIP_ENG from (S,255,0) to (S,20,0)	Ignore the message ONLY for the table T_DTO_FNC	FIS-ID:10291 - solved with upg3.0.39 – new plm601upgref dump. See the error descriptions 12 or 13 for further details.
12.	Had no permission to change the type of the field T_CHK_PAR.REC_ID from (I,10,0) to (S,40,0). OR Permission OK. Ready to change the type of the field T_CHK_PAR.REC_ID from (I,10,0) to (S,40,0).	Do NOT ignore the message. Use the upgrade tool: upgtool 3.0.24	At the end of the upgrade the data type has to be correct. FIS-ID:10152 - solved with upg3.0.34
13.	Missing permission to change the type of the field BVB_ARTMEHUFK.NACHMEHR from (S,4,0) to (I,10,0).	Do NOT ignore the message. Use the upgrade tool: upgtool 3.0.38	At the end of the upgrade, the data type has to be correct. FIS-ID:10276, 10262 - solved with upg3.0.38. In the course of these FIS-IDs, the file special.xml has been modified. Change the datatype BVB_Tables to FALSE.
14.	Did not delete DB index A_T_ORD_HIS.	Ignore the message	