Agile Product Lifecycle Management

Database Upgrade Guide

Release 9.3.5

E61168-01

January 2016

ORACLE

Agile Product Lifecycle Management Database Upgrade Guide, Release 9.3.5

E61168-01

Primary Author: Jacinda Johnson

Contributing Author: Frank Mocek

Contributor:

Contents

Preface

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

Audience

This document is intended for administrators and users of the Agile PLM products.

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) website 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 website, 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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

What's New

This chapter introduces additional upgrade paths and enhancements in release 9.3.5.

AUT Tools

There are several tools included in AUT version 9.3.5 help to further organize your database after an upgrade. The tools are located in the AUT_HOME\bin directory.

The tools and their descriptions are as follows:

- gap_collector: Identifies unused or orphan ID numbers that can be reused.
- reorder_query: Clears out temporary records and gaps to compact the query table to reuse sequence IDs.
- update_content_url: Updates content URLs for Agile PLM databases before release 9.3.2.

Supported Agile PLM Upgrade Paths

Upgrade of your Agile PLM database to version 9.3.5 is supported from any of the earlier versions listed below.

- Agile 9.3.4
- Agile 9.3.3
- Agile 9.3.2
- Agile 9.3.1.2
- Agile 9.3.1.1
- Agile 9.3.1
- Agile 9.3.0.3
- Agile 9.3.0.2
- Agile 9.3.0.1
- Agile 9.3
- Agile 9.2.2.7
- Agile 9.2.2.6
- Agile 9.2.2.5
- Agile 9.2.2.4

- Agile 9.2.2.3
- Agile 9.2.2.2
- Agile 9.2.2.1
- Agile 9.2.2
- Agile 9.2.1.6
- Agile 9.2.1.5
- Agile 9.2.1.4
- Agile 9.2.1.3
- Agile 9.2.1.1
- Agile 9.2.1
- Agile 9.2.0.2
- Agile 9.2.0.1
- Agile 9.2
- Agile 9.1 SP1-SP4
- Agile 9.0 SP1-SP7
- Agile 8.5 SP1-SP7

Supported Agile Advantage Upgrade Paths

Upgrade of your Agile Advantage (AA) database to version 9.3.5 is supported from the following version only:

- AA 2006

2

Preparing for the Upgrade

You can upgrade your database from your currently installed version to Agile PLM 9.3.5 using the Automated Upgrade Tool.

The upgrade tool takes inputs from a property file and completes the upgrade using your existing database as the source database. When complete, the destination database is upgraded to the 9.3.5 release.

Preparing the System

To become familiar with the upgrade procedure, it is recommended that you prepare a separate test environment and perform the upgrade there first. It is highly recommended that you run the upgrade on the system that is hosting the destination database.

Agile PLM supports Oracle 11g and 12c databases. You MUST have the Oracle Database Server on the system where you are installing the Agile database before starting the upgrade. For information on installing Oracle databases, refer to the related documentation available on the Oracle Technology Network (OTN) website http://www.oracle.com/technetwork/documentation/agile-085940.html.

If you are upgrading from an 8.5 product and your Oracle 8i home is not installed on the system where you are performing the upgrade, you must map (or mount) a local drive on the destination system to point to the existing Oracle 8i home. Also, if you are upgrading from an 8.5 product, you must create a new Oracle database instance for the new Agile PLM product version and try to reuse the existing 8.5 Oracle instance.

Note: Ensure binary compatibility of the mounted binaries (for example, ensure that you map a drive containing Windows binaries to a Windows system only).

You must also have JRE 1.8 installed. The JAVA_HOME environment variable must be set to the path for that JRE on the system where AUT is run.

Database Upgrade Planning

The entire upgrade process may take a considerable amount of time depending on the size of the database and the hardware configuration. Based on past experience, 75% of the upgrade process is spent on upgrading and backing up the database. We strongly recommend running the schema backup and upgrade on a test system so that appropriate time can be scheduled for the production deployment.

There are two types of upgrade:

- **Upgrade destination in place** - Source database is upgraded with the existing database user. In the *aut.properties* file, the parameter sourceEqualsDestination is set to true.
- **Upgrade source to destination** - A new database user is created in the destination database instance. In the *aut.properties* file, the parameter sourceEqualsDestination is set to false.

For Agile PLM or Agile Advantage (AA) upgrade, a new database user is created in the destination database instance using *useragile.sql*, and then the source database is upgraded to that user.

Task Overview

The sequence of actions to be performed for an upgrade is as follows:

1. Undeploy the Agile PLM application.
2. Upgrade the database to 9.3.5:

   a. Download the Agile PLM database installer and Automated Upgrade Tool (AUT). For instructions on downloading the database installer, see the *Agile PLM Database Installation Guide*. For instructions on downloading the AUT, see "Downloading the Automated Upgrade Tool" on page 2-2.

   b. Configure and run AUT. For instructions on configuring AUT for each type of upgrade, see "Understanding the Property Files" on page 3-1 and "Whether Source Equals Destination" on page 3-2.

   c. Generate maintenance scripts for the target release: Run the database installer, choosing the **Generate Maintenance Scripts only** option. See "Generating Database Maintenance Scripts Only" on page 2-3.

   d. Configure the database as described in "Configuring the Agile PLM Database" on page 5-1.
3. Install the Agile PLM application software.
4. Deploy the Agile PLM application.

Downloading the Automated Upgrade Tool

The Automated Upgrade Tool can be downloaded from Oracle Support Services.

To download the software:

1. Log in to My Oracle Support (https://support.oracle.com).
2. Under **Patches & Updates**, click **Product or Family (Advanced Search)**.

   1. Select the **Include all products in a family** check box.

   2. Enter the following search parameters:

   Product: Oracle Agile Applications

   Release: Agile PLM Tools 9

   Platform: Select the appropriate platform, for example, Microsoft Windows (32-bit).

   3. Click **Search**.

   4. Links to Averify and AUT are displayed in the search results.

   5. Select the Patch ID and click **Download**.

3. From the Patchset zip file, extract the contents of the AUT.zip file to your destination database server. After the AUT is extracted, all of the files are located in an AUT directory on the local drive. In this document, this directory is named the AUT_HOME.

Creating the Agile Database Instance

Unless you plan to upgrade in-place, a new database instance must be created before you can upgrade your existing database. This database instance will be used as the destination database during the upgrade process.

To create the Agile PLM 9.3.5 database instance, follow the instructions in the Agile PLM *Database Installation Guide*.

After the 9.3.5 database instance is created, run the *useragile.sql* script, located in the **<ORACLE_HOME>/admin/<SID>/create/<agile schema user>** directory, to create an empty user in the database.

Note: Change the token values inside the @ symbols in the *.sql* file before running the script. Replace username **agile** with the desired username to create the user.

Generating Database Maintenance Scripts Only

To upgrade an existing Agile schema, you do not need to recreate the database instance. You only need to generate the database maintenance scripts for the target release.

The procedure for generating maintenance scripts is detailed below.

On Windows:

1. Run the database installer file, agile9350db_oracle.exe, and enter appropriate information for Destination Location, Database Size and Oracle home, as described in the *Agile PLM Database Installation Guide.*
2. When prompted for the Oracle SID, change the default SID **agile9** to the existing SID.
3. Select the **Generate database maintenance scripts only** option. Click **Next**.
4. Follow on-screen directions to generate scripts.

A confirmation message appears when you have successfully generated the maintenance scripts.

On UNIX:

1. Run the agile9database.sh script as described in the *Agile PLM Database Installation Guide.*

   $ chmod u+x agile9database.sh

   $ ./agile9database.sh [Enter]
2. When prompted to choose the installation mode, enter **S** to generate maintenance scripts only. This action generates the database scripts but does not execute them.
3. Follow on-screen directions to generate scripts.

A confirmation message appears when you have successfully generated the maintenance scripts.

Validating the Databases

You should validate the source and destination databases to ensure a successful upgrade by performing the following checks:

Run the latest averify script on the source database and fix any errors.

For each database upgrade, you should know the following:

- Source and destination database user and password
- Destination sys user and password
- TNS configuration for source and destination database on the respective database systems
- Paths to the Oracle homes
- Map the source and destination Oracle homes, if necessary.
- Validate the TNS entries of the source and destination databases.

  On a command line, change ORACLE_HOME to point to the Oracle home of the source database, then try to connect to it using the TNS name. Repeat this procedure for the destination database.
- Make sure the destination database user exists and there are no objects in the schema.

  Note: This does not apply if your source database is also your destination database.

- Make sure the source and destination database users have the same roles and privileges.
- Make sure the Agile tablespaces in the destination and source databases match. Also, the destination database should have unlimited tablespace.
- Verify that the available tablespace in the destination database is sufficient to complete the database upgrade.

3

Understanding the Property Files

AUT uses property files to supply the input information to the database scripts. If you are not upgrading a Product Cost Management 8.5 or Program Execution 8.5 database, you should only complete the *aut.properties* file.

If you are also upgrading a Product Cost Management 8.5 or Program Execution 8.5 database, you should complete either the *psupgrade.properties* file or the *peupgrade.properties* file, in addition to the *aut.propertie*s file.

The aut.properties File

The *aut.properties* file, located in the $AUT_HOME\config directory, contains user-supplied information on the source and destination databases. The source database parameters contain information about the database to be upgraded. The destination database parameters contain information about the user and host details of the database where the source database is being upgraded. The database scripts use the parameters during the upgrade process.

Defining the aut.properties Parameters

You can edit the *aut.properties* file to define the following parameters:

- RMW Only Details
- Whether Source Equals Destination
- Destination Details
- Source Database Backup
- Sequence Project
- Source Details
- Averify Details
- Ignore Import Warnings
- Language Details
- Character Set Encoding Details
- File Manager URL
- Destination Version
- 8.5 Only Details
- PPM 9.3.1 Details

Each parameter is followed by an example displaying the correct format for each parameter value. You must complete the source and destination details for AUT to proceed with the upgrade.

RMW Only Details

Parameter	Value Definition
rmwUpgrade	Specifies whether you want to upgrade the RMW database. Default value is **false**. Set this value to **true** to upgrade the RMW database to the dest.version specified.

Whether Source Equals Destination

Parameter	Value Definition
sourceEqualsDest	Determines if the source database is upgraded to the destination database with or without the database user. If this value is set to **true** (upgrade destination in-place): The source database is upgraded without exporting and importing the database user to the destination database. The existing database user is upgraded instead of importing to a new database user and then upgrading. AUT does not perform an export unless a value for the src.db.backup parameter is specified. The source database details are not required. **Note** This does NOT apply if your source and destination databases are on different versions of the Oracle database.

Destination Details

Parameter	Value Definition
dest.jdbc.url	The JDBC connection string to the destination database. The format is **jdbc:oracle:thin@<DESTINATION_DB_HOST_NAME>:<PORT>:<SID>** where **DESTINATION_DB_HOST_NAME** is the host name of the destination database, **PORT** is the listener port of the destination database (1521 is the default), and **SID** is the database instance.

Parameter	Value Definition
dest.jdbc.driver	The database driver to use for JDBC connections to the destination database. The default is **oracle.jdbc.driver.OracleDriver**.
dest.tns.name	The TNS name for the destination database.
dest.oracle.home	The Oracle home of the destination database. This value can be a mapped drive on the system where the AUT is run which points to the shared folder on the host system. **Note** Use either double back slashes or a slash for directory separation, for example, **d:/oracle/ora102** or **D:\\oracle\\ora102**

Source Database Backup

Parameter	Value Definition
db.backup	Specifies whether a backup of the source database is required. Set this value to **true** to take a backup.

Sequence Project

Parameter	Value Definition
gap_threshold	Size of block of unused or orphaned sequence IDs than can be reused. The default minimum value is 5000.

Source Details

Parameter	Value definition
src.jdbc.url	The JDBC connection string to the source database. The format is **jdbc:oracle:thin@<SOURCE_DB_HOST_NAME>:<PORT>:<SID>** where **SOURCE_DB_HOST_NAME** is the host name of the source database, **PORT** is the listener port of the source database (1521 is the default), and **SID** is the database instance.
src.jdbc.driver	The database driver to use for JDBC connections to the source database. The default is oracle.jdbc.driver.OracleDriver.
src.tns.name	The TNS name for the source database.

Parameter	Value definition
src.oracle.home	The Oracle home of the source database. This value can be a mapped drive on the system where the AUT is run which points to the shared folder on the host system. **Note** Use either double back slashes or a slash for directory separation, for example, d:/oracle/ora102 or D:\\oracle\\ora102.

Note: If you are upgrading from Product Cost Management or Program Execution only, leave all Source detail properties blank in the *aut.properties* file.

Averify Details

Parameter	Value Definition
averify.error.count	Specifies the number of averify errors to occur before the AUT stops the upgrade. Specify **-1** to ignore the errors and continue with the upgrade. Specify **+1** to stop the upgrade if an error occurs. Once you fix the error, you can run AUT again to continue the upgrade.

Ignore Import Warnings

Parameter	Value Definition
ignore.imp.warnings	Decide if the AUT ignores the import warnings or not while importing the database. If set to true, the warnings are ignored. AUT will stop if there is a critical error.

Language Details

Parameter	Value Definition
i18n.lang	Specifies the language of the operating system where the database is located. The values are English, Japanese, Chinese, or French.

Character Set Encoding Details

Parameter	Value Definition
nls.lang	Specifies the character set encoding being used during database import and export. For example, AMERICAN_ AMERICA.AL32UTF8. *Note*: Customers upgrading from AA and 8.5 releases to Agile PLM should set this variable to 'American_America.AL32UTF8' when running AUT on a non-English client. This is necessary to ensure that the following database users - ifsuser, superadmin, and agileuser are successfully upgraded.

File Manager URL

Parameter	Value Definition
file.manager.url	Specifies the location of the file manager. The content_url column in the files table is upgraded with an encrypted URL based on the file type. This URL is used after upgrading for full text search configuration. The format is **http://<FILE_MANAGER_ HOST>:<PORT>/Filemgr/AttachmentServlet.**

Destination Version

Parameter	Value Definition
dest.version	The version of the upgraded Agile PLM database.

8.5 Only Details

Parameter	Value Definition
isPC isPSI isPCM isPE	Each of the parameters must have a value if the source database is either 8.5 or 8.5 SPx only. Set the parameter to true for the components that are installed on the source database and to false for the components that are not installed.

Parameter	Value Definition
username.migration.action	Determine how you want the Agile users in the source database to appear in the destination database. The value is set as the number of one of the following choices: 1. FIRSTNAME LASTNAME 2. LASTNAME FIRSTNAME 3. FIRSTNAME, LASTNAME 4. LASTNAME, FIRSTNAME 5. None of the above. FIRSTNAME will be migrated to LASTNAME. If your database contains mixed cases, choose the option that reflects the format for most of your users. The remaining users must be migrated manually after the upgrade.
gmt.timezone.difference	The GMT time zone of your database. The database and Agile application server must be on the same GMT time zone. The GMT time zone format is six characters, (+ or -)hh:mm, for example, GMT would be -00:00.
dest.users.timezone	The time zone to be used for the upgraded database. A text file listing the time zones that are supported is provided in the **<AUT_HOME/config** directory . Each time zone has a corresponding ID. Enter the ID that corresponds to the applicable time zone as the value for this parameter.
dest.users.encode.type	The encode type to be used for the upgraded database. All the supported encode types are specified in comments immediately above this parameter in the *aut.properties* file. Each encode type has a corresponding ID number. Enter the ID number that corresponds to the applicable encode type as the value for this parameter. For example, to set the default encode type to Unicode (UTF-8), set dest.users.encode.type= 5

PPM 9.3.1 Details

PPM 9.3.1 supports timestamps for Schedule, Estimated, and Actual Dates, allowing users to define tasks with durations in hours, minutes or even seconds.

To ensure that the correct durations are reflected in programs that were created with earlier releases, the following parameters need to be specified. These specifications should match related entries in the *agile.properties* file. If the default start and end times

specified in the *agile.properties* file do not match your entries here, change the entries manually.

Parameter	Value Definition
dest.ppm.start.time	Start time of the work day in 24-hr format. For example, 08:00.
dest.ppm.end.time	End time of the work day in 24-hr format. For example, 17:00.
dest.ppm.working.hours	Total working hours in a day, in 24-hr format. For example, 08:00.

Note: Although PPM supports timestamps with hours, minutes and seconds, the time span defined as working hours in the *agile.properties* file should be specified only in hours and minutes, not in seconds. For example, you can use the time span 08:30:00 to 17:30:00, but not 8:30:25 to 17:30:25.

Customers upgrading from a previous release to 9.3.1 may notice a discrepancy in the schedule dates of upgraded programs which have an Activity-Milestone-Activity construct with FS dependencies and time buffers, as described in the following scenario: Task 1 (1-day duration) has an FS dependency with Gate1, and Gate 1 has an FS dependency with Task 2 (1-day duration). Both dependencies have a 1-day time buffer also specified. During upgrade to 9.3.1, an extra day is added to the schedule. The duration for these tasks in 9.2.2.x will be 3 days, in 9.3.1, it will appear as 4 days. Upgrading customers who encounter this issue are requested to contact Oracle Support Services for assistance in fixing the data.

The psupgrade.properties File

The *psupgrade.properties* file, located in the $AUT_HOME\config directory, contains user-supplied information on the existing Product Cost Management database. During an upgrade of a Product Cost Management database, the data is merged into the Agile PLM database. The database scripts use the parameters in the file during the upgrade process to determine how the data is migrated. This file should be completed in addition to the *aut.properties* file if you are migrating a Product Cost Management database.

It is recommended to perform a test upgrade of the Product Cost Management database with reports to evaluate the database and make any necessary corrections before performing the actual data migration.

Defining the psupgrade.properties Parameters

Parameter	Value Definition
pcm.jdbc.url	The JDBC connection string to the Product Cost Management database. The format is **jdbc:oracle:thin@<SOURCE_DB_HOST_NAME>:<PORT>:<SID>** where **SOURCE_DB_HOST_NAME** is the host name of the Product Cost Management database, **PORT** is the listener port of the database (1521 is the default), and **SID** is the database instance.
pcm.jdbc.driver	The database driver of the Product Cost Management database. The default is oracle.jdbc.driver.OracleDriver.
pcm.tns.name	The TNS entry name in the Oracle home of the source database.
pcm.oracle.home	The Oracle home of the Product Cost Management 8.5 database. **Note** Use either double back slashes or a slash for directory separation, for example, d:/oracle/ora102.

Parameter	Value Definition
generate.pcm.reports	Set this parameter to true to generate a report about the data contained in the Product Cost Management database before migration. The report will display data matching the following rules: ■ Update firstname, lastname, and user organization for the matched user login ID, if you set the update.matched.user.data parameter to true. ■ Migrate the Product Cost Management Items not found in Agile 8.5, if you set the migrate.pcm.items.not.found.in.pc parameter to true. ■ Convert all manufacturer names to uppercase, if you set the convert.mfrName.to.upperCase parameter to true. ■ Update all Agile 8.5 manufacturer data with the Product Cost Management manufacturer data for the same manufacturer, if you set the update.matched.mfr.data parameter to true. ■ Migrate Product Cost Management manufacturer parts not found in Agile 8.5, if you set the migrate.pcm.mpns.not.found.in.pc to true. If you set this parameter to false, migration will proceed without a report being generated. If you choose to have a report generated, AUT will stop. After viewing the report and you are satisfied with the results, change this parameter to false and run AUT again.
report.folder.location	Location where you want the report to be generated.
update.matched.user.data	If set to true, the firstname, lastname, and user organization are updated based on matching users in the Agile 8.5 database.
migrate.pcm.items.not.found.in.pc	If set to true, all Product Cost Management items not found in the Agile 8.5 database are migrated.
convert.mfrName.to.upperCase	If set to true, all manufacturer names are converted to uppercase.
migrate.pcm.mpns.not.found.in.pc	If set to true, all Product Cost Management manufacturer parts not found in the Agile 8.5 database are migrated.

Parameter	Value Definition
update.matched.mfr.data	If set to true, all Agile 8.5 manufacturer data is updated with the Product Cost Management data for the same manufacturer.
expiry.date.to.remove.prices	Removes prices based on the entered expiration date. All prices before the entered date are removed. If no date is entered, the prices are checked against the current date. Enter a date in the following format: MM-DD-YYYY.
migrate.suppliers.with.autonumber	If set to true, all suppliers are given autonumbers. If set to false, you must provide a supplier code map file and add its location to the supplier.code.crossref.filepath parameter. The file should be in the form of comma separated text file with the heading in the first line as supplierName,supplierCode Agile,agil EMS1,emsCode1
supplier.code.crossref.filepath	Location of the supplier code map file.
iFS Details*	
copy.files	If set to true, the file attachments are copied.
src.ifs.folder	The mapped drive location of the local iFS files folder.
src.ifs.schema	The database user of the Product Cost Management database.
dest.ifs.folder	The path where the upgraded file vault will be located.
dest.ifs.schema	The database user of the destination database.

*After the database is upgraded, you will need to run the IFS Reorganization tool to restructure the file vault to match the current format. For more information on this tool, see the *Agile Product Lifecycle Management Application Installation Guide*.

The peupgrade.properties File

The *peupgrade.properties* file, located in the $AUT_HOME\config directory, contains user-supplied information on the existing Program Execution database. During an upgrade of a Program Execution database, the data is merged into the Agile PLM database. The database scripts use the parameters in the file during the upgrade process to determine how the data is migrated. This file should be completed in addition to the *aut.properties* file if you are migrating a Program Execution database.

It is recommended to perform a test upgrade of the Program Execution database with reports to evaluate the database and make any necessary corrections before performing the actual data migration.

Defining the peupgrade.properties Parameters

Parameter	Value Definition
pe.jdbc.url	The JDBC connection string to the Program Execution database. The format is **jdbc:oracle:thin@<SOURCE_DB_HOST_NAME>:<PORT>:<SID>** where **SOURCE_DB_HOST_NAME** is the host name of the Program Execution database, **PORT** is the listener port of the database (1521 is the default), and **SID** is the database instance.
pe.jdbc.driver	The database driver of the Program Execution database. The default is oracle.jdbc.driver.OracleDriver.
pe.tns.name	The TNS entry name in the Oracle home of the Program Execution database.
pe.oracle.home	The Oracle home of the Program Execution database. Use either double back slashes or a slash for directory separation, for example, d:/oracle/ora102.
create.del.subclass	If set to true, the Deliverables subclass will be created after upgrade for data currently in that subclass. If set to false (default, the data currently in the Deliverables subclass will be migrated to the Tasks subclass after upgrade.
generate.pe.reports	If set to true, a report on any circular dependencies in the data is generated and no data is migrated. If set to false, no report is generated and the data is migrated. If you choose to have a report generated, AUT will stop. After viewing the report and correcting the dependencies, change this parameter to false and run AUT again.
report.folder.location	Location where you want the report to be generated.
pe.weekend.days	Comma-separated numbers representing the weekend days in a week. The default days are Saturday and Sunday. Format: 1,7
currency.description	Base currency. The default is US Dollars (USD).
iFS Details*	
copy.files	If set to true, the file attachments are copied.

Parameter	Value Definition
src.ifs.folder	The mapped drive location of the local iFS files folder.
src.ifs.schema	The database user of the Program Execution database.
dest.ifs.folder	The path where the upgraded file vault will be located.
dest.ifs.schema	The database user of the destination database.

*After the database is upgraded, you will need to run the IFS Reorganization tool to restructure the file vault to match the current format. For more information on this tool, see the *Agile Product Lifecycle Management Application Installation Guide*.

4

Upgrading the Agile PLM Database

Before you upgrade your database, you should always make sure you are using the latest version of the AUT. Changes to scripts such as averify may occur after the initial version has been released.

If updates are needed to the AUT, you can always find the latest version on My Oracle Support. Download the AUT.zip file and extract the file contents into the same directory where you originally extracted the files.

To ensure your database is correctly upgraded, always check the My Oracle Support website for updates.

Running the Automated Upgrade Tool

Before running the AUT:

- Make sure you have completed all of the pre-requisites listed in "Preparing for the Upgrade" on page 2-1.
- Edit the property file with the correct values for your upgrade. See "Understanding the Property Files" on page 3-1 for more information.
- On UNIX, change the user permissions on all of the files under AUT_HOME in order for the upgrade log files to be created and the shell scripts under AUT_HOME/bin to be run.
- On AIX or systems with non-Sun JVMs, ensure that you remove the following parameters in the files mentioned below. You must do this before running Averify as well.

Remove this parameter	From this file
-hotspot -ms128m -mx640m	aut.sh located in <AUT>/bin
averify.sh located in <Averify>/bin	
-hotspot -ms64m -mx64m	encryptpwd.sh located in <AUT>/bin
encryptpwd.sh located in <Averify>/bin	

To run the AUT in interactive mode:

Go to the $AUT_HOME\bin directory and run the **aut.bat** (Windows) or **aut.sh** (UNIX) script.

You are prompted to enter the following usernames and passwords:

- Destination database user and password—user and password of the destination database.
- Destination database SYS user and password—name and password of the user account with SYS privileges in the destination database.
- Source database user and password—user and password of the source database.

To run the AUT in command line mode:

Usage: aut.bat/sh :<destination db user> <destination db user password> <sys user> <sys password> <source db user> <source db user password>

Checking the Status of the Upgrade

You can monitor the status through the log files that are generated during the upgrade. The **$AUT_HOME\logs** directory contains all of the upgrade-related log files. *AUT.log* is the main log file for the upgrade. Separate directories are created for the logs generated by averify and the database scripts. These log files are located in the **averifylogs** and **oracle** subdirectories.

What's Not Upgraded

Markup restrictions that were enforced through privileges are not retained when you upgrade. With the new security model implemented in 9.2.2.2, **Markup** is a subclass under the **File Folders** class and all markup privileges are automatically enabled. To prevent users from viewing or modifying the existing markups, you must *disable* blanket privileges at File Folders class level in Java Client and *enable* Markup privilege for relevant users only.

Troubleshooting the Upgrade

The following list describes some problems that can occur during the database upgrade:

What if I cannot resolve my TNS service name?

If the TNS service name cannot be resolved, then the service name specified in the *aut.properties* file is not correctly defined in the tnsnames.ora file.

Perform the following checks to fix the error:

- Verify that a tnsnames.ora file exists and is in the correct place. See your Oracle documentation for more information on the required name and location.
- Verify that the service name exists in one of the tnsnames.ora files. Add the service name, if necessary.
- Make sure there are no syntax errors in the tnsnames.ora file, especially unmatched parentheses or stray characters.

Why did I receive an Invalid User Name or Password error in the AUT.log file?

You may receive this error if the database connection was not successful. The following Oracle error messages may be displayed:

EXP-00004: invalid username or password

Cause: An invalid username or password was specified.

Action: Retry with a valid username and password.

ORA-01017: invalid username/password; logon denied

Cause: An invalid username or password was entered in an attempt to log on to Oracle. The username and password must be the same as was specified in a GRANT CONNECT statement. If the username and password are entered together, the format is username/password.

Action: Enter a valid username and password combination in the correct format.

What errors would I see if an import fails?

An import can end abnormally due to the following reasons:

IMP-00009: abnormal end of export file

Cause: The export file is probably from an aborted Export session.

Action: If so, retry the export and import.

IMP-00013: only a DBA can import a file exported by another DBA

Cause: The privileges needed to import an export file generated by a database administrator do not exist. Only a database administrator can import such files.

Action: The source and destination database users should have the same roles and privileges. Because the source database has the DBA role, the database destination should also have the role. If you do not want the destination database to have the DBA role, then remove the role from the source database and retry the upgrade.

IMP-00041: Warning: object created with compilation warnings

Cause: The object in the SQL statement following this error was created with compilation errors. If this error occurred for a view, perhaps the base table of the view was missing or altered.

Action: This is a warning. The object may have to be recompiled before being used.

Why is AUT hanging?

AUT may not be hanging. Processes may be running that seem to take longer than others. For example:

- Indexes may be missing on the source database. If this is the case, averify may be running slowly.
- The import may be taking a longer time because it's a large database.

What should I do if AUT suddenly exits?

Check the AUT.log file. If no information is available, then the AUT could not start because of invalid settings. Check the pre-requisites and the contents of the *aut.properties* file before retrying.

If I have database connection problems, what Oracle errors will I get?

Database connection errors can occur if the SQL connection to the source or destination database is lost. Ensure that the database connections are available before running the AUT. If connection problems occur, you may see the following Oracle error messages:

ORA-01034: ORACLE not available

Cause: Oracle was not started. Possible causes include the following:

- The SGA requires more space than that was allocated for it.

- The operating system variable pointing to the instance is improperly defined.

Action: Refer to accompanying messages for possible causes and correct the problem mentioned in the other messages. If Oracle has been initialized, then on some operating systems, verify that Oracle was linked correctly.

ORA-01089: immediate shutdown in progress - no operations are permitted

Cause: The SHUTDOWN IMMEDIATE command was used to shut down a running Oracle instance, terminating any active operations.

Action: Wait for the instance to be restarted or contact the database administrator.

ORA-01090: shutdown in progress - connection is not permitted

Cause: The SHUTDOWN command was used to shut down a running Oracle instance, disallowing any connects to Oracle.

Action: Wait for the instance to restart or contact the database administrator.

ORA-12541: TNS: no listener

Cause: The connection request could not be completed because the listener is not running.

Action: Ensure that the supplied destination address matches one of the addresses used by the listener - compare the tnsnames.ora entry with the appropriate listener.ora file (or tnsnav.ora if the connection is by way of an interchange). Start the listener on the remote system.

I received an ORA-00955 error in my AUT.log file. What should I do?

ORA-00955: name is already used by an existing object

Cause: An attempt was made to create a database object (such as a table, view, cluster, index, or synonym) that already exists. A user's database objects must have distinct names.

Action: Enter a unique name for the database object or modify or drop the existing object so it can be reused.

You can ignore this in the Agile schema because the object already exists. The upgrade process takes care of dropping the object and recreating it if there are any changes.

I received an ORA-00904 error. What should I do?

ORA-00904: MS_JAVA "."LONGNAME":Invalid Identifier.

Solution: Log in as the **sys** user and run the $ORACLE_HOME\javavm\install\initdbj.sql script.

I received an ORA-01555 error. What should I do?

ORA-01555: "snapshot too old (rollback segment too small" when using Automatic Undo Management (AUM).

Solution: The UNDO tablespace is too small. Increase the size of the UNDO tablespace. The UNDO tablespace should be large enough to store the undo data generated by active transactions and those preserved to honor the undo retention setting.

Increase the value of the Undo_retention parameter. This is important for systems running long queries. The parameter's value should at least be equal to the length of the longest running query on a given database instance. This can be determined by querying the V$UNDOSTAT view once the database has been running for a while:

```
SQL> select max (maxquerylen) from v$undostat;
```

How long does it take the AUT to run?

The amount of time it takes the AUT to run depends on the database size. The amount of time needed to upgrade is also based on the database version of the source database. Import and export steps can be avoided by using the sourceEqualsDest property which also reduces the time.

Data Migration for Signoff User Dual Identification

Agile provides optional data migration scripts that can be used by customers who choose to implement the Signoff User Dual Identification feature for approval signoffs. The Signoff User Dual Identification feature was introduced to address FDA regulations laid out in 21 CFR Part 11 Section 11.200. The system now facilitates the usage of two forms of identification from the user when signing off on a document such as a change order.

If the Login Password is to be used as a second form of identification, the following scripts automate the required administration procedures.

Details of these scripts and the actions they perform are provided in the following table:

#	Script	Action
1.	**UseLoginPwdYesList.bat**	Identifies the list of users who have set the "Use Login Password for Approval" option to **Yes** in Preferences and writes the output to a comma separated (.csv) file. **Note** The default file name for the output file is Output.csv. To change the file name, suffix a file name to the command as shown: **UseLoginPwdYesList <File Name>**. Where <File Name> is the name of the .csv file.
2.	**ChangeUseLoginPwdToNo.bat**	Resets the "Use Login Password for Approval" option to **No** for all users who have set it to **Yes** in Preferences.
3.	**LoginPwdAppPwdMatchList.bat**	Identifies the list of users who have identical Login and Approval passwords and have set the "Use Login Password for Approval" option to **No**. Writes the output to a .csv file.

#	Script	Action
4.	**SetApprovalPwd.bat**	Resets the approval passwords to an administrator-supplied value for users who have identical Login and Approval passwords and have set the "Use Login Password for Approval" option to **No**. To change the password, suffix the new password value to the command as shown: **SetApprovalPwd <password>**. Where <password> is the new value for the approval password. (The password is in cleartext.)

To migrate data to meet second signature requirements:

1. Locate the second signature migration scripts stored at the following path:

   <Agile installation folder>/ agileDomain/tools/ SecondSignature.zip

2. Unzip the **SecondSignature.zip** file and extract the scripts to a new folder (temp directory) in the same location.

3. Run each bat/shell file in the sequence listed in the table above.

> **Note:** For Solaris/Linux installations, the corresponding **.sh** files are provided. You may have to change the format of the script files from DOS to UNIX.

Once approval passwords have been reset, users should be asked to change the administrator-provided approval password to a password of their choice. (This should not be the same as the login password.)

Running Averify

Averify is a scripting tool that performs an integrity check against the PLM database and reports on specific errors. Averify includes a check for missing indexes and inconsistent data stored in the various database tables. Averify only reports errors, and does not attempt to fix them or modify your database. You should run Averify periodically as part of your regular database maintenance procedures.

Running Averify and using the results to eliminate known errors is a prerequisite for diagnosing problems. Averify is not comprehensive, however, and finding zero errors does not certify a database as perfect.

To run Averify:

1. Make sure you have downloaded the latest version of Averify from My Oracle Support (https://support.oracle.com).

2. Modify the properties in the averify.properties file, located in the Averify_Home\config directory.
3. Make sure you have backed up your database.
4. Clone your Production database to a Test environment. This can be done using export/import utilities, Recovery Manager, or copying backup database files from one file system to another and recreating the control file.
5. Shut down the application server if it is connected to the Test environment to prevent users from connecting to the database.
6. Go to the *Averify_Home\bin* directory and run the *averify.bat* script on Windows or *averify.sh* script on UNIX in interactive mode. You are prompted to enter the database user name and password.

   Or you can run the averify script in command line mode as follows:

   Usage: averify.bat/sh <db user> <db user password>

   Note: If you run Averify as a Cron job, add the following line to the *averify.sh* file after the # Set up the environment comment line:

   cd /export/home/oracle/averify/bin

7. Go to the $AUT_HOME\scripts\oracle\utilities\averify directory.
8. Connect to the Agile database from the command line using SQL*Plus.
9. Run the *oracle_averify9x.sql* script.

```
SQL> @oracle_averify9x.sql
```

Note: If errors are generated in the oracle_averify_report.log file located in the $AUT_HOME\scripts\oracle\utilities\averify directory, contact Agile Support.

Checking the Status of Averify

You can monitor the status through the log file that is generated while Averify is run. The *\logs* directory contains the *averify.log* file.

If you chose to have the log files sent as an email attachment, the *averify.zip* file, containing the *averify.log* and *oracle_averify_report.log*, is sent upon completion.

Notes on Agile Advantage (AA) Upgrade

Take note of the following when you upgrade to Agile PLM from Agile Advantage (AA) Releases:

- When you upgrade from any AA release or 8.x releases, obsolete privileges of the following types will not get migrated: Create PDX Package, FTS, Reports, Specify Output columns, and Agile Server Monitor.
- Organizations in AA will not be migrated as Suppliers in Agile 9.3, as AA has both Organization and Supplier objects. Organization Contacts will be migrated as inactive users without any Roles and Privileges associated. The Agile Administrator will need to assign Roles and Privileges manually.

- AA backup tables are not automatically removed during upgrade from AA. Customers must run the script CleanupBackuptables.sql available in **<AUT_Home>\Scripts\Oracle\Utilities\Script_tools** to clean up these tables.
- Full Text Search field properties are migrated as detailed in the table below. In Agile Java Client, the **Indexing** field value is set to 'Manual' by default. If you want it be 'Scheduled', the administrator must edit the **Recurrence** field properties appropriately.

AA Field Name	AA Field Value	A9 Field Name	A9 Field Value
Enable Stem Search	Yes	**Stem Searching**	Enabled
Enable Stem Search	No	**Stem Searching**	Disabled
Indexed File Types	Comma-separated list	**Index File Types**	Comma-separated list
Indexing	Manual	N/A	N/A

Notes on Variant Management Upgrade

In Agile PLM 9.3.1, the fields Relation Type and Configuration Graph were moved from the **Relationships** tab to the **Instances** tab. Customers who already have a Variant Management patch installed should take the following actions after upgrade to 9.3.4:

1. Call Oracle Consultant Services to migrate the user data from the Relationships tab to the Instances tab.
2. Once migration is successful, run the following SQL statements to remove the redundant data in the Relationships tab:

   --Drop legacy data from the Relationships table:

   ```
   Delete from RELATIONSHIP where relation_type = 1;
   ```

   --Drop relation_type column

   ```
   Alter table RELATIONSHIP drop COLUMN "RELATIoN_TYPE";
   ```

   --Drop configuration graph table:

   ```
   Alter table CONFIGURATION_GRAPH_DATA drop
   constraint CONFIGURATION_GRAPH_DATA_FK;
   Drop table CONFIGURATION_GRAPH_DATA;
   ```

   --Update dot on Relationships tab if it is empty now:

   ```
   update ITEM set flags=
   NVL(subStr(flags,1,29),'') || '0' ||
   NVL(subStr(flags,31),'')
   where id not in
   ( select distinct r.ctr_objid from relationship
   r ) and NVL(subStr(flags,30,1),'') != '0';
   ```

3. Next, run the following SQL statements to remove unnecessary metadata:

   --Delete Relation Type listname and corresponding entries:

   ```
   delete listentry where parentid=2000009331;
   delete listname where id=2000009331;
   ```

   --Delete Relation Type attribute:

```
delete from nodetable where id-2000009332;
delete from propertytable where parentid=2000009332;
DELETE FROM APPLIEDTO WHERE attid in (2000009332);
```

--Remove Relation Type attribute from part's Relationships table

```
delete from nodetable where id=2000009333;
delete from propertytable where parentid=2000009333;
```

--Delete configuration graph attribute

```
delete from nodetable where id=2000009334;
delete from propertytable where parentid=2000009334;
DELETE FROM APPLIEDTO WHERE attid in (2000009334);
DELETE FROM TABLEINFO WHERE att in (2000009332, 2000009334);
```

--Remove configuration graph attribute from the part's Relationships table

```
delete from nodetable where id=2000009467;
delete from propertytable where parentid=2000009467;
```

5

Configuring the Agile PLM Database

Reorganizing the Database

> **Important:** The Agile PLM database includes two user accounts, CTXSYS and AGILE. The following upgrade procedure is required for the database sizing and configuration of the CTXSYS and AGILE accounts.

After AUT is run, the upgraded schema and data have been validated. However, the data must be analyzed and statistics generated to enable Cost-Based optimizer (CBO) and Full Text Search (FTS) support. There are eight Agile-specific tablespaces required for optimization. The Agile schema has to be reorganized for this optimization to occur.

> **Note:** If you are using Oracle Datapump for import and export, use **agile9expdp** and **agile9impdp** scripts instead of **agile9exp and agile9imp** scripts in the steps outlined here.

To reorganize the upgraded schema:

1. On the database server, change to the following directory:

   (Windows) **\oracle\admin\<Oracle SID>\create\<agile schema user>**

   (UNIX) **$ORACLE_BASE/admin/$ORACLE_SID/create/agile**

2. Run the **agile9exp** script.

   The agile9exp.dmp file is created in the current folder. This dump file is used with the agile9imp script. Make sure the dump file can be imported successfully before proceeding to the next step.

   > **Note:** If you are using Datapump for export, this file will be called agile9expdp.dmp.

3. Run the **recreateagile** script to drop the existing account and recreate the agile account and schema.

4. Run the **agile9imp** script to import the upgraded Agile schema, including setup of CBO and FTS.
5. Rename the existing agile9exp.dmp file to agile9exp_upgrade935.dmp for backup. Also rename the agile9exp.log file to agile9exp_upgrade935.log.
6. Run the **agile9exp** script again.

   The Agile9exp.dmp file created can be used as a backup of the system after the Agile PLM 9.3.5 database reorganization.
7. Rename the agile9exp.dmp file to agile9exp_reorg935.dmp for backup. Also rename the agile9exp.log file to agile9exp_reorg935.log.
8. Start SQL*Plus from a command line and log in as **agile/tartan** or **agile/tartan@<Oracle SID>**.
9. Run the *agile9_check.sql* file to validate the schema integrity and confirm integrity of the database reorganization.

   ```
   SQL> @agile9_check.sql
   ```

   Note: Any warnings related to non-Agile tables can be ignored if you want to retain these tables in the Agile schema. If not, you can drop the non-Agile tables and run *agile9check.sql* again. If any other errors are displayed, contact Oracle Support Services http://www.oracle.com/agile/support.html. All errors must be fixed before proceeding to the next step.
10. Start the Oracle listener.

    Note: The following steps should only be performed after you have upgraded the Agile PLM application:
11. Start the Agile application server.
12. Shut down the database.
13. Perform a cold backup of all the database-related files.
14. Restart the database, listener, and application server.

Upgrading the Schema without Database Reorganization

The Agile PLM database installer generates a shell script *agile9postupgrade.bat/sh* that can be run after upgrading the Agile PLM schema with AUT. If you choose not to perform a database reorganization, then you must at least run the *agile9postupgrade* shell script.

The *agile9postupgrade* script isolates the SQL statements that compile invalid objects, so you can independently compile, validate, and gather statistics on the schema after an upgrade. The *agile9postupgrade* script references three SQL scripts:

SQL Scripts	Description
compile_invalid_objects.sql	Compiles invalid objects after running AUT

SQL Scripts	Description
agile9_check.sql	Validates the Agile PLM schema.
agile9stats.sql	Gathers statistics on the upgraded schema.

If there are any fragmentation or corruption issues in the indexes, you can also choose to perform a partial database reorganization using the following scripts:

agile9_index_recreate.sql	Drops and creates Normal or function-based Normal, non-unique, non-PK indexes.
agile9_ctx_recreate.sql	Drops and creates CTX indexes (except for FILES_CONTENT_IDX which is handled by *agile9_fts.sql).*

Updating Content URLs for File Attachments

Importing or reorganizing the database sets all values in FILES.CONTENT_URL to NULL which prevents existing attachment content from being searchable. To allow Full Text Search (FTS) on existing attachments, their Content URLs must be updated in order for the files to be indexed by Oracle Text and searchable within Agile PLM. For Agile PLM 9.3.2 and later databases, updating content URLs is accomplished within the Java Client. For more information, see the "Full Text Search" chapter of the *Agile PLM Administrator Guide*.

> **Note:** For Agile PLM 9.3.1.x and prior databases, content URLs are updated using AUT. Perform the following steps.

Before using the update_content_url.bat/sh script to update FILES.CONTENT_URL:

1. Set the following parameters in *aut.properties*.

   sourceEqualsDest = true

   file.manager.url = <http://<HOST>:<PORT>/Filemgr/AttachmentServlet>

   To obtain the **file.manager.url,** log on to Agile Java Client and navigate to **Admin > Server Settings > Locations > File Manager**. Double-click the Primary File Manager (iFS) entry and copy the URL from the **File Manager URL** field.

2. Open a command prompt on Windows or a terminal window on UNIX, and change to the **AUT/bin** directory:

3. If using AUT version 1.7.5 or later:

   On Windows, run the following batch file:

   .\update_content_url.bat

   On UNIX, execute the following shell script:

   ./update_content_url.sh

   > **Note:** You are prompted to enter the DB username and password.

4. If using an AUT version before 1.7.5:

On Windows, run the following batch file:

.\aut.bat execute-fts-post-tasks-921

On UNIX, execute the following shell script:

./aut.sh execute-fts-post-tasks-921

5. To synchronize the Oracle Text domain index associated with files (causes Oracle Text to begin downloading and parsing files from Agile PLM and storing their content), open a SQL*Plus session, connect as schema owner (AGILE), and issue the following command:

   call agile_server_fts.sync_index('files_content_idx');

   Note: This final step synchronizes the attachments domain index and requires that the application server and primary file manager are running. The synchronization process may take a long time to complete depending on the number and size of attachments being indexed. After the process completes, searches performed on attachment content should result in appropriate files being returned in the search results within Agile PLM.

Configuring the Agile SDK

If you are using the Agile SDK, delete the **AgileSDK.cache** directory from your client systems after you have upgraded the Agile PLM database and application server. The **AgileSDK.cache** directory can be found under the **temp** directory (%temp%AgileSDK.cache).

Configuring PG&C

If you have the PG&C component installed, additional configuration may be needed for substance migration.

Substance migration is necessary if you have met ALL of the following requirements:

- Upgraded your database from Agile PLM 9.2 to Agile PLM 9.3.5.
- Imported the JGPSSI substances and substance groups.

Note: In Agile PLM 9.3.5, you should use IPC declarations with IPC substances and substance groups. The IPC list of substances and substance groups is slightly different than the JGP list of substances and substance groups.

Important: Contact Oracle Support to obtain the files needed for substance migration.

If you imported the JGPSSI substances in Agile PLM 9.2, but do not want to use the IPC substances, you can continue to use the JGPSSI substances without migrating.

Configuring PPM

With PPM objects, there should always be at least one user in the team, such as Owner. For every user on the team, a corresponding Access Control List (ACL) in the Share should exist. In the 9.0 and 9.1 releases of Agile PLM, a logged in user was allowed to delete all users from Share which included the owner's ACL.

You must clean the corrupted data in the ObjectACL database table to access the shared objects. The data cleanup involves replacing the deleted role with a valid, non-deleted equivalent role.

The following SQL statement cleans all of the corrupted rows in ObjectACL table where a deleted role is assigned to a user with the valid role:

```
DELETE FROM TEAM WHERE ACTIVITY_ID<=0;
INSERT INTO OBJECTACL SELECT TEAM.activity_id,(SELECT CLASS FROM ACTIVITY WHERE
id=TEAM.activity_id),TEAM.user_id,',9506,',sysdate,sysdate FROM TEAM WHERE
TEAM.user_id NOT IN(SELECT USERID FROM OBJECTACL WHERE
OBJECTACL.objid=TEAM.activity_id AND OBJECTACL.objclass IN (18022,18387))
```

In the previous SQL statement, the Program Team Member role is generally assigned to a user and the ID for this out-of-box role is 9506.

The roleid values may differ if you have customized your database. In this case, use the following SQL statement to obtain roleid values for each role in your database:

```
select ID from nodetable where parentid=5006 and value='<name of role>';
```

The Agile Administrator is still allowed to delete a role assigned to a user in the object's ACL. If a role is deleted, an error message stating "Node <number> does not exist in cache." displays when the object is accessed.

Also, in earlier releases of Agile PLM, the Dashboard or other PPM-related home pages were allowed to be selected as the start page. In Agile PLM 9.2.2.1, a license check was added for default home pages. If any users have previously set their preferred Start Page and preferred Inbox to values that are related to PPM objects and PPM is no longer installed, run the following SQL statements to set the pages:

To set the Preferred Start Page to Home for all users:

```
update agileuser set PREFSTARTPAGE=1;
```

To set the Preferred Inbox View to Notifications if the Preferred Inbox View is Activities

```
update agileuser set PREFINBOXVIEW=3 where PREFINBOXVIEW=2;
```

Running the PPM Post-Upgrade Utility

After you have upgraded your database and installed the Agile PLM 9.3.5 application, you must run the PPM Post Upgrade Utility to accommodate the business rule changes introduced in this new version. For information on how to run this utility, see the *Agile Product Lifecycle Management Application Installation Guide*.

A

Upgrading from Agile Advantage 2006

You can upgrade an Agile Advantage 2006 file vault to Agile PLM with the Upgrade File Vault Utility.

Upgrade File Vault Utility for Agile Advantage

When upgrading from Agile Advantage 2006, the file vault structure must be reorganized. An Upgrade File Vault utility is provided for this purpose and can be found in the AGILE_HOME\AgileDomain\tools\ directory.

To upgrade the Agile Advantage file vault:

1. Unzip the UpgradeFVUtils.zip file to a temporary directory.
2. Backup the existing Agile Advantage file vault to a safe location.
3. Modify the following options and parameters in the config.properties file, located in the <UpgradeFVUtils_Home>\config directory:

Parameters	Description
SourceStoreDir	Agile Advantage 2006 file vault store directory The value of SourceStoreDir is located in the Agile Advantabe 2006 registry key {HKEY LOCAL MACHINE\SOFTWARE\Agile Advantage\AgileIFS\StorDirs}.
DestStoreDir	Agile PLM file vault base directory
FilePrefix	File prefixes of the Agile Advantage and Agile PLM files. The format is FilePrefix=AAschema1,934fileprefix 1; AAschema2,934fileprefix2 You can use this option to upgrade multiple file vaults. All existing files will be renamed using the PLM file name prefix. If the PLM file name prefix is omitted, the default value is the same as the Agile Advantage file prefix.

4. Run the UpgradeFVUtils command.

   Windows: UpgradeFVUtils.cmd **UNIX:** UpgradeFVUtils.sh

Parameters:

Parameter	Description
-help	Displays the readme file.
-upgrade [-configfile]	Upgrades the file vault with the named configuration file. If no configuration file is specified, the config.properties file located in the config directory is used. To specify a different configuration file, use the -configfile option. The option and path of the file must be enclosed in quotation marks; for example, "-configfile=c:\myconfig.properties"
-rename <*old file name prefix*> <*new file name prefix*>	Only used to rename the file name prefix in the Agile PLM file vault located in the DestStoreDir defined directory.

5. View the fvu.log file, located in the UpgradeFVUtils directory, for detailed information and error messages. The log file is overwritten each time the utility is run.

Post Upgrade Considerations

When upgrading from Agile Advantage 2006 to Agile PLM, existing objects must change to reflect the new structure. The following sections discuss how some existing Agile Advantage objects will appear in Agile PLM.

Supplier Objects

Supplier objects on Agile Advantage are migrated to Agile PLM with the following changes:

1. Suppliers are migrated as non-Web suppliers.

   Geography details are not migrated because of how the data is stored in the Agile PLM database. In Agile Advantage, this attribute points to a single geographic location. In Agile PLM, geographic details are obtained from a cascading list containing Continent, Country, State, and Region information.

2. Contact users on the Users tab of Agile Advantage are not migrated as contact users of Agile PLM.

   Users are migrated as normal users without any association to a specific supplier. The migrated users are disabled and have no assigned roles and privileges. You must enable the users and associate them to a supplier. In Agile Advantage, the same user can be added to multiple suppliers. In Agile PLM, a user can be added to only one supplier.

3. The Manufacturer Parts tab in a Supplier object is not supported in Agile PLM.

   In Agile PLM, there is a Manufacturers tab in Supplier. This tab displays the manufacturer line cards for each manufacturer part in Agile Advantage. The price information that was available in the Manufacturer Parts of a Supplier is migrated to Quote History objects with the material and non-material costs migrated as Page Two fields of the Quote History object. The prices are visible in the Prices tab

of the Manufacturer Part object. The Manufacturer Parts from the Supplier object are migrated to the Suppliers tab in the Manufacturer Part object.

Manufacturer Part Objects

The Manufacturer Part object is migrated with the following changes:

1. The status of a supplier on the Supplier tab in Agile Advantage is not migrated to the Supplier tab of the Manufacturer Part in Agile PLM.

   The pricing details on the Supplier tab are moved to Prices in Agile PLM as Quote History objects.

2. The standard cost and the target cost on the Cover Page of a Manufacturer Part object in Agile Advantage are moved to Page Two of a Supplier object in Agile PLM.

   Other price-related information, like Min Qty and Max Qty, are moved to the Cover Page tab of a Quote History object.

Item Objects

Price information is revisable in Agile Advantage, but not in Agile PLM. This information is moved to Page Two of the Item object.

Agile PLM does not support the summation functionality like Agile Advantage. Instead, you can run the Assembly Cost Report as part of Product Collaboration's report. If there is a missing price, Agile Advantage displayed the price with an asterisk (*), while the Assembly Cost Report in Agile PLM displays the price as 0.

Unlike Agile Advantage, there is no feature available out-of-box to set the target cost based on the Supplier. You can manually set the costs in the Page Two fields of Items and Manufacturer Parts or this can be automated through a Process Extension.

Compliance

In Agile Advantage, the compliance information on Manufacturer Part, Item, and Document objects is stored in 14 separate lists with the list label detailing the type of compliance this particular object meets. In Agile PLM, compliance information is a combination of Specification object and compliance status. During migration, the 14 lists on each object specified are moved as a Specification object with the specification name as the label of the enabled compliance list.

The migrated specification of Item and Document objects have a prefix of Item_ and Doc_ to differentiate between the specifications migrated from each object. The flex field information is moved to the Compliance tab, along with the status of each specification.

Unlike Agile Advantage, Agile PLM does not support the Audit Report functionality for calculating compliance. Instead, customers can use the BOM Compliance Report which displays similar compliance information. Likewise, Agile PLM does not support calculating compliance of an ECO or generating an audit report of an ECO in the out-of-box product. You could view the compliance of the parts in the ECO manually or this could be automated through a Process Extension.

Glossary

ACP

See **Agile Configuration Propagation (ACP)**

Activity

A project activity in Agile Product Portfolio Management, such as a program, task, or phase.

Affected Files

Similar to Affected Items, these objects are EC files that are Design Release Candidates.

Agile Configuration Propagation (ACP)

Propagating existing configuration the PLM to the newly installed version of PLM.

ACS

See **Agile Content Service (ACS)**

Agile Content Service (ACS)

ACS is an event-driven XML-based publishing service that makes the product record available to a wide variety of business applications and users, internally and across the global manufacturing network

Agile Destination

A package created by an Agile PLM system in the target PLM using Web Services to import from the Attachments tab of the package in the target system.

Agile Integration Services (AIS)

A collection of predefined Web Services in the Agile Integration Framework that enable communication between the Agile Application Server and disparate systems

Agile Product Portfolio Management

The Agile PLM project management solution that is integrated with the product information in PLM.

AI

Affected Items tab on Change objects in Agile.

AIS

See **Agile Integration Services (AIS)**.

Approved Manufacturer Parts List (AML)

List of approved manufacturer parts associated with an item.

AML

See **Approved Manufacturer Parts List (AML)**.

API

See **Application programming interface (API)**.

Application programming interface (API)

A set of routines, protocols, and tools for building software applications. An API expresses a software component in terms of its operations, inputs, outputs, and underlying types.

Assembly

A product assembly lists the parts in a product and shows the substances and materials that comprise those parts. It is linked to specifications that can restrict how much of a particular substance that product assembly may contain

Automated transfer orders (ATO)

Content published by Agile PLM users in real time with a content transfer order (CTO) or set up subscribers to automatically create automated transfer orders (ATO) based on a schedule or triggered by a workflow status change.

Baseline

A snapshot of a project, usually in its initial stage, used as a reference for future comparison in Agile Product Portfolio Management.

Bill of Material (BOM)

A hierarchical representation of a product that is made up of other products.

Bill of Substances (BOS)

A hierarchical list of substances that are contained in the parts and assemblies that make up a BOM.

BOM

See **Bill of Material (BOM)**.

BOS

See **Bill of Substances (BOS)**.

CAD

See **Computer-aided design (CAD)**

Commodity

A class of goods that is in demand, that is supplied without qualitative differentiation regardless of supplier.

Computer-aided design (CAD)

The use of computer systems to assist in the creation, modification, analysis, or optimization of a design. CAD software is used to increase the productivity of the designer, improve the quality of design, improve communications through

documentation, and to create a database for manufacturing. CAD output is often in the form of electronic files for print, machining, or other manufacturing operations.

Contract (Price)

This is a subclass of the Published Prices class. Contract prices are prices provided by the supplier for a specific item or manufacturer part. This price information applies only for the specified duration and can apply to any project.

Co-Sourcing

The process of leveraging product cost across suppliers.

DCO

See **Design Change Order (DCO)**

Deliverable

A unit of work required for a project's success, usually fulfilled by generating a digital file. (Word processing documents, spreadsheet documents, PDFs, presentation documents, and so on.) Deliverables can also be Agile PLM objects and processes. Also called 'content' in Agile Product Portfolio Management.

Design Change Order (DCO)

A Change Order subclass that is available when the effected File Tab is enabled and provides access to all Agile PLM Workflow functions.

Design File Folder

An EC file folder that is integrated with CAD and PLM files, providing full access to PLM Workflow function.

EC

See **Engineering Collaboration (EC)**

EC Client

A Java-based UI to access, administer and operated the EC solution.

ECO

See **Engineering Change Order (ECO)**

Engineering Change Order (ECO)

An object that carries with it all the proposed changes to a product and/or its BOM. When approved and implemented, the proposed changes become effective.

Engineering Collaboration (EC)

An application that provides data and process integration between CAD applications and Agile PLM. It allows CAD designers and engineers to capture and control the data representing a primary source of the product record.

Extensible Markup Language (XML)

A markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable

File Manager

The File Manager manages files in a repository or vault in the file system and provides a place to store and retrieve files locally or remotely. You can install it on the same

server as the Agile Application Server or on a separate one. You can also install the File in a redundant configuration and/or distributed across geographic regions.

File Transfer Protocol (FTP)

A standard network protocol used to transfer computer files from one host to another host over a TCP-based network, such as the Internet.

FQPN

See **Fully qualified path name (FQPN)**.

FTP

See **File Transfer Protocol (FTP)**

Fully qualified file name

The exact name of a file on a computer that is completely specified such that it is unambiguous and cannot be mistaken for any other file on that system.

Fully qualified path name (FQPN)

The full path of a resource, directory or file, stored in a computer. It is composed by the full path to the resource and its syntax depends on the operating system.

Gantt Chart

A project management tool that shows project activities and schedule as a bar chart. The chart lists project activities in sequence, and presents critical information such as the start and end dates of each activity, as well as interdependencies between activities.

Item Master

The product record. It is the entire collection of Items - Parts, Documents, and any other user-defined subclasses of the Items class maintained under change control in the Agile system.

Java Message Service (JMS)

The Java Message Service (JMS) API is a Java Message Oriented Middleware (MOM) API for sending messages between two or more clients.

JMS

See **Java Message Service (JMS)**.

Lifecycle Phase

Current state in an object's workflow.

LRR

Latest Released Rev - concerning a Part or Document.

NCNR

Non-Cancelable Non Returnable. Applies to an item. NCNR can be a Yes or No, depending on the supplier. You can ask for the NCNR information in the supplier response. This is one of the critical factors in finding the best deal among the supplier responses.

PDX

See **Product Definition eXchange (PDX)**.

PLM

See **Product Lifecycle Management (PLM)**.

Percent allocation or % allocation

The percentage of a resource's time allocated to a specific task or tasks in Agile Product Portfolio Management..

Percent complete or % complete

Amount of time and effort expended on a project measured as a percentage of the time and effort required to complete the whole project. Used in Agile Product Portfolio Management.

Product Definition eXchange (PDX)

A standard designed for the e-supply chain. This standard is based on the XML format because it provides a simple yet powerful and flexible way to encode structured data into a format that is both human- and computer-readable. In PLM, PDX packages contain product content, such as items.

Product Lifecycle Management (PLM)

The process of taking parts/documents from inception to production to phase-out, and all the stages in between.

Protocol

A system of digital rules or agreed-upon format for data exchange within or between devices. It determines the type of error checking and data compression used.

Published Price

This is a subclass of the Published Prices class. Published prices are prices provided by the suppliers in response to an RFQ and published from the project. The published price information can also be used in other projects.

PCO

See **Price Change Order**

Price

An object that carries with it all the proposed changes to a product and/or its BOM. It can be approved and implemented to make the proposed changes effective.

Price Change Order

It is an object that carries with it all the proposed changes to a price. It can be approved and implemented to make the proposed changes effective.

Quote history

A subclass of the Quote Histories class. Quote history prices are the stored prices from supplier responses that you can use. Any change in the response line of an RFQ is stored in the historical response and is usable at any time.

Request for Information (RFI)

A material declaration that lists the parts in a product assembly and shows the substances and materials contained in the part.

Request for Quote (RFQ)

A standard business process whose purpose is to invite suppliers into a bidding process to bid on specific products or services.

Request for Proposal (RFP)

A solicitation, often made through a bidding process, by an agency or company interested in procuring a commodity, service or valuable asset, to potential suppliers.

Response Line

A response line has information about only one item. The negotiation of price and terms for items is dealt with in a response line.

Resource Pool

A group of users who can be bulk assigned as resources for a particular project or task in Agile Product Portfolio Management.

RFI

See**Request for Information (RFI)**.

RFP

See **Request for Proposal (RFP)**

RFQ

See **Request for Quote (RFQ)**

RFQ Response

A medium of communication between the user and the supplier. One response from a supplier can contain multiple response lines for different items. Price data is added to the project automatically when the supplier submits the response.

Schedule Editor

The scheduling engine that handles updates to the project schedule in Agile Product Portfolio Management.

Schema

In computer programming, a schema is the organization or structure for a database. The activity of data modeling leads to a schema.

SDK

See **Software Development Kit (SDK or "devkit")**

Software Development Kit (SDK or "devkit")

A set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform.

Sourcing Project

The entry point of sourcing and product pricing. A sourcing project tracks data required for sourcing and pricing, to perform data analysis for effective pricing.

Standard Cost

Applies to an item. This is the market cost of the item. It is site-specific. The standard cost is for a unit.

Supplier

A supplier of one or several commodities.

Target Cost

Applies to item. This is the expected cost of the item by you or the supplier. This can be a percentage of the standard cost. Target cost is for a unit.

Timesheet

The time entry system in Agile Product Portfolio Management, used to track actual hours spent by resources on project activities and to calculate corresponding labor cost.

TLA

See **Top Level Assembly (TLA)**

Top Level Assembly (TLA)

The level in a BOM that indicates the ultimate product being manufactured.

Transfer order

Every time Agile Content Service (ACS) publishes product content, it produces a transfer order that keeps track of what, where, and when product content is transferred.

UPK

See **User Productivity Kit (UPK)**

User Productivity Kit (UPK)

The Oracle online help system used in some Oracle products.

Web Service Extensions (WSX)

A Web service engine that enables communication between Agile Product Lifecycle Management system and disparate internal and external systems.

WSX

See **Web Service Extensions (WSX)**.

XML

See **Extensible Markup Language (XML)**.

XML Schema

Description of a type of XML document, typically expressed in terms of constraints on the structure and content of documents of that type, above and beyond the basic syntactical constraints imposed by XML rules.