



Bulk Migrator Installation and Usage Guide

11.1 M020

Copyright © 2018 PTC Inc. and/or Its Subsidiary Companies. All Rights Reserved.

User and training guides and related documentation from PTC Inc. and its subsidiary companies (collectively "PTC") are subject to the copyright laws of the United States and other countries and are provided under a license agreement that restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed software user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the PTC copyright notice and any other proprietary notice provided by PTC. Training materials may not be copied without the express written consent of PTC. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of PTC and no authorization is granted to make copies for such purposes. Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from PTC.

UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.

PTC regards software piracy as the crime it is, and we view offenders accordingly. We do not tolerate the piracy of PTC software products, and we pursue (both civilly and criminally) those who do so using all legal means available, including public and private surveillance resources. As part of these efforts, PTC uses data monitoring and scouring technologies to obtain and transmit data on users of illegal copies of our software. This data collection is not performed on users of legally licensed software from PTC and its authorized distributors. If you are using an illegal copy of our software and do not consent to the collection and transmission of such data (including to the United States), cease using the illegal version, and contact PTC to obtain a legally licensed copy.

Important Copyright, Trademark, Patent, and Licensing Information: See the About Box, or copyright notice, of your PTC software.

UNITED STATES GOVERNMENT RIGHTS

PTC software products and software documentation are "commercial items" as that term is defined at 48 C.F.R. 2.101. Pursuant to Federal Acquisition Regulation (FAR) 12.212 (a)-(b) (Computer Software) (MAY 2014) for civilian agencies or the Defense Federal Acquisition Regulation Supplement (DFARS) at 227.7202-1(a) (Policy) and 227.7202-3 (a) (Rights in commercial computer software or commercial computer software documentation) (FEB 2014) for the Department of Defense, PTC software products and software documentation are provided to the U.S. Government under the PTC commercial license agreement. Use, duplication or disclosure by the U.S. Government is subject solely to the terms and conditions set forth in the applicable PTC software license agreement.

Contents

About This Guide	6
Related Documentation.....	7
Technical Support.....	7
Documentation for PTC Products.....	7
Comments	8
Introducing the Windchill Bulk Migrator	9
Windchill Bulk Migrator Utilities	10
Windchill Bulk Migrator Checklists.....	13
Installing the Windchill Bulk Migrator.....	21
Installation Overview.....	22
Media Content.....	23
Modifications to Windchill System Files	23
Configuring the Loading Infrastructure and Object Loaders	24
Database Settings	26
Using the wbmgm Command.....	27
Windchill Bulk Migrator Object Model	30
Installing the Staging Schema.....	30
Installing the Audit Tables on the Target Windchill System.....	32
Installing Windchill Bulk Migrator Extraction Utilities	32
Uninstalling and Upgrading the Windchill Bulk Migrator	37
Upgrading the Windchill Bulk Migrator	38
Uninstalling Loaders	38
Uninstalling Extraction Utilities	44
Terminating an Installation	45
Windchill-to-Windchill Data Mapping	46
Windchill Bulk Migrator Mapping Definitions.....	47
The classMap.csv File.....	54
Configuring the Windchill Bulk Migrator	55
Configuring the Windchill Bulk Migrator Loading Infrastructure	56
Configuring Extraction Utilities	66
Deferred Constraint Enforcements	69
Extracting Data	70
Windchill Bulk Migrator Extraction Utilities Overview.....	71
Running the Windchill Bulk Migrator Extraction Utilities.....	71
Data Not Extracted	81
Windchill Bulk Migrator Extraction Utilities Status Reports.....	81
One-to-Many Extraction	83

Extraction Sequence	84
Filter Options	85
Installation	86
Starting the Filtered Extraction GUI	86
Filtered Extraction GUI Overview	86
Windchill Bulk Migrator Enhanced Filtered Extraction Utilities Overview	90
Running the Windchill Bulk Migrator Filtered Extraction Utilities	91
Enhanced Filtering Commands	91
Extraction Mode	92
Filtered Extraction Report	92
Options and Variants	93
Normalizing Data	95
Normalizing Data Overview	96
Using the wbldr Command	97
Normalization Sequence	109
Pre-loading Validation	110
Pre-loading Validation Process	111
The Windchill Bulk Migrator Pre-Load Verification Utilities	111
Loading Data	129
Loading Data Overview	130
Loading Sequence	136
Post-loading	137
Post-loading Verification Overview	138
Generating Post Load Verification Reports	138
Execution of WinRU during post migration	141
Migrating Non-Windchill Data into Windchill	142
Non-Windchill Data Migration Overview	143
Extraction from Non-Windchill Systems	144
Understanding the Windchill Bulk Migrator Staging Schema	145
Populating the MigrationSourceSite Table	148
Loading Data to De-Normalized (Cache) Staging Tables	150
Unsupported Use Cases	154
Loading CAD Data from File System Directories	155
Overview	156
CAD Migration	157
Migration Checklist	165
Supported Hardware and Software Notes	165
Installing the CAD Metadata Extractor	166
CAD Data Extraction Sequence	167
Configuring the CAD Metadata Extractor	170
Data Preparation	172
Parallel Execution of CAD Metadata Extractor	187
Custom Mapper Interface	190
Custom System Attributes	193

CAD Metadata Extraction Log Files.....	194
CAD Metadata Extractor Settings File Properties	194
Using Incremental Migration	198
Migration Failure Resolution Options.....	202
 Workflow Migration.....	206
Overview.....	207
Workflow Migration Scope	207
Workflow Migration Sequence	208
Windchill Bulk Migrator Mapping Client.....	210
Attribute Definition Analyzer Client	210
Updating the WBMWFProcessMapping.xml File	210
Validating the Workflow Template Mapping: Pre-Load Validation	214
Extracting, Staging, and Loading All Other Objects to be Migrated Using the Windchill Bulk Migrator	214
Extracting Workflow Data into the Staging Database Using the Windchill Bulk Migrator.....	215
Validating the Workflow Data Prior to Loading	217
Loading Workflows into the Target Windchill System	217
Publishing PBO Data From the RMD to the Public Domains.....	218
Initializing Workflow Tasks for Running Workflow Instances	219
Workflow Migration Staging Tables.....	220
Supported Workflow Process Variables for Migration.....	221
Windchill Bulk Migrator Utilities Performance Considerations.....	223
Extraction Utilities	224
Loader Infrastructure.....	225
Customizing the Windchill Bulk Migrator.....	227
Understanding the classMap.csv File	228
Appendix A.Additional Information on Select Use Cases	234
File Content Migration	235
NavigationCriteria - Selective Class Loading.....	256
Incremental Migration	257
Installing the Staging Schema on Cross-Platform Configurations	261
EPM Data Loaders	268
Enterprise Systems Integration Data Loaders	273
Classification Attribute Configuration.....	275
Representation Migration	279
Change Object Migration.....	282
Flexible Change Item Delegate Configuration	284
Managing Life Cycle Services and Life Cycle History During Loading	285
Migrating into a Live Production System	287
Understanding the Staging Database Table Attributes	294
Security Labels.....	295
Supplier Management Data Loaders	296
Appendix B.Supported Object Types.....	298

1

About This Guide

Related Documentation	7
Technical Support.....	7
Documentation for PTC Products	7
Comments	8

This guide provides the instructions to install, configure, and use Windchill Bulk Migrator (loader, extractor, and WBM FF) in order to migrate data to Windchill.

Before installation, be sure you have the most up-to-date version of this manual. It will be posted on the PTC Web site:

<https://www.ptc.com/en/support/refdoc>

Related Documentation

The following documentation may be helpful:

- *Windchill Installation and Configuration Guide*

If books are not installed on your system, see your system administrator.

Technical Support

Contact PTC Technical Support via the PTC Web site, phone, fax, or e-mail if you encounter problems using the Windchill Bulk Migrator or the product documentation.

For complete details, refer to Contacting Technical Support in the PTC Customer Service Guide. This guide can be found under the Self Help section of the PTC Web site at:

<https://www.ptc.com/en/support/>

The PTC Web site also provides a search facility for technical documentation of particular interest. To access this page, use the following URL:

<https://www.ptc.com/en/support/refdoc>

You must have a Service Contract Number (SCN) before you can receive technical support. If you do not have an SCN, contact PTC Maintenance Department using the instructions found in your *PTC Customer Support Guide* under Contacting Your Maintenance Support Representative.

Documentation for PTC Products

You can access PTC documentation using the following resources:

- **Windchill Help Page**—The Windchill Help Center is an online knowledgebase that includes a universal index of all Windchill documentation; you can access it by clicking a help icon or the Help link in any Windchill page header. You can browse the entire Windchill documentation set, or use the advanced search capability to customize your keyword search.
- **Reference Documents Web Site**—All books are available from the Reference Documents link of the PTC Web site at the following URL:

<https://www.ptc.com/en/support/refdoc>

A Service Contract Number (SCN) is required to access the PTC documentation from the Reference Documents Web site. For more information on SCNs, see Technical Support:

<https://www.ptc.com/en/support/>

Comments

PTC welcomes your suggestions and comments on its documentation. Send comments to the online survey form at the following address:

documentation@ptc.com

Please include the name of the application and its release number with your comments. For online books, provide the book title.

2

Introducing the Windchill Bulk Migrator

Windchill Bulk Migrator Utilities	10
Windchill Bulk Migrator Checklists	13

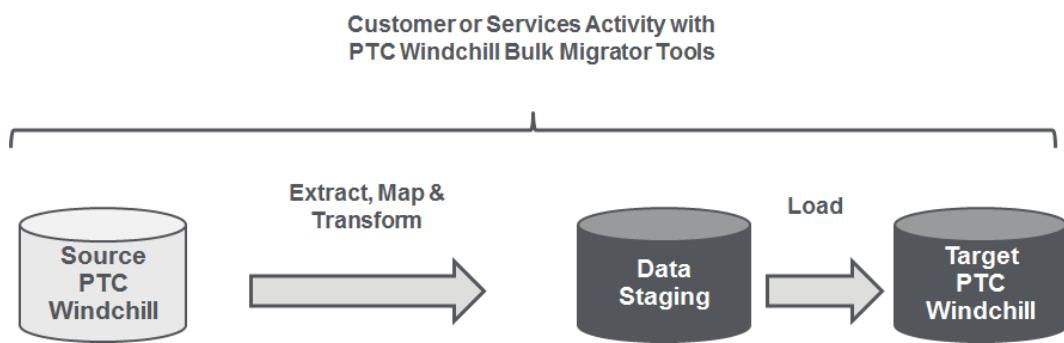
This chapter provides an overview of the Windchill Bulk Migrator.

Windchill Bulk Migrator Utilities

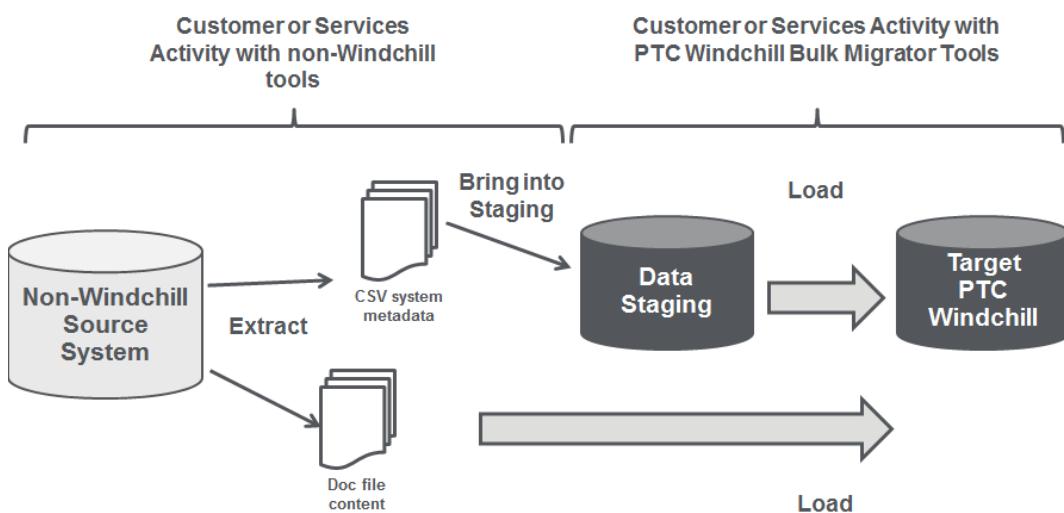
The Windchill Bulk Migrator is a collection of tools that are meant to facilitate the permanent, one-way migration of data into Windchill. The tools work within an overall Extract, Transform, Load and Validate (ETLV) migration approach whereby data is pulled from one or more source systems, mapped and transformed to conform to the Windchill application and then loaded into the Windchill database and file vaults.

Use the Windchill Bulk Migrator to extract data from Windchill source systems, and/or load the standardized data from the staging database into the target Windchill system. After migration, you can retire or archive the source systems that originally held the data. The following diagram provides an overview of the process:

Windchill-to-Windchill Migration Example



Non-Windchill System to PTC Windchill Migration Example



Note

Non-Windchill legacy systems require separate extraction tools to extract source system data to a format usable by the Windchill Bulk Migrator. The development, implementation and testing of extractors for non-Windchill systems is left to either Customer or professional services migration teams.

The tools are not meant to support two-way integration, federation, synching, or replicating of data between two or more separate systems or related use cases such as the movement of data packages back and forth from one system to another. Rather, the intent of bulk migration is to migrate specific data sets into Windchill.

Example use cases include the following:

- Legacy system replacement.

A legacy system is a system that has been outdated or may be no longer supported. The obsolete system is a candidate to be replaced by a separate system.

- System consolidation.

System consolidation includes combining or merging two or more separate source systems into one target system.

- Splitting off or diverging one system into more than one system.

This example can be illustrated by a corporation selling or spinning off a business unit and wanting to set up a separate Windchill instance containing only that business unit's data.

In general, ETLV migrations can be used in the following situations:

- When a business wants to select specific data sets to move or split off.
- When the source system is very different from the target system (implying that data mapping and transformation is needed).
- When data from two or more separate sources needs to be combined.

The source system and data can be varietal. The source systems can be from legacy PTC systems, non-PTC systems, or home grown applications.

The Windchill Bulk Migrator includes the following components which support some aspects of ETLV migrations:

- Staging database schema definition and installation infrastructure.

The staging database is where object metadata is stored prior to being loaded into Windchill. The Windchill Bulk Migrator staging schema is the neutral data format for mapping data to the target Windchill system. For details on what is provided, see [Installing the Staging Schema on Cross-Platform Configurations on page 261](#).

- Data loading architecture.

The utilities include mechanisms to control, route, and dispatch specific load jobs to the Windchill method server. For details on how to configure and use the utilities, see [Wbm.properties Configurations Overview on page 56](#).

- Windchill object type loaders.

Object loaders are called by the data loading framework to create the actual object data in the target Windchill system. The object type loaders use Windchill APIs to create objects in the target system. For a list of out-of-the-box object type loaders, see [Supported Object Types on page 298](#).

- Data extraction infrastructure for Windchill as a source system.

The utilities include mechanisms to query for source system data model and attribute information. The infrastructure controls, routes and dispatches extraction jobs to the Windchill method server. It also accepts and reads map file definitions to write data directly to the staging tables. For details on how to configure and use the utilities, see [Extracting Data on page 70](#).

- Windchill object type extractors

Object extractors are called by the extraction framework to query for and pull data from the source system. The classes in the `classMap.csv` file are supported by extraction tools.

- Windchill mapping definitions

Mapping definitions are created when a worm generation command is executed. The options to source or obtain definitions from are Source Windchill, or Non-Windchill, or Third Party PLM systems. The definitions are created as a `classmap.csv` file. For more information on `classMap.<Windchill version>.csv` and `classMap.csv`, see [Installing Extraction Utilities on page 32](#).

The definitions are extendable for new or custom objects. In addition to object and attribute mappings, the tools also provide administrative data mapping definitions to enable users to map such object types as folders, users and life cycles.

- Data normalization and verification infrastructure

Within the staging environment, the tools provide automated mechanisms to transform metadata from a de-normalized format into a normalized format that aligns closely to the target Windchill schema. The normalization process ensures that data is in a format that is ready to be loaded, as well as enables automated verification of data referential integrity.

- Oracle bulk loader (SQLLDR) control files.

The SQLLDR control files are used to load the raw metadata extracted from a non-Windchill source system into the staging database. A control file is provided for each object type supported by the migration utilities used to load data into the target Windchill system. Out-of-the-box control files are located in the following directory:

```
<Windchill>/db/[sql|sql3]/wbm/stage/control
```

Windchill Bulk Migrator Checklists

Windchill Bulk Migrator offers the capability to load data into a target Windchill system from a pre-populated staging area. Therefore, this document assumes that the following activities have been completed prior to using the migration tools:

- Target Windchill system design, installation, and configuration.

The target system must be available to the migration tools prior to performing the migration. All target system information required to define and locate the migrated data such as object types (and subtypes), object attributes, contexts, folders, and users must be configured and available prior to the migration.

- Data mapping

The data that is to be loaded into the target Windchill system must conform to the staging area schema definition. For non-Windchill source data, this means that feature and object solution design and mapping must occur to define how data should be created in the target Windchill system.

For Windchill source systems, out-of-the-box mappings are provided for supported object types and attributes. Any new or custom mappings that are required for a specific project should be added to the map files before migration. In addition, the administrative data mappings must be updated and configured before migration occurs.

- Staging schema configuration.

Most migration use cases will require updates to be made to the staging tables to add object attribute information.

- Source system data extraction (for non-Windchill source systems).

The data that is to be loaded into the target Windchill system must be extracted from the source system and loaded into the staging database. For non-Windchill source systems you must choose the technique to extract the data. Examples of techniques include using SQL queries or Java programs. Extraction of the metadata must be in the form of a delimited file that can then be loaded to the staging database using the Oracle or SQLServer bulk loader.

- Resolve any conflicting objects. For this, execute the WinDU task **Report All Organization Owned Namespace Objects in External Organizations** to detect conflicting objects before you begin with WBM process. A report is generated which lists the organization-owned conflicting objects such as products, libraries, projects, organizations, or site. The report contains links to the individual report for each object which can be used to resolve the conflicting objects. The conflicting objects are listed for site and organization level uniqueness. The conflicts must be resolved either by renumbering the objects or by changing the object's organization manually. To renumber a conflicting object, update the column 'WTKEY' of Objects-Key and 'Object Number' of the Objects-Master table to prevent the conflicting objects to occur in the Windchill system. If SQL queries are required to update the conflicting object, contact the customer support team.

After you have populated the data in the schema defined by Windchill Bulk Migrator, data can be loaded by Windchill Bulk Migrator.

Installing Windchill Checklist

Windchill Bulk Migrator is installed as an optional module during Windchill installation or when using Windchill PSI. Therefore, the source and target Windchill systems must be available and accessible by the installer tools. Before installing Windchill:

- Read the PTC software support matrices documentation for Windchill and third party software.
- Read the PTC software support matrices documentation for the Windchill Bulk Migrator and Windchill.
- Install Windchill and third party software according to the *Windchill Installation and Configuration Guide*.

Configuring Windchill Checklist

The Windchill Bulk Migrator creates product data in the target Windchill system. The process assumes administrative data and configurations have been made so that the data to be migrated can be mapped and loaded into the target system after Windchill is installed. The following list provides examples of configurations that should be made prior to migration, according to the intended system design.

- Create the Windchill container and folder structure.
- Create Windchill users.
- Create the data object and subtype definitions.
- Create object attributes.
- Create life cycles and workflows.
- Create access control policies.
- Create Windchill file vaults and vaulting rules.
- Validate the system design and business use cases using test data.

Installing Windchill Bulk Migrator Loaders Checklist

The Windchill Bulk Migrator loaders and loading infrastructure are installed on the target Windchill application load point. Complete the following steps on the target Windchill system before configuring and using the Windchill Bulk Migrator:

1. Read the PTC software support matrices documentation for the Windchill Bulk Migrator and Windchill and verify that your migration path is supported.
2. Download the Windchill Bulk Migrator media content.
3. Obtain the intended target Windchill host name, staging database instance URL, the staging database user and password, and the database type (Oracle or SQLServer). This information will be used as input for the lightweight installer in the next step.
4. Install Windchill Bulk Migrator using Windchill PSI as an optional module.
5. Verify the installation by running the Windchill version command.
6. Proceed to the installation of the staging schema.

For more information about these steps, see [Configuring the Loading Infrastructure and Object Loaders on page 24](#).

Installing Staging Schema Checklist

The Windchill Bulk Migrator uses its own staging schema as a neutral data format for Windchill object types. In order to stage and load data, it is necessary to create the Windchill Bulk Migrator staging schema.

1. Create a staging database user.
2. Create the staging database schema tables using the Windchill Bulk Migrator creation scripts.
3. Populate the MigrationSourceSite table.

Note

For migrations from non-Windchill systems into Windchill only.

4. Proceed to create target Windchill Audit tables.

For more information about these steps, see [Configuring the Loading Infrastructure and Object Loaders on page 24](#).

Installing Audit Tables on the Target Windchill System Checklist

As the Windchill Bulk Migrator loads data into Windchill it registers the migration events with Windchill in database Audit tables. These Audit tables must be created as part of the installation process.

Connect to the Windchill database and run the appropriate schema creation scripts for either Oracle or SQLServer.

Installing Windchill Bulk Migrator Extractor Checklist

The Windchill Bulk Migrator extraction utilities are installed on the source Windchill system. The extractor infrastructure extracts data directly into the Windchill Bulk Migrator staging schema and is intended to only work with that schema.

1. Read the PTC software support matrices documentation for the Windchill Bulk Migrator and Windchill and verify that your migration path is supported.
2. Download the Windchill Bulk Migrator extractor media content.
3. Obtain the intended Windchill Bulk Migrator staging database instance URL,

the staging database user and password, and your user-defined working directory that will contain supplemental files that the extractors will use. This information will be used as input for the installer in the next step.

4. Install the Windchill Bulk Migrator extractor by running the installer.

For more information about these steps, see [Installing Extraction Utilities on page 32](#).

Data Mapping Checklist

Data mapping allows you to map administrative data on the source system to administrative data on the target system. Often this mapping is desired in order to change the location or data behavior in the target system. In order to facilitate the mapping process, the extraction tools provide a method to collect source system configurations and output editable mapping xml files. Complete the following steps to map data for Windchill-to-Windchill migrations using the Windchill Bulk Migrator extractor.

1. Define how source system administrative data should map to the target system.
2. Generate the default mapping files by running the `com.ptc.windchill.migration.wbm.wc2wc.mapping.client.WBMMappingClient` command and update the following files:
 - `WBMWTUserMapping.xml`
 - `WBMWTViewMapping.xml`
 - `WBMTTeamTemplateMapping.xml`
 - `WBMSiteMapping.xml`
 - `WBMTTypeDefMapping.xml`
 - `WBMLifeCycleMapping.xml`
 - `WBMOrganizationMapping.xml`
 - `WBMClassificationMapping.xml`
 - `WBMWfProcessMapping.xml`
 - `WBMWTFormulaSetMapping.xml`

For more information, see [Windchill Bulk Migrator Mapping Definitions on page 47](#).

Configuring Windchill Bulk Migrator Checklist

All properties used by the Windchill Bulk Migrator are stored in the `wbm.properties` file. Use the `xconfmanager` utility to set and propagate property values for the `wbm.properties` file.

Review the various properties and options for the Windchill Bulk Migrator and use the `xconfmanager` utility to set `wbm.properties` values as appropriate for your migration. Properties that you may want to set include (but are not limited to):

- Supported object types
- Locale
- Content migration type
- Restricted migration domain settings
- Log file settings

For more information, see [Configuring the Windchill Bulk Migrator on page 55](#).

Extracting Data Checklist

1. Copy the corresponding `classMap.csv` file from the target Windchill loader installation to source Windchill extractor installation.
2. Generate the data dictionary files from the source system.
3. Generate and update the mapping files.
4. Run the extractor command for each Windchill object class you want to extract.
5. Review log and output reports.

For more information, see [Extracting Data on page 70](#).

Normalizing Data Checklist

Once data is extracted and staged in the staging database, the next step in the migration process normalizes the data to transform it into a state that it is ready to be loaded. Normalization includes data integrity checks and places the metadata into internal tables that align with the target Windchill system.

1. Run the `wbmldr` command (-s option) to normalize the data.
2. Verify that there are no conflicts reported. If there are conflicts, resolve them before continuing to load data.

For more information on data normalization and using the `wbmldr` command, see [Normalizing Data Overview on page 96](#).

Pre-load Validation Checklist

The Windchill Bulk Migrator provides pre-load validation tools to ensure that the data to be loaded is clean and conforms with the target system's administrative configurations. The pre-load validation tools are launched through a command line and can be used through a user interface.

1. Verify that data normalization is complete.
2. Launch the pre-load validation user interface.
3. Run the pre-load validation checks and resolve any reported data conflicts.

For more information, see [Pre-loading Validation Process on page 111](#)

Loading Data Checklist

Before running the Windchill Bulk Migrator, determine the correct order of loading for the object classes. In general, versioned objects should be loaded first and then links.

1. Run the family table pre-verifier command (if migrating CAD family table data).
2. Start the DataLoadRouter.

Note

You can choose to place the target Windchill system into Read Only mode by running the ROAPController command. However, this can only be done for migrations into an active production system, not for offline migrations.

3. Run the loading command for each object class to create the objects and object metadata in the target Windchill system.
4. Run the ContentMetaDataLoader command to load content metadata into the target Windchill system. If using the enhanced content migration option, you must also create the ContentMap information and manually complete the content migration of the data into the target system file vaults.
5. Review the log reports and verify that there are no errors.

Note

At this point, you can choose to take the target Windchill system out of Read Only mode by running the ROAPController command.

6. Stop the DataLoadRouter.

For more information, see [Loading Data on page 129](#).

Post-load Checklist

The following is a checklist for post-loading verification:

1. Generate post-load verification reports.
2. Publish data from the Restricted Migration Domain to the Public Domains (if using the Restricted Migration Domain).
3. Run WinDU, and perform the diagnostic tasks for Family Table Pre-Verification.
4. Validate the data by performing and validating end user business cases.

For more information, see [Post-loading Verification Overview on page 138](#).

3

Installing the Windchill Bulk Migrator

Installation Overview	22
Media Content	23
Modifications to Windchill System Files	23
Configuring the Loading Infrastructure and Object Loaders.....	24
Database Settings	26
Using the wbmgen Command	27
Windchill Bulk Migrator Object Model	30
Installing the Staging Schema	30
Installing the Audit Tables on the Target Windchill System	32
Installing Windchill Bulk Migrator Extraction Utilities	32

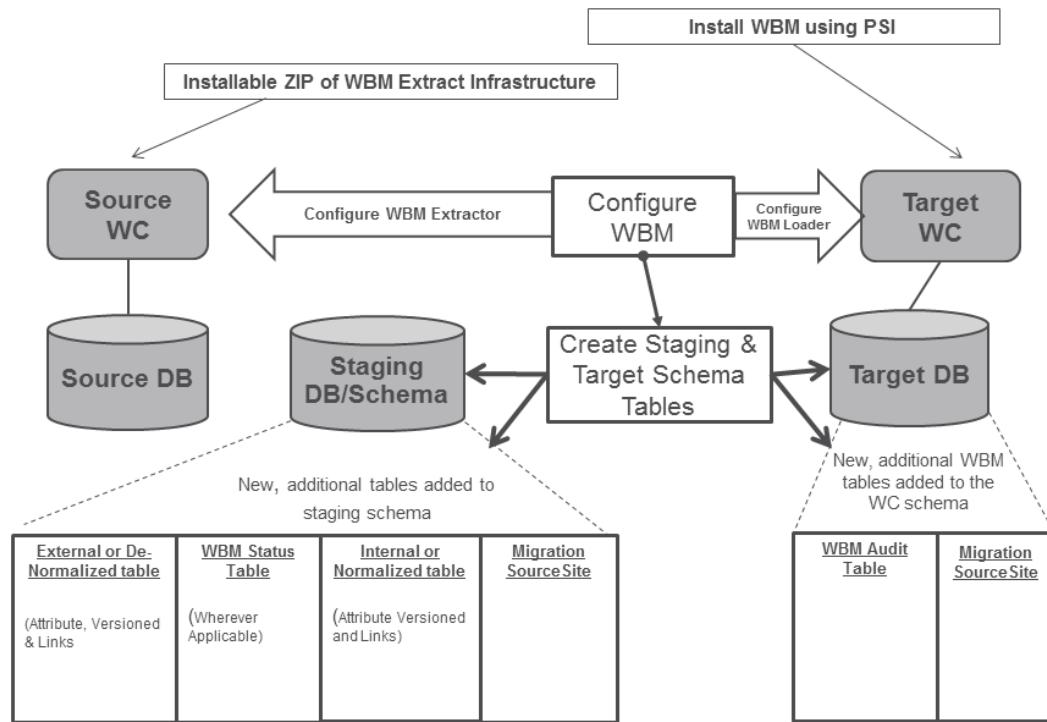
Note

Before continuing, make sure that you have completed the required actions listed in the [Installing Loaders Checklist on page 15](#) and the [Installing Staging Schema Checklist on page 16](#).

This chapter describes the Windchill Bulk Migrator file contents, the Windchill Bulk Migrator installation steps, and schema generation activities that are required before configuring and using the Windchill Bulk Migrator.

Installation Overview

The following image shows a graphical representation of the installation process:



Complete the following general steps before configuring and using the Windchill Bulk Migrator:

1. Install Windchill Bulk Migrator. See [Configuring the Loading Infrastructure and Object Loaders on page 24](#).
2. Install the Windchill Bulk Migrator Extraction Utilities. See [Installing Extraction Utilities on page 32](#).
3. Generate the staging schema. See [Installing the Staging Schema on Cross-Platform Configurations on page 261](#).

Note

Once you uninstall Windchill Bulk Migrator and if you are able to see latest Windchill Bulk Migrator Build under Display Label as already installed components. Delete temp files if Windchill Bulk Migrator has been already installed using previous version from the <USER HOME>\.windchill\site\.xconf.properties location.

Delete temp file : %temp%

Delete files from : Administrator\AppData\Roaming\PTC\Windchill

Media Content

WBM Extraction Utilities

Refer to the Windchill Bulk Migrator Product Compatibility Matrices document for information on supported Windchill releases for extraction utilities. The document is available at <http://support.ptc.com/WCMS/files/173805/en/WindchillBulkMigratorProductCompatibility.pdf>

You can download the file containing the media content from the PTC Software Downloads page. Start by accessing the Technical Support page using the following URL:

<http://www.ptc.com/support/index.htm>

From the support page, you can navigate to the page where you can download the Windchill Bulk Migrator. For example:

1. Click the **Download Center** tab.
2. Click **Order or Download Software Updates**.
3. Enter your customer number and click **Next**.
4. Select **Windchill Bulk Migrator**.

Modifications to Windchill System Files

The installation of the Windchill Bulk Migrator adds tool components and loader source code into the Windchill codebase. Properties that are used are maintained in a separate XCONF file.

Note

Property values set for the Windchill Bulk Migrator are propagated using the xconfmanager utility.

Windchill Bulk Migrator modifies the following properties:

File	Modification
site.xconf	Adds properties for the Windchill Bulk Migrator values that are entered as input on a command line when running Windchill Bulk Migrator.
declarations.xconf	Adds cross reference to wbm.xconf.

Configuring the Loading Infrastructure and Object Loaders

The Windchill Bulk Migrator loading infrastructure and object loaders are installed by PSI and configured to use on the target Windchill application. To complete the configuration of the Windchill Bulk Migrator on the target system, complete the following steps:

Target System

Pre-requisites

1. Target Windchill system installed and available
2. Windchill Bulk Migrator should be installed on target system using PSI
3. Staging database created and available
4. Documented list of source and target type and attribute mappings

Configure Windchill Bulk Migrator by running ant script as follows:

1. Run ANT installer from a Windchill shell:
 - On Windows, enter the following:
`ant -f bin\configure_WBM.xml`
 - On UNIX, enter the following:
`ant -f bin/configure_WBM.xml`
2. Restart Windchill
3. From a Windchill shell, run the `wbmgen` command.
 - On Windows, enter the following:
`wbmgen`
 - On UNIX, enter the following:
`./wbmgen.sh`

Loading Infrastructure and Object Loader –ant Script Installation Example

When run, the `-ant` script prompts you for information regarding the target Windchill system and the staging system.

For this example, assume the following:

- Staging database instance URL:
 - Oracle: Oracle Hostname:Port:Oracle SID (i5311.ptcnet.ptc.com:1521: migora)

- SQL Server: SQL Server Hostname\instance:Port:staging db username
(i5792-vm2.ptcnet.ptc.com\sql2008r2:1433:staging db)
fully qualified host name, port, and Oracle SID supplied in the database
instance URL is MIG11.ptcnet.ptc.com:1521:wind.
- Staging database user is staging and corresponding password is staging.

 **Note**

The password that you enter is not displayed in the output and is encrypted
when saved.

The following output shows the installation prompts and input values entered
(formatted to fit on the page):

Oracle

```
[input] Enter valid staging database instance URL {e.g. i4605.ptcnet.ptc.com:  
1521:SID}  
host:1521:wind  
  
[input] Enter valid staging database user : staging  
  
Enter valid staging database password :  
staging password  
  
Re-enter valid staging database password :  
staging password  
  
[input] Enter staging database type(Oracle|SQLServer) : Oracle  
  
[echo]  
  
[echo] Select the modules as explained below to include from the source system:  
  
[echo] Option 1: [Foundation] Basic PDM/PJL functionality  
[echo] Option 2: [Optional] Foundation and other optional modules supported by WBM  
[echo] Option 3: [All] Foundation and all WBM-supported optional modules.  
This is the default input, if you do not enter any value  
  
[echo] System generation is based on the module available on the target system  
  
[input] Enter the valid source system module option (Foundation, Optional, All)  
  
If Optional is selected, enter the optional modules in the next step.
```

```
[Input] Enter the optional modules, separated by a comma.  
Select from [ESI, MPMLINK, SUMA, SIMSP]
```

SQL Server

```
[input] Enter valid staging database instance URL  
{e.g. i4527.ptcnet.ptc.com\\sql2008r2:1433: stg}  
host.ptcnet.ptc.com\\instance:1433: stg  
[input] Enter valid staging database user :  
stg  
Enter valid staging database password :  
staging db user  
Re-enter valid staging database password :  
staging password  
[input] Enter staging database type(Oracle|SQLServer) :  
staging password  
SQLServer  
  
[echo]  
  
[echo] Select the modules as explained below to include from the source system:  
  
[echo] option 1: [Foundation] Basic PDM/PJL functionality  
[echo] option 2: [Optional] Foundation and other optional modules supported by WBM  
[echo] option 3: [All] Foundation and all WBM-supported optional modules.  
This is the default input, if you do not enter any value  
  
[echo] System generation is based on the module available on the target system.  
  
[input] Enter the valid source system module option (Foundation, Optional, All)  
  
If Optional is selected, enter the optional modules in the next step.  
  
[Input] Enter the optional modules, separated by a comma.  
Select from [ESI, MPMLINK, SUMA, SIMSP]
```

Database Settings

The property values that you must set for your staging database are unique to your environment. It is recommended to use different DB instances to host staging and Windchill schema as a best practice. Use the following property formats to update the set of properties for your site:

Oracle DB

```
com.ptc.windchill.migration.wbm.load.Staging_DB_Instance=<host>  
\:<port>\:<oracle_sid>
```

SQL Server DB

```
com.ptc.windchill.migration.wbm.load.Staging_DB_Instance=<host>
\\<Instance>\:<port>\:<staging DB>
com.ptc.windchill.migration.wbm.load.Staging_DB_User=<user_name>
com.ptc.windchill.migration.wbm.load.Staging_DB_PW=<password>
```

where:

<host> is the fully qualified host name of the server where the staging database is running.

<port> is the port number of the port used by the database.

<oracle_sid> is the Oracle SID set up for the database.

<user_name> is the staging database user typically created by a site database administrator for loading the data.

<password> is password of the staging database user.

<Instance> is instance name for SQL server.

<Staging DB> is staging database user.

Although you enter the password of the staging database user as plain text when you set property value using the xconfmanager, the Windchill Bulk Migrator encrypts the password before saving the property. The following is the property that is stored in wbm.properties:

```
com.ptc.windchill.migration.wbm.load.Staging_DB_PW=
encrypted.com.ptc.windchill.migration.wbm.load.Staging_DB_PW
```

Using the wbmgen Command

After installing the Windchill Bulk Migrator, complete the following steps:

1. Run wbmgen.bat from a Windchill Shell. This will create worm XML files for all WBM-supported worm classes and all necessary SQL scripts with migration option list.

Note

For Unix, execute wbmgen.sh from a Windchill shell in <WT_HOME>\bin.

2. Select the appropriate option for wbmgen from the table below:

Option	Description
1. Start with a new source Windchill database	<p>Select this option for Windchill-to-Windchill migrations. Provide the Source Windchill information as follows:</p> <ol style="list-style-type: none"> Enter a value for Database Type: [sql, sql3, sqlServer] : <Database type>. Type sql for Single byte source data, sql3 for multi-byte or Unicode Data, or sqlServer for SQLServer source database. Enter a value for Database Connection Properties: [<host>:<port>:<sid> for Oracle source database, or <host>\<instance-optional if default>:<port optional if 1433> for SQL Server database.] Enter a value for Database User: <Windchill DB_User>. Enter a value for Database Password: <Password for Windchill DB_User>. The password will be encrypted and not visible. Enter a value for Database JDBC Protocol: <tcp or tcps> If you enter tcps, you will need to enter the JDBC Name list and Value list as follows: Enter a value for Database JDBC Name List: <value of wt.pom.dbConnectionPropertiesNameList>. Refer to the Windchill db.properties. Enter a value for Database JDBC Value List: <value of wt.pom.dbConnectionPropertiesValueList>. Refer to the Windchill db.properties. <p>Selecting this option will map source and target modules and available data in source Windchill to generate the WORM XML files, WORM Model, SQL files, Views, and the</p>

Option	Description
	ClassMap.csv . It will optimize the worm model generation. Hence, it will only create the WORM XML files, WORM Model, SQL files, Views, and ClassMap.csv for possible Hybrid Link combinations.
2. Generate WBM staging schema for a non Windchill source system	<p>For non-Windchill data migration, select this option. This will eliminate analyzing source database and only read the target Windchill database to generate XML files, WORM Model, SQL files, and Views.</p> <p> Note</p> <p>wbmgen -g All is used to create WORM for both SQL Server and Oracle data base, this is recommended when caching the CSV files using BCP command for SQL Server</p>
3. Exit	Select this option to exit from wbmgen and do a fresh re-run.

Note

Loading CAD Data from File System Directories is considered as Windchill-to-Windchill Migration.

For such migration, provide target Windchill system information as source Windchill system information for wbmgen command.

Based on the database connection, there can be following scenarios:

1. When Regular JDBC connection over TCP is used.
2. When Secure JDBC connection is used.

Both these cases will get distinguished by value of the protocol that is being used. So, you will need to provide a new input against - Database JDBC Protocol prompted in console by wbmgen operation. Make sure you enter `tcp` when using the Regular JDBC connection over TCP and `tcps` for Secure JDBC connection.

In the case of a Windchill-to-Windchill migration, if any of the source and Target has been updated or in case user has to migrate from second Windchill source to same target Windchill system (site consolidation), re-execute wbmgen in target Windchill shell to re-calculate and generate the updated WORM models , XML file, SQL files and `classMap.csv` again.

Windchill Bulk Migrator Object Model

Along with wbmgen Windchill creates .html files. It can be accessed through Windchill HTTP server. This .html pages displays the following information:

- **Object Model**- It will list all the existing objects in Windchill Bulk Migrator. You can select **Worm** view or the **Class** view. All generic loadable and legacy loadable objects will be displayed as per the Windchill installation.
- **Hybrid Checkbox**-It will display only the hybrid objects.
- **SQL data**-It will list the create table list for Oracle and SQL server as per the installation selection of staging.
- **Data and Reference fields**-It will display the data and reference fields of the objects. It also displays the details regarding the fields.
- **Search Reference by Roles**-In case of hybrid objects there will be multiple objects references in Windchill. The **Search Reference by Roles** gives you an option to provide role A and role B objects and Windchill displays the relevant combinations.

Installing the Staging Schema

The Windchill Bulk Migrator staging schema represents a neutral format for Windchill object types. The schema consists of various object and attribute tables as well as internal tables used to track the migration status, pre-load validation data, and transformed, normalized data that represents the state of the data that is ready to be sent to the loaders.

It is recommended that the staging schema be created on a different, separate database for performance issues. The database user should be unique for migration activities.

When you install the Windchill Bulk Migrator, and run the `wbmgen` command, database SQL scripts are created within the Windchill codebase (*<Windchill>*). These scripts are used to create the staging schema and update the target Windchill schema to include migration-specific tables.

Separate sets of SQL scripts located in *<Windchill>/db/[sql|sql3|sqlserver]/wbm/stage* support the following use cases:

- Migrations involving single byte English or non-English locale data sets in the staging database and the target Windchill system running on Oracle (sql directory).
- Migrations involving single and multi-byte English locale data sets in the staging database and the target Windchill system running on SQL Server (sqlserver directory).
- Migrations involving multi-byte English or non-English locale data sets in the staging database and the target Windchill system running on Oracle (sql3 directory).

It is possible to have different combinations of the database type and operating system for the staging area and the source and target Windchill applications. For information about creating the staging schema and target Windchill schema, see [Additional Information on Select Use Cases on page 234](#).

Staging System

Create the Staging Schema

1. Create a staging database user
2. Create staging database schema object tables

Oracle: <Windchill>/db/[sql or sql3]/wbm/create_staging_schema.sql

SQLServer: <Windchill>/db//sqlServer/wbm/make_staging.bat (Windows)
or <Windchill>/db//sqlServer/wbm/make_staging.csh (UNIX)

For more information, see [Installing the Staging Schema on Cross-Platform Configurations on page 261](#).

3. If you are performing a non-Windchill or legacy migration, you must populate the MigrationSourceSite table. For Windchill-to-Windchill migrations, this table is automatically populated by the Windchill Bulk Migrator.

Single and Multi-Byte Character Sets and Locale Support for Migration

The Windchill Bulk Migrator supports the migration of source data sets that contain either single or multi-byte characters. The following migrations are supported:

Note

If the staging database has been created using multi-byte SQL3 scripts, you must use UTF-16 or Unicode for CSV files.

- Single byte staging to single byte Windchill.
- Multi-byte staging to multi-byte Windchill for supported locale values.

For example, a source data set containing Korean characters migrating to a Windchill system with the Korean language pack installed and the locale set to English, would be supported.

It does not matter that the target Windchill locale is set to English as long as the Windchill system has been installed and configured with multi-byte character support enabled.

Installing the Audit Tables on the Target Windchill System

The Windchill Bulk Migrator uses new tables in the target Windchill schema to audit and track data that is created by the tools. These new audit tables must be created on the target Windchill system as part of the installation process. Complete the following steps to create the target system audit tables:

Target System

- Creating Schema Objects in target Windchill

1. Create Audit tables:

Oracle: <Windchill>/db/[sql or sql3]/wbm/create_audit_schema.sql

SQLServer: <Windchill>/db/sqlServer/wbm/make_staging.bat (Windows)
or <Windchill>/db/sqlServer/wbm/make_staging.csh (UNIX)

For more information, see [Installing the Staging Schema on Cross-Platform Configurations](#) on page 261.

Note

To customize tablespace names, use the /t option in wbmgen. For more information about the option, see the help file generated with wbmgen -h.

Installing Windchill Bulk Migrator Extraction Utilities

Installation Tasks

Note

The Windchill Bulk Migrator extraction utilities are installed on the source Windchill system; whereas, the Windchill Bulk Migrator loading infrastructure and object loaders are installed on the target Windchill system.

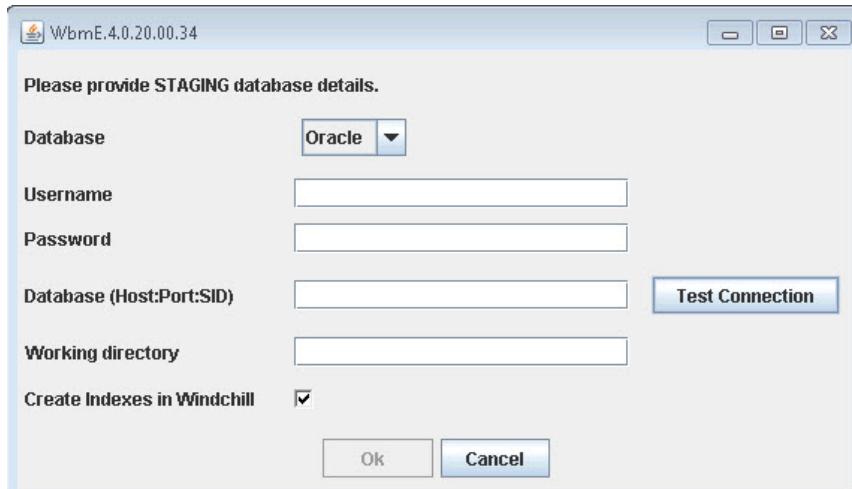
Complete the following steps to install the Windchill Bulk Migrator extraction utilities:

- Source System

1. Download the file containing the installation media from the PTC web site. If you are not sure which file you are to download, see [Media Content on page 23](#).
2. Unzip the contents of the file into a directory on the server where the source Windchill system is installed.
3. In source Windchill shell, change the current directory to the directory where the Windchill Bulk Migrator extraction file is unzipped.
4. From the shell, enter the following command to stop Windchill:

```
windchill stop
```
5. From the shell, run the following command:

```
ant install
```
6. Enter the appropriate information on the following screen:



Input	Description
Database	Specify whether the staging database is Oracle or SQL Server.
Username	The staging database user name.
Password	The staging database password.
Database (Host:Port:SID)	Enter the database information in the format Host:Port:SID for Oracle and HostInstance:Port:Stagingdb for SQL Server. Clicking Test Connection checks whether the staging database can be successfully connected or not. You will not be able to proceed with the installation if connection to the staging database fails. See Loading

Input	Description
	Data Overview on page 130 for more information on creating the staging database and initializing the schema.
Working Directory	The root location for the data dictionary files, administrative data mapping files and report files that WBM extraction generates. The directory is represented as <WBMExtractWorkingDir> in the rest of the documentation.
Create indexes in Windchill	<p>If this option is selected, the installer creates an index on the Windchill schema to speed up the queries to extract data from the source.</p> <p>Currently, there is only one non-unique index that the installer creates in the Cabinet table, column name personalcabinet. The name of the index is WBM_CABINET_1.</p> <p>If this index exists, the query execution time for extracting EPMDocuments, WTDocuments, and a few other Windchill objects can be significantly reduced.</p> <p> Note</p> <p>If the extractor is uninstalled, the index is dropped.</p>

7. Click **OK**.
8. The `classMap.csv` file provides object class and modelled property mappings from a specific source Windchill version to the target Windchill version. Depending on the Windchill system you are extracting data from, copy one of the mapping files listed below to the source system. For example, if you are extracting from a Windchill 10.2 system, you would copy the `classMap.10.2.csv` file.

The following `classMap.<Windchill version>.csv` files are located on the target system after running `wbmgen` following the Windchill Bulk Migrator installation on the target system:

- `<WC_Home>\Windchill\loadFiles\wbm\extractor\classMap.10.2.csv`
- `<WC_Home>\Windchill\loadFiles\wbm\extractor\classMap.11.0.csv`
- `<WC_Home>\Windchill\loadFiles\wbm\extractor\classMap.11.1.csv`

Once you have copied the correct file, place the file in the following directory on the source system and rename the file to `classMap.csv`:

```
<Source_WC_Home>/codebase/com/ptc/windchill/  
migration/wbm/wc2wc/extractor/classMap.csv
```

9. Restart Windchill so the properties changes and `classMap.csv` can take place.

Upon successful installation, the following things have happened:

- All `WBMEtraction` class files are inserted under `<Windchill>/codebase`. All extraction classes reside under `<Windchill>/codebase/com/ptc/windchill/migration/wbm/wc2wc`.
- `<Windchill>/codebase/com/ptc/windchill/migration/wbm/wc2wc/extractor/extract.xconf` is updated with the installation inputs.
- `<Windchill>/codebase/com/ptc/windchill/migration/wbm/wc2wc/extractor/extract.xconf` is added to the Windchill declarations.xconf.
- `<Windchill>/codebase/com/ptc/windchill/migration/wbm/wc2wc/extractor/extract.xconf` is propagated to `<Windchill>/codebase/com/ptc/windchill/migration`

wbm/wc2wc/extractor/extract.properties and <Windchill>/codebase/wt.properties, respectively, using xconfmanager.

- The extraction utilities release information is in <Windchill>/codebase/com/ptc/windchill/migration/wbm/wc2wc/extractor/buildcode.txt.

Note

Before modifying the files (declarations.xconf, wt.properties and extract.xconf), a backup of the original files is taken and the backup can be found in <Windchill>\codebase\com\ptc\windchill\migration\wbm\wc2wc\extractor\wbmextractor\install.

4

Uninstalling and Upgrading the Windchill Bulk Migrator

Upgrading the Windchill Bulk Migrator.....	38
Uninstalling Loaders	38
Uninstalling Extraction Utilities.....	44
Terminating an Installation.....	45

Upgrading the Windchill Bulk Migrator

Note

An in-place upgrade of the Windchill Bulk Migrator from a previous release is not supported.

To implement a more recent version of the Windchill Bulk Migrator, you must perform a complete uninstallation of your existing Windchill Bulk Migrator, and a complete re-installation of the more recent version. Uninstallation includes removing the staging schema as well as Windchill Bulk Migrator Codebase.

The following are examples of unsupported upgrade paths:

- Windchill Bulk Migrator 2.2 M010 to Windchill Bulk Migrator 2.2 M020
- Windchill Bulk Migrator 2.2 M020 to Windchill Bulk Migrator 4.0 F000
- Windchill Bulk Migrator 4.0 F000 to Windchill Bulk Migrator 4.0 M010

Uninstalling Loaders

This section describes how to manually uninstall the Windchill Bulk Migrator.

First, verify that your Windchill server and Windchill Bulk Migrator server are stopped.

In the following steps, use a text editor to manually remove content from the Windchill files that are identified and save all changes. In the file paths listed, *<Windchill>* is the directory path of your Windchill installation:

1. Execute `Windchill\db\sql\wbm\WbmCore\Drop_module_WbmCore.sql`, and log in as a target Windchill database user. This step is not applicable for target Windchill with SQLServer database.
2. Delete the following files and directories:

Path	File or Directory	Ignore if Not Available
<code>Windchill\wbm</code>	Directory	
<code>Windchill\lib\wbm.jar</code>	File	Yes
<code>Windchill\srclib\wbm</code>	Directory	
<code>Windchill\installer\wbm</code>	Directory	
<code>Windchill\installer\</code>	File	

Path	File or Directory	Ignore if Not Available
wbmconfigured.success		
Windchill\bin\configure_WBM.xml	File	
Windchill\codebase\wbm	Directory	
Windchill\codebase\com\ptc\wbm	Directory	
Windchill\codebase\com\ptc\windchill\migration\wbm	Directory	
Windchill\loadFiles\wbm	Directory	
Windchill\loadXMLFiles\wbm	Directory	
Windchill\bin\wbmgen.bat	File	
<p> Note This is specific to Windows.</p>		
Windchill\bin\wbmgen.sh	File	
<p> Note This is specific to Unix.</p>		
Windchill\bin\wbmldr.bat	File	
<p> Note This is specific to Windows.</p>		
Windchill\bin\	File	

Path	File or Directory	Ignore if Not Available
wbmldr.sh		
 Note This is specific to Unix.		
Windchill\db\sql\com\ptc\windchill\migration\wbm	Directory	
Windchill\db\sql\wbm	Directory	
Windchill\db\sql3\com\ptc\windchill\migration\wbm	Directory	
Windchill\db\sql3\wbm	Directory	
Windchill\db\sqlServer\com\ptc\windchill\migration\wbm	Directory	
Windchill\db\sqlServer\wbm	Directory	
Windchill\utilities\wbm	Directory	

3. Back up and modify the following files:

Path	Instructions	Ignore if Not Available
Windchill\site.xconf	Remove any and all properties related to WBM where: <code>'targetFile= "codebase/com/ptc/windchill/migration/wbm/wbm.properties"'</code>	Yes
Windchill\declaration.s.xconf	Remove <code>'<Configuration Refxlink:href= "codebase/wbm/</code>	

Path	Instructions	Ignore if Not Available
	WbmCore/ wbm.xconf"/>'	
Windchill\ codebase\ modelRegistry. properties	Remove any and all properties related to WBM where: 'com.ptc.wind chill.migra tion.wbm.loa ders.wc.epm='	
Windchill\ codebase\ descendentRegis try.properties	Remove any and all properties related to WBM where: <ul style="list-style-type: none"> • 'wt.fc.Ab- stractDatastor eArray= com.ptc.wind chill.migra tion.wbm.loa ders.wc.epm.' • 'wt.fc.Data- storeArray= com.ptc.wind chill.migra tion.wbm.loa ders.wc.epm.' • wt.fc.Ab- stractDatastor eStruct= com.ptc.wind chill.migra tion.wbm.loa ders.wc.epm.' 	Yes
Windchill\db\ sql\Make_DDL_ install.sql	Remove '@wbm/ WbmCore/Drop_ module_WbmCore_ Array.sql @wbm/' WbmCore/Drop_ module_WbmCore_ Struct.sql @wbm/' WbmCore/Make_ module_WbmCore_ Struct.sql @wbm/'	

Path	Instructions	Ignore if Not Available
	WbmCore/Make_module_WbmCore_Array.sql'	
Windchill\db\sql\Drop_DDL_install.sql	Remove '@wbm/WbmCore/Drop_module_WbmCore_Array.sql @wbm/WbmCore/Drop_module_WbmCore_Struct.sql'	

Path	Instructions	Ignore if Not Available
Windchill\db\sql3\Make_DDL_install.sql	Remove '@wbm/WbmCore/Drop_module_WbmCore_Array.sql @wbm/WbmCore/Drop_module_WbmCore_Struct.sql @wbm/WbmCore/Make_module_WbmCore_Struct.sql @wbm/WbmCore/Make_module_WbmCore_Array.sql'	
Windchill\db\sql3\Drop_DDL_install.sql	Remove '@wbm/WbmCore/Drop_module_WbmCore_Array.sql @wbm/WbmCore/Drop_module_WbmCore_Struct.sql'	

-
4. From a Windchill shell, execute `xconfmanager-p` to propagate the above changes.

Note

It is possible to reinstall the Windchill Bulk Migrator on the same system. When reinstalling, do not inadvertently overwrite or delete the existing `MigrationSourceSite` table and audit tables. These tables serve as a record of migration activity.

Additional steps before reinstalling Windchill Bulk Migrator:

Modify the property file `psi_iir.xml` from

```
<SelectedProduct assemblyId="wbm" dir="<Windchill_HOME>\Windchill" isInstalled="true" dataLoaded="false" schemaLoaded="false" isDeleted="false"\> to false.
```

Rerun WNC installer, select **Update Existing Installation** instead of **New Product Installation**, (now user can see Windchill Bulk Migrator as selectable file for installation). Proceed with Windchill Bulk Migrator selection in optional module installation and give path of extracted Windchill Bulk Migrator media at staging location. Upon successful installation, Windchill Bulk Migrator version gets updated in Windchill version and new binary/jars are deployed. Continue with post installation and configure Windchill Bulk Migrator script as usual.

Uninstalling Extraction Utilities

1. Open a Windchill shell on the source system.
2. Change the directory to the directory where the extraction files were unzipped.
3. From the shell, enter the following command to stop Windchill:

```
windchill stop
```

4. Type the following command:

```
ant uninstall
```

This will retract all the files that were deployed.

Note

The `ant uninstall` command will not remove the staging database. All the files that were modified during installation, like `wt.properties` and `declarations.xconf`, are backed up to the following location before uninstalling:

```
<Windchill>\codebase\com\ptc\windchill\migration\
wbm\wc2wc\extractor\.wbmextractor\uninstall
```

5. Restart Windchill so the properties changes can take place.

Note

The previous version of the extractor should be uninstalled before installing a new version.

Terminating an Installation

If you terminate a Windchill Bulk Migrator installation, remove the following files and directories:

- `<WBM_Home>/windchill`
- `<Windchill>/installer/wbm.site.xconf.properties`
- `<Windchill>/installer/wbm`
- `<Windchill>/installer/logs/wbm`

After removing those files and directories, you can re-install the Windchill Bulk Migrator.

5

Windchill-to-Windchill Data Mapping

Windchill Bulk Migrator Mapping Definitions	47
The classMap.csv File	54

Windchill Bulk Migrator Mapping Definitions

The Windchill Bulk Migrator provides out-of-the-box data mappings for various administrative data. For Windchill-to-Windchill migrations, the data mappings are defined in XML files. The XML files with default values, are generated by the Windchill Bulk Migrator extraction utilities. After generating the default XML files, update the files to complete the mapping to the desired target system solution design.

The following contains the mapping files and their descriptions:

WBMWTUserMapping.xml

Provides the mapping of user database names between the source Windchill system and the target Windchill systems.

Note

The disabled/deleted user entries are included in the map as well.

This process makes the target value the same as the source out-of-the-box. It is the migration administrator's responsibility to modify the target value according to their target Windchill system.

The following is a detailed example for disabled/deleted user configuration in WBMWTUserMapping.xml:

1. Create all the required objects on the source server [WTpart] with the users you have to disable or delete [users disable and delete].
2. Disable/delete these users from the source system.
3. Create same users as disabled/deleted users on the target system and disable/delete them [user=disable and delete].
4. Run the following extraction command:

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.mapping.
client.WBMMappingClient -u wcadmin -p wcadmin
```

5. Map these users under WBMWTUserMapping.xml.

Note

The numerical values are `ida2a2` from the respective Windchill system.

```
<User srcDbName=" {wt.org.WTUser:10358}deleted" disabled="1" targetName=" {wt.org.WTUser:50107}deleted" />
<User srcDbName=" {wt.org.WTUser:10359}disable" disabled="1" targetName=" {wt.org.WTUser:50108}disable" />
```

6. Run the following command:

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.util.AttDefA
nalizerClient -a

<?xml version="1.0" encoding="UTF-8" ?>
- <WBMEtractorMapping>
- <Users>
  <User srcDbName="test_user" disabled="0" targetName="test_user" />
  <User srcDbName="Administrator" disabled="0" targetName="Administrator" />
  <User srcDbName="testuser1" disabled="0" targetName="testuser1" />
</Users>
```

WBMWTViewMapping.xml

Provides the mapping of WTPart View Names between the source and target Windchill systems. This process makes the target value the same as the source out-of-the-box. It is the migration administrator's responsibility to modify the target value according to their target Windchill system.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <WBMEtractorMapping>
- <Views>
  <View srcName="Manufacturing" targetName="Manufacturing" />
  <View srcName="Design" targetName="Design" />
</Views>
```

WBMTTeamTemplateMapping.xml

Provides the mapping of TeamTemplate names between the source and target Windchill systems. This process makes the target value the same as the source out-of-the-box. It is the migration administrator's responsibility to modify the target value according to their target Windchill system.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <WBMEtractorMapping>
- <Organizations>
  - <Organization name="ptc" type="wt.org.WTOrganization" id="445">
    - <Container type="wt.inf.container.ExchangeContainer" name="Site">
      <TeamTemplate srcName="Default" targetName="Default" />
      <TeamTemplate srcName="Change Activity Team" targetName="Change Activity Team" />
      <TeamTemplate srcName="Problem Report Team" targetName="Problem Report Team" />
      <TeamTemplate srcName="Change Notice Team" targetName="Change Notice Team" />
      <TeamTemplate srcName="Change Request Team" targetName="Change Request Team" />
      <TeamTemplate srcName="Variance Team" targetName="Variance Team" />
      <TeamTemplate srcName="Promotion Request Team" targetName="Promotion Request Team" />
    </Container>
  </Organization>
</Organizations>
```

WBMSiteMapping.xml

Use the `WBMSiteMapping.xml` file to map local and remote servers. The file contains entries for one master Windchill site, as well as entries for each remote site that the mapping generation tool finds for the source server. In the entry, extraction is controlled by the `use` attribute, which is set to `true` by default. When set to `true`, the host is used for extraction.

Note

Only one host is allowed per site and an exception will be thrown in runtime if more than one host per site is found.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<WBMEExtractorMapping>
  <Sites>
    <Site srcName="master">
      <Url srcUrl="http://15792-vm11.ptcnet.ptc.com:8080/Windchill/servlet/WindchillGW" targetUrl="" />
      <host hostName="15792-vm11.ptcnet.ptc.com" use="true" />
    </Site>
  </Sites>
</WBMEExtractorMapping>
```



```
<WBMEExtractorMapping>
  <Sites>
    <Site srcName="master">
      <Url
srcUrl="http://glepsky0d1.ptcnet.ptc.com:20600/Windchill/servlet/WindchillGW"
targetUrl="" />
      <host hostName="glepsky0d1.ptcnet.ptc.com" use="true" />
      <host hostName="fake.com" use="false" />
    </Site>
    <Site srcName="rfs1">
      <Url
srcUrl="http://glepsky0d.ptcnet.ptc.com:30600/Windchill/servlet/WindchillGW"
targetUrl="" />
      <host hostName="remotefake" use="true" />
    </Site>
  </Sites>
</WBMEExtractorMapping>
```

WBMTTypeDefMapping.xml

Use the `WBMTTypeDefMapping.xml` file to map all types, subtypes, and attributes from the source Windchill system to the target Windchill system. Use this file to map all Windchill global attributes as well as standard attributes.

In most situations, the `srcAttrName` is the attribute's logical form under the type. However, `srcAttrName` is

`IBA|someGlobalAttributeDefinitionName` when the attribute is not added to the type, but there are objects of the corresponding root class with the value for the specific attribute.

If you only want to extract subsets of attributes, you can identify this in the XML file by setting the `toBeMigrated` tag accordingly.

Example:

```
<Type srcTypeName="Wt.eppn.EPMDocument|com.ptc.ptclnet.DefaultEPMDocument|com.ptc.ptclnet.ECADDocument"
      targetTypeName="Wt.eppn.EPMDocument|com.ptc.ptclnet.DefaultEPMDocument|com.ptc.ptclnet.ECADDocument">
  <Attribute srcAttrName="PTC_ECAD_ASSEMBLY_PART_NUMBER" srcAttrGlobalDefName="PTC_ECAD_ASSEMBLY_PART_NUMBER" srcClassifyBindingAttr="false" srcDataType="java.lang.String"
    targetAttrName="PTC_ECAD_ASSEMBLY_PART_NUMBER" toBeMigrated="true" />
  <Attribute srcAttrName="PTC_ECAD_BOARD_PART_NAME" srcAttrGlobalDefName="PTC_ECAD_BOARD_PART_NAME" srcClassifyBindingAttr="false" srcDataType="java.lang.String"
    targetAttrName="PTC_ECAD_BOARD_PART_NAME" toBeMigrated="true" />
  <Attribute srcAttrName="PTC_ECAD_ASSEMBLY_PART_NAME" srcAttrGlobalDefName="PTC_ECAD_ASSEMBLY_PART_NAME" srcClassifyBindingAttr="false" srcDataType="java.lang.String"
    targetAttrName="PTC_ECAD_ASSEMBLY_PART_NAME" toBeMigrated="true" />
```

For File Based attributes, `isFileBased` is set to `true` on attribute definition for EPMDocument, EPMDocumentMaster, EPMMemberLink, EPMReferenceLink, ModelItem, ModelItemMaster, and ModelItemLink. This attribute will help user to identify that attribute is file based attribute and if `toBeMigrated` is set to `false` then it gets ignored.

The new behaviour will ignore `toBeMigrated` attribute for file based attributes on objects of type EPMDocument, EPMDocumentMaster, EPMMemberLink, EPMReferenceLink, ModelItem, ModelItemMaster, and ModelItemLink. It also ignores `toBeMigrated` attribute for Ad-Hoc attributes on WTPart, WTPartMaster and WTPartUsageLink.

Summary:

1. `toBeMigrated` flag gets ignored and attribute will be migrated only if attribute is file based attribute for objects of type EPMDocument, EPMDocumentMaster, EPMMemberLink, EPMReferenceLink, ModelItem, ModelItemMaster, and ModelItemLink.
2. `toBeMigrated` flag does not get ignored for general attributes on EPMDocument, EPMDocumentMaster, EPMMemberLink, EPMReferenceLink, ModelItem, ModelItemMaster and ModelItemLink.
3. `toBeMigrated` flag gets ignored and attribute will be migrated only if attribute is AD-Hoc attribute on WTPart, WTPartMaster and WTPartUsageLink.
4. `toBeMigrated` attribute does not ignored for general attributes on WTPart, WTPartMaster and WTPartUsageLink.
5. If user do not want to ignore `toBeMigrated` attribute then hidden property `data.extract.OverrideToBeMigrated` can be set to `true` in `extract.properties`. This property will always consider `toBeMigrated` flag during migration irrespective of type of attribute or object.

WBMTTypeDefMapping.xml snippet with `toBeMigrated` set to `false` for file based and non-file based attributes on EPMDocument .

```
<Type srcTypeName="wt.epm.EPMDocument" targetTypeName="wt.epm.EPMDocument">

    ...

    <Attribute srcAttrName="ABC" srcAttrGlobalDefName="ABC" srcClassifyBindingAttr="false" srcDataType="java.lang.String" targetAttrName="ABC" isFileBased="true" toBeMigrated="false" />

    <Attribute srcAttrName="DEF" srcAttrGlobalDefName="DEF" srcClassifyBindingAttr="false" srcDataType="java.lang.String" targetAttrName="DEF" toBeMigrated="false" />

    ...

</Type>
```

WBMTTypeDefMapping.xml snippet with `toBeMigrated` set to `false` for ad-hoc attributes on WTPart

```
<Type srcTypeName="wt.part.WTPart" targetTypeName="wt.part.WTPart">

    ...

    <Attribute srcAttrName="IBA|ABC" srcAttrGlobalDefName="ABC" srcClassifyBindingAttr="false" srcDataType="java.lang.String" targetAttrName="IBA|ABC" toBeMigrated="false" />

    <Attribute srcAttrName="IBA|DEF" srcAttrGlobalDefName="DEF" srcClassifyBindingAttr="false" srcDataType="java.lang.String" targetAttrName="IBA|DEF" toBeMigrated="false" />

    ...

</Type>
```

For more information on mapping source system modeled attributes to target system soft type global attributes, see “Attribute Extraction” in [Running the Windchill Bulk Migrator Extraction Utilities on page 71](#).

Note

Classification attributes are not included in this mapping file. They are located in the `WBMClassificationMapping.xml` file.

WBMLifeCycleMapping.xml

Provides the mapping of the lifecycle (that is defined on the container) between the source and target Windchill systems. This process makes the target value the same as the source out-of-the-box. It is the migration administrator's responsibility to modify the target value according to their target Windchill system.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <WBMLifeCycleMapping>
- <Organizations>
- <Organization name="ptc" type="wt.org.WTOrganization" id="445">
- <Container type="wt.inf.container.ExchangeContainer" name="Site">
- <LifeCycleTemplateMaster srcName="Default" targetName="Default">
<PhaseTemplate srcPhaseState="UNDERREVIEW" targetPhaseState="UNDERREVIEW" inLatest="true" />
<PhaseTemplate srcPhaseState="RELEASED" targetPhaseState="RELEASED" inLatest="true" />
<PhaseTemplate srcPhaseState="INWORK" targetPhaseState="INWORK" inLatest="true" />
</LifeCycleTemplateMaster>
```

WBMOrganizationMapping.xml

Provides the mapping of the organizations, containers (Products, libraries and projects) and folders between the source and target Windchill systems. The file maintains the hierarchy of organizations, containers and folders. This process makes the target value the same as the source out-of-the-box. It is the migration administrator's responsibility to modify the target value according to their target Windchill system.

Example:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <WBMOrganizationMapping>
- <Organizations>
- <Organization srcName="ptc" targetName="ptc">
- <Container type="wt.inf.container.ExchangeContainer" srcName="Site" targetName="Site">
<Folder srcPath="/Default" targetPath="/Default"/>
<Folder srcPath="/Default/Change Log" targetPath="/Default/Change Log"/>
<Folder srcPath="/Default/General" targetPath="/Default/General"/>
<Folder srcPath="/Default/Policies" targetPath="/Default/Policies"/>
<Folder srcPath="/System" targetPath="/System"/>
<Folder srcPath="/System/ProductPlan" targetPath="/System/ProductPlan"/>
<Folder srcPath="/System/Reports" targetPath="/System/Reports"/>
<Folder srcPath="/System/Reports/ChangeMonitor" targetPath="/System/Reports/ChangeMonitor"/>
<Folder srcPath="/System/Reports/ChangeMonitor/Custom" targetPath="/System/Reports/ChangeMonitor/Custom"/>
<Folder srcPath="/System/Workflows" targetPath="/System/Workflows"/>
<Folder srcPath="/Type Definitions" targetPath="/Type Definitions"/>
</Container>
```

A new attribute named `Profile` with three different values `LICENSE`, `NONE`, and `STANDARD` has been added to the element `Group`.

```
</Organization>
<Organization targetName="PTC1" srcName="PTC1">
<Container targetName="Site" srcName="Site"
type="wt.inf.container.ExchangeContainer">
<Folder targetPath="/Default" srcPath="/Default"/>
<Group targetName="PTC Design Control License" srcName="PTC Design Control License"
profile="LICENSE"/>
<Group targetName="Replication Managers" srcName="Replication Managers"
profile="NONE"/>
<Group targetName="Customer" srcName="Customer" profile="STANDARD"/>
</Container>
</Organization>
```

WBMCClassificationMapping.xml

Provides the mapping of classification namespace, node path, and classification attributes between the source and target Windchill system. This process makes the target value the same as the out-of-the-box source. It is the migration administrator's responsibility to modify the target value according to their target Windchill system.

Example:

```

<?xml version="1.0" encoding="UTF-8" ?>
- <WBMEtractorMapping>
- <NameSpace srcNameSpace="" targetNameSpace="">
- <NodePath srcNodePath="Electronic Parts" targetNodeName="Electronic Parts">
- <Attribute srcAttrGlobalDefName="xje136" srcAttrDisplayName="Weight" targetAttrName="IBA|xje136" toBeMigrated="true" />
- <Attribute srcAttrGlobalDefName="RoHS_Compliant" srcAttrDisplayName="RoHS Compliant" targetAttrName="IBA|RoHS_Compliant" toBeMigrated="true" />
- <Attribute srcAttrGlobalDefName="ptc_general_description" srcAttrDisplayName="General Description" targetAttrName="IBA|ptc_general_description" toBeMigrated="true" />
</NodePath>

```

WBMWFProcessMapping.xml

Provides the mapping of workflow process template between the source and the target Windchill system. This includes mapping of all activity templates, connector templates, and process variables. Unlike other mapping, workflow process template mapping requires mapping of source ID and the target ID (idA2A2-to-idA2A2 mapping). Out-of-the-box, the source ID is populated and the target ID is left blank. However, it populates target process template name, activity template name, and process variable names the same as source system. It is the migration administrator's responsibility to update the target ID and template name to their target Windchill system.

Example:

```

<?xml version="1.0" encoding="UTF-8" ?>
- <WBMEtractorMapping>
- <WPProcessTemplates>
- <WPProcessTemplate toBeMigrated="true">
- <Container>
- <Source id="15894" Name="Submit">
- <ContainerName>Site</ContainerName>
- <ContainerType>SITE</ContainerType>
- <ContainerOrgName />
</Source>

```

WBMTranslationDictionaryMapping.xml

After running wbmggen, complete the following steps on the target Windchill system to generate the WBMTranslationDictionaryMapping.xml file:

1. On the target Windchill system, execute the following command:

```

windchill
com.ptc.windchill.migration.wbm.mapping.WBMDictionaryMappingClient -u <username> -p <password>

```

The mapping file can be found at <WT_HOME>\loadFiles\wbm\extractor\WBMTranslationDictionaryMapping.xml.

2. Copy the mapping file from the target Windchill system to <wbm_working_directory>\mapping folder on source Windchill.
3. Execute the extraction on the source system.

WBMWTFormulaSetMapping.xml

Provides the mapping of FormulaSet names between the source and target Windchill systems. This process makes the target value the same as the source out-of-the-box. It is the migration administrator's responsibility to modify the target value according to their target Windchill system.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<WBMEtractorMapping>
  <FormulaSets>
    <FormulaSet targetName="Standard Time and Cost" srcName="Standard Time and Cost"/>
    <FormulaSet targetName="Periodic Setup Time and Cost" srcName="Periodic Setup Time and Cost"/>
  </FormulaSets>
</WBMEtractorMapping>
```

The list describes the default mapping behavior implemented with the class com.ptc.windchill.migration.wbm.wc2wc.mapping.server.MappingHandlerDefault. You can have your own mapping behavior by implementing the com.ptc.windchill.migration.wbm.wc2wc.mapping.server.MappingHandler interface. Register the customized mapping class by modifying the value of the key data.extract.mapping.mappingHandlerClass in the file <Windchill> /codebase/com/ptc/windchill/migration/wbm/wc2wc/extractor/extract.xconf. Then, run the xconfmanager utility to propagate the change to <Windchill> /codebase/com/ptc/windchill/migration/wbm/wc2wc/extractor/extract.properties.

The classMap.csv File

The Windchill Bulk Migrator extractor requires the `classMap.csv` file that provides the mapping of a Windchill class to its staging table, as well as the mapping between Windchill class object properties and columns in the staging table.

For more information, see [Understanding the classMap.csv File on page 228](#).

6

Configuring the Windchill Bulk Migrator

Configuring the Windchill Bulk Migrator Loading Infrastructure	56
Configuring Extraction Utilities	66
Deferred Constraint Enforcements	69

This chapter describes how to configure the Windchill Bulk Migrator. Configuring the Windchill Bulk Migrator includes setting property values for properties in the `wbm.properties` file and updating the create schema scripts.

Configuring the Windchill Bulk Migrator Loading Infrastructure

Wbm.properties Configurations Overview

All properties used by the Windchill Bulk Migrator are stored in the `wbm.properties` file. Use the `xconfmanager` utility to set property values for the `wbm.properties` target file and then propagate the settings to `wbm.properties`.



Tip

When setting a large number of property values, create a text file containing the properties and values you want to set. Enter one `name=value` pair on each line. Then use the `--setfromfile` parameter on the `xconfmanager` command to name the file you have created. For additional information on using the `xconfmanager` utility, see the section Specialized Administration in the Windchill Help Center.

Ensure that you have set the following types of information set in `wbm.properties`:

- Supported object types for legacy loaders and optional modules
- Base directory for SQL query files and specific query file names by object type
- Locale setting
- Restricted domain settings
- Content loading settings
- Incremental migration (loading)
- Staging database connection information: host:port:SID
- Staging database user and password settings

Other loading infrastructure configurations:

- Configuring log file information
- Subtype and attribute settings
- Average object history size for versioned objects
- ConnectionPool settings

The following sections describe the properties to set.

Supported Object Type Setting for Optional Modules and Legacy Loaders

The following property lists legacy loaders that are supported by Windchill Bulk Migrator for various modules on the target Windchill system:

```
com.ptc.windchill.migration.wbm.supportedObjectsTypes=$(com.ptc.windchill.migration.wbm.supportedModuleObjectsTypes_FOUNDATION)
```

```
com.ptc.windchill.migration.wbm.supportedObjectsTypes=$(com.ptc.windchill.migration.wbm.supportedModuleObjectsTypes_FOUNDATION)
```

The list of object types supported by each module are established through the following properties:

```
com.ptc.windchill.migration.wbm.supportedModuleObjectsTypes_FOUNDATION=CONTENT  
METADATA,Folder,PDMLinkProduct,LibraryContainer,OrgContainer,WTOrganization,EPMFamilyTable,EPMFTProp,EPMAsStored,SHAREDCONTAINERMAP,EPMParamMapForDoc,EPMParamMapForML,EPMParamMapForRL,EPMDocumentUnitInfo,EPMMemLinkUnitInfo,EPMRefLinkUnitInfo,DERIVEDIMAGE,WFPROCESS
```

Each of the above enlists comma separated corresponding legacy loader object types for the module.

Settings for Query File Base Directory and Names

You must establish the base directory for SQL query files, identify which query files to use of each top-level legacy object type.

For Windchill Bulk Migrator to locate the SQL query files, you must set the base directory under which the query files are stored. The default base directory is:

```
<Windchill>/db/sql/com/ptc/windchill/migration/wbm/staging/query
```

where *<Windchill>* is the Windchill installation directory.

If you have stored the query files in a different base directory, set the following property:

```
com.ptc.windchill.migration.wbm.config.sql.query.basedir
```

Additionally, you must identify the set of query files used by Windchill Bulk Migrator for processing the data. To do this, identify the top-level object types associated with the data that will be loaded and set a property for each type that identifies the query file that will be used to process the data of that type. The format of the properties to set is as follows:

```
com.ptc.windchill.migration.wbm.config.sql.query.file.<obj_type>=<query_file>
```

where:

- <obj_type> is the last part of internal name of the top-level object type. For example, the internal name for OrgContainer type is wt.inf.container.OrgContainer. Therefore, use OrgContainer for <obj_type>.
- <query_file> is the file name of the SQL query file that has been set up for the object type. The file must reside in the base directory set in the basedir property described earlier.

For example, if the query file for the Part type is OrgContainerQuery.sql, then enter the following property:

```
com.ptc.windchill.migration.wbm.config.sql.query.file.OrgContainer=OrgContainerQuery.sql
```

Locale Setting

To set up source locale, set the following property in wbm.properties:

```
com.ptc.windchill.migration.wbm.datasource.locale=<locale>
```

where <locale> identifies the locale of the data source and can be any value supported by Windchill. For example, the following property sets the locale value to the Korean language:

```
com.ptc.windchill.migration.wbm.datasource.locale=ko
```

The default locale value is blank after the initial installation. Set this property when a different locale is required for the data being migrated.

Restricted Domain Configurations in wbm.properties

Restricted migration domains can be used on the target system to support use cases for migrating into an active production system. For more information about migrating into an active production system, see [Migrating Into an Active Production System](#).

To use restricted domains on the target system when migrating data, the following property in wbm.properties must be set to true:

```
com.ptc.windchill.migration.wbm.domain.restricted
```

By default, this property is set to true and unless changed, restricted domains are used in the migration process.

Use the following property to identify the list of object types that can be migrated into a restricted domain:

```
wbm.objectType.supported.restricted.domain=<obj_type_list>
```

In the object type list, separate each object type in the list using a comma. For individual object types in the list, use the last part of the internal name of the top-level object type. For example, if the internal name of the part type is wt.part.WTPart, you would include WTPart in *<obj_type_list>*. If additional types are added using modeled subclasses, those types must be added to the list.

The `wbm.objectType.supported.restricted.domain` property is also used to identify top-level object types when moving objects from a restricted domain to a public domain.

File Content Migration Settings

There are two options for migrating file content using the Windchill Bulk Migrator. Standard content migration loads content into the Windchill default cache vault, and subsequent revaulting occurs during post-migration through normal Windchill content services. Enhanced content migration provides a means to move file content directly into Windchill vaults.

Standard Content Migration

No property settings need to be modified to enable standard content migration. By default, the following property that controls the selection of the content migration option is provided in `wbm.properties`:

```
com.ptc.windchill.migration.wbm.xferContent=true
```

Enhanced Content Migration

To implement enhanced content migration, use the xconfmanager to set the following properties in the `wbm.properties` file:

```
com.ptc.windchill.migration.wbm.xferContent=false
com.ptc.windchill.migration.wbm.processor.postload.CONTENTMETADATA=
com.ptc.windchill.migration.wbm.processor.ContentPostLoadProcessor
com.ptc.windchill.migration.wbm.processor.preload.CONTENTMETADATA=
com.ptc.windchill.migration.wbm.processor.ContentLoaderPreLoadProcessor
com.ptc.windchill.migration.wbm.vaultconfig.single=false
com.ptc.windchill.migration.wbm.loaders.threadpool.max.size=5
```

To use the vault setting for multiple sites with a single master vault per site, change the following property to true in the `wbm.properties` file:

```
com.ptc.windchill.migration.wbm.vaultconfig.single=
false
```

In addition, use the xconfmanager to set the following properties in the `wt.properties` file on target Windchill system:

```
wt.fv.useVaultsForAllContent=False
```

```
wt.fv.forceContentToVault=false  
  
wt.fv.useFvFileThreshold=true  
  
wt.fv.fvFileThreshold=4000  
  
wt.fv.uploadtocache.ignoreFileMoveCheck=true
```

For more information about use cases for standard and enhanced content migration, see [File Content Migration on page 235](#). It is highly recommended that you review this section before using the enhanced content migration approach.

Incremental Migration Settings

Enabling Incremental Migration

You must enable the incremental migration mode by setting the following property in `wbm.properties` to `true`:

```
com.ptc.windchill.migration.wbm.mode.incremental=true
```

By default, this property is set to `false`.

Note

Ensure that this property is set to `false` for the migration of new data and object masters into Windchill.

Configuring Log File Information

In the Windchill Bulk Migrator, log configuration XML files reside in the `<Windchill>/codebase/com/ptc/windchill/migration/wbm/common/log` directory. There are three configuration XML files:

- `log4jBulkMigration.xml` – Contains the loader settings
- `log4jBulkMigration_loader.xml` – Contains loader-specific logger settings

Note

Out of the box, these XML files are configured to capture complete debugging information in specified log files. Changing settings in these files is optional.

To change settings in the log configuration XML files, use either an XML or text editor. Since these files are Windchill Bulk Migrator files, modifying the content of the files is considered a customization. Be sure to backup the original files before making changes and save copies of the updated files according to established site maintenance practices.

The following sections describe the log files generated by the Windchill Bulk Migrator and the settings within log configuration XML files that you can modify.

Logger Definitions

Loading specific logger names are defined in the `log4jBulkMigration_loader.xml` file.

The following items provide information about the loggers used by Windchill Bulk Migrator:

- Error Logger

Error appender logger name is: `com.windchill.migration.wbm.error`

Default logging level is: `ERROR`

- Report Logger

Report appender logger name is: `com.windchill.migration.wbm.report`

Default logging level is: `INFO`

- Migration Loader Logger

This logger is associated with the migration loader appender.

By default, a higher level logger is set from the `com.ptc.windchill.migration.wbm` logger and the default level is `DEBUG`:

```
<logger name="com.ptc.windchill.migration.wbm" additivity="false">
<level value="DEBUG" />
<appender-ref ref="migration" />
<appender-ref ref="console" />
</logger>
```

Every class has its own logger defined. To define logging for a class, add the logger name as the fully qualified class name associated with logging level and appender. For example:

```
<logger name="com.ptc.windchill.migration.wbm.DataLoadRouterImp" additivity="false">
<level value="DEBUG" />
<appender-ref ref="migration" />
<appender-ref ref="console" />
</logger>
```

Error File Configuration

Errors generated when running the Windchill Bulk Migrator are logged in a separate file. The following are error file appender settings that are defined in the `log4jBulkMigration_loader.xml` file:

- The `FieldSeparator` setting controls column separators.

The default setting creates tab-separated fields:

```
<param name="FieldSeparator" value="\t"/>
```

Possible values are either `\t` (the tab character) or `|` (the pipe symbol).

- The `IncludeRecord` setting controls whether the error file includes error records.

The default setting includes error records:

```
<param name="IncludeRecord" value="true" />
```

Possible values are either `true` or `false`.

- The `IncludeConflictMsg` setting controls whether conflict messages are include in the error file.

The default setting includes conflict messages:

```
<param name="IncludeConflictMsg" value="true" />
```

Possible values are either `true` or `false`.

- The `MaxFileSize` setting controls maximum error file size.

The default setting sets the maximum error file size at 10,485 KB:

```
<param name="MaxFileSize" value="10485760"/>
```

The name of the error file has the following format:

```
loadErrors-<JVM_start_timestamp>-<JVM_process_id>-  
log4j.log
```

where `<JVM_start_timestamp>` is the date formatted as `yyMMddHHmm` and `<JVM_process_id>` is the process ID of the JVM.

When the size of an error file reaches the maximum size (as defined in the `MaxFileSize` setting), the existing file is renamed by adding a numeric suffix and a new log file is started with the existing file name. The format of the name used for the renamed files is as follows:

```
loadErrors-<JVM_start_timestamp>-<JVM_process_id>-  
log4j.log.{1..n}
```

where `{1..n}` identifies order in which the files were created.

For example, assume the log folder has three error log files:

```
loadErrors-1102071731-1284-log4j.log
```

```
loadErrors-1102071731-1284-log4j.log.1  
loadErrors-1102071731-1284-log4j.log.2
```

Then log entries are contained in the file as:

```
loadErrors-1102071731-1284-log - Latest entry  
loadErrors-1102071731-1284-log4j.log.2 - Next older entry  
loadErrors-1102071731-1284-log4j.log.1 - Oldest entry
```

Load logs for generically loadable objects can be found in the `cirouter.log` file at `<WT_HOME>\wbm\cirouter\<DateCode>\logs` and Method Server logs. For a list of generically and non-generically loadable objects, execute the `windchill`

```
com.ptc.windchill.migration.wbm.DataLoadController -l  
command on the target system.
```

Report File Configuration

Each Windchill Bulk Migrator worker thread produces a summary of processed records. The processed record summaries are logged in separate file. The report file definition is defined in the `log4jBulkMigration_loader.xml` file.

Each record summary in the report file contains the following fields:

- `WorkerThread = <thread_name>`
- `objectType = <object_name_processed>`
- `objectCount = <number_records_processed>`
- `startTime = <start_timestamp>` (formatted as `yyyy-MM-dd HH:mm:ss:mmm`)
- `successCount = <number_record_successes>`
- `failCount = <number_records_failed>`
- `elapsedTime = <time_to_process_all_records>`

For example:

```
WorkerThread = T-2, objectType = EPMDocument, objectCount = 3,  
startTime = 2011-01-14 16:44:41.97, successCount = 3, failCount = 0,  
elapsedTime = 343598
```

The name of the report file has the following format:

```
loadReport-<JVM_start_timestamp>-<JVM_process_id>-  
log4j.log
```

where `<JVM_start_timestamp>` is the date formatted as `yyMMddHHmm` and `<JVM_process_id>` is the process ID of the JVM.

The report file is rolled on daily basis. Each day, the existing file name is renamed by appending the date of the gathered content in the following format:

.YYYY-MM-dd

For example, assume that the record summaries are gathered on March 17, 2011 in a file named:

`loadReport-1103041644-5616-log4j.log`

At the end of the day, the file is renamed:

`loadReport-1103041644-5616-log4j.log.2011-03-17`

Migration Loader File Configuration

A separate migration log file contains logs generated from all client-side data loading operations. The following are migration loader file appender settings that are defined in the `log4jBulkMigration_loader.xml` file:

- The `ConversionPattern` setting controls what is saved in each log entry:

```
<param name="ConversionPattern" value="%d{ISO8601} %-5p [%t] %C - %m%n" />
```

By default, the conversion pattern contains the following fields:

- `%d{ISO8601}` – Output date uses the ISO8601 format
- `%-5p` – Logging level format uses spaces to the right of the level if logging level is less than 5 characters
- `%t` – Thread name
- `%C` – Caller class name
- `%m` – Application supplied message
- `%n` – Platform dependent line separator

- The `MaxFileSize` setting controls maximum migration loader file size.

The default setting sets the maximum migration loader file size at 10,485 KB:

```
<param name="MaxFileSize" value="10485760"/>
```

The name of the migration loader file has the following format:

`load-<JVM_start_timestamp>-<JVM_process_id>-log4j.log`

where `<JVM_start_timestamp>` is the date formatted as `yyMMddHHmm` and `<JVM_process_id>` is the process ID of the JVM.

When the size of a migration loader file reaches the maximum size (as defined in the `MaxFileSize` setting), the existing file is renamed by adding a numeric suffix and a new log file is started with the existing file name. The format of the name used for the renamed files is as follows:

```
load-<JVM_start_timestamp>-<JVM_process_id>-
log4j.log.{1..n}
```

where `{1..n}` identifies order in which the files were created.

For example, assume the log folder has three migration loader log files:

```
load-1103122441-1875-log4j.log  
load-1103122441-1875-log4j.log.1  
load-1103122441-1875-log4j.log.2
```

Then log entries are contained in the file as:

```
load-1103122441-1875-log4j.log – Latest entry  
load-1103122441-1875-log4j.log.2 – Next older entry  
load-1103122441-1875-log4j.log.1 – Oldest entry
```

ConnectionPool Settings

ConnectionPool handles all the connections to the Staging Database.

These properties help in defining the size and cleanup of connections in pool.

- You can define the minimum number of connections with `minPoolSize` to start the process. The pool initializes with `minPoolSize` and as demand grows new connections are added to the pool.

```
data.loader.commonConnectionPool.minPoolSize=5
```

- Maximum number of connections in use will never be above `maxPoolSize` as defined below

```
data.loader.commonConnectionPool.maxPoolSize=10
```

Note

Always select the value of `maxPoolSize` higher than `minPoolSize`. If the process needs more connection than the `minPoolSize` number at the same time, then it has to wait for other process to finish. WBM always try to maintain `minPoolSize` connections in the pool and hence need to remove the idle connections after some time from the pool.

- If the following property is set to true, it enables the CleanUp Schedule. The property is disabled by default.

```
data.loader.commonConnectionPool.enableIdleConnectionCleanup=false
```

- `enableIdleConnectionCleanupInterval` defines the seconds after which the connection will be idle (not in use).

```
data.loader.commonConnectionPool.enableIdleConnectionCleanupInterval=60
```

If any connection is idle for more than `enableIdleConnectionCleanupInterval` seconds, it is considered for cleanup only if the total connections are higher than `minPoolSize`.

 **Note**

The values given above for every property is the default value.

Configuring Extraction Utilities

Windchill Bulk Migrator Extraction Utilities Logging

The Windchill Bulk Migrator extraction utilities utilize the log4j framework to provide the logging mechanism. WBM Extraction logging produces a client log file, server log file and server error log file. The server error log file only contains those statements that have logging level ERROR or higher.

To facilitate the user collecting all logs and other info, Windchill Bulk Migrator extraction utilities provide the target “collectlogs” in the ant build script. In a Windchill shell, change to the directory where the Windchill Bulk Migrator extraction files are unzipped. Type the following command:

```
ant collectlogs
```

It collects all the relevant files including various log files, `classmap.csv`, administrative data mapping files, data dictionary XML files and others, and creates a ZIP file. The ZIP file is stored as `<WBMExtractWorkingDir>/WBMExtract-Log-<timestamp>.zip`.

The following table describes the log properties specified in the `<Windchill>/codebase/com/ptc/windchill/migration/wbm/wc2wc/extractor/extract.xconf`:

Log Property	Description
<code>client.log4j.appendер.rolling.File</code>	The location on the file system where the client log file will be created. The default is <code><Windchill>/logs/WBM/extract/client/Client-\${user.name}- \${wbm.jvm.startTime}-log4j.log</code> .
<code>client.log4j.appendер.rolling.MaxValueSize</code>	The maximum size (in MB) the client log file can grow up to. For example, 50MB.

Log Property	Description
client.log4j.appendер.rolling.MaxBackupIndex	The number of backups to preserve for the client log files to keep. For example, 5.
data.extract.ciRouterBatchIds	Defines the number of batches while extracting the data into staging area. For example, if the value is set to '2', objects are extracted in 2 batches with BATCH_ID=1 and BATCH_ID=2 in staging area.
server.log4j.appendер.rolling.File	The location on the file system where the server log file will be created. The default is <Windchill>/logs/WBM/extract/server/Server- \${wt.jvm.startTime.formatted.short}-\${wt.jvm.id}-log4j.log.
server.log4j.appendер.rolling.MaxFileSize	The maximum size (in MB) the server log file can grow up to. For example, 50MB.
server.log4j.appendер.rolling.MaxBackupIndex	The number of backup to preserve for the server log files to Keep. For example, 5.
server.log4j.appendер.errorLogging.File	The location on the file system where the server error log file will be created. The default is <Windchill>/logs/WBM/extract/server/Server- \${wt.jvm.startTime.formatted.short}-\${wt.jvm.id}-Error-log4j.log.
server.log4j.appendер.errorLogging.MaxFileSize	The maximum size (in MB) the server error log file can grow up to. For example, 50MB.

Log Property	Description
server.log4j.appenders.errorLogging.MaxBackupIndex	The number of backup to preserve for the server error log files to keep. For example, 5.
server.log4j.category.com.ptc.windchill.migration.wbm.wc2wc	Define the logger level with three appenders writing to console, rolling and errorLogging. For example, “INFO,CA,rolling,errorLogging”. The allowed values for logger level are: DEBUG, INFO, WARN, ERROR, FATAL

Extraction Properties of Connection Pooling

ConnectionPool handles all the connections to the Staging Database and Source Database. These properties help in defining the size and clean-up of connections in the pool.

Connection Pooling Property	Description
data.extract.commonConnectionPool.minStagingPoolSize=1	Minimum Connections with staging database to start with. WBM initializes the pool with this value. As demand grows new connections are added to the pool. (It is always greater or equal to 1)
data.extract.commonConnectionPool.maxStagingPoolSize=5	Maximum number of connections with Staging Database in use. It will never be above maxStagingPoolSize.
data.extract.commonConnectionPool.minWindchillPoolSize=1	Minimum Connections with Source Database to start with. WBM initializes the pool with this value. As demand grows new connections are added to the pool. (Always greater or equal to 1).
data.extract.commonConnectionPool.maxWindchillPoolSize=5	Maximum number of connections with Source Database in use. It will never be greater than maxWindchillPoolSize.

Connection Pooling Property	Description
data.extract.commonConnectionPool.enableIdleConnectionCleanup=false	This property helps in the clean-up process (by default it's disabled).
data.extract.commonConnectionPool.enableIdleConnectionCleanupInterval=60	enableIdleConnectionCleanupInterval defines the seconds after which the connection is idle (not in use). So if some connection is idle for more than enableIdleConnectionCleanupInterval seconds, it is considered for clean-up only if total connections are higher than minPoolSize. (The clean-up will start for both the pools).

Deferred Constraint Enforcements

The Windchill Bulk Migrator loads soft type attributes of an IBAHolder object during loading. The soft attribute may be constrained on the target system. The Windchill Bulk Migrator supports cases when the attribute value may need to conform to the constraints on target system attributes, and when it needs to support the historical values bypassing the constraints during the load time. In the latter case the constraints are deferred during the load time persistence, and becomes active as soon as the soft type is saved. The desired behavior can be controlled using the following property:

```
com.ptc.windchill.migration.wbm.lwc.deferredEnforcement=true <default>
```

The applicable values are Boolean value <true|false>. By default, this property is set to true, indicating loading behavior that supports deferring the constraints on the attributes until after the object is saved or loaded. When the value is set to false, the loader successfully loads those objects whose attribute values are consistent with the constraints defined in the target system.

7

Extracting Data

Windchill Bulk Migrator Extraction Utilities Overview	71
Running the Windchill Bulk Migrator Extraction Utilities	71
Data Not Extracted	81
Windchill Bulk Migrator Extraction Utilities Status Reports	81
One-to-Many Extraction	83
Extraction Sequence	84

This chapter contains information on extraction data using the Windchill Bulk Migrator.

Windchill Bulk Migrator Extraction Utilities Overview

The Windchill Bulk Migrator extraction utilities provide the tools to extract from the source Windchill system to the staging database

This release of the Windchill Bulk Migrator supports two types of extraction:

1. Normal/Full extraction: Every iteration of every object, by type that exists in the source system This is the default behavior and normal operation of the tools
2. Filtered extraction: The user can provide a filter criteria through the filter extraction user interface and can extract desired data from source Windchill system to the staging database. For more information, refer to the Chapter 8.

After successfully installing the Windchill Bulk Migrator extraction utilities and starting the source Windchill system, the following high-level steps can be performed to extract data:

1. Generate Extraction Mapping
2. Generate data dictionary files
3. Create and modify the administrative data mapping files
4. Extract data for object types

Two additional features will be useful:

- Extraction status reports
- Windchill Bulk Migrator extraction logging that includes a client log file, server log file and server error log file For more information on logging, see [Configuring Log File Information on page 60](#)

Running the Windchill Bulk Migrator Extraction Utilities

The following sections describe the procedure to run the Windchill Bulk Migrator extraction utilities.

Generate data dictionary files

Use the following procedure to generate the data dictionary files:

1. Start the Windchill server.
2. From a Windchill shell, execute the following command:

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.util.AttDefAnalyzerClient -a
```

This outputs the data dictionary XML files of all object classes listed under <Windchill>\ codebase\com\ptc\windchill\migration\wbm\wc2wc\extractor\classMap.csv to <WBMEExtractWorkingDir>/DataDictionary. For other uses of the command, execute:

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.util.AttDefAnalyzerClient -h
```

For example, an alternative argument generates the data dictionary XML file for a specific Windchill class.

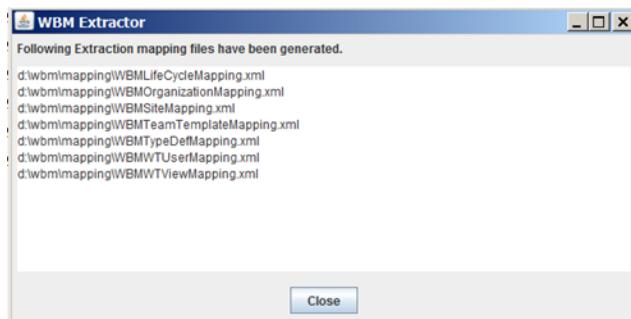
The data dictionary file of the object type is referenced when the specific object type (Windchill class) is extracted.

Create and modify the administrative data mapping files

The mapping files provide the transformation capability from the source Windchill administrative data value to the target Windchill administrative data value. From a Windchill shell, execute the following command:

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.mapping.
client.WBMMappingClient -u <wtuser> -p <wtpassword>
```

The following window appears listing the mapping files generated:



The mapping files are generated under <WBMEExtractWorkingDir>/mapping. Those mapping files provide the transformation capability from the source Windchill administrative data value to the target Windchill administrative data value. You should only modify the value of the tag if it begins with "target" (for example, "targetName", "targetPath", "targetUrl", "targetAttrName", "targetTypeName", and "targetPhaseState") in the mapping XML files if needed. You can also modify the value of the tag if it begins with "use" or "toBeMigrated". The mapped target windchill administrative data value will be the one populated into the staging database.

Administrative data here include users, part views, teams, sites, types (soft types), lifecycles, organizations, products, libraries, projects and folders. For more information, see [Windchill Bulk Migrator Mapping Definitions on page 47](#)

Extract data for specific object types

Once data dictionary files and mapping files are in place, the procedure of extracting data to the staging database can begin. You can still add more mapping entries into administrative data mapping files during the entire extraction procedure; however, you should not modify any existing mapping entries once extraction procedure takes place. The Windchill Bulk Migrator extractor framework utilizes Windchill introspection to get the values of Windchill attributes and populate those values into the staging database.

Extraction Commands

The following list shows information that can be extracted using the Windchill Bulk Migrator Extraction Utilities. All commands should be executed from a Windchill shell:

- To extract the information of a specific Windchill business class

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.extractor.
client.WindchillDataExtractor -o <WindchillClassName>
-u <wtuser> -p <wtpassword>
```

In this command, <WindchillClassName> is the full Java class name of the Windchill class; for example, wt.part.WTPart, wt.epm.attributes, EPMPParameterMap, etc. The <WindchillClassName> and its mappings should exist in the `classMap.csv`.

Note

During Extraction One-off EPMDocuments will get filtered out, whereas one-off EPMFamilyTable will get extracted.

- To extract the file content metadata

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.extractor.
client.WindchillDataExtractor -o
<ContentHolderClassName> -content -u <wtuser> -p
<wtpassword>
```

In this command, <ContentHolderClassName> is the full Java class name of the content holder; for example, wt.doc.WTDocument, wt.epm.EPMDocument, wt.change2.WTChangeOrder2

The Windchill Bulk Migrator Extraction Utilities only extract the metadata of the master copies of file content residing in External File Vaults. Prior to Windchill Bulk Migrator 2.2, the extractor would not extract any files from Windchill if some of them were in a BLOB or cache vault. Starting in Windchill Bulk Migrator 2.2, the extraction utility allows users to extract file content that is in external file vaults. This is allowed even if some content is BLOB content or content in the cache vault. The behavior is governed by the value of data.extract.verifyContentInVault property in extract.xconf file.

When the property data.extract.verifyContentInVault is set to true (the default value), the Windchill Bulk Migrator Extraction Utilities will not extract any file content if any content is in a BLOB or cache vault. The Extractor will create reports in the Windchill Bulk Migrator extractor working folder.

If the property data.extract.verifyContentInVault is set to false, the extraction utility will do the extraction of the contents in the external vault and mark the BLOB contents and cache vault contents as a failure. The content failure report includes these failures. If the user re-vaulted the files in the BLOB to external file vaults or synced the files in the cache vaults later, subsequent extraction is able to extract those file contents which were marked as a failure in the previous run.

See the section Enterprise Administration in the Windchill Help Center for instructions on revaulting the files.

Note

The Windchill Bulk Migrator extraction utilities do not copy source Windchill file content to the staging area, and expect the user to use the enhanced content migration option that does not need the files in the staging area in the later WBM loading phase. See [Using the Enhanced Content Migration Option on page 241](#) for more information.

- To extract the URL content:

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.extractor.
client.WindchillDataExtractor -o
<ContentHolderClassName> -urlContent -u <wtuser> -p
<wtpassword> -urlContent
```

In this command, <ContentHolderClassName> is the full JAVA class name of the content holder; for example, wt.doc.WTDocument.

- To extract the family table information:

 **Note**

The family table files are also extracted when wt.epm.EPMDocument is provided as the content holder class name.

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.extractor.
client.WindchillDataExtractor -ft -u <UserName> -p
<Password>
```

- To extract the asStored information

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.extractor.
client.WindchillDataExtractor -assstored -u <UserName>
-p <Password>
```

- To extract Workflow information

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.extractor.
client.WindchillDataExtractor -o <WindchillClassName>
-wf -u <UserName> -p <Password>
```

<WindchillClassName> is the fully qualified Windchill class name of the Primary Business Object (PBO).

Note

All OPEN_RUNNING workflows can be suspended at the end of successful a workflow extraction. This can be done by adding the below property in the extract.properties file:

```
data.extract.workflow.suspendProcess=true
```

By default, this value is set to false.

All suspended workflows due to extraction can be resumed using the following command:

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.extractor.
client.WfProcessResumeHandler
```

- Fault Tolerance and Resume Extraction

All of the previous extraction commands can be run repetitively. The Windchill Bulk Migrator extraction utilities record the extracted counts in the staging database and the next run will continue from where the previous run had left off. Some of the object records may fail to get extracted during one extraction command run. The Windchill Bulk Migrator extraction utilities record the failure and the failure reason. The reason will be shown by running a status report. You can fix the issue that caused the failure. When the extraction command for the same type is run again, the utility will attempt to extract the failed objects. It should succeed if the issue that caused the failure is eliminated.

- Attribute Extraction

By default, the Windchill Bulk Migrator extracts all attributes and classification attributes, even if they are not included in the classMap.csv. For non-classification attributes, the source attributes are mapped to the target attributes in the WBMTTypeDefMapping.xml file. The WBMClassificationMapping.xml file maps the source classification attributes to the targets.

The Windchill Bulk Migrator extractors support extraction from multi-value attributes for all attribute data types that the Windchill Bulk Migrator supports.

The following advanced features are supported:

- Extract a subset of attributes—The source attribute will not be extracted if the `toBeMigrated` property is set to `False` in the `WBMTypedefMapping.xml` or `WBMClassificationMapp.xml` file.
- Extract source attributes to target modelled properties—You must map the source attribute to a staging column of the main staging table for the object type in the `classMap.csv` file.

For example, to extract the `myContractNumber` soft attribute of a part to the `contractNumber` modelled property of the part, you must modify the following entry of `wt.part.WTPart` mapping in the `classMap.csv` file:

```
COL_TYPE:NORMAL,contractNumber,CONTRACTNUMBER  
to  
COL_TYPE:NORMAL,myContractNumber,CONTRACTNUMBER
```

Note

The `toBeMigrated` property should remain set to `True` for the `myContractNumber` entry in the `WBMTypedefMapping.xml` file.

- Extract source modelled properties to target attributes—for example, if you want to migrate the modelled part attribute `contractNumber` to the target soft attribute `SomeTargetIBANName`, you must manually edit the `COL_TYPE:NORMAL,contractNumber,CONTRACTNUMBER` entry of the part mapping in `classMap.csv` file to:

```
COL_TYPE:NORMAL,contractNumber,WBM.TO_SOFT_ATTR
```

In the above, `WBM.TO_SOFT_ATTR` is the keyword.

In the `WBMTypedefMapping.xml` file, you must manually add the mapping for `contractNumber` for `wt.part.WTPart` type and its subtypes:

```
<Type srcTypeName="wt.part.WTPart" targetTypeName="wt.part.WTPart"]>  
    <Attribute srcAttrName="contractNumber" srcAttrGlobalDefName="contractNumber"  
    targetAttrName="SomeTargetIBANName" toBeMigrated="true" />  
  
<Type srcTypeName="wt.part.WTPart|mySubType" targetTypeName="wt.part.WTPart">  
    <Attribute srcAttrName="contractNumber" srcAttrGlobalDefName="contractNumber"
```

```
targetAttrName="SomeOtherTargetIBAName" toBeMigrated="true" />
```

For more information about the `classMap.csv` file, see [Understanding the classMap.csv File on page 228](#).

- Extraction of Iterations from Projects

The Windchill Bulk Migrator will not extract one-off iterations of an item from a project. One-off iterations mean the iterations in a project which is checked out to the project from Windchill and iterated in the project.

- Working Copy Extraction

The Windchill Bulk Migrator does not extract an iteration in a Working Copy state. However, by default, it does extract an iteration in a Checked Out state. Since incremental extraction supports extraction of iterations, there is a data. `extract.FailIfCheckedOut` property in the `extract.xconf` file that, if set to true, allows you to prevent extraction of a master with an iteration in Checked Out or Working Copy state. After the iteration is checked in, the whole item could be extracted by running the Windchill Bulk Migrator Extractor for the given type again.

- Incremental Extraction

Incremental extraction is supported in the following context beginning with Windchill Bulk Migrator 2.2:

1. Any new iteration, new objects, or new links created in the source Windchill system after the previous extraction will be extracted in the subsequent incremental extraction of the same object type.
2. Any deletion that happened in the source Windchill system after the previous extraction will not be reflected in the staging database in the subsequent incremental extraction.
3. Any in-place update (for example, `WTPartMaster` attribute modification) in the source Windchill system that happened after the previous extraction will not be reflected in the subsequent incremental extraction.

The extraction commands for the first extraction and any subsequent incremental extraction are the same.

- Appender Mechanism

The Windchill Bulk Migrator extractor provides an appender mechanism so you can plug in additional extraction code for any specific Windchill class.

The appender class needs to implement the interface

`com.ptc.windchill.migration.wbm.wc2wc.extractor.ap`
`pender.WCOBJECTExtractorAppender` and the class has to be registered in `extract.xconf`. The `xconfmanager` utility needs to be run to populate the `extract.xconf` change to

<Windchill> /codebase/com/ptc/windchill/migration/wbm/wc2wc/extractor/extract.properties.

The out-of-the-box appender class,

com.ptc.windchill.migration.wbm.wc2wc.extractor.appenders.PartUsesOccurrenceAppender, provides the extraction of the rotation and translation matrix that is difficult to extract through the general extraction introspection framework. The following is an example of the class file:

```
package com.ptc.windchill.migration.wbm.wc2wc.extractor.appenders;

import javax.vecmath.Matrix4d;
import wt.part.PartUsesOccurrence;
import com.ptc.wbm.extractor.util.StringToObjectMap;
import com.ptc.windchill.migration.wbm.wc2wc.extractor.ExtractedData;
import com.ptc.windchill.migration.wbm.wc2wc.extractor.ExtractionContext;
import com.ptc.windchill.migration.wbm.wc2wc.util.WBMEException;
import com.ptc.windchill.migration.wbm.wc2wc.mapping.server.MappingHandler;
public class PartUsesOccurrenceAppender implements WCObjectExtractorAppender
{
    private static String amountAttr = "quantity.amount";
    private static double defaultAmount = 1;
    private static String unitAttr = "quantity.unit";
    private static String defaultUnit = "ea";
    public ExtractedData getObjectAttributes(Object a_object,ExtractedData a_objectData
    MappingHandler mappingHandler,
    ExtractionContext cache) throws WBMEException
    {
        String attNamePrefix = "PartUsesOccurrence";
        StringToObjectMap attributesData = a_objectData.getAttributes();
        PartUsesOccurrence puOccurrence = (PartUsesOccurrence)a_object;
        Matrix4d matrix4d = puOccurrence.toMatrix4d();

        double rotation1_0 = matrix4d.getElement(0, 0);
        attributesData.put(attNamePrefix + ".rotation1_0", rotation1_0);

        double rotation1_1 = matrix4d.getElement(0, 1);
        attributesData.put(attNamePrefix + ".rotation1_1", rotation1_1);

        double rotation1_2 = matrix4d.getElement(0, 2);
        attributesData.put(attNamePrefix + ".rotation1_2", rotation1_2);

        double rotation2_0 = matrix4d.getElement(1, 0);
```

```

        attributesData.put(attNamePrefix + ".rotation2_0", rotation2_0);

        double rotation2_1 = matrix4d.getElement(1, 1);
        attributesData.put(attNamePrefix + ".rotation2_1", rotation2_1);

        double rotation2_2 = matrix4d.getElement(1, 2);
        attributesData.put(attNamePrefix + ".rotation2_2", rotation2_2);

        double rotation3_0 = matrix4d.getElement(2, 0);
        attributesData.put(attNamePrefix + ".rotation3_0", rotation3_0);

        double rotation3_1 = matrix4d.getElement(2, 1);
        attributesData.put(attNamePrefix + ".rotation3_1", rotation3_1);

        double rotation3_2 = matrix4d.getElement(2, 2);
        attributesData.put(attNamePrefix + ".rotation3_2", rotation3_2);

        double translation_0 = matrix4d.getElement(3, 0);
        attributesData.put(attNamePrefix + ".translation_0", translation_0);

        double translation_1 = matrix4d.getElement(3, 1);
        attributesData.put(attNamePrefix + ".translation_1", translation_1);

        double translation_2 = matrix4d.getElement(3, 2);
        attributesData.put(attNamePrefix + ".translation_2", translation_2);

        // set the quantity attributes if not in the source
        if (!attributesData.containsKey(amountAttr)) {
            attributesData.put(amountAttr, defaultAmount);
        }
        if( !attributesData.containsKey(unitAttr)) {
            attributesData.put(unitAttr, defaultUnit);
        }
        return a_objectData;
    }

    public StringToObjectMap getMasterAttributes(Object a_object, StringToObjectMap
a_objectData)
throws WBMEException {
    // TODO Auto-generated method stub
    return null;
}

    public String needsToBeFiltered(Object a_object) throws WBMEException {
    // TODO Auto-generated method stub
}

```

```
        return null;
    }

}
```

Data Not Extracted

The following extractions are not supported:

- Extraction of Iterations from Projects

Windchill Bulk Migrator does not extract one-off iterations of an object from a project. A one-off iteration is an iteration in a project that is checked out to the project from Windchill and iterated in the project.

- Working Copy Extraction

Windchill Bulk Migrator does not extract an iteration in a working copy state. However, by default, it does extract an iteration in a Checked Out state. Since incremental extraction supports extraction of iterations, the `data.extract.FailIfCheckedOut` property in the `extract.xconf`, if set to true, allows you to prevent extraction of a master with an iteration in Checked Out or Working Copy state. After the iteration is checked in, the whole object can be extracted by running the Windchill Bulk Migrator Extractor for the given type again.

Windchill Bulk Migrator Extraction Utilities Status Reports

The Windchill Bulk Migrator extraction utilities provide a reporting mechanism for users to learn the status of extraction activities. The following list shows the command to execute from a Windchill shell to receive a status report for various extraction activities:

- All object types in the `classMap.csv`

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.report.Report
Generator -all
```

- A specific type

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.report.Report
Generator -o <objectType>
```

For example,

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.report.Report
Generator -o wt.part.WTPart
```

- Family tables

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.report.Report
Generator -ft
```

- asStored Configuration

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.report.Report
Generator -as
```

- File content of all supported content holder types

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.report.Report
Generator -co
```

- URL content for all supported content holder types

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.report.Report
Generator -ur
```

- Workflow

```
windchill
com.ptc.windchill.migration.wbm.wc2wc.report.Report
Generator -wf
```

The reports are generated under <WBMExtractWorkingDir>/Report-<TimeStamp> when –all is specified and <WBMExtractWorkingDir>/Report-<TimeStamp>/<objectType> for other options.

Three formats of the reported are generated:

- WBMExtractorStatus.html
- WBMExtractorStatus.xls
- WBMExtractorStatus.xml

These files contain the following information:

Column	Description
TotalCount	The number of objects for the specific object type in the source Windchill system.
Processed	Number of source objects processed for the specific object type in the extraction process.

Column	Description
	Processed should always be equal to the sum of Succeeded, Failed and Filtered.
Succeeded	The number of objects for the specific object type that Windchill Bulk Migrator extraction utilities have extracted.
Failed	The number of objects for the specific object type that the Windchill Bulk Migrator extraction utilities fail to extract. Click the failed number (if > 0) to see the detailed failure report.
Filtered	The number of objects for the specific object type that have been filtered by Windchill Bulk Migrator extraction utilities. The following objects are filtered: one-off objects, working copies, objects in personal cabinet and any versioned object that are filtered out by implementing additional PTC or custom-developed appenders.
One-Offs	The number of filtered One-Off iterations is shown here. The number is included in the overall Filtered. The One-Offs for Family Table, As Stored, and Contents are 0 in the report as the One-Offs have not been in the totalCount for those object types.

One-to-Many Extraction

The Windchill Bulk Migrator provides the ability to create multiple objects in the staging database that correspond to a single object in the source Windchill instance. The additional object classes should be listed as a secondary class using the keyword SECONDARY_CLASSES in the mapping entries of the main class in the `classMap.csv` file. The additional classes should also have their own regular mapping entries in the `classMap.csv` file as well. For more information, refer to class `wt.workflow.engine.WfProcess` and its secondary classes `wt.workflow.work.WfAssignedActivity`.

You must write a `CustomExtractedDataCreator.java` class that inherits from
`com.ptc.windchill.migration.wbm.wc2wc.extractor.appender.CustomExtractedDataCreator`.

You can refer to the `com.ptc.windchill.migration.wbm.wc2wc.extractor.append.wf.CustomWorkFlowDataExtractor.java` file that extracts the `wt.workflow.work.WfAssignedActivity` and other secondary classes in the same extraction session that extracts `wt.workflow.engine.WfProcess`.

The `CustomExtractedDataCreator.java` that you created must be registered in the `extract.xconf` file. You can refer to the following sample registration in `extract.xconf`:

```
<Property  
name="data.extract.customDataCreator.wt.workflow.engine.WfProcess"  
default="com.ptc.windchill.migration.wbm.wc2wc.extractor.append.  
wf.CustomWorkFlowDataExtractor"  
targetFile="codebase/com/ptc/windchill/migration/wbm/wc2wc/  
extractor/extract.properties"/>
```

Extraction Sequence

There is no required extraction sequence of object types. Data dictionary generation and mapping file generation must be completed before extraction.

8

Filter Options

Installation	86
Starting the Filtered Extraction GUI.....	86
Filtered Extraction GUI Overview.....	86
Windchill Bulk Migrator Enhanced Filtered Extraction Utilities Overview.....	90
Running the Windchill Bulk Migrator Filtered Extraction Utilities	91
Enhanced Filtering Commands	91
Extraction Mode	92
Filtered Extraction Report	92
Options and Variants	93

This chapter contains information on filtered extraction data using the Windchill Bulk Migrator.

Installation

The filter extraction user interface is installed along with the WBM Extractor. Filtered option used in filtered extraction should be created and saved from user interface.

It has a front end developed in NodeJS which needs to be used for creating the filter option for running filtered extraction.

Starting the Filtered Extraction GUI

In windchill shell run the following command to start the node server:

```
startFilterOption
```

Note

The front end can be accessed through URL <http://localhost:8080> and supports only below browsers:

- Google Chrome
 - Mozilla Firefox
-

Filtered Extraction GUI Overview

The filtered extractor GUI salient features are listed below:

- GUI based filter option creation.
- Save/Retrieve filter option to/from database.
- Download Filter Option in XML format for debugging purpose.
- Seed Class name and Lifecycle State selection guidance.

The below GUI can be used to generate the Filter Options criteria:

 Filter Options

21 - Options ▾

Seed - Class

Seed 1

Name * wt.part.WTPart

Iterations Latest ▾

Lifecycle State In Work

Collect Objects From Type

Container Folder Object

Container Type Container 1

Product ▾

Container Name * Container 1

Container Organization * Organization 1

EPM Dependency Type

All ▾

As Stored Type

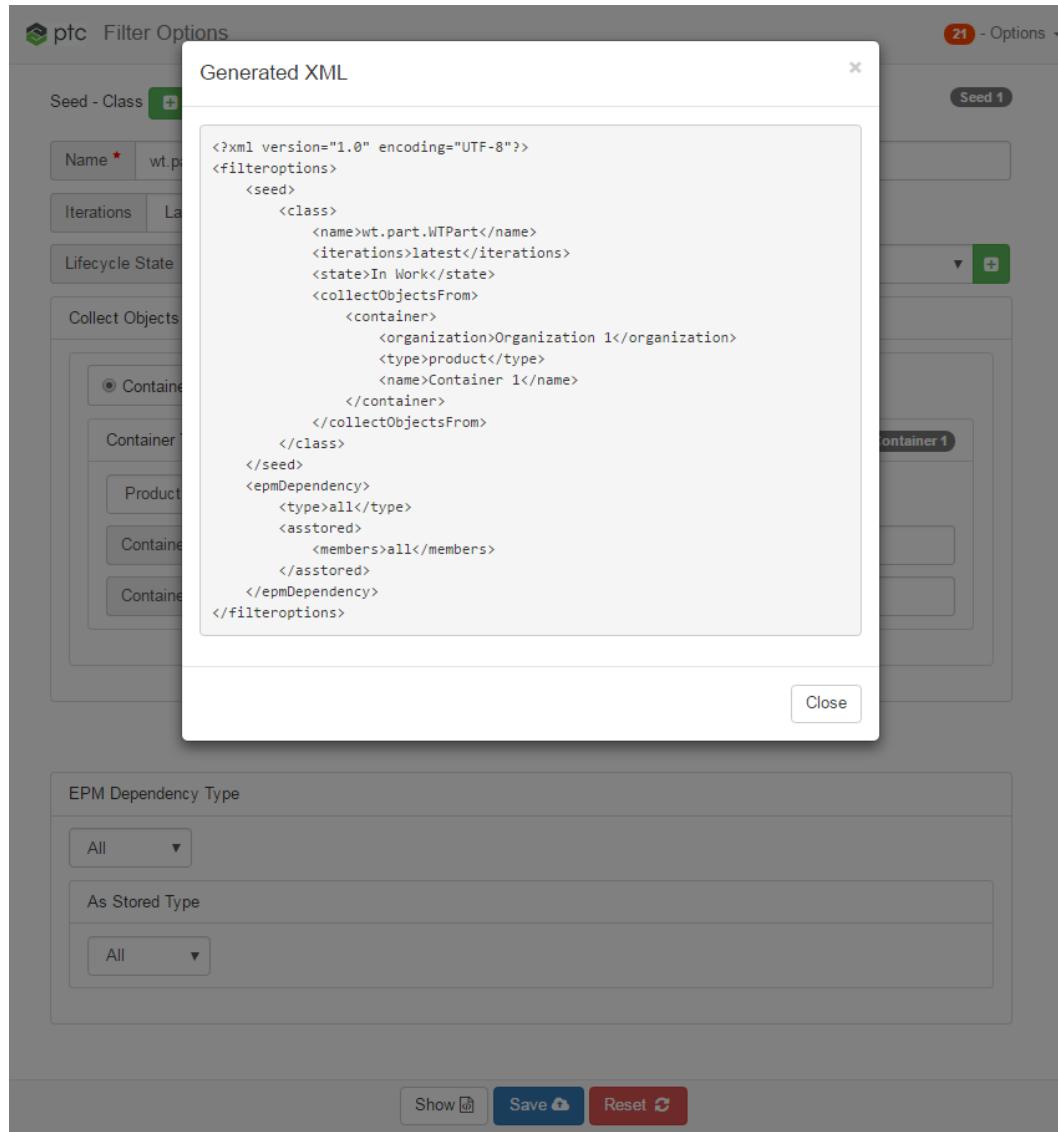
All ▾

Show  Save  Reset 

Show — Display the Filter Option with parameters from the current page.

Save — Save the Filter Option to Staging Database.

Reset — Reset the form.



The saved filter options are displayed in descending order of creation and it is downloadable.

 ptc Filter Options

21 - Options ▾

Created Date	
2017-04-13 13:37:05.358 GMT	
2017-04-13 13:27:10.267 GMT	
2017-04-12 09:13:04.451 GMT	
2017-04-12 09:02:59.183 GMT	
2017-04-12 08:54:44.056 GMT	
2017-04-12 08:53:21.446 GMT	
2017-04-12 08:52:25.46 GMT	
2017-04-12 08:38:10.191 GMT	
2017-04-12 08:34:39.263 GMT	
2017-04-12 08:34:07.504 GMT	
2017-04-12 08:33:50.329 GMT	
2017-03-27 12:15:21.484 GMT	
2017-03-24 12:13:30.351 GMT	
2017-03-24 09:59:53.001 GMT	
2017-03-21 09:48:02.955 GMT	
2017-03-21 09:44:28.729 GMT	
2017-03-13 12:54:41.325 GMT	
2017-03-07 09:06:03.393 GMT	
2017-03-06 09:25:43.32 GMT	
2017-02-24 09:18:25.414 GMT	
2017-02-17 13:07:49.78 GMT	

Seed - Class 

Name * wt.part.WTPart

Iterations Latest ▾

Lifecycle State In Work

Collect Objects From Type

Container Folder Object 

Container Type

Product ▾

Container Name * Container 1

Container Organization * Organization 1

EPM Dependency Type

All ▾

As Stored Type

All ▾

Show  Save  Reset 

Note

The latest file (latest entry of the Filter Option table) will be used when running filtered extraction.

Windchill Bulk Migrator Enhanced Filtered Extraction Utilities Overview

The Windchill Bulk Migrator Extractor provides extraction mechanism with filtering capabilities. The filtering mechanism is based on a configuration file in which you can specify the filter criteria on the various seed types, selection criteria, and dependency gathering options. Based on the specified filter criteria, the utility selects the seeds and transverses the dependencies that include various links, dependents, related complete family tables, and relevant files. You can specify the filter criteria using the Filtered Extraction GUI.

After successfully installing the Windchill Bulk Migrator extraction utilities and starting the source Windchill system, the high-level steps to extract data are as follows:

1. Generate Extraction Mapping.
2. Generate data dictionary files.
3. Create and modify the administrative data mapping files.
4. Create the Filter Options file through GUI as per requirement.
5. Extract data for object types.

Additional information that will be useful for filtered extraction:

WBMEExtractWorkingDir

- `reset_filtering_schema.sql` If a user drops the `tf$` table from staging database and runs fresh filtered extraction, user has to log in as staging DB and run the above SQL file.
- `GenFilterDD.xml` Contains dictionaries of the classes, links reachable from the given seed class. This also includes the temporary table - `tf$` table associated to the seed class.

`<Windchill>\codebase\com\ptc\windchill\migration\wbm\wc2wc\extractor\filter`

- `foreignKeyReferences.properties`—Contains the bridging information between the concrete class and extended class. PTC recommends that content of these properties file should not be changed.
- Windchill Bulk Migrator extraction logging that includes a client log file, server log file and server error log file. For more information on logging, see Configuring Log File Information on page 64 of Windchill Bulk Migrator Installation and Usage Guide.

Running the Windchill Bulk Migrator Filtered Extraction Utilities

Prerequisites:

- Run the
`com.ptc.windchill.migration.wbm.wc2wc.util.AttDefAnalyzerClient -a` command once for the unchanged source Windchill. You must run the command again if the source system is changed.
- Run the
`com.ptc.windchill.migration.wbm.wc2wc.mapping.client.WBMMappingClient` command once and edit, if needed for the unchanged source Windchill. You must run the command again if the source system is changed.
- Create the Filter Option in database through the GUI.

Enhanced Filtering Commands

There is one command to run filtering, extraction, and reporting: <WT_SHELL>windchill
`com.ptc.windchill.migration.wbm.wc2wc.extractor.filter.WindchillFilteredExtractor -u <wcUser> -p <wcPassword>`.

1. To find other options for this command, use `-h` with the command. Once you run the command with the default option, it completes the steps listed below: Generates the `<ExtractWrkDir>\Filtering\GenFilterDD.xml` file that indicates the network of classes relating to the seed class. All classes are extracted. For example, if the seed class is `wt.epm.EPMDocument`, the content of `wt.epm.EPMDocument`, family tables, `wt.epm.structure`, `EPMMemberLink`, `wt.epm.build.EPMBuildRule`, `wt.part.WTPart`, and `wt.part.WTPartUsageLink` will be extracted as well.
2. Creates filtering tables like `TF$<shortClassName>` in the staging table.
3. Generates the `<ExtractWrkDir>\Filtering\reset_filtering_schema.sql` file to drop the tables created in step 2. The script should only be used after calling the `drop_staging_schema.sql` file to clean the filtering tables.
4. Populates the filtering tables with `ida2a2` of the objects you want to extract.
5. Extracts the objects for the `ida2a2s` in the filtering table.
6. Generates a report.

The command also supports stop, resume, and incremental extraction. Use `-h` to find all other options for the command.

Extraction Mode

Normal extraction or Filtered Extraction is supported at any time for extraction.

If the system is in Filtered Extraction Mode following objects are supported by using Normal Extraction command.

- Folder
- PDMLinkProduct
- LibraryContainer
- OrgContainer
- WTOrganization

Note

If the system is in Filtered Extraction Mode and you try to run normal extraction for other than those listed above, the following error is displayed:

Error message: - Extraction initiated on : YYYY-MM-DD HH:mm:ss.
XXX | Filter Mode : true.

Filtered Extraction Report

The filtered extraction report is stored in the WBM Extract Working Directory.

Run the Filtered extraction command to update the data under **Staged** column.

WBM Filter Extraction Status Report

Object Type	Total Count	Staged	Failed	Pending
wt.part.WTPartUsageLink	43	43	0	0
wt.viewmarkup.DerivedImage	63	63	0	0
wt.lifecycle.ObjectHistory	430	430	0	0

Data will be seen only under **Pending** if the filtered extraction will be run with **-f** Mode.

WBM Filter Extraction Status Report

Object Type	Total Count	Staged	Failed	Pending
com.ptc.windchill.suma.axl.AXLEntry	6	0	0	6
com.ptc.windchill.suma.part.ManufacturerPart	2	0	0	2
com.ptc.windchill.suma.axlrule.AVLPolicyRule	1	0	0	1

Note

: In case of data failure the **Failed** column gets updated.

Options and Variants

Companies use a variety of business strategies to deliver products with multiple configurations, that share a base model that can be produced with optional components and features. To understand how your business processes can be implemented using Windchill, you need to define the preferred logical workflow that enables you to produce configurable products, and then align it to Windchill capabilities that support variability.

For more information on Options and Variants, See [Windchill Options and Variants Capabilities](#).

Creating Options and Variants Objects

1. Create the required variant WTPart structure with the associated CAD Document.
2. Create the configurable product structure in Windchill. For more information, see [Setting up Configurable Product Structures](#).
3. Add variants to the configurable modules.
4. Define Options and Choices. For more information, see [Options and Choices](#).
5. Define Exclude and Include rules. For more information, see [Defining Global Rules for Options and Choices](#).
6. Create and assign the option set.

-
7. Assign choices to the module variants from Windchill.
 8. Create the variant specification in Windchill and request deliverables. For more information, see [Variant Specification Step: Defining a Variant Specification](#).

Filtered Extraction for ATO Objects

Filtered extraction of Options and Variants objects can be performed in the same manner as other Windchill objects.

To extract an object and its related objects, upload the object class as seed class in filter GUI and then perform the filtered extraction.

9

Normalizing Data

Normalizing Data Overview	96
Using the wbmldr Command	97
Normalization Sequence	109

Normalizing Data Overview

For many object types, the Windchill Bulk Migrator enforces a data preparation process that includes data integrity checks and data processing to enable eventual consumption by the object loaders. The term used for the data preparation in this document is Normalization. Normalization encompasses all of the internal data preparation processes that are enabled by the Windchill Bulk Migrator to put data in a state that is ready to be used by or sent to the object loaders.

Note

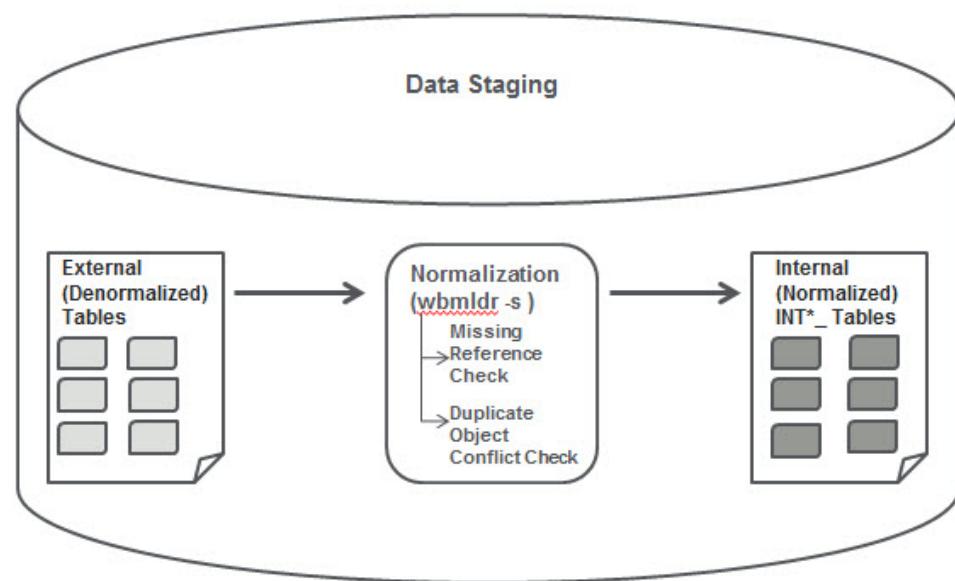
The Windchill Bulk Migrator does not require normalization for all object types that are supported out-of-the-box. The objects that do not require normalization include:

- Administrative data object types:
 - Folder
 - LibraryContainer
 - OrgContainer
 - WTOrganization
 - PDMLinkProduct
 - Legacy Loadable objects:
 - CONTENTMETADATA
 - DERIVEDIMAGE
 - SHAREDCONTAINERMAP
 - WFPPROCESS
 - REGISTEREDOPTIONSETLINK
 - NAVIGATIONCRITERIA
-

The Windchill Bulk Migrator staging schema consists of tables that represent the normalized data condition as well as tables that represent the de-normalized condition. The de-normalized tables usually align with the corresponding Windchill object class name. The normalized tables have the same root name as the de-normalized tables with the additional identification of an INT_* prefix. The de-normalized tables are considered tables that are only used by the Windchill Bulk Migrator infrastructure. These tables should never be directly modified or altered.

For Windchill-to-Windchill migrations, the extractors will extract, map, and transform data directly to the de-normalized tables according to rules defined in the `classMap.csv` file and the user defined mapping XML. For non-Windchill migrations, you must first populate the de-normalized tables. For more information, see [Migrating Non-Windchill Data into Windchill on page 142](#).

The loading process begins after the de-normalized tables are populated with the object metadata that corresponds with the Windchill object types that are to be loaded. Data normalization moves object metadata from the de-normalized tables to the normalized tables.



The `wbmldr` command utility facilitates the normalization process. For more information, see [Using the `wbmldr` Command on page 97](#).

Note

You must normalize master objects and then the iterations.

Using the `wbmldr` Command

The `wbmldr` command is used to facilitate and execute the data normalization process. It can also be used to perform other actions such as populating the de-normalized staging tables (for non-Windchill migrations) as well as data cleansing actions in the staging tables.

Wbmldr Configurations

The `wbmldr` command has various configuration options. You can set the configuration options using a property file located at:

`Windchill/codebase/com/ptc/wbm/wbmldr/core/meta/wbmldr.config`

The following table describes key options to consider before and while using the command:

Property	Description
<code>REPORT.QUANTITY.VALUE</code>	<p>The percentage of data reported to <code>wbmldr</code> conflict files. This is located in the <code>wbmldr</code> logs directory.</p> <p>Set this property to a higher value if you want to display a higher percentage of conflicts in the log files.</p> <p>By default, this property is set to 10.</p> <p> Note</p> <p>If the report quantity threshold has been exceeded, the log files will indicate that not all of the conflicting data is shown.</p>
<code>WBMLDR.CACHE.COLUMN.DELIMITER</code>	<p>Specifies the CSV delimiter when working with CSV files.</p> <p>By default, this property is set to (pipe character).</p>

Property	Description
WBMLDR.CACHE.OPTIONALLY.ENCLOSEDBY	Specifies the enclosed by character when working with CSV files. By default, this property is set to " (quotes character).
WBMLDR.STAGE.WC2WC	For Windchill-to-Winchill migrations, set this property to true. For non-Windchill migrations, set this property to false. If set to false, WBMSOURCEIDENTIFIER will become optional. However, the value of MASTERWBMSOURCEIDENTIFIER on the iterated object must match the value of WBMSOURCEIDENTIFIER on the master table whether it is an actual value or null. If WBMLDR.STAGE.WC2WC is set to false, Windchill Bulk Migrator will use a different query that will join all the identity columns (MASTERORGANIZATION_NAME, MASTEROBJECTNUMBER which are not needed when WBMLDR.STAGE.WC2WC is set to true) between the master and the iterated object table. Therefore those tables should be populated with correct value or the join will fail.

The logs and reports (if any) are located under Windchill/wbm/wbmldr/<timestamp folder>/<logs or reports>. For example, F:\ptc\Windchill\wbm\wbmldr\<date>\<timestamp>\logs on Windows and /disk1/ptc/Windchill/wbm/wbmldr/\<date>\<timestamp>/logs on UNIX.

Executing the wbmldr Command

The wbmldr command should be executed from a Windchill shell with Windchill/bin as the working directory using the below platform-specific command:

- Windows: wbmldr.bat <command flag>
- UNIX:wbmldr.sh <command flag>

More help on wbmldr can be obtained by executing it with a -h or -help option.

Note

If out of memory exception occurs while executing the wbmldr command, edit the wbmldr.bat or wbmldr.sh file, as applicable, to adjust the Java memory arguments.

Wbmldr Command Line Options

Command Flag	Description	Usage
-h -help	Displays information about using the wbmldr command.	-h
-c -cache	Populates the de-normalized (cache) staging tables by reading the CSV files.  Note This command flag is only supported on an Oracle staging database. SQL Server staging databases must use the BCP (Bulk Copy Program) utility to load CSV data into the cache tables.	-c <path to folder containing the CSV files> Where the name of the CSV files will be <OBJECTTYPE>_*.*.csv. For example, EPMDOCUMENT_1.csv will be associated with the EPMDOCUMENT de-normalized table.
-s -stage	Initiates the normalization process. This results in the transfer and staging of metadata to the internal INT* normalized tables. Once the data for a particular type is staged, the de-normalized (cached) data for that object type is truncated.  Note This command flag is supported for staging databases on both Oracle and SQL Server.	-s <All or OBJECTTYPE> For example, you could specify -s All or -s EPMDOCUMENT.

Command Flag	Description	Usage
<p>-r -remove</p>	<p>Removes data from staging tables.</p> <p>This command flag is commonly used for data cleansing, and results in the removal of specified data from both the normalized and de-normalized (cache) tables.</p> <p>Note This command flag is supported for staging databases on both Oracle and SQL Server.</p> <p>For more information, see Data Cleansing Using the wbmldr Command on page 102.</p>	<p>-r <All, OBJECTTYPE, or selective removal></p> <p>When using the -r or -remove option, the process is self-cascading. That is, all references to the data being removed are also deleted.</p> <p>For example, EPMMEMBERLINK -> wheel.prt A.1 <->car.asm, will automatically be removed when either EPMDOCUMENT wheel.prt A.1 or EPMDOCUMENT MASTER car.asm is removed.</p>
<p>-v</p>	<p>Spools information about specified objects to a CSV file. The output can be used to generate CSV files to help use in conjunction with the wbmldr -r command to remove data from the normalized tables, and for data cleansing.</p>	<p>wbmldr -v all—Creates all object CSVs from the normalized table.</p> <p>wbmldr -v <ObjectType>—Creates the specific object CSV from the normalized table.</p> <p>wbmldr -v all /d—Creates all object CSVs from de-normalized table.</p> <p>wbmldr -v <ObjectType> /d—Creates the</p>

Command Flag	Description	Usage
		specific object CSVs from the de-normalized table.

Data Cleansing Using the wbmldr Command

There are three scenarios where it may become necessary to remove data out of the normalized tables so you can perform data cleansing.

- If there are conflicts reported by the normalization (`wbmldr -s`) process
- If there are conflicts reported by the pre-load verification utility
- If there are other conflicts reported during loading

Data cleansing steps are unique depending on the conflict or issue. However, the general cleansing process fits into a migration as follows:

1. Remove the conflicting data from the normalized tables.
2. Cleanse the data. Cleansing can be performed on the source Windchill system, the user defined map files, or in external CSV files for a given object type.
3. Re-cache the data to the de-normalized tables.
4. Re-run the normalization process.
5. Load the data.

Note

Data cleansing operations should never occur in the normalized `INT_*` tables. The normalized tables are internal tables that should never be modified manually.

Types of Conflicts Reported by wbmldr

The following tables lists the types of conflicts that may be reported when executing the `wbmldr -s` command:

Conflict Type	Description
Not all references found conflicts for all other supported object types	<p>All conflicting data is reported to <OBJECTTYPE>.bad files in the reports location.</p> <p>This conflict indicates that there are missing objects in the staging database and the defined link is therefore invalid.</p> <p>In order to ensure data integrity, this conflict is an all or nothing situation. Even if the cause of a conflict is because of one cached row, none of the link data is staged to the normalized tables.</p> <p>However, the link data will continue to remain in the internal cache.</p> <p>For example, when staging EPMMEMBERLINK -> wheel.prt A.1 <-> car.asm, none of the cached EPMMEMBERLINK data is staged if either EPMDOCUMENT wheel.prt A.1 or EPMDOCUMENTMASTER car.asm was not found by the tool and subsequently staged to the normalized tables. The EPMMEMBERLINK data remains in the de-normalized cache tables.</p> <p>In the example, the data remains in the cache tables because one possible resolution is to locate the missing data and add it to the data that is to be migrated. In the situation where you add the missing data, you can simply re-run wbmldr normalization for the link object and normalization will succeed.</p>
Duplicate object conflicts	<p>If this conflict exists, all conflicting data will be reported to <objecttype>.dup file in the reports location. This includes duplicates within the current de-normalized (cache) table or duplicates</p>

Conflict Type	Description
	<p>from the current cache compared with any previously normalized (staged) data. The conflicts reported to <code><objecttype>.dup</code> during conflict checking are grouped into conflicting sets with precedence in a conflict set given to the rows deleted from the current cache.</p> <p>If duplicates are discovered, all conflicting duplicate data is deleted from the de-normalized tables.</p> <p>However, any previously staged normalized data is only reported to <code><objecttype>.dup</code> and is not deleted from the normalized tables. The conflict checking is performed based on the following criteria.</p> <p>For all non-project contexts:</p> <ul style="list-style-type: none"> • Objects cannot have duplicate CADNAMES irrespective of any other master properties. • Objects cannot have duplicate OBJECT_NUMBER within an organization irrespective of any other master properties. <p>For all project contexts:</p> <ul style="list-style-type: none"> • Objects cannot have duplicate CADNAMES irrespective of any other master properties within the same project. However this is allowed across multiple projects. • Objects cannot have duplicate OBJECT_NUMBER irrespective of any other master properties within the same project. However this is allowed across multiple projects. • Duplicate iterations within a master. For ex: Master 'a.prt' cannot have two iterations of A.1.

Removing Data from the Normalized Tables for Data Cleansing

To cleanse data reported as not valid by the any of the normalization (wbmldr -s) process, the pre-load verification utility, or from other loading errors, you can use the removal from the normalized tables option with the wbmldr -r command flag.

The process of removing data from the normalized tables could be different depending upon if the staging database is either Oracle or SQLServer.

The following table describes in more detail the wbmldr <-r | -remove><ALL, ObjectType, Selective Removal> command line options.

Option	Description
ALL	<p>Wipes out all non-migrated data both from the cached and/or staged data for all the supported object types.</p> <p>This option is fully supported for staging databases on Oracle and SQLServer.</p>
ObjectType	<p>Wipes out all non-migrated data both from the cached or staged data for this supported object type and cascades to other supported object types if referenced.</p> <p>This option is fully supported for staging database on Oracle and SQLServer.</p>
Selective Removal	<p>This option behaves differently depending on whether you are using Oracle or SQL Server.</p> <p>Oracle</p> <ul style="list-style-type: none"> • Requires path to a folder containing CSV with data to be removed as an additional argument to <code>-r</code> or <code>-remove</code>. • The CSV files need to end with <code><OBJECTTYPE>.csv</code> for them to be associated with the correct object type. • For ex: <code>foo_EPMDOCUMENT.csv</code> and <code>barEPMMEMBERLINK.csv</code> will automatically be associated with EPMDOCUMENT and EPMMEMBERLINK object types respectively. • Multiple files containing data belonging to the same object type will be combined and removed in a single process.

Option	Description
	<ul style="list-style-type: none"> All tokens of a row in a CSV file should exactly match what was cached for removing the data successfully. <p>SQL Server</p> <ul style="list-style-type: none"> Requires NO ADDITIONAL ARGUMENT to <code>-r</code> or <code>-remove</code>. The CSV files needs to be loaded manually using 'bcp' into RMV_<OBJECTTYPE> tables. For example, in order to selectively remove for EPMDOCUMENT object type, load the CSV with tokens exactly matching what was cached using 'bcp' into RMV_EPMDOCUMENT and then execute <code>wbmldr.[bat sh] -r</code> or <code>-remove</code> to complete the remove process.

Data Removal and Cleansing for Oracle Staging Databases

For normalization conflicts reported by the `wbmldr -s` process or the family table pre-verifier utility, it may be possible to use the output conflict reports that are spooled out for the user defined input file for the `wbmldr remove` command.

Data that is cleansed either in the source system, or in a CSV file should then be repopulated to the de-normalized (cache) tables. For Windchill-to-Windchill migrations this could involve re-extracting the data, or through manual healing and correction in a .csv file, and re-caching to the de-normalized tables using the `wbmldr -c` command.

For conflicts reported by the Windchill Bulk Migrator pre-load verification utility or for other conflicts reported during the loading process, use the following technique to remove conflict data from the normalized tables, if appropriate, based upon your desired corrective action.

- Spool out the results of a SQL select statement on the read-only `VEW <objecttype>` view staging tables for the records that fail pre-load validation. The output can be spooled to a CSV file using the `wbmldr -v` option that can then be used as input to the `wbmldr -r` command. For more information, see [Executing the wbmldr Command on page 99](#).
- Run the `wbmldr -r` command as described in the `wbmldr` command line options section.

Data Removal and Cleansing for SQL Server Staging Databases

For SQL Server, the `wbmldr -r` command flag operates using special internal tables that are intended to contain the data that you want to remove from the normalized tables. You must populate the internal tables manually with the information the `wbmldr -r` command is looking for. The internal tables are denoted in the staging schema with a `RMV_<objecttype>` naming convention. The overview of the process is as follows:

- Create a CSV file that contains the object information that you want to remove from the normalized tables.
- Use the SQL Server BCP command to load the CSV data into the `RMV_*` internal tables.
- Run the `wbmldr -r` command as described in the `wbmldr` command line options section for SQL Server selective removal.

For normalization conflicts reported by the `wbmldr -s` process or the family table pre-verifier utility, it may be possible to use the output conflict reports that are spooled out for the user-defined input file used to populate the `RMV_*` tables using the BCP command.

Data that is cleansed either on the source system, or in a CSV file should be repopulated to the de-normalized (cache) tables. For Windchill-to-Windchill migrations this may involve re-extracting the data, or through manual healing and correction in a CSV file, then re-caching to the de-normalized tables using the SQL Server BCP command.

For conflicts reported by the pre-load verification utility or for other conflicts reported during the loading process, use the following technique to remove conflict data from the normalized tables based on your desired corrective action:

- Spool out the results of a SQL select statement on the read-only `VIEW <objecttype>` view staging tables for the records that fail pre-load validation. The output can be spooled to a CSV file using the `wbmldr -v` option that can then be used as input for the SQL Server BCP command to populate the `RMV_*` tables. For more information, see [Executing the wbmldr Command on page 99](#).
- Run the `wbmldr -r` command as described in the `wbmldr` command line options section.

Normalization Sequence

Follow the normalization sequence by running the `wbmldr -h` command.

10

Pre-loading Validation

Pre-loading Validation Process	111
The Windchill Bulk Migrator Pre-Load Verification Utilities.....	111

Pre-loading Validation Process

Pre-load validation improves the quality of the data migration by providing you the opportunity to find and resolve errors before encountering them in the loading process. The Windchill Bulk Migrator provides validation capabilities to ensure that the staged data is ready to be loaded, as well as that the target system administrative data is available to enable the migration.

Validation utilities and data cleansing is an iterative process. That is, the validation checks should occur, and the log files should be analyzed for conflicts. Any conflicts found will result in corrective action, which can in turn be verified by the verification utilities.

The following validation tools are available to help with migration and are explained in this section:

- The Windchill Bulk Migrator Pre-Load Validation: An optional, but recommended tool to be run after data normalization but before data loading. This tool collects information about the target system, and analyzes the staged data looking for conflicts that could impact the migration
- Family Table Validation: A mandatory tool that is used to check for known issues which could potentially impact migration or upgrade events. This tool is also available on the source Windchill system via the Windchill Diagnostic Utility (WinDU) and can be run independently on the active source system. This tool is also mandatory for all CAD documents as well.

The Windchill Bulk Migrator Pre-Load Verification Utilities

The pre-load validation utility is installed with the Windchill Bulk Migrator loaders and is launched from a Windchill shell on the target system. The utility helps guide you through the various validation checks.

The utility performs the following types of validation tasks:

- Data Mapping: Validates that the Windchill-to-Windchill migration user defined data mapping XML files are valid. Ensures that the target Windchill system contains the values specified in the mapping files.
- Data Validation: Validates that the staged data conforms with the Windchill data model, administrative configurations, and has valid file content.

Note

Running the utility is optional for all data types except for workflow. If you are migrating workflows for a Windchill-to-Windchill migration, it is mandatory to run pre-load validation tasks on the staged and extracted workflow data. However, it is possible to make it mandatory for the checks to be executed and completed successfully for all object types before allowing data loading to occur. You can control this behavior with the following property in `wbm.properties`:

```
com.ptc.windchill.migration.wbm.preloadvalidator.en  
forceValidation=true
```

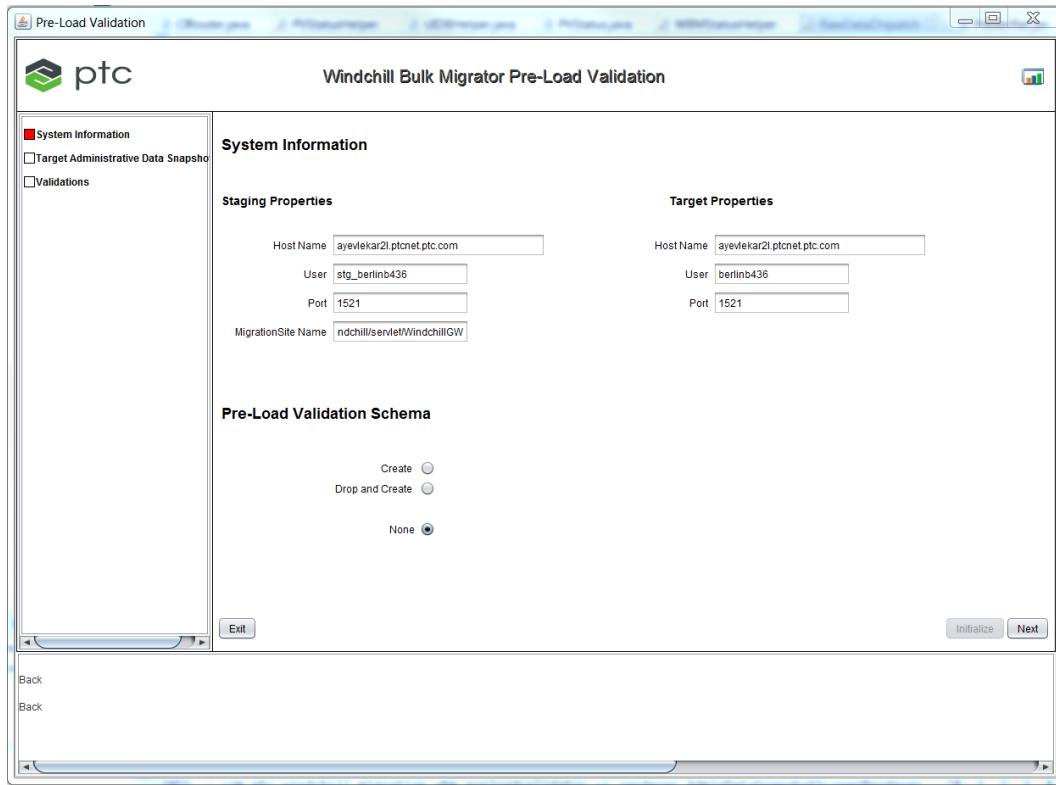
By default, this property is set to false (pre-load validation is optional).

Pre-requisites Before Running the Pre-Load Validation Utility

- The source and target Windchill systems are installed and available.
- The Windchill Bulk Migrator loaders are installed on the target Windchill system.
- Administrative data mapping has been completed and the mapping XML files are available.

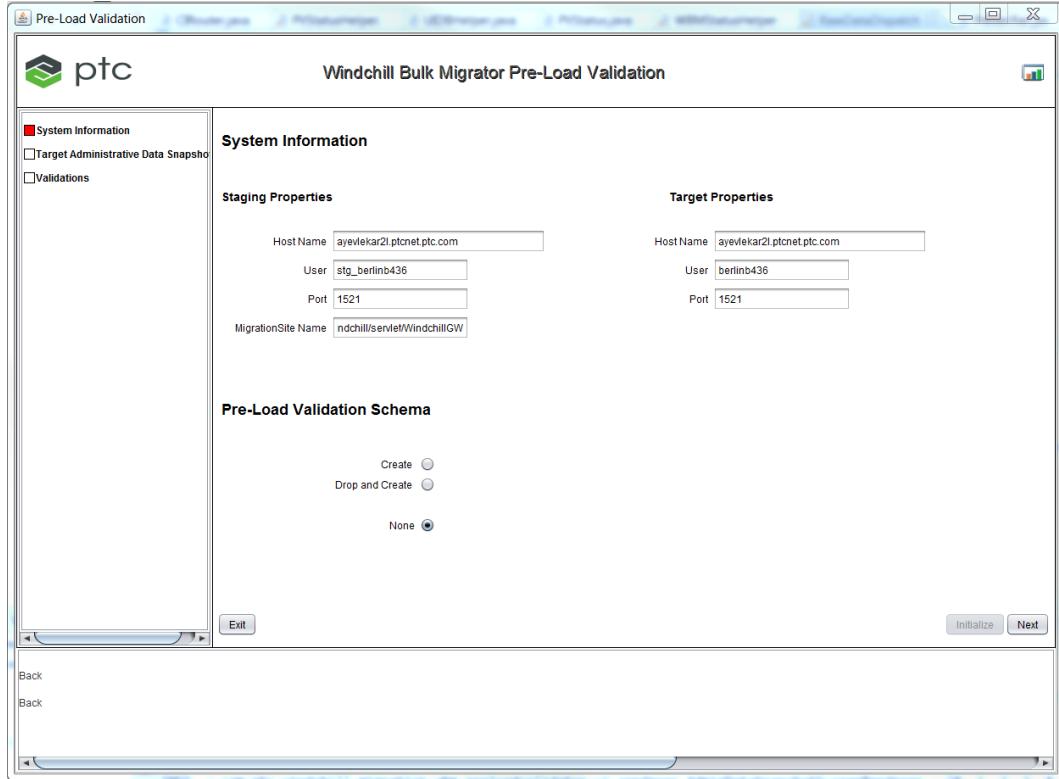
The pre-load validation utility is launched from a Windchill shell using the following command:

```
windchill  
com.ptc.windchill.migration.wbm.preloadvalidator.ui.Pre  
LoadValidationManager
```



Entering System Information and Generating the Pre-Load Validation Schema

When launched, the **System Information** page appears in the pre-load validation utility. The utility automatically populates the staging database properties and the target Windchill system **Host Name**, **User** and **Port**.



The verification tasks log, stores and maintains status information about the pass/fail results for the various object types. By design, this information works with the existing Windchill Bulk Migrator staging schema. However, the verification

utility implements and uses new schema that must first be initialized using the user interface. There are three options for the pre-load verification schema interaction:

Action	Description
Create	Create the pre-load verification schema. Use this option the first time you are using the verification tasks.
Drop and Create	Drop existing pre-load verification schema and recreate/reinitialize schema. Use this option to clean and refresh the schema and results from prior runs. This option does not affect other staged data in any way.
None	Use existing schema and output results from a prior run. You could use this option when coming back to the utility for example after cleansing tasks or to run additional verification checks.

Note

If you choose either the **Create** or **Drop and Create**, you must click **Initialize** to perform the schema initialization.

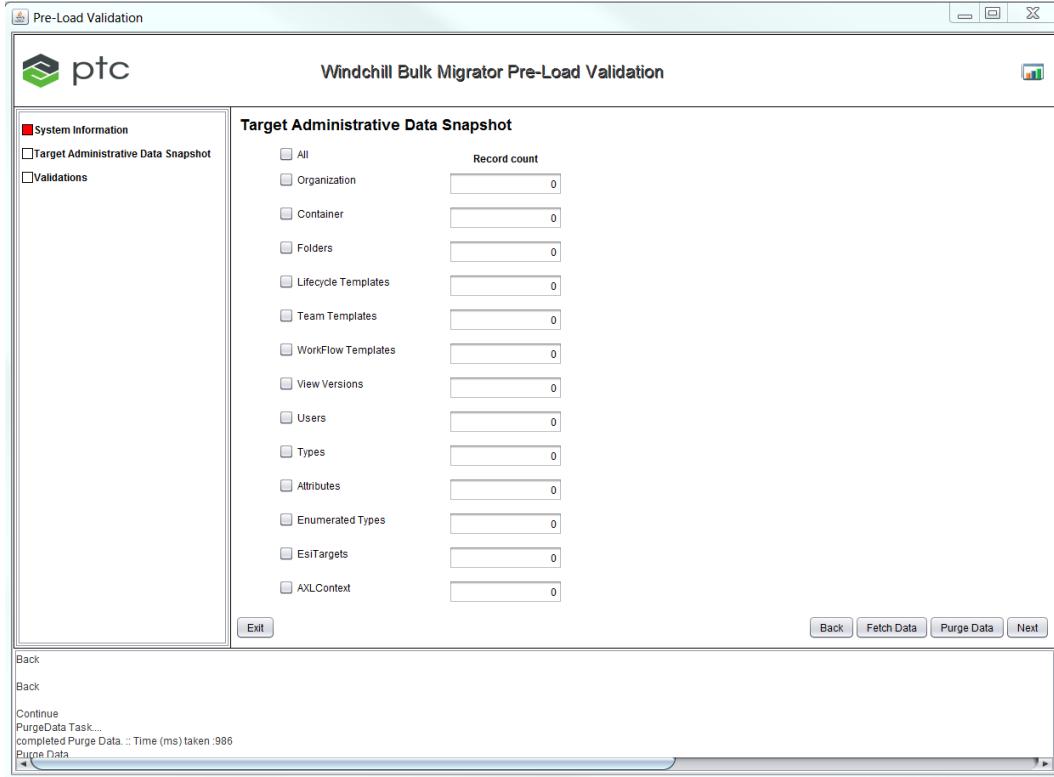
The following table summarizes the window buttons from the System Information page:

Button	Description
Initialize	Start and complete the schema.
Next	Go to the next verification task page
Exit	Close and exit the pre-load verification utility.

After you ensure that the **Staging Properties** and **Target Properties** are correct, and you have selected the desired option for schema initialization, click **Next** to continue.

Creating a Target Administrative Data Snapshot

From the **Target Administrative Data Snapshot** page, you can query the target Windchill system for important information that must be configured correctly and corresponds to object values in the staging database.



Note

Fetching data results in displaying the number of objects found, which is shown in the **Record Count** column on the page.

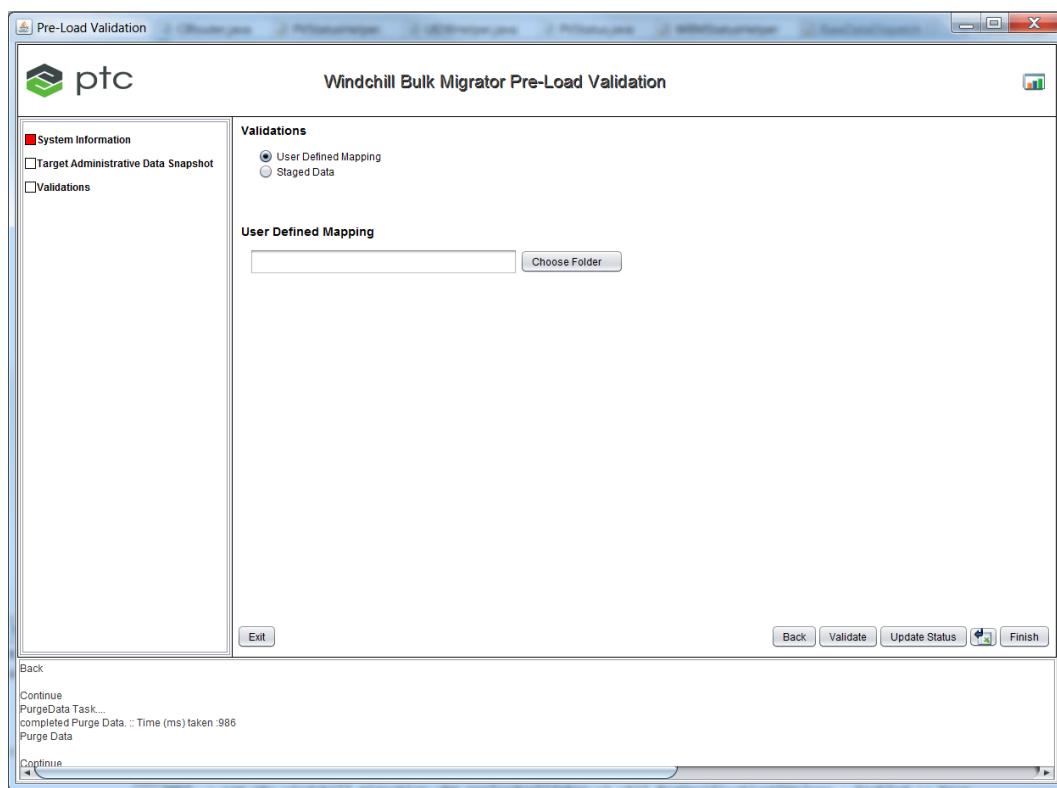
The following table summarizes available actions from the **Target Administrative Data Snapshot** page:

Button	Description
Back	Go back to the System Information page.
Fetch Data	Purges data from the snapshot tables and retrieves selected administrative data from the target Windchill system. Note Fetching data requires Windchill administrative authentication with the target server.
Purge Data	Removes target administrative snapshot

Button	Description
	data from the respective snapshot tables in the staging database. Use this option when you change or update administrative data on the target Windchill system.
Next	Go to the next verification task page.
Exit	Close and exit the pre-load verification utility user interface.

User Defined Mapping Validation

After creating the target system administrative data snapshot, you must verify the data. By default, the utility allows you to verify the user-defined mappings, as specified in the mapping XML files for Windchill-to-Windchill migrations.



The following table describes options that are available for the above page:

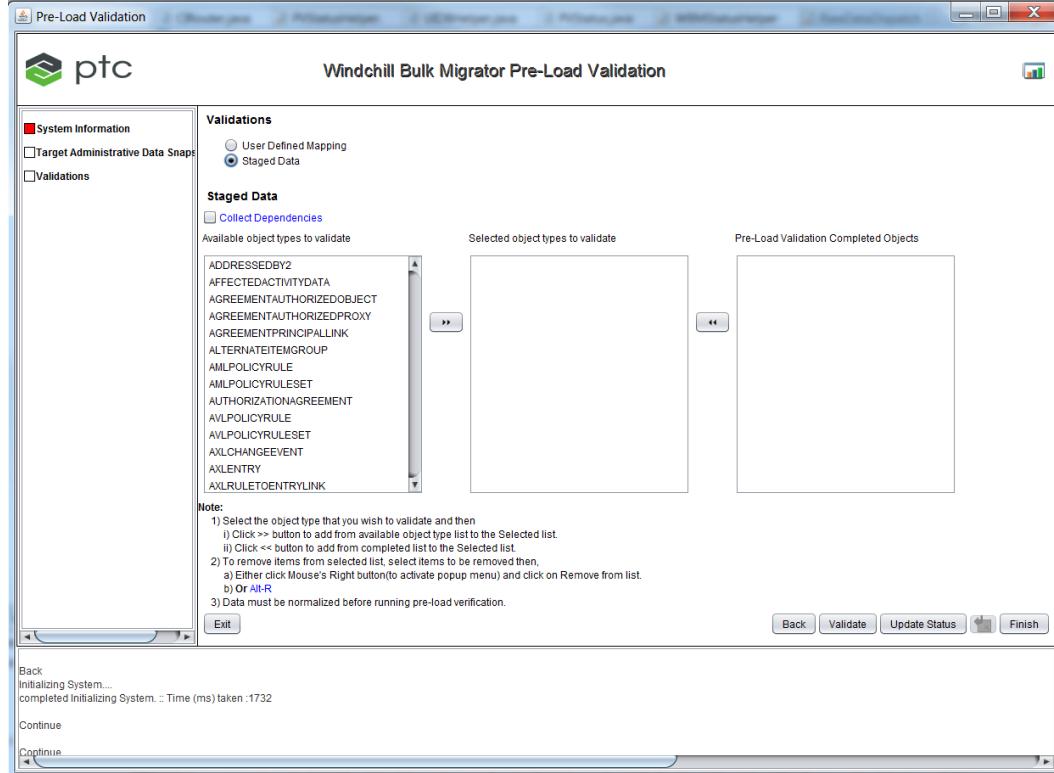
Option	Description
Validations	In the Validations section, you can select one of the following options: <ul style="list-style-type: none">• User Defined Mapping—Validates the mapping XML files with the target system administrative data snapshot. Selecting this option enables you to select the location of the mapping XML files.• Staged Data—Verifies the staged data with the target Windchill object model and target system configurations.
User Defined Mapping (folder selection)	Click Choose Folder and navigate to the directory where the mapping.xml files are located.

After making the appropriate selections for user defined mapping validation, click one of the following:

Button	Description
Back	Go back to the Administrative Data Snapshot page.
Validate	Start and execute the verification tasks for the user defined mappings.
Update Status	Used for the Staged Data verification option. See the following section Validation of Normalized Data for more information.
	Export pre-load validation error records to a .csv file.
Finish	Close and exit the pre-load validation utility.
Exit	Close and exit the pre-load validation utility.

Validation of Normalized Data

Validation of the staged data occurs when you select **Staged Data** on the Data Validation page.



The following table describes options that are available for the above page:

Option	Description
Collect Dependencies	Select this checkbox to automatically include other object types in the Available object types to validate list, as defined by a dependency graph.
Available object types to validate	Displays all object types that can be loaded, as identified from the staging schema tables. Select object classes that you want to validate from this field, and click >> to add the object classes to the Selected objects types to validate list or drag the object type and drop the selected object types to the validate list.

Option	Description
Selected objects types to validate	Displays the object classes selected for validation.
Pre-Load Validation Completed Objects	Displays object classes that completed validation.  Note Object classes with failures are displayed in red. Object classes that pass validation display in black.

After making the appropriate selections for data validation, click one of the following:

Button	Description
Back	Go back to the Administrative Data Snapshot page.
Validate	Start and execute the validation tasks for the selected object classes.
Update Status	Used for the Staged Data validation option. See the following section Validation of Normalized Data for more information.
	Export pre-load validation error reports to a CSV file.
Finish	Close and exit the pre-load validation utility. All updates and status are saved.
Exit	Close and exit the pre-load validation utility.

Note

If pre-load validation property is set to true (`com.ptc.windchill.migration.wbm.preloadvalidator.forceValidation=true`), you must click **Update Status** in order to load data into target Windchill system. By default, this property is set to false.

Generating Validation Reports

You can generate validation reports for the verification task results. To generate the validation reports, click the  report icon in the upper, right-hand corner of the pre-load validation window.

The reports are generated in the following location:

<WT_Home>\logs\wbt\reports\preloadverification\
<datestamp>\<timestampl>

The filenames for the reports are in the following format:

PreLoadVerificationReport <verification type>, and can be generated as HTML, PDF, and XML files. The verification type can be either Data or Mapping.

If errors occur, review the reports to assess the nature of errors.

Interpreting Verification Reports

The following table describes available verification reports:

Report	Description
PreLoadVerificationReport_Mapping	<p>Summarizes the results of the mapping verification tasks. The following columns are available:</p> <ul style="list-style-type: none"> • Object Type—Displays the type of administrative data that was verified against the target system administrative data snapshot. • Counts—The number of conflicts found. A count of 0 (zero) indicates that no conflicts were found. • Database table(s) to reference for detailed information—The internal pre-load verification staging table to reference if errors are discovered.
PreLoadVerificationReport_Data	<p>Summarizes the results of the data verification tasks. The following columns are available:</p> <ul style="list-style-type: none"> • ObjectType—Displays the type of administrative data that was verified against the target system administrative data snapshot. • Counts—The number of conflicts found. A count of 0 (zero) indicates that no conflicts were found. • Database table(s) to reference for detailed information—The internal pre-load verification staging table to reference if errors are discovered. • Object Type—The object type in the staging schema that has a conflict. • Class Name—The Windchill object class corresponding to the object type name. • Objects In Staging Table—The number of objects in the staging table with conflicts. • Failed Due to Administrative data checks—The number of conflicts for the given object class that have conflicts related to administrative data where the administrative data type checks are listed on the first

Report	Description
	<p>page of the report.</p> <ul style="list-style-type: none"> • Failed due to Identity collision checks—The number of conflicts due to duplicate identities for the given object class. • Failed due to other checks—The number of conflicts due to reasons other than administrative data checks and identity check. • Database view(s) to reference for detailed information—The internal pre-load verification staging table refer to if errors are discovered.

The pre-load verification reports are intended to provide an understanding of where in the staging schema you can look for additional information. If conflicts are identified, refer to the staging database table indicated in the report and assess the information in the tables to determine the nature of the conflict.

Data Cleansing

Data cleansing can take a number of forms depending upon the error reported in the pre-load verification tables. In most cases, the steps to resolve the issue can be found in the conflict reports. If you need further assistance to resolve conflicts, contact PTC Technical Support.

Information about specific errors is contained in the pre-load verification schema in the staging database. Depending on the issue, you may need to look in multiple tables. The following describes the naming conventions for the tables and the information they contain:

Pre-load Verification Table Naming Prefix	Usage	Examples
Z_ (views created by PLV tool)	<p>Displays an enumerated list of the errors found, and a description for the given object class. This table lists errors for the various types of verification tasks.</p> <p>For a given conflict, the table also provides details on the identity of the object so you can perform further investigation.</p>	Z_WTPart
PV_ERR_M (table for mapping errors)	<p>Displays additional information about the specific type of mapping conflict, and why it was reported as a conflict.</p> <p>Errors reported in this table indicate that something is invalid with the mapping file definitions. For example, the error could be that the target Windchill system does not have the administrative object configured as desired or as indicated from the mapping file. It could also indicate a typo error in the mapping file.</p>	PV_ERR_M_Folder PV_ERR_M_User PV_ERR_M_Lifecycle
PV_ERR_D (table for data error)	<p>Displays additional information about the specific type of data conflict, and why it was reported as a conflict.</p> <p>Navigate to this tables after looking at the Z_ views to determine the nature of the conflict and determine possible resolutions.</p>	PV_ERR_D_Enumeration PV_ERR_D_Folder PV_ERR_D_Lifecycle

Some data cleansing or conflict resolution actions may result in changes to the target Windchill system, the user-defined mapping XML files, or the staged data itself.

If changes to the staged data are needed (for example, modifying metadata information on the staged record) complete the following steps:

1. Remove the record from the normalized INT_* table using the `wbmldr` command with the `-r` command flag. For more information about using this command for Oracle or SQL Server staging databases, see [Data Cleansing Using the `wbmldr` Command on page 102](#).
2. Correct the entry for the record in a CSV file that corresponds to the type of object. For example, `EPMDocument.csv`.
3. Re-populate the de-normalized (cache) staging table using the `wbmldr` command with the `-c` command flag for Oracle. For SQL Server, use the BCP command to re-populate the cache table. For more information, see [Data Cleansing Using the `wbmldr` Command on page 102](#).
4. Re-normalize the data.
5. Re-run pre-load verification. You must drop and re-create the pre-load verification schema before running the verification tasks again in order to clear the records of the old conflicts.
6. Look at the verification reports and ensure that the conflict was resolved.

 **Note**

Object types that do not go through the normalization process can be cleansed either directly in the de-normalized tables, or in the source system.

If changes to the target Windchill system are needed (for example, adding attribute definitions) complete the following steps:

1. Update the target Windchill system with the corrective action.
2. Re-run pre-load verification. You must generate a new administrative data snapshot, and drop and recreate the pre-load verification schema before running the verification tasks again. This ensures the records are clear of the old conflicts.

If changes to the mapping files are needed (for example, fixing mapping file XML entries), fix the entries and re-run pre-load verification. You must re-load the updated mapping files within the utility.

Family Table Data Validation

The Windchill Bulk Migrator supports migration of family table data. To ensure that data extracted from the source system is error free, execute error detection tasks (if available) and data correction actions in the source system before extracting data to the staging database.

Executing Family Table Error Detection Tasks on the Source System (Windchill PDMLink Only)

If the source system is Windchill PDMLink, PTC strongly recommends executing the Validate EPM Family Tables (ValidateFamilyTableData) task before extracting the source data into the staging database.

1. Open a Windchill shell.
2. Enter the following command:

```
windu
```

3. From the Windchill Diagnostic Utility (WinDU), expand the **EPM Validators** task and select **Validate EPM Family Tables**.
4. Clear all other WinDU tasks.
5. Click **Run**.
6. In the **Results** window, the diagnostic tasks that were run are displayed. If any problems are reported in the <Windchill>\WinDU\logs\<date>\ValidateFamilyTableData.log file, open a call with PTC Technical Support, upload the log file and get assistance in resolving the errors.

Family Table Pre-Load Validation in the Staging Database

After EPMDocument and family table data has been successfully populated into the staging database using the wbmldr -s command, as explained in [Executing the wbmldr Command on page 99](#), follow the steps below in validating family table data.

1. Executing Preverifier

From a Windchill shell, execute the following command:

```
windchill
com.ptc.windchill.migration.wbm.loaders.wc.epm.Pre
verifier
```

2. Reviewing the results

Carefully review the output printed on the console or the log file (located in <Windchill>\wbm\preverifier) to determine if any errors were detected. Each Preverifier execution generates a single CSV file per staging database table (containing family table data or associated objects) if errors

were detected for that table. These CSV files will be located at <Windchill>\wbt\preverifier\<date>\<timestamp>\reports\review.

3. Resolving the errors

Erroneous data should be reviewed to determine if it can be healed in the source system. In order to remove erroneous data from staging database, use the wbmldr -r command and provide the folder path <Windchill>\wbt\preverifier\<date>\<timestamp>\reports\remove as an input argument. Removal of erroneous data should be done after each Preverifier execution. If erroneous data is corrected in the source, it should be extracted and loaded into staging database again. Subsequent execution of Preverifier should not report such corrected data again.

11

Loading Data

Loading Data Overview.....	130
Loading Sequence	136

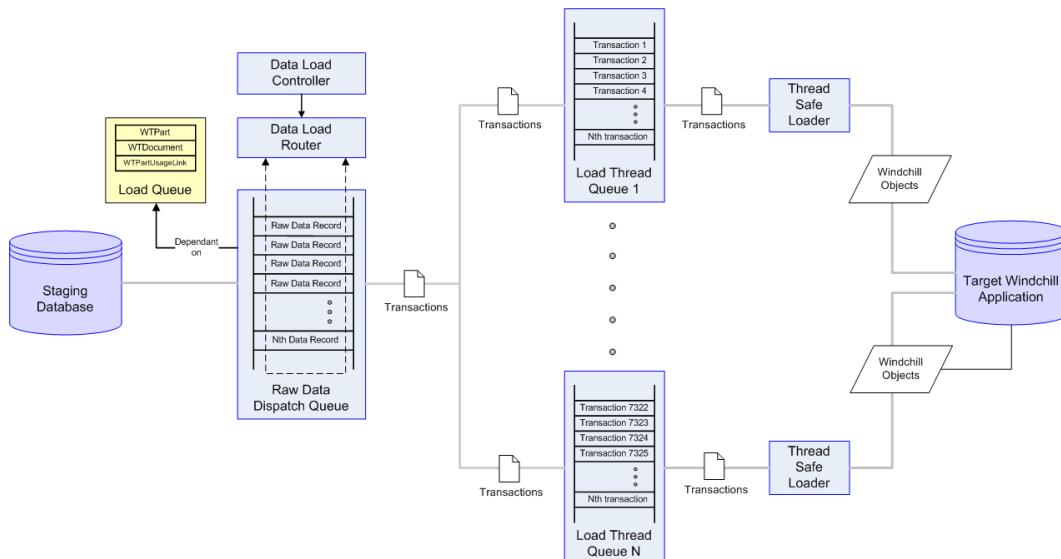
Loading Data Overview

The Windchill Bulk Migrator responsible for loading the data consists of the following:

- The load router, which is responsible for obtaining a result set from a staging database and instantiating an appropriate data loader to process the data. The load router is the Windchill Bulk Migrator RMI server and communicates directly with the target Windchill system.
- The load controller, which dispatches commands to the load router. Multiple requests are pushed onto a load queue and as data loading requests are completed, the load router looks at the load queue for subsequent requests. The load controller is also known as the Windchill Bulk Migrator client.

To load data into a target Windchill system, you start the load router. With the load router running, you issue load commands to the router through the use of the load controller. You can dispatch multiple load commands to the load router. Subsequent commands are added to a load queue. The load queue then queues the commands in a first in first out sequence.

The following diagram shows the data loading process.



As the load router processes a request, data is retrieved from the staging database and this raw data is assembled in a raw dispatch queue. The dispatch queue organizes data into transactions. For versioned objects, the size of the transaction is governed by the `OBJECTS_PER_THREAD.VERSIONED` property in `<WT_HOME>\codebase\com\ptc\wbt\core\meta\cirouter.config`. For example if set to 500, 500 master objects will be processed in one transaction. If one fails, the entire set will be rolled back. If set to 1, only one master object will be processed per transaction.

For non-versioned objects, the size of the transaction is governed by the following property:

```
com.ptc.windchill.migration.wbm.loaders.nonVersionedObjectTransactionSize
```

You can load data in batches using the CIRROUTER_BATCH_ID property in <WT_HOME>\codebase\com\ptc\wbm\ci\core\meta\cirrouter.config. For example, if the value is set to '1', the loader picks the objects with BATCH_ID=1 from the staging area.

The following sections describe common loader behavior, data loading steps, and data loading command information.

Loading Data into Windchill

To load data using the Windchill Bulk Migrator, complete the following steps:

1. Record the current parameter value for optimizer_dynamic_sampling on the target database and update it to 4 just before beginning the load process. After loading data, you will revert this parameter back to its original value.
2. For optimum performance, it is recommended to gather optimizer statistics in the target and staging databases before you start loading.
3. Ensure that your Windchill system is running and has been prepared as described earlier in this section.
4. Open a Windchill shell.

For example, on a Windows system where Windchill program shortcuts have been created, navigate to the Windchill menu under **Programs** and click **Windchill Shell**.

Otherwise, from a command prompt, navigate to the Windchill bin directory and enter:

```
windchill shell
```

5. Specify whether you want the DataLoadController to handle reconciliation explicitly or implicitly by entering one of the following commands:

Explicit

- windchill
com.ptc.windchill.migration.wbm.DataLoadController
-rc <object type>
- windchill
com.ptc.windchill.migration.wbm.DataLoadController
-RC

Implicit

```
windchill  
com.ptc.windchill.migration.wbm.DataLoadController -l  
OrgContainer
```

For more information on data load reconciliation, see [Generating Data Load Reconciliation Reports on page 133](#).

6. From this shell, start the load router using a `DataLoadRouterImp` command similar to the following:

```
windchill com.ptc.windchill.migration.wbm.DataLoadRouterImp
```

You can also include additional arguments on this command for user credentials, heap size maximum value, and heap size minimum value.

7. Open a second Windchill shell.
8. From the second Windchill shell, enter load controller commands which dispatch requests to the load router that is running in the first Windchill shell. The format of the `DataLoadController` command is as follows:

```
windchill com.ptc.windchill.migration.wbm.DataLoadController -l <loader_alias>
```

where `<loader_alias>` identifies a registered loader that has been defined in the

`com.ptc.windchill.migration.wbm.loaderRegistry.<loader_alias>` property. Loader alias names are listed in the **Loader Alias** column of the supported object types tables found in [Supported Object Types on page 298](#). To determine the loader alias to use, identify the business class of the objects you want to load and then use the loader alias listed in the same row.

The loader alias is also last part of the internal name of the top-level object type (for example, `WTPart` or `WTDocument`) of the data you want to load. The loader aliases specified in load controller commands are the same as the object types that you have configured for query files in the properties defined in `wbm.properties`. For details about the properties set for query files, see [Settings for Query File Base Directory and Names on page 57](#).

For example, to load part data associated with the top-level `WTPart` object type, enter the following command:

```
windchill com.ptc.windchill.migration.wbm.DataLoadController -l WTPart
```

The order in which the loading commands are entered for the object types is not important. The Windchill Bulk Migrator uses an internal object dependency graph to ensure that data is loaded in the target Windchill system in the correct order. If a loading command for a particular object type is entered, but the object is not yet cleared to be loaded by the Windchill Bulk Migrator, the loading request is queued and then re-evaluated at a later time as part of a loop sequence.

9. Review log files, reports, and results of the load job in the console output of the running load router.

Review the logs and Windchill Bulk Migrator reports for the completion status of requested load jobs. For details on where logs and reports are located, see [Configuring Log File Information on page 60](#).

10. After all requests that you are interested in complete, enter the following load controller command to shutdown the load router:

```
windchill com.ptc.windchill.migration.wbm.DataLoadController -shutdown
```

This command shuts down the load router. The command window remains open.

11. If your site has family table data, PTC recommends that you validate the data using the Windchill Diagnostic Utility (WinDU). For information on downloading, installing, or using WinDU, see <http://www.ptc.com/support/windu.htm>.

WinDU should be executed after all the CAD data has been published (moved from restricted domain to the public domain) and before using this data.

Perform the following steps after installing WinDU:

- a. Launch a Windchill shell and enter the following command:

```
WINDU
```

- b. From the Windchill Diagnostic Utility (WinDU) window, select the following diagnostic tasks to run:

- Select the **EPMValidators** checkbox.
- In the **EPMValidators** checkbox section, select the **Validate EPM Family Tables (ValidateFamilyTableData)** checkbox.
- Select the **All** checkbox.
- In the **All** checkbox section, select the **Validate EPM Family Tables (ValidateFamilyTableData)** checkbox.

- c. Click **Run**.

- d. In the **Results** window, the diagnostic tasks that were run are displayed.

- e. Check the `Windu.log` record in your Windchill logs folder. For example, `D:\ptc\Windchill_10.1\logs\Windu.log`.

12. Change the parameter value for `optimizer_dynamic_sampling` on the target database back to its original value.

Generating Data Load Reconciliation Reports

The `DataLoadController` can reconcile an inconsistent data load either implicitly or explicitly and it will generate a report. Implicit reconciliation happens prior to any data load, occurs on all object types and generates a report under

<Windchill>\logs\wbm\loadReconcile-<yyMMddHHmm>-<jvmid>-log4j.log. The following is a sample report that could be generated from an implicit reconciliation or an explicit reconciliation for all object types:

OBJECT TYPE	SOURCE ID	TARGET ID	TABLE NAME	RECONCILED LOAD STATUS
WTPartUsageLink	20	281704	WTPartUsageLink	SET TO SUCCESS
WTPartUsageLink	19	281703	WTPartUsageLink	SET TO SUCCESS
WTPart	22	292146	WTPart	SET TO SUCCESS
WTPart	23	292151	WTPart	SET TO SUCCESS
WTPart	24	292156	WTPart	SET TO SUCCESS
WTPart	25	292161	WTPart	SET TO SUCCESS
WTPart	26	null	WTPart	RESET TO NULL
WTPart	27	null	WTPart	RESET TO NULL
WTPart	28	null	WTPart	RESET TO NULL
WTPart	29	null	WTPart	RESET TO NULL
EPMDocument	23	287031	EPMDocument	SET TO SUCCESS
EPMDocument	24	287033	EPMDocument	SET TO SUCCESS
Folder	2	null	Folder	RESET TO NULL
Folder	1	null	Folder	RESET TO NULL

The next example is an explicit report generated for WTPart objects:

OBJECT TYPE	SOURCE ID	TARGET ID	TABLE NAME	RECONCILED LOAD STATUS
WTPart	22	292146	WTPart	SET TO SUCCESS
WTPart	23	292151	WTPart	SET TO SUCCESS
WTPart	24	292156	WTPart	SET TO SUCCESS
WTPart	25	292161	WTPart	SET TO SUCCESS
WTPart	26	null	WTPart	RESET TO NULL
WTPart	27	null	WTPart	RESET TO NULL
WTPart	28	null	WTPart	RESET TO NULL
WTPart	29	null	WTPart	RESET TO NULL

In both examples, the **Target ID** value “null” indicates that those objects have been submitted to WBMServices on Windchill but are not yet loaded.

When the reconciliation begins, the reconciliation client picks all the objects for which the load status = STARTED and migrated = 0 . However for worm generated loaders the migrated status for processed objects are 50. The reconciliation client sends the source IDs to the Windchill Bulk Migrator verification service on Windchill. The Windchill Bulk Migrator verification service checks the source IDs against the WBMAudit table and sends back a mapping file with the source ID and the target object ID, or else it assigns a value of null. Once the reconciliation client gets the map of the source and target object

IDs, it will update the load status of WBMStatus and the migrated flag of object table, accordingly, as well as generating a report for those objects. The report is stored under the logs\wbm folder.

Moving Data from Restricted Domain to Public Domain

When the data is loaded into a restricted domain, the following wbm.properties property is set to true:

```
com.ptc.windchill.migration.wbm.domain.restricted
```

The load then uses a pre-defined restricted domain for the data.

After verification that the migrated data is good, you can move data from a restricted domain into a public domain. By entering the Java command for publishing data, data that is associated with a specific object type can be moved or data associated with applicable object types can be moved. Applicable object types are identified in the following wbm.properties property:

```
wbm.objectType.supported.restricted.domain
```

To move data, complete the following steps:

1. Execute the following command from within a Windchill shell:

```
windchill com.ptc.windchill.migration.wbm.DataPublishController <obj_type>
```

The <obj_type> argument is optional:

- When not included, all applicable object types identified in the wbm.objectType.supported.restricted.domain property are moved.
- When included, the <obj_type> argument must be the last part of the internal name of the top-level object type (for example, WTPart or WTDocument) of the data you want to move. The object types specified in DataPublishController commands are the same object types that you have configured for query files in the properties defined in wbm.properties.

Note

Depending on how your system is configured, you may need to include additional java arguments when executing the above DataPublishController command.

2. In the **Authorization Request** window, enter the user name and password of a user who has the required permissions to move the data and click **Yes**.

Possible users include users who are in the Administrators group (also known as site administrators).

The command uses the information stored in the MigrationSourceSite table to determine which restricted domains to use when identifying data to move. A result of this command is that the foldered iterable objects related to the restricted domain are moved into the public domains inherited from the folder in which they reside. After objects are moved into the public domain, they become accessible to the active users according to the access rules on the public domain of the parent folder. For details about populating the MigrationSourceSite table, see [Installing the Staging Schema on Cross-Platform Configurations on page 261](#).

To verify that the move has occurred, log on as a user who is able to see the moved objects and view the objects.

Loading Sequence

For objects loaded using generic loaders, follow the loading sequence by running the DataLoadController -l command.

For objects loaded using Legacy loaders:

- EPMDocument object must be loaded before loading the below legacy object types:
 - EPMFamilyTable
 - EPMFTProp
 - EPMAStored
 - EPMPParamMapForDoc
 - EPMPParamMapForML
 - EPMPParamMapForRL
 - EPMDocumentUnitInfo
 - EPMMemLinkUnitInfo
 - EPMRefLinkUnitInfo
 - DERIVEDIMAGE
- All Content Holder object types (EPMDocument, WTDocument, EPMFamilyTable, WTChangeIssue, WTVariance etc.) must be loaded before CONTENTMETADATA loading.
- SHAREDContainerMap requires that administrative objects, such as Folders, be in place, and “Shareable” persistables must be loaded first.
- WFPROCESS must be loaded last, as it requires all LifecycleManaged objects to be loaded and published first.

12

Post-loading

Post-loading Verification Overview.....	138
Generating Post Load Verification Reports	138
Execution of WinRU during post migration.....	141

Post-loading Verification Overview

After loading data, complete any additional steps necessary as part of your test or go-live strategy. These steps include, but are not limited to:

1. Generate and review the Windchill Bulk Migrator post-load verification reports.
2. Perform a system backup.
3. Initialize services or queues that may have been user-disabled as part of the migration process.
4. Deploy new client applications (if applicable).
5. Validate end-user business use cases.

Generating Post Load Verification Reports

The Windchill Bulk Migrator includes the `PostLoadVerificationController` command that can be used to generate reports that identify the number of objects that have been successfully loaded.

You can generate reports after the initial migration run and after incremental migration runs. After the initial report is generated, subsequent reports can be generated to show the number of objects successfully loaded during incremental migration runs.

Record the current parameter values for processes and sessions on the staging and target databases and update it to approximately 1000 or above for processes and 700 or above for sessions on both the databases just before beginning the post load verification process. After completing the post load verification, you must revert this parameter back to its original value.

The following topics describe how to execute the command, the result of executing the command, the details of the reports, and report customization options.

Running the Post Load Verification Command

The following defaults are used when generating a post load verification report:

- A maximum of three threads are used in parallel when generating post load verification reports (set in `wbm.verification.post.thread`).
- A maximum of five object types can be process per each thread (set in `wbm.verification.post.objecttype.per.thread`).
- An in-clause chunk size of 500 characters for Oracle queries (set in `com.ptc.windchill.migration.wbm.inClauseBaseChuckSize`).

If the defaults need to be modified, set the corresponding properties in `wbm.properties` using the `xconfmanager` utility.

Command Syntax

To generate a post load verification report, enter the following Java command (on one line) from within a Windchill shell:

```
windchill <heap_size_args> com.ptc.windchill.migration.wbm.PostLoadVerificationController  
-v <obj_type> -u <username> -p <password>
```

where all arguments are optional:

- When specified, `-v <obj_type>` identifies a specific object type on which to report. Valid object types are those types specified in the `com.ptc.windchill.migration.wbm.supportedObjectSTypes` property. See [Supported Object Type Setting for Optional Modules and Legacy Loaders on page 57](#).

For example, to report on part data migration, use the following command:

```
windchill com.ptc.windchill.migration.wbm.PostLoadVerificationController -v WTPart
```

When omitted, all object types are include in the report.

- `-u <username>` specifies the user under which the command is run. The user name specified in `<username>` must be a site administrator (who is a member of the Windchill Administrators group).
- `-p <password>` specifies the password of the user specified in `-u <username>`.
- `<heap_size_args>` indicates one or more arguments for managing heap size values. Using the Windchill Bulk Migrator on some platforms requires additional Java arguments.

If you omit the user credential arguments on the command, the report generator opens a window where you must enter the credentials.

As a result of running the report generation command, a report is generated (as described in [Post Load Verification Report Details on page 139](#)) and the post verification status in the staging database `<obj_type>WBMStatus` tables. The updates to the `WBMStatus` tables are described in the next section.

Post Load Verification Report Details

The following sections provide details on the formats, content, and location of reports.

Report Formats

The format of the post load verification report is determined by the value of the `wbm.postload.report.type` property in `wbm.properties`.

Valid formats are HTML and PDF. By default, the report format is HTML. To generate two reports, one as an HTML file and the other as a PDF file, you can specify both values (separated by a comma) in the property.

Report Content

The post load verification report content is dictated by the template used to generate the report. The default report templates produce reports that list the reported object types grouped by the categories: **Non Versioned**, **Binary Link**, and **Versioned**. For each object type, the report shows the number of objects in the staging database, the number of objects that successfully loaded, the number of objects that failed to load, and corresponding totals. For example, the following image shows report information where there were a total of 781 objects in the staging database of which 767 were loaded successfully and 14 failed to load:

Post Load Verification Report

Report generated on: 2011-08-12 11:12:44 IST

Object Type		Class Name	Objects in Staging Database	Objects Successfully Loaded	Objects Failed to Load
Non-Versioned	Folder	wt.folder.FolderMemberLink	1	0	1
		wt.folder.SubFolder	9	9	0
	LibraryContainer	wt.inf.library.WTLibrary	4	3	1
	OrgContainer	wt.inf.container.OrgContainer	26	25	1
	PDMLinkProduct	wt.pdmlink.PDMLinkProduct	5	4	1
		Subtotal	45	41	4
Versioned	EPMDocument	wt.epm.EPMDocument	81	81	0
		com.ptc.ptcnet.Note	60	60	0
	WTDocument	wt.doc.WTDocument	87	87	0
	WTPart	wt.part.WTPart	216	215	1
			Subtotal	444	443
Binary Link	EPMBuildRule	wt.epm.build.EPMBuildRule	30	30	0
	EPMDescribeLink	wt.epm.structure.EPMDescribeLink	31	30	1
	EPMMemberLink	com.ptc.ptcnet.DefaultEPMMemberLink	30	30	0
	NoteHolderNoteLink	wt.note.NoteHolderNoteLink	61	61	0
	WTDocumentUsageLink	wt.doc.WTDocumentUsageLink	22	21	1
	WTPartDescribeLink	wt.part.WTPartDescribeLink	24	24	0
	WTPartUsageLink	wt.part.WTPartUsageLink	94	87	7
			Subtotal	292	283
		Total	781	767	14

To identify the objects that were not successfully loaded, see `loadErrors-Timestamp-jvmid-log4j.log[x]` at the following location: `C:/ptc/Windchill_10.0/Windchill/logs/wbm`

Report Location

Post load verification reports are stored in the following directory:

`<Windchill>/logs/wbm/reports/postloadverification/<date>/<time>`

where:

- <Windchill> is the Windchill installation directory.
- <date> is the date when the report was created. The date format is yyyy-MM-dd.
- <time> is the time when the report was created. The time format is HH.mm.ss, where the hours are in a 24-hour format.

For example, assume that the Windchill installation directory is opt/ptc/Windchill_10 and a report was successfully created on April 25, 2011 at 1:40:50 PM. Then, the generated report can be found in the following directory:

opt/ptc/Windchill_10/logs/wbm/reports/postloadverification/2011-04-25/13.40.50

Execution of WinRU during post migration

When the following Windchill object types are involved in Windchill Bulk Migration process, then following WinRU tasks must be executed during post migration.

Windchill Object types	WinRU repair tasks
Service Effectivity	Delete duplicate Service Effectivity records and update SE_HASH
Options and Variants (Expressionable classes)	Expression Update Repair Task

13

Migrating Non-Windchill Data into Windchill

Non-Windchill Data Migration Overview	143
Extraction from Non-Windchill Systems.....	144
Understanding the Windchill Bulk Migrator Staging Schema	145
Populating the MigrationSourceSite Table	148
Loading Data to De-Normalized (Cache) Staging Tables	150
Unsupported Use Cases	154

Non-Windchill Data Migration Overview

The Windchill Bulk Migrator assists with migrating data from non-Windchill source systems. For non-Windchill source systems, the Windchill Bulk Migrator provides the data staging format for source data mapping, and executes the standard data normalization, pre-load verification, and loading process steps.

Executing the data load process is only one part of performing a successful migration. Other project tasks include:

1. Target Windchill solution design—designs and verifies how the target Windchill systems and data are to be used. Solution design provides a fixed point for migration data mapping. It is highly recommended that the target solution design be frozen before the desired production go-live event. This leaves plenty of time to rehearse the data migration process as well as to perform full end-user functional validation of the migrated data and the business use cases in the target system.
2. Data mapping—defines how the source data maps to Windchill object types and schema. This step may require custom data transformation steps in order to map and define the data as desired, to the target Windchill system.

Note

Custom data mapping and transformation may add significant time to the project plan.

3. Data extraction tool design and implementation—designs and develops the data extraction technique and tools. PTC does not define how the data should be extracted, only that the data is eventually able to be staged to the de-normalized tables which represent a neutral format for Windchill object types. A common output format of data extraction are CSV files that correspond with the Windchill object type and the object attribute data. The CSV format uses standard Oracle and SQLServer tools, that can be used to load CSV data into defined database tables such as what is represented by the Windchill Bulk Migrator staging schema.
4. Pre-load Validation: Running Windchill Bulk Migrator Pre-Load Validation (PLV) utility is mandatory for data extracted from non-source Windchill systems before data migration steps in target Windchill. Please refer Pre-Load Validation section [Pre-loading Validation Process on page 111](#) for more information.

Note

For non-source Windchill, the following property should be set manually to `WBMLDR.STAGE.WC2WC=false` (The default value of the property is true). PLV should be completed, otherwise the Loader will throw the following exception: Cannot proceed with loading as 3rd party migrations need PLV to be completed <PLV Status>

5. Data migration testing—it is critical to fully test and rehearse the complete migration process multiple times before the production go-live event.
6. Data cleansing—it is common that data coming from non-Windchill systems need to be cleansed in order to conform with the Windchill data model. Cleansing steps are often unique to the data step and nature of the conflict.
7. Post-load user verification—user verification consisting of functional testing of the Windchill business use cases is also a critical project task.
8. User training—user training is highly recommended to ensure the successful adoption of the Windchill system.

Migrating data from non-Windchill source systems is more complex than a Windchill-to-Windchill migration because data mapping, transformation, and cleansing are unique for each migration. There are also more requirements for an in-depth understanding of the Windchill object model.

Extraction from Non-Windchill Systems

The Windchill Bulk Migrator does not include extractors for specific non-Windchill or home-grown systems. For non-Windchill systems, it is up to your implementation team to determine and implement the data extraction strategy. It is also up to your implementation team to define data mappings from the source data to the target Windchill system or Windchill Bulk Migrator staging schema.

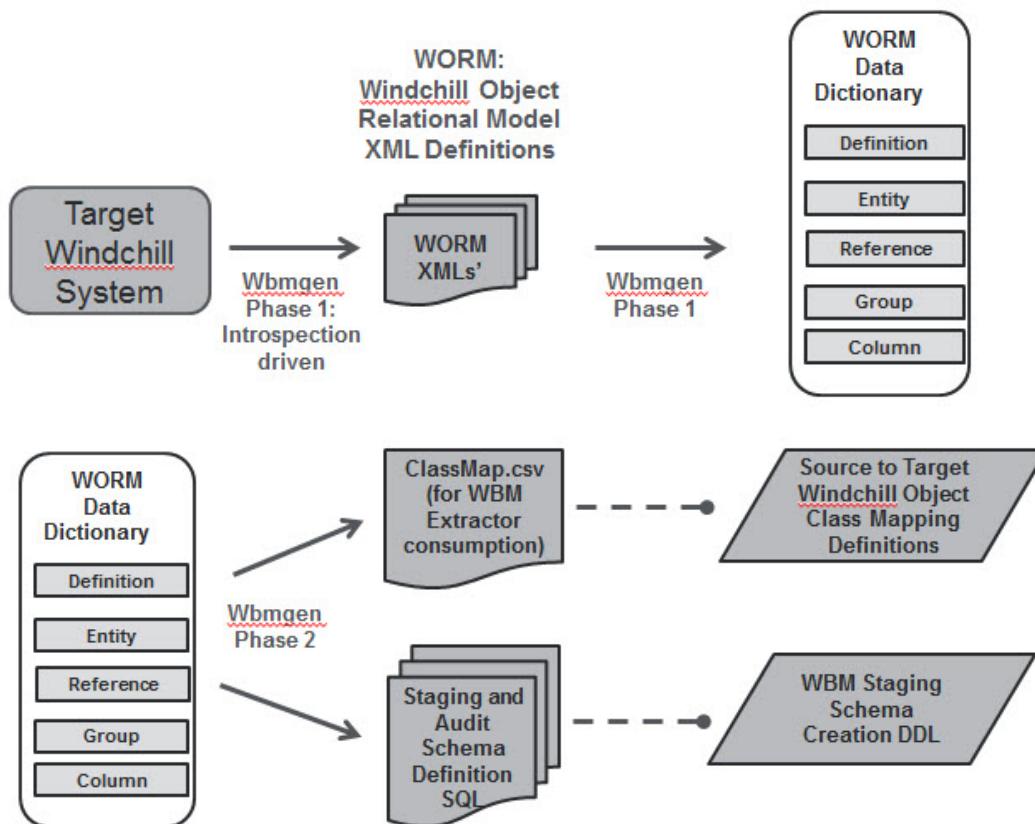
Note

Professional services that have relevant data extraction and data mapping experience can help with data extraction and data mapping. It is recommended that the professional services team also have experience with the Windchill object model and the Windchill Bulk Migrator.

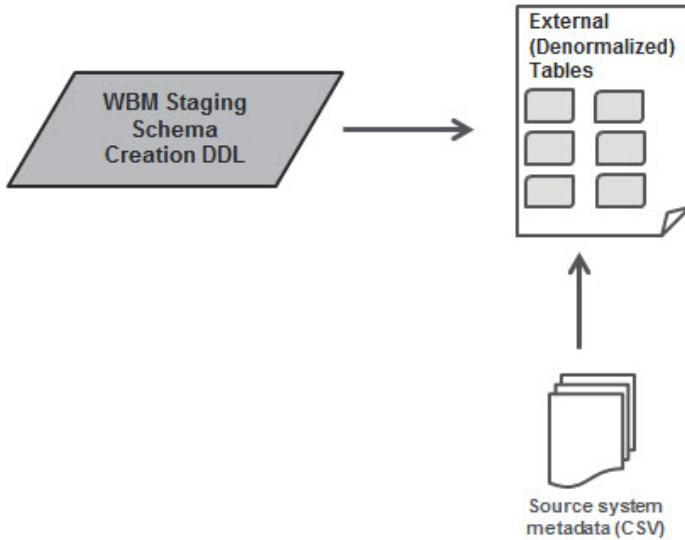
Understanding the Windchill Bulk Migrator Staging Schema

Most supported object types in the Windchill Bulk Migrator staging schema are dynamically generated during installation based upon the target Windchill system. The staging schema generation is driven by Windchill introspection, and it is straightforward to generate a data dictionary for different object types and relational dependencies. For non-Windchill migrations, it can be helpful to understand the infrastructure components that create and define the Windchill Bulk Migrator staging schema, as it will help you better understand the schema tables and dependencies as well as how to dynamically generate new schema for customized objects (if applicable).

The following diagram displays the logical overview for how the Windchill Bulk Migrator generates the staging schema:



For non-Windchill migrations, metadata from the source system must be mapped and imported to the staging schema de-normalized tables in order to allow the Windchill Bulk Migrator to execute the loading process into the target Windchill system.



Tip

Use the `classMap.csv` file to understand the staging table relationships and column formats. The `classMap.csv` can help you understand which metadata from the source system should map to which tables and columns in the staging schema.

Staging Table Naming Conventions and Usage

When preparing the source system CSV load files, refer to the following table naming conventions:

Staging Table Naming Convention	Description
<code><Windchill Class Name></code>	The de-normalized form of the staging table. Use this table as a target table for the population of the raw, formatted object metadata from your input CSV files. Data is typically populated through SQLLDR (Oracle) or BCP (SQLServer).
<code>ATT_<Windchill Class Name></code>	The de-normalized form of the staging table for attributes of the given object class. Use this table as a target table for the population of the raw, formatted attribute metadata from your input CSV

Staging Table Naming Convention	Description
	files.
<code>INT_<Windchill Class Name></code>	The normalized form of the Windchill object class staging tables. These tables are internal tables only and should not be directly modified in any way by the person performing the migration.
<code>VEW_<Windchill Class Name></code>	Internal staging tables that are used to provide a historical record of the view of the de-normalized staging table as it existed before the normalization (<code>wbmldr -s</code>) step. This table can be used for some data cleansing use cases. For more information, see Data Cleansing Using the wbmldr Command on page 102 .
<code>ATV_<Windchill Class Name></code>	Internal staging tables that are used to provide a historical record of the view of the de-normalized attribute staging able as it existed before the normalization (<code>wbmldr -s</code>) step. This table can be used for some data cleansing use cases. For more information, see Data Cleansing Using the wbmldr Command on page 102 .
<code>R_<Windchill Class Name></code>	Internal staging tables that are used to facilitate data cleansing actions for SQL Server staging databases. For more information, see Data Cleansing Using the wbmldr Command on page 102 .
<code>Z_<Windchill Class Name></code>	An internal table used to contain output from pre-load verification tasks. This table is not used for loading data into the staging tables for non-Windchill migrations.

Staging Table Naming Convention	Description
PV_ERR_M_<Windchill Class Name>	An internal table used to contain output from pre-load verification tasks. This table is not used for loading data into the staging tables for non-Windchill migrations.
Z_ERR_D_<Windchill Class Name>	An internal table used to contain output from pre-load verification tasks. This table is not used for loading data into the staging tables for non-Windchill migrations.

In general, it is expected that the person implementing the Windchill Bulk Migrator is familiar with the Windchill object model, the staging schema tables, and table columns in order to understand the correct usage of the tables. If further definition of staging column values and usage is needed, you can review the `classMap.csv` file used for Windchill-to-Windchill migrations. For a description of the `classMap.csv` file, see [The classMap.csv File on page 54](#).

Understanding Object Attribute Tables

The Windchill Bulk Migrator staging schema now has separate de-normalized tables that are intended to contain attribute information for various object types. The attribute tables as indicated in the above table begin with an ATT_* prefix and must be populated in order to migrate attribute information. There is an attribute table for each object type that can be configured to have attributes in Windchill.

Populating the MigrationSourceSite Table

The information provided in the `MigrationSourceSite` table uniquely identifies where the source data came from, where the source data has been extracted from and is an identity to the source that is chosen by the Migration Administrator. The following columns make up the `MigrationSourceSite`:

Column Name	Data Type	Description
SITENAME	VARCHAR2 (2000)	Name for the source data site.
SITEDESCRIPTION	VARCHAR2 (2000)	Description of source data site (optional).
SITEID	VARCHAR2 (2000)	A string that uniquely identifies the source site (the source of the data).

Column Name	Data Type	Description
PRODUCTNAME	VARCHAR2 (2000)	Name of the product where the source data was managed (For example, Windchill PDMLink).
PRODUCTVERSION	VARCHAR2 (2000)	Version of the product (For example, 9.1).

Note that the MigrationSourceSite table may only have one Source Site ID name per staging database and must not change for the migration once the name is selected.

Note

For Windchill-to-Windchill migrations, this table is populated automatically by the Windchill Bulk Migrator on extraction. For a non-Windchill source, the migration administrator may populate the table using SQL update statements with administrator-selected values that are unique.

Additionally, the SITEID in the above table provides the naming pattern for the restricted domains that are created in the target Windchill system by the Windchill Bulk Migrator. When loading data into a restricted domain in the site context or in an organization context, the domain name is arrived at for the site level and the organization level as follows:

- Site level Domain: siteRMDName = "WBM_SiteId UUID(" + UUID.nameUUIDFromBytes(siteId.getBytes()) + ")"
- Org level Domain: orgRMDName = "WBM_OrgContainerName UUID ("+UUID.nameUUIDFromBytes((orgName+','+parentDomainName).getBytes())+ ")";
- The siteId is the value for the SITEID field in the MigrationSourceSite Table. The Restricted Migration Domain naming thus uniquely associates itself with the SiteID.

During a data loading exercise, there is a corresponding persistable object that gets created from the values in the above table. The MigrationSourceSite becomes important, in part, because of the auditing that the Windchill Bulk Migrator performs during the migration.

For example, the following columns make up each audit table:

Column Name	Data Type	Description
MIGRATIONSITE	NUMBER	Primary key of MigrationSourceSite object in Windchill. This key identifies the source of the migrating data.
PLMSILOOBJECTID	NUMBER	Primary key of the source data from staging database. This key is the sequence number field in the staging database.
TARGETOBJECTID	NUMBER	Primary key (identified as IDA2A2) of the migrated data into the target Windchill system.
Class	Varchar2(2000)	This column contains the object's internal type representation stored on the target Windchill system. For example, wt.epm.EPMDocument org.rnd.DefaultEPMDocument.
LoadTime	Date	This column contains the date and time at which the object was migrated to target Windchill.
WBMSOURCEDESCRIPTION	Varchar2(2000)	This column contains the object's internal type representation stored on the target Windchill system. For example, wt.part.WTPart.
WBMSOURCEIDENTIFIER	Varchar2(2000)	It's the identifier of the extracted object. For Windchill extractions, the value is the ida2a2 of the extracted source object.

Loading Data to De-Normalized (Cache) Staging Tables

The Windchill Bulk Migrator supports the migration of objects that must go through the normalization process as well as objects that do not have to go through the normalization process.

The objects that must go through the normalization process (objects supported by the generic loading infrastructure) are the majority of the supported object types, and are those that also have INT_* staging tables in the staging schema. The

objects that do not have to go through the normalization process are only represented by de-normalized tables in the staging schema and consist of Windchill optional module object types.

Objects that are normalized before loading should have metadata loaded to the staging database using the wbmldr command. Objects that are not normalized before loading can have metadata loading using standard utilities in Oracle and SQLServer. The following sections describe the techniques to load metadata to staging tables for different object types.

Instructions for Objects that are Normalized Before Loading

Loading data to the staging tables is performed by using the wbmldr command with the appropriate command flag and user input. The wbmldr command is run from a Windchill shell and the usage of the cache command for an Oracle staging database is as follows:

```
Wbmldr [-c|-cache < path to folder containing the CSV files>]
```



Note

To load data to SQL Server staging databases, use the BCP command.

For more information on the wbmldr command, see [Using the wbmldr Command on page 97](#).

This is the first step in the loading process that caches the data from CSV file(s) into a set of de-normalized tables (cache). The CSV files should end with <OBJECTTYPE>_*csv. For example, EPMDOCUMENT_foo.csv and EPMMEMBERLINK_bar.csv will automatically be associated with EPMDOCUMENT and EPMMEMBERLINK object types, respectively.

It is also necessary to populate the ATT_<objecttype> tables for each object type in order to enable attribute migration. Populating attribute tables also implements the wbmldr -c command.

Note

- Multiple files containing data belonging to the same object type will be combined and cached in a single process.
 - Once a CSV file has been cached, it will not be cached again as long as the data within the file remains the same.
 - An <OBJECTTYPE>.log summarizing the cache results will be generated in the same folder as the source CSV files.
 - An <OBJECTTYPE>.bad will be generated in the same folder as the source CSV files which contains all un-cacheable and discarded rows from the CSV files.
 - For SQLServer, use 'bcp' to manually cache the data from the CSV files.
For more information on using 'bcp', refer to <http://msdn.microsoft.com/en-us/library/ms191479.aspx>.
 - The order of tokens to be specified in the CSV files for a particular object type and all database types (including SQLServer) can be found under Windchill/db/<sql or sql3>/wbm/stage/control/<OBJECTTYPE>.ctl.
-

Instructions for Objects Not Normalized Before Loading

Examples of objects that are not normalized (non-generically loaded) with the Windchill Bulk Migrator include Windchill optional module object types. For these objects you can use standard database tools such as SQLLDR (Oracle) and BCP (SQLServer) to populate the de-normalized tables.

Note

Non-generically loaded objects do not have separate ATT_* and INT_* staging tables. Attributes for these objects must be added as columns directly to the staging table for the object type.

Non-generically loaded object types must have their attributes directly created in the staging table. The attributes can be created by manually editing the corresponding .sql and .ctl for the object type and then executing the new schema DDL scripts in the staging database.

Schema File Locations

The following location contains the Oracle SQL and SQLLDR control files necessary for creating and populating data into the staging database:

- Oracle SQL: <Windchill>/db/[sql|sql3]/wbm/stage
- SQLLDR Control Files: <Windchill>/db/[sql|sql3]/wbm/stage/control

Note

To generate CTL (Control) files while generating WORM on SQL Server Refere option 2 from table under section [Using the wbmgen command on page 27](#)

Loading Staging Tables with Object Metadata Information

Once the attribute columns have been added to the staging tables, you can populate the staging tables with the metadata from the source system CSV files.

For Oracle:

Use SQLLDR commands like the following:

```
SQLLDR userid=<oracle_user/password@SID>
control=<table.ctl_file>
log=<table.log> bad=table.bad
```

In the example above:

- <oracle_user/password@SID> is the user created during the installation process for the staging schema.
- <table.ctl_file> is an Oracle bulk loader (SQLLDR) control file, which specifies how data should be loaded into the staging database. For more information, see the Schema Control File Locations section above.
- <table.log> is the name of a log table that you want to use.

Note

If the staging database has been created using multi-byte SQL3 scripts, you must use UTF-16 or Unicode for CSV files.

Unsupported Use Cases

This section describes specific use cases of note for non-Windchill-to-Windchill migrations, and is not intended to be all-encompassing for the definition of supported and unsupported use cases.

The Windchill Bulk Migrator does not support the following use cases and must not be used, adapted, or customized with the intent of applying the tool for the specific migration:

- Loading AutoCAD data from file system—this use case also includes AutoCAD data that may have been extracted from specific non-Windchill source systems and staged in file system directories.
- Loading and initializing Windchill Workflows—the workflow migration capability available in the Windchill Bulk Migrator is intended for Windchill-to-Windchill migrations only.
- Incremental/Delta migration—this use case is includes use cases that involve bi-directional or federated data exchange, or where the same data is mastered in more than one location. The Windchill Bulk Migrator is not intended to resolve conflicts or force version history merges from two different systems of record. It is also not intended to support long-term one way data transfer in the form of an integration-like solution.

Note

These use cases may be supported in future releases. If you have a requirement for the migration of data that is not supported, PTC recommends contacting third party professional services businesses that can help with these scenarios.

14

Loading CAD Data from File System Directories

Overview	156
CAD Migration	157
Migration Checklist	165
Supported Hardware and Software Notes	165
Installing the CAD Metadata Extractor	166
CAD Data Extraction Sequence	167
Configuring the CAD Metadata Extractor	170
Data Preparation	172
Parallel Execution of CAD Metadata Extractor	187
Custom Mapper Interface	190
Custom System Attributes	193
CAD Metadata Extraction Log Files	194
CAD Metadata Extractor Settings File Properties	194
Using Incremental Migration	198
Migration Failure Resolution Options	202

This chapter describes how to implement the CAD Metadata Extractor, a component of the Windchill Bulk Migrator. The extractor helps with migrating CAD data that resides in a file system directory structure into Windchill.

Overview

The CAD metadata extractor:

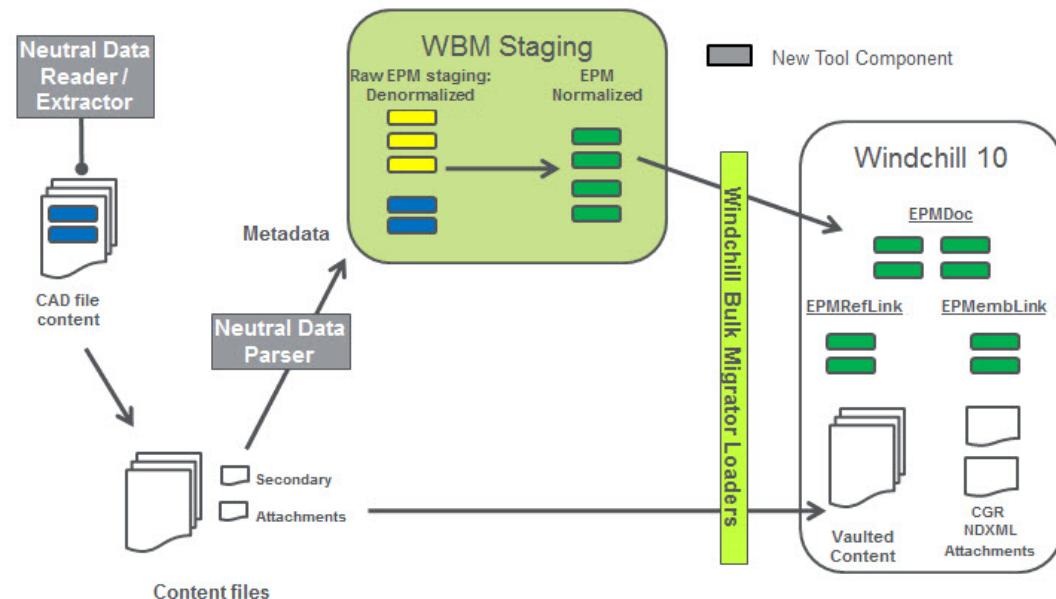
- Analyzes and processes the CAD data to generate the neutral data required to fully define the CAD (EPM) objects in the target Windchill system.
- Maps and updates the CAD files to enable specific desired conditions in Windchill. For example, it includes mapping file system folder structures to Windchill containers and folders, and updates or modifies system attributes and file parameters.
- Passes metadata information to the Windchill Bulk Migrator to enable and control subsequent loading into Windchill.

The CAD metadata extractor supports use cases where CAD data was not managed in a PLM system (user workgroup model), as well as use cases such as non-Windchill migrations.

Note

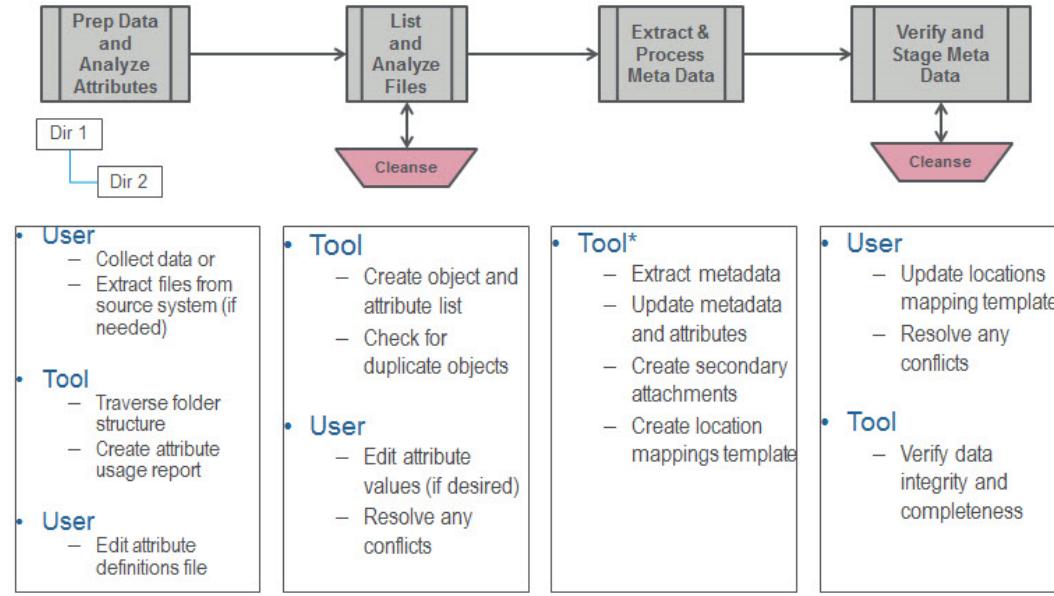
For legacy system migrations, such as non-Windchill migrations, it is assumed that data has already been extracted from the source system and is in a condition that is ready to be analyzed and loaded into Windchill. PTC does not offer data extractors for non-Windchill source systems.

The following is a high-level overview of the process:



The CAD Metadata extractor is responsible for executing the process of extracting the CAD-neutral data as directed by the user. The process workflow provides a number of points for user analysis and interaction with the tool.

The following shows the CAD metadata extraction workflow:



CAD Migration

The CAD migration capability described in this section supports the execution of the migration in two modes: Standard and Incremental.

Caution

Incremental Migration mode is not certified for NX, Autodesk Inventor and SolidWorks.

Note

To implement Incremental Migration mode, you must first implement a first-time standard migration event. Therefore, it is important to read all of the information in this section in the following order:

1. [Unsupported Use Cases and Migration Constraints on page 158](#)
 2. [Standard Migration Mode Overview on page 160](#)
 3. [Incremental Migration Mode Overview on page 162](#)
-

Unsupported Use Cases and Migration Constraints

This section describes use cases and situations that are not handled and supported by the tool.

Each item should be assessed by the migration implementation team and be accounted for appropriately when defining the overall migration plan, process and target system permissions.

1. To-Be migrated data sets must be uniquely identified from any pre-existing data in the target Windchill system that was not migrated by the Windchill Bulk Migrator. The Windchill Bulk Migrator only works with data that the tool itself has migrated.
2. To-Be migrated data sets cannot be linked, via migration, to any pre-existing data in the target Windchill system that was not migrated by using the CAD Metadata Extractor and the same migration source site as defined in the staging area. For example, an assembly that is to be migrated with a reference to a to-be migrated library component cannot exclude the to-be migrated library component and link with pre-existing library component by the same ID in the target system.
3. Incremental Migration cannot use separate, different Windchill Bulk Migrator staging schemas. All incremental migrations must use the same staging schema that was used for the first migration. The reason is because the staging schema retains information about the migrated data that is used for subsequent incremental migrations.
4. Windchill Bulk Migrator CAD Metadata Extractor Working Directories must not be deleted for the duration of the complete migration project and for all incremental migration events.
5. Incremental Migration Mode does not synchronize changes in source and target data for Rename events. Rename events of source files will be treated like new object IDs in subsequent processing of incremental data sets.

-
- 6. Incremental Migration does not synchronize changes in source and target data for Delete events. Delete events of source files will not have any impact on data that has already been migrated.
 - 7. Changes in attribute definitions, location mappings and version/iteration mapping for incremental migrations is not supported.
 - 8. It is mandatory to map the Creo 'Real Number with Units' attributes to corresponding Windchill 'Real with Units' attributes only. Mapping the Creo 'Real Number with Units' attributes to Windchill 'Real' attributes is not allowed.
 - 9. Standard and Incremental data sets must not be modified in the target system at any time until any, each and all migration events are complete. Data that is or could be impacted by migration events must be considered quarantined until the conclusion of any, each and all migration events. The reason quarantine is required is because data loading transactions are on an object by object basis such as EPMDocument, then EPMMemberLink. So data migration and resulting CAD structures are not complete until all object types are loaded.

Note

Implementation and enforcement of data quarantine is relevant for both Standard and Incremental Migration s and is up to the implantation team. The strategy to implement and communicate an appropriate quarantine strategy must be part of the project planning task.

- 10. Neither the Windchill Bulk Migrator, nor the target Windchill system may be upgraded during any each and all migration events. Static staging schema and target Windchill schema is required.
- 11. PTC supports the use of incremental extraction only to append objects to the version history in the target. PTC does not support using incremental extraction to extract lower iterations For example, in the initial extraction, the following objects for EPMDocument are migrated:



-
- In the next phase of the incremental extraction, A.3 cannot be extracted.
12. .xpr and .xas files, which are secondary content for a given CAD file, are not considered for migration.

Limitations with NX, Autodesk Inventor and Solidworks Data Migration

1. The DATE attribute must be migrated with type as STRING.
2. Client system for extractor should have only one version (installation) of Autodesk Inventor which is supposed to be used for migration.
3. For NX, if Family Table (FT) is migrated with Read-only processing, instance properties do not have the Windchill system attributes (PTC_WM_*) and can't be seen on retrieving in the CAD session after migration. As a remedy, after migration, such FTs can be regenerated and iterated from the CAD session to populate the system attributes in instance properties automatically.
4. For correct migration of Autodesk Inventor FT instances, the file structure must follow Autodesk Inventor standards, where instance files are stored in a Folder with a generic name. Relocating instances out of the folder would result in extraction failure.
5. For successful extraction of NX the part family instances, keep the Generic and Instances in the same folder. If Generic and Instances are in different folders, only the Generic file will be extracted.

Note for SolidWorks

Note

In some cases, manual intervention will be required as native SolidWorks messages are observed during extraction of SolidWorks data. User can select “Don’t show again” option, if those messages need not be shown during extraction.

Standard Migration Mode Overview

Standard migration mode can be considered as a complete data set approach whereby all of data is pre-staged, extracted, and loaded in a one-time or first-time migration loading event. The data set must have all required dependencies in order to migrate it, and each object must be uniquely identified from any data that may already be residing in the target Windchill system. The use case involves migrating new object masters and iterations into the target system. The complete

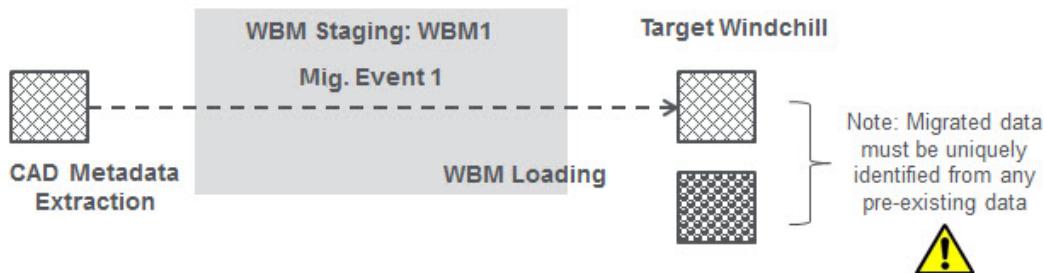
data set is allowed to be separated into one or more partitions or working directories to enable parallel processing of some CAD Metadata Extraction steps, but ultimately, the data must be staged in one event and loaded in one migration event.

The following graphics illustrate different applications of the Standard Migration Mode:

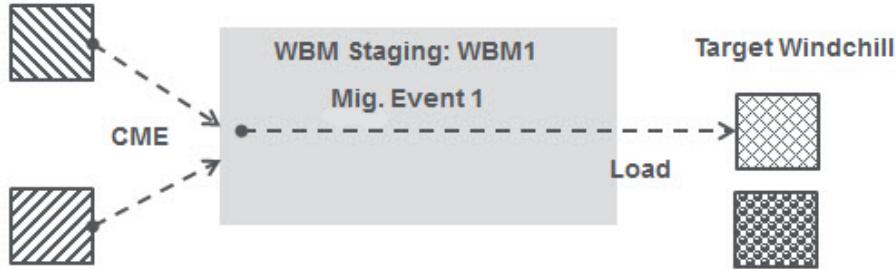
Standard Migration Graphics Key

		Examples
	1st data partition	MyAsm1.asm A.1 ↳ MyPart1.prt A.1
	2nd data partition	MyAsm1.asm A.2 ↳ MyPart2.prt A.1
	Complete data set to be migrated	MyAsm1.asm A.1 MyAsm1.asm A.2 MyPart1.prt A.1 MyPart2.prt A.1
	Pre-existing data in the target system, not migrated by the WBM	

Standard Use 1: Complete Data Set



Standard Use 2: Complete data set, separated into one or more data partitions



In both of the Standard Mode use cases, the most important thing to note is that the loading event into the target Windchill system is a one-time event. The CAD Metadata extraction can occur over different partitions, or even iteratively with new data added to the partitions at different times. However, the data is only staged into the Windchill Bulk Migrator staging database by the tool once, and loaded once as a single-time or first-time event.

If you choose to use data partitions for Standard Migration, they must be created manually. Data files can be distributed according to the strategy you have defined; there are no restrictions imposed by the CAD Metadata Extractor other than once you get to the preload step, the data must be preloaded, staged, and loaded together by the Windchill Bulk Migrator.

Incremental Migration Mode Overview

Incremental migrations occur after a Standard Migration. During an incremental migration, the incremental data set may contain new object masters and/or new iterations to object masters that were migrated during the standard migration. The incremental data set may also have references to data that was migrated during the standard migration.

The following graphic describes the application of the Incremental Migration mode:

Incremental Migration Graphics Key

		Examples
	1st data set to be migrated	MyAsm1.asm A.1 ↳ MyPart1.prt A.1
	2nd or incremental data set to be migrated	MyAsm1.asm A.2 ↳ MyPart2.prt A.1
	Full data set to be migrated	MyAsm1.asm A.1 MyAsm1.asm A.2 MyPart1.prt A.1 MyPart2.prt A.1
	Pre-existing data in the target system, not migrated by the WBM	

Incremental Use 1: Incremental Migration



During incremental migration, the migration events may occur at different times, and in phases per the schedule determined by the migration administrator. For incremental migration, the Windchill Bulk Migrator will remember what has been migrated before and maintain referential integrity of any new dependency objects that are incrementally migrated.

Note

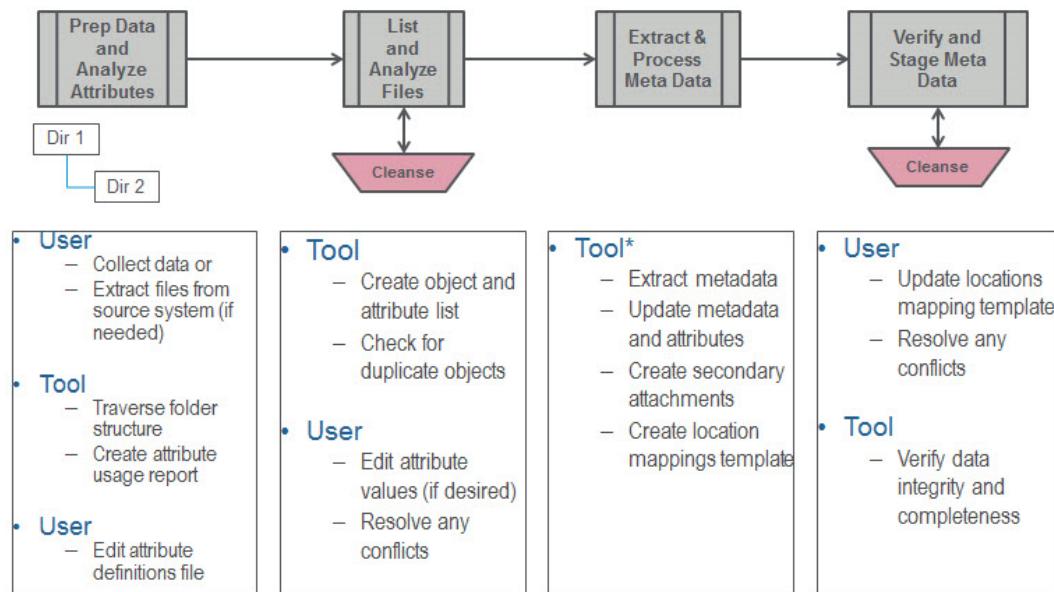
To implement Incremental Migration mode, you must first implement a first-time standard migration event. Therefore, it is important to read all of the information in this section in the following order as noted in [CAD Migration on page 157](#):

1. [Unsupported Use Cases and Migration Constraints on page 158](#)
2. [Standard Migration Mode Overview on page 160](#)
3. [Incremental Migration Mode Overview on page 162](#)

Using Standard Migration

The CAD Metadata extractor is responsible for executing the process of extracting the CAD-neutral data as directed by the user. The process workflow provides a number of points for user analysis and interaction with the tool.

The following shows the CAD metadata extraction workflow:



Migration Checklist

The following list provides a high-level look at the steps that you must perform to complete the migration of CAD data in file system directories into Windchill using the Windchill Bulk Migrator.

- Ensure that the target Windchill system is installed and available.
- Ensure that the hardware that will host the CAD Client, Workgroup Manager, and CAD Metadata Extractor is available and is among the supported hardware platform combinations.
- Ensure that the Windchill Bulk Migrator is installed and the staging schema is created by selecting the option for Windchill-to-Windchill migrations in wbmggen.
- Ensure that the CAD worker/authoring application is installed and available.
- Prepare source CAD data. Collect distributed data into a shared file system folder structure.
- Use the CAD Metadata Extractor to extract metadata and populate the Windchill Bulk Migrator staging tables.
- Load data into Windchill using the Windchill Bulk Migrator.
- (Optional) Execute the Incremental Migration steps for new data sets.
- Perform user validation of the migrated data on the target Windchill system.

Supported Hardware and Software Notes

The CAD Metadata Extractor (Source Client) has the following hardware environment requirements:

- Windows 7 (64-bit)
- 64-Bit Java Runtime Environment, version 1.8
- Microsoft Visual C++ 2010 SP1 Redistributable Package (x64) <http://www.microsoft.com/en-us/download/details.aspx?id=13523>
- CAD Application and Windchill WGM

Windchill Bulk Migrator data staging requirements:

Note

Refer to the [Windchill Bulk Migrator Product Compatibility Matrices](#) for staging schema.

The CAD file content must be prepared, staged and available before running the tools. The content may reside on file systems using any operating systems, but the file directories must, at minimum, be able to be mounted to the hardware system that is hosting the metadata extractor.

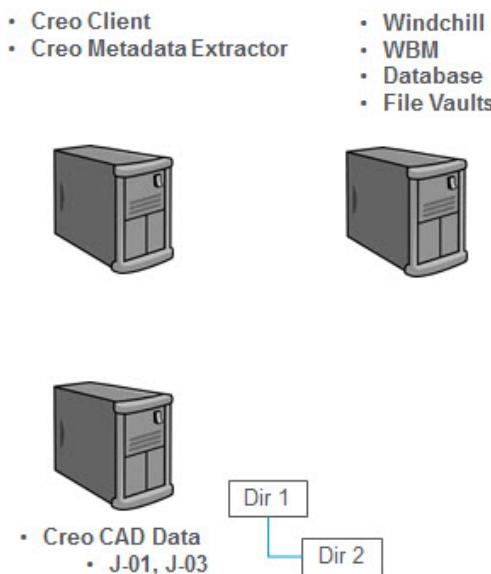
Note

For optimum performance for file processing file content is located on the Metadata Extractor and CAD client systems. It would be optimum to minimize client-worker-content I/O traffic during data processing.

Installing the CAD Metadata Extractor

The CAD Metadata Extractor is included with the Windchill Bulk Migrator media. It is a stand-alone component that can be installed on hardware systems other than the Windchill server. For example, it could be installed where the CAD file data is located. The CAD Metadata Extractor works in coordination with other applications including: Windchill, Creo, and the Windchill Bulk Migrator.

The following graphic displays an example component installation strategy:



The compatibility with Windchill is documented in the Windchill Bulk Migrator software matrices documentation.

Complete the following steps to install the CAD Metadata Extractor:

1. Locate the CAD Metadata Extractor installation ZIP file at *<Windchill Home>\utilities\wbm\extractor\WBMFF.zip*.
2. Copy the *WBMFF.zip* file to the system where you want to install the client. Unzip the file to the desired installation directory *<CAD_ME_Install>*.
3. Open and edit the *settings.txt* file in the installation directory. Review the configuration options and enter the property values accordingly. For more information, see [Configuring the CAD Metadata Extractor on page 170](#)

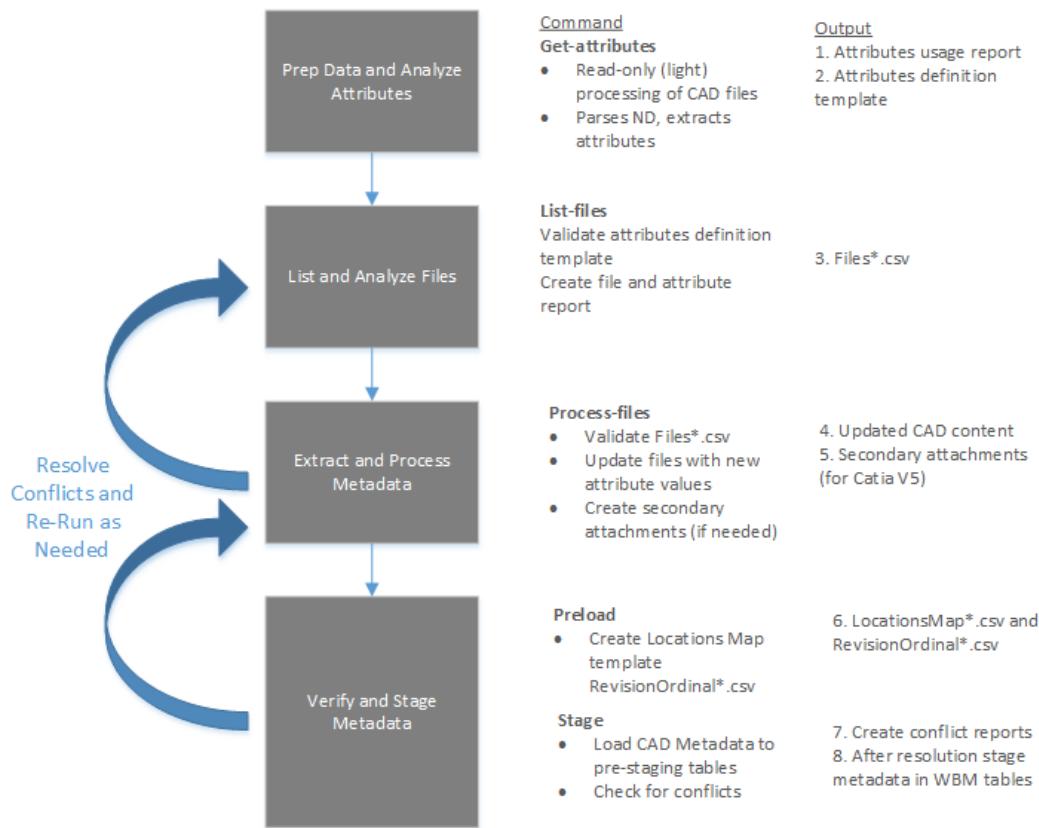
CAD Data Extraction Sequence

This section describes the intended order of migration and process workflow for the CAD metadata extraction.

Pre-requisites

- The CAD data has been collected into file directories that are accessible by the CAD Metadata extractor.
- The CAD Metadata extractor has been installed.
- The Creo or CATIA V5, or NX or Autodesk Inventor or SolidWorks Client has been installed.
- The target Windchill system has been installed.
- For CATIA V5, NX, Autodesk or SolidWorks inventor migrations, the Windchill Workgroup Manager must be installed.
- The Windchill Bulk Migrator loaders have been installed and the migration staging schema has been created.
- The migration user is familiar with the Windchill Bulk Migrator loading process for EPM data.

The migration workflow is an interactive process that includes steps for user review and actions. The following graphic shows an overview of the tool commands and outputs:



Migration Sequence

1. Prepare and collect CAD data that is to be migrated into a directory structure.
2. Configure the CAD Metadata Extractor.
3. Open a command prompt window on the extractor client machine and navigate to the install directory location. Run the `getattributes` command to generate a list of attribute definitions in the CAD files.
4. Review the output reports of the `getattributes` command. Reports are located in the `<CADME_Work>` directory. For more information, see [Get Attributes Command on page 172](#).
5. Run the `listfiles` command to generate a list of the CAD files and the corresponding attribute values for the data set. The tool will traverse the directory structure to generate the list.
6. Review the output reports of the `listfiles` command located in the `<CADME_Work>` directory. For more information, see [List Files Command on page 175](#).

-
7. Run the `processfiles` command to create CAD file neutral data, add/update file attributes, and create secondary attachments (if needed by the CAD type). The resulting neutral data is needed to fully define the Windchill EPM data model for the CAD files.
 8. Run the `preload` command to extract metadata to intermediate tables. This command will also create the `LocationsMap.csv` file which allows you to map file system folders to the target Windchill system folders. The `revisionOrdinals--<timestamp>.csv` file is required to order EPM document revisions.
 9. Update the `<CADME_Work>\locationsMap<timestamp>.csv` file to define folder mappings from the file system to Windchill.
 10. Review, and if necessary, update the `<CADME_Work>\revisionOrdinals--<timestamp>.csv` file to include missing numbers in the ordinal column. The numbers are pre-populated if all encountered revisions are from the default letter based revision scheme in Windchill.
 11. Run the `stage` command to verify the data for invalid conditions.
 12. Check the `<CADME_Work>\conflicts<timestamp>.csv` and `<CADME_Work>\missingLinksReport<timestamp>.csv` files for output from the `stage` command. Take appropriate cleansing actions to resolve any conflicts and execute the `stage` operation directly to resolve the conflicts.

Note

The target Windchill containers and folders are not created as part of the CAD loading migration process. If the target locations already exist on the target system, data can be loaded as normal and described in this section. Optionally, if the target locations do not exist, the Windchill Bulk Migrator can be used to create these objects on the target system. This is because the staging tables for OrgContainer, PDMLinkProduct, LibraryContainer and Folder are populated by the CAD Metadata Extractor stage command and the Windchill Bulk Migrator contains loaders for those object types.

13. Execute the Windchill Bulk Migrator loading process to populate Windchill with the EPM (CAD) objects and content.
14. (Optional) Enable Incremental Migration within `wbm.properties` and run the `flagmigrated` command. Repeat the previous steps for the incremental data set.

For more detailed information on these steps, see [Data Preparation on page 172](#) and [Using Incremental Migration on page 198](#).

Configuring the CAD Metadata Extractor

Preparing the Working Directory

The working directory should be an empty directory with sufficient room to store extracted metadata XML files and other working files of Creo ME. The path to the working directory should be assigned to the `workingDir` settings file property.

Preparing the Settings File

By default, the `setting.txt` settings file is located in the `<CAD_ME_Install>` folder. To list properties supported in the settings file, execute the following command from the command window with the `<CAD_ME_Install>` directory:

```
java -jar Wbmff.jar -help
```

For more information, see [CAD Metadata Extractor Settings File Properties on page 194](#).

The following are required settings file properties:

- Common: `workingDir`, `stagingDbConnectionString`,
`cadJniDllPath`, `config.TargetSiteUrl`,
`config.SoftTypePrefix`
- Required for staging area database on Oracle: `dbUserName`, `dbPassword`
- Required for migration of Creo CAD files: `creoAdapterDllPath`
 - For extraction using Creo 3 or Creo 4, add following entries to the path variable:
 - ◆ `[Creo_Installation]\Common Files\x86e_win64\obj`
 - ◆ `[Creo_Installation]\Common Files\x86e_win64\lib`
 - ◆ `C:\Program Files\Internet Explorer; [Path_to_InternetExplorer_Folder]`
- Required for migration of CAD files for NX, Autodesk Inventor, SolidWorks and CATIA V5: `cadAdapterDllPath`
 - For NX:
`<WGM_HOME>/x86e_win64/obj/ugnxadapter_migrator.dll`
 - For Autodesk Inventor:
`<WGM_HOME>/x86e_win64/obj/aiadapter_migrator.dll`
 - For CATIA V5:
`<WGM_HOME>/x86e_win64/obj/cv5adapter_migrator.dll`
 - For SolidWorks:

```
<WGM_HOMME>/x86e_win64/obj/swadapter_migrator.dll
```

- For NX, if client machine has multiple versions of NX installed, set the following environment variables for specific versions. UGII_BASE_DIRECTORY=<NX INSTALLATION DIR>
UGII_ROOT_DIRECTORY=<NX INSTALLATION DIR>\UGII

Setting up Path Environment Variable

The path environment variable needs to be set to include the folder location of the cv5adapter_migration.dll which is the x86e_wind64/obj folder where the Workgroup Manager is installed.

Configuring System Attributes

System attributes are configured through the properties in the settings file. The property line has the following structure:

```
sysparam.<standard-name>=<CAD-name>,<column-name>,<default-value>
```

In the example above:

- <standard-name> is a Windchill standard system attribute name. It can be one of the following: PTC_WM_REVISION, PTC_WM_ITERATION, PTC_WM_LIFECYCLE_STATE, PTC_WM_LIFECYCLE, or PTC_WM_ORGANIZATION_ID.
- <CAD-name> is a name of the parameter in ND file.
- <column-name> is the name of the column in the fileAttributes--<time-stamp>.csv generated by the listfiles operation. Currently, the following are supported: PTC_WM_REVISION, PTC_WM_ITERATION, and PTC_WM_LIFECYCLE_STATE.
- <default-value> is a value that is used when no value is provided by other attribute value sources.

The following table displays system supported attributes:

Standard Windchill Name (same as default CAD name)	Default Column Name	Default Value
PTC_WM_REVISION	revision	A
PTC_WM_ITERATION	iteration	1
PTC_WM_LIFECYCLE_STATE	lifecycle_state	INWORK
PTC_WM_LIFECYCLE	lifecycle	Default
PTC_WM_ORGANIZATION_ID	organizationId	an organization

System attribute properties examples:

```
sysparam.PTC_WM_REVISION=CAT:REVISION,rev,C  
sysparam.PTC_WM_ITERATION=CAT:ITERATION,iter,44  
sysparam.PTC_WM_LIFECYCLE=CAT:LIFECYCLE,LIFECYCLE,the Lifecycle  
sysparam.PTC_WM_ORGANIZATION_ID=CAT:ORGID,ORGID,Zy
```

Data Preparation

Data preparation consists of collecting CAD files that you want to migrate into a directory structure on a file system. The directory structure can consist of one or more root directories.

In general, it is not necessary to create multiple root directories. However, in some situations it may be useful in order to break up or stop processing very large data sets. If multiple root directories are used, the `getattributes`, `listfiles`, and `processfiles` commands must be executed for each root directory. After processing each root directory separately, the `preload` command will consolidate the conflict analysis steps and check to ensure data integrity. For example, it will check for any missing links in the whole data set.

Note

You must fix the missing links in the dataset before you can proceed. You cannot fix the missing links in the `<CADME_Work>\missingLinksReport<timestamp>.csv.report`.

Get Attributes Command

The `getattributes` command generates a list of attribute definitions in the CAD files. The command is a lightweight processing of the files used only to generate a summary report of the attributes that are candidates for migration in the data set. The command requires a CAD application adapter to be locally installed and configured.

The command can be run from the command line window with the current directory `<CAD_ME_Install>`:

```
java -jar Wbmff.jar -operation getattributes -content  
<root content directory path>
```

The command produces the following files:

- Report
 - attributesUsage--<time-stamp>.csv
- Conditional Reports (not created when empty)
 - cadProcessingFailures--<time-stamp>.csv
 - renames--<content-ID>.csv
- Metadata files generated
 - attributeDefinitions.csv

The attribute definitions file is pre-populated with designated parameters (attributes) found in the processed CAD files. The file can be modified to do the following:

- Suppress attribute migration
- Map CAD parameter to a Windchill global attribute (IBA) with different name
- Allow attribute value modification
- Other advanced attribute manipulations like mapping of multiple CAD parameters to a single global attribute, or changing parameter value type
- In case of attributes having same name but different types, set the value to *true* of the attribute to be migrated in attributeDefinitions.csv and the remaining as *false*.

Make sure the attribute to be migrated is the first among the attributes with same name in attributeDefinitions.csv

If the attributes are of the same name but one is with ‘FLOAT’ and the other is with ‘FLOAT with units’, then the attribute that is to be migrated is the first among them and the other is set as false.

For Example, if the attributes are listed as below:

cadName	wncName	column	proename	valueType	unit	defaultValue	migrate
XX_WF_REAL1	XX_WF_REAL1	XX_WF_REAL1	XX_WF_REAL1	float	m		TRUE
XX_WF_REAL1	XX_WF_REAL1	XX_WF_REAL1		float			TRUE

and if the attribute ‘REAL NUMBER’ needs to be migrated then the file needs to be changed as:

cadName	wncName	column	proename	valueType	unit	defaultValue	migrate
XX_WF_REAL1	XX_WF_REAL1	XX_WF_REAL1		float			TRUE
XX_WF_REAL1	XX_WF_REAL1	XX_WF_REAL1	XX_WF_REAL1	float	m		FALSE

Note

If desired, edit the CSV file to define which attributes will be migrated into the Windchill Bulk Migrator staging tables, and which default attribute values should be passed to the files during migration. Save the file when you are finished reviewing and editing the CSV file. When the file is modified in Microsoft Excel, make sure it is saved in “CSV (Comma delimited) (*.csv)” format.

The following table lists the `AttributeDefinitions.csv` column definitions:

Column	Description
cadName	Attribute name in the CAD file.
wncName	The corresponding Windchill attribute name mapping.
column	Name of the EPMDocument global attribute in the target Windchill system.
proename	This is attribute's internal name used in Creo.
valueType	The attribute value type. This value can be: <ul style="list-style-type: none">• Bool• Int• Float• String
unit	Measurement unit for an attribute of float type.
defaultValue	The default value, if desired, to be populated if an attribute value is not discovered in the CAD file.
migrate	TRUE/FALSE command flag to the tool to migrate the attribute value (TRUE) or not (FALSE).

File data sample:

cad-Name	wncNa-me	column	value-Type	unit	default-Value	migrate
AP-PRDATE	AP-PRDATE		string			true
	is_	approved	bool			true

cad-Name	wncName	column	value-Type	unit	default-Value	migrate
AP-PROVED	approved					
CAL_WEIGHT	Weight		float	kg		true
MC_ERRORS	MC_ERRORS	MC_ERRORS	integer			true
MOD-ELED_BY	MOD-ELED_BY		undefined			false

The `getattributes` command detects instance flushing conflicts. If conflicts are detected, an instance flushing conflict report is created.

The conflict occurs when CAD adapter flush (create) instance files overwrite standalone files with the same name. For example, conflicts occur if:

- There are two files, one for standalone part `bolt_M10.prt`, and another for generic part `bolt_M.prt` in the same folder.
- Generic `bolt_M.prt` has an instance named `bolt_M10.prt`. Under these conditions processing of `bolt_M.prt` will flush all its instances, and instance file `bolt_M10.prt` will overwrite standalone file having the same name resulting in the loss of this standalone part.

The CAD Metadata Extractor detects and reports these conflicts in the `instanceFlushingConflict.<contentId.csv` file which allows you to review and resolve the conflicts.

Options for resolving the above conflicts include:

- Removing the `bolt_M10.prt` standalone CAD file
- Removing the `bolt_M10.prt` instance from the generic `bolt_M.prt`

List Files Command

The `listfiles` command generates a report with a list of the CAD files, where the list also displays the attribute values of the CAD files. The intent of the report is to allow you to review the files that are to be migrated as well as to update specific file attributes, providing instructions to the tool in later steps where the CAD-neutral data is created.

The command can be run from a command line window:

```
java -jar Wbmff.jar -operation listfiles -content <root content directory path>
```

The output is written to the `<CADME_Work>\fileAttributes<timestamp>.csv` file.

Note

The `fileAttributes.csv` file can be edited to alter system or user attribute values. This includes system attributes such as Revision and Iteration. Altering Revision or Iteration values can be used to create an object history order when the data is loaded to the target Windchill system.

Process Files Command

The `processfiles` command validates the `fileAttributes.csv` file and creates CAD file neutral data that maps to the Windchill EPM object model. Depending on the CAD type, it may also update the CAD files to add Windchill system attributes as well as create secondary attachments that may be needed to ensure optimum behavior once loaded into Windchill.

The command can be run from the command line window:

```
java -jar Wbmff.jar -operation processfiles -content  
<content directory root folder>
```

The neutral data output is written to the `<CADME_Work>\<Data Set>.output\` directory. The command creates other files used by other migration steps.

Invalid data in the `fileAttributes.csv` file causes abnormal termination of the command. Reasons for the termination are reported in the console window and in a log file.

Preload Command

The `preload` command collates the results from the process files steps and loads it into the database. It also generates a list of file system folders so they can be mapped to Windchill containers and Windchill folder locations using a `LocationsMap` template. The other files created during the operation is `RevisionOrdinal*.csv`, which provides the ordinals to the revisions in the extracted data.

The command can be run from an extractor client system command window:

```
java -jar Wbmff.jar -operation preload
```

The output is written to the following reports.

Report	Description
<CADME_Work>\locationsMap<timesamp>.csv	<p>Contains two lists:</p> <ul style="list-style-type: none"> • Product/library containers list • Source CAD file folders list with default mapping to the target folder <p>Allows you to map file system folders to target Windchill containers and folders.</p>
<CADME_Work>\revisionOrdinal<timesamp>.csv	<p>Contains two columns:</p> <ul style="list-style-type: none"> • Revision • Ordinal <p>The column revision is fully populated. The column ordinal is a number defining the order of the revision. It is populated when all encountered revisions are from the Windchill default letter based revision scheme (A-Z, AA-ZZ, etc.). When ordinals are not pre-populated it must be manually populated.</p>
<CADME_Work>\issues.DuplicateUniqueNDId<timestamp>.csv	<p>Duplicate UniqueNDIDs are found for dependencies.</p> <p>Recommended resolution: Listed CAD files are invalid and should be excluded or repaired.</p> <p>To repair the files consider removing or modifying duplicate dependencies.</p>
<CADME_Work>\issues.DuplicateFTDependency<timestamp>.csv	<p>FTs containing duplicate dependencies are found.</p> <p>Recommended resolution: Listed CAD files are invalid and should be excluded or repaired.</p> <p>To repair the files, consider removing or modifying duplicate dependencies from family tables.</p>

Report	Description
<CADME_Work>\linkAttributes.csv	<p>This file lists MemberLink attributes similar to attributeDefinitions.csv</p> <p>Set the value to <i>true</i> of the attribute to be migrated in linkAttributes.csv and the remaining <i>false</i> in case of two attributes with same name but different data types.</p> <p>In case the attribute name is shared between attributeDefinitions.csv and linkAttributes.csv then make sure both have the same data type.</p>
<CADME_Work>/ItemsExcludedForDesignatedParams<timestamp>.csv	<p>This will list the files having excluded designated attributes - attributes mapped with proename column and these files will be excluded from migration. The files would be identified based on attribute definition of excluded designated parameters. The list of attribute definitions to be excluded is configured in the attributeDefinitions.csv file.</p>

Preparing the Locations Map

The Locations Map file is prepared during the preload operation, and is used by the operation stage. In the following example, fields that can be modified are designated with (*):

@add-type	container	ff_containerMap		
container-Header	Id	Name	Type	Container Organization
container	5*	Prototype_1*	PRODUCT	Org_A*
container	12*	Fasteners*	LIBRARY	Org_A*
@add-type	folder	ff_		

		<code>folderMap</code>		
folder-Header	<code>originId</code>	<code>SourceLocation</code>	<code>destContainerId</code>	<code>destFolderPath</code>
folder	<code>Proto1</code>	<code>proto\frame (A)</code>	5*	<code>prototype\frame*</code>
folder	<code>Proto1</code>	<code>proto\frame (B)</code>	5*	<code>prototype\frame*</code>
folder	<code>Proto1</code>	<code>proto\core</code>	5*	<code>prototype\core*</code>
folder	<code>Catalog</code>	<code>fasteners\screws\type1</code>	12*	<code>screws*</code>

The Locations Map file contains two tables:

- Containers
- Folders

Container table rows have **container** in the first column, and folders table have **folder** in the first column. Containers define target system products and libraries. Folders define target system folders within referenced products or libraries.

Note

You can create additional container rows, but not additional folder rows.

Table	Column Name	Description
container	Id	Container ID (any number).
container	Name	Name of the product or library in the target system.
container	Type	Container type: PRODUCT or LIBRARY
container	ContainerOrganization	Parent organization for the container.
folder	originId	ID of the source data set content root.
folder	SourceLocation	Relative path of the source content folder.

Table	Column Name	Description
folder	destContainerId	Folder's parent container ID. It must be one of the values in the containers Id column.
folder	destFolderPath	Target system folder path.

Stage Command

The `stage` command loads revision ordinals, analyzes the data, and checks for conflicts. It populates staging tables so they can be used immediately by the Windchill Bulk Migrator loader. It also validates and applies the `locationsMap.csv` file, and runs some verifications of the staged data.

The following command can be run from an extractor client system command window:

```
java -jar Wbmff.jar -operation stage
```

The output is written to the following reports:

Report	Description
<code><CADME_Work>\conflicts<timestamp>.csv</code>	Contains a list of files that are duplicate objects that have the same unique identity as identified by Name/Number, Version, and Iteration. It is possible that the Name-Version conflicts exist for stand-alone objects and for family table (FT) objects (generic and instances) in comparison to stand-alone objects.
<code><CADME_Work>\missingLinksReport<timestamp>.csv</code>	Contains a list of files that have missing references (to other objects), meaning a file that references another file that could not be found. Missing references can be Required or Non-Required objects.
<code><CADME_Work>\issues.locationsMap<timestamp>.csv</code>	Invalid records in <code>locationsMap.csv</code> file

After the data is staged, proceed to the data loading process.

Type Version Conflicts Report

The <CADME_Work>\conflicts<timestamp>.csv report file contains a list of conflicting CAD objects. Types of conflicts reported include:

- Multiple types—This conflict indicates that CAD objects with the same name have a different document type. For example, consider two CAD files named nx1.prt with neutral data specifying that one file type is part, but another assembly. This conflict resolution requires removing or renaming conflicting files, and restarting or resuming the migration.
- FT-instances having more than one top-generic – This conflict indicates that two or more top generics have instances with the same name. This conflict resolution requires exclusion of all but one conflicting top generic, or rename of instances in conflicting top generic.
- FT-standalone mix (for NX and Autodesk Inventor CAD files)—This conflict indicates that CAD objects with the same name have a different family table status. For example, one object may be standalone and the other generic, or one may be generic and the other instance. This conflict resolution requires removing or renaming conflicting files, and restarting or resuming the migration.
- Duplicate version—This conflict indicates that CAD objects with the same name also have the same revision and iteration.

Duplicate object conflicts are identified in the Conflicts.csv report. The following table lists the available columns:

Column	Description
ID	CAD Metadata Extractor internal ID for the CAD file for migration.
Master ID	Internal ID of the CAD file master.
Origin ID	ID of the root content path for the object.
Source Relative Location	CAD file path relative to the root content path.
Source Name	CAD model name as it appears in the file system.
Model Name	The model name extracted from the CAD neutral data. ☞ Note Model Name also reflects the name of a family table instance.
Library Name	The family table generic model name. This column is populated if the object

Column	Description
	in the row of the report is a family table instance. This is null if it not a family table instance.
Type	The CAD file type including: .prt, .asm, or .drw.
Subtype	The CAD sub-type, such as sheet metal, bulk item, etc.
Revision	The version value provided in the <code>filesAttributes.csv</code> file.
Iteration	The iteration value provided in the <code>filesAttributes.csv</code> file.
Family Table Status	Status = 0 indicates the object is not a family table generic or instance. Status = 1 indicates the object is a family table instance. Status = 2 indicates the object is a family table generic object. Status = 3 indicates the object is an intermediate family table generic object (for nested family tables).

Column	Description
Conflict Type	<p>The conflict type can be one of the following:</p> <ul style="list-style-type: none"> • Duplicate Version—indicates that a duplicate identity exists between two or more files. • FT-standalone mix—indicates a name of a family table instance conflicts with the name of a standalone object. • Multiple types— indicates that CAD objects with the same name have a different document type. • FT-instances having more than one top-generic – indicates that two or more top generics have instances with the same name. This conflict resolution requires exclusion of all but one conflicting top generic, or rename of instances in conflicting top generic. <p> Note</p> <p>Some files may have both types of conflicts and can appear in multiple places in the report.</p>
Resolution	<p>This field should be manually updated. Enter one of the following to remove the objects from subsequent processing:</p> <ul style="list-style-type: none"> • exclude—select this option if the object is not a Family Table member. ○ excludeft—select this option in the case of a Family Table generic/instance which will remove the entire Family Table. ○ update—select this option to change the value of revision or

Column	Description
	iteration column.

- FT-Standalone mix conflict between Standalone Object and FT Instance (For Catia and Creo only)—In this case, update standalone objects as newer iterations or Rename the item through CAD application. This conflict can be observed in the following cases:
 - Standalone and FT member have only one version—Update FT member as latest version
 - Standalone and FT member have more than 1 version

The diagram illustrates three scenarios of FT-Standalone mix conflicts:

- Scenario 1:** The first table shows a standalone object with versions 1, 2, and 3, all labeled as "Standalone".
- Scenario 2:** The second table shows a standalone object with versions 1, 2, and 3, all labeled as "Instance".
- Scenario 3 (OR):** The third table shows a standalone object with versions 3, 4, and 5, all labeled as "Instance".

The criterion for finding this conflict is if the maximum version of FT instance is greater than or equal to the minimum version of standalone object.

To resolve this conflict, update the versions of the FT member and standalone so that the standalone object will receive newer versions.

a.prt	
Version	FT Status
1	Instance
2	Instance
3	Instance
4	Standalone
5	Standalone
6	Standalone

- FT-Status conflict between FT Top-generic and FT instance or FT mid-generic (for Catia and Creo only)—For this conflict, update Top-generic objects as newer versions or Exclude them from migration. In Windchill, FT instances (& Mid-generic) can become top-generic objects; but standalone object cannot become a FT instance.

a.prt	
Version	FT Status
1	Instance
2	Instance
3	Instance

a.prt	
Version	FT Status
1	top--generic
2	top--generic
3	top--generic

a.prt	
Version	FT Status
3	top--generic
4	top--generic
5	top--generic

OR

The criterion for finding this conflict is if the maximum FT instance version is greater than or equal to the minimum top-generic iteration.

To resolve this conflict, update the versions of the FT member and standalone so that the standalone object will receive newer versions.

a.prt	
Version	FT Status
1	Instance
2	Instance
3	Instance
4	top-generic
5	top-generic
6	top-generic

- To resolve a Name-Version conflict, user can populate **Resolution** column value with the following options:
 - **exclude**—Select this option if the object is not a family table member.
 - **excludeft**—Select this option in the case of a family table generic or instance which will remove the entire family table.
 - **update**—Select this option to change the value of revision or iteration column.

Missing Links Report

The `<CADME_Work>\missingLinksReport<timestamp>.csv` report file contains a list of missing references (links). That is, a reference defined in an existing file that references another file that could not be found. Depending on the

reference type that is missing, a reference can be Required or Non-Required. Missing references can be ignored through defining the settings file property `allowMissingLinks` that can have one of the following values:

- NONE—migration is allowed when no missing links are found.
- ALL—migration is allowed despite missing links of any type.
- NON-REQUIRED—migration is allowed when no required missing links are found.

Note

- Migration will not continue until all conflicts and missing link related issues have been resolved.
 - MigrationSourceSite table is automatically populated after executing the stage command.
-

Parallel Execution of CAD Metadata Extractor

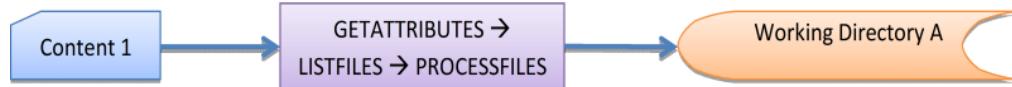
The CAD Metadata Extractor supports independent processing of CAD files in multiple working directories. This allows you to process portions of overall CAD files migration dataset in any order (parallel or sequential) on one or more computers. The benefits include faster processing due to parallelization and simplified handling of invalid CAD files and other CAD files processing failures.

Only three Windchill Bulk Migrator From Files operations can be parallelized: `GETATTRIBUTES`, `LISTFILES` and `PROCESSFILES`.

These operations must be performed in sequence on the same set of parameters. The set parameters are content root directory and working directory. The working directory holds intermediate results of content directory processing.

The following scenarios are possible:

- One content (root), one working directory:

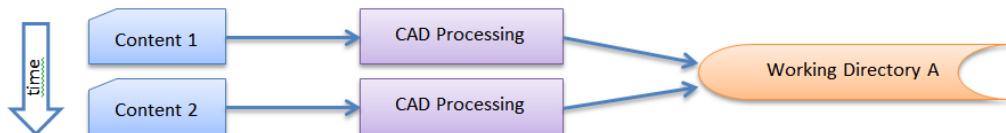


One working directory can accumulate processing results of the multiple content root directories, but such processing cannot be performed in parallel.

Note

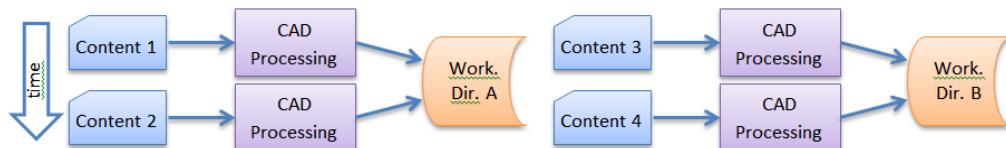
In this section, the working directories can be considered the same as “data partitions” as referenced in the description of Standard and Incremental migration.

- Multiple contents, one working directory:



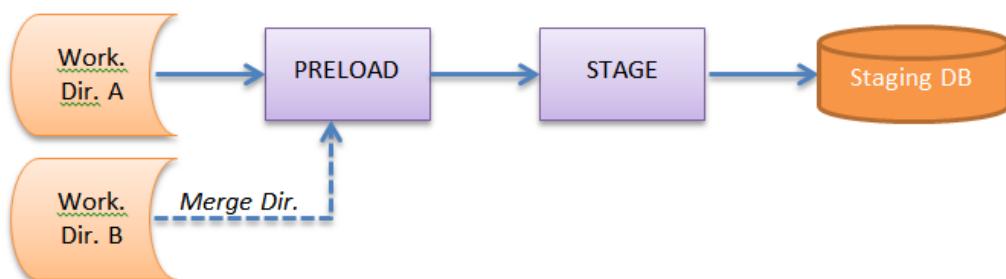
“CAD Processing” on this diagram represents the sequence of three operations (GETATTRIBUTES, LISTFILES and PROCESSFILES) each storing interim data files in the working directory. Two or more working directories can accumulate results of the processing of the multiple content roots.

- Multiple contents, multiple working directories:



Two remaining Windchill Bulk Migrator From Files operations, PRELOAD and STAGE, must be executed on the entire migration dataset. In the case of multiple working directories, one of them serves as working directory for these operations the others serve only as sources of intermediate CAD processing data.

- Multiple working directories during PRELOAD and STAGE:



Constraints for the Directory Paths Used in Parallel Processing

The following are constraints for the directory paths used in parallel processing:

- The content root directory name is used to identify content and it must be unique within the total dataset.
- When multiple computers are used for CAD processing, content and working directories must be mapped so that their paths appear the same on each computer.

Command Line Options for Parallel Execution

Operations GETATTRIBUTES, LISTFILES, PROCESSFILES and STAGE do not have special options for the parallel execution. Yet it may be convenient to provide working directory not through the settings file, but through command line. PRELOAD command is one combining data from multiple working directories using the following command line options:

- `-workdir<work-dir-path>`—Path to the (primary) working directory that is used as a source of intermediate CAD processing data and as a working directory for PRELOAD.
- `-mergedir<work-dir-path>`—Path to a working directory that is used as a source of intermediate CAT processing data.

Example

If you are performing a migration using dataset divided into five partitions and three working directories, you would execute the following commands:

>Note

- Partitions content roots are: N:\CAD\content11; N:\CAD\content12; N:\CAD\content21; N:\CAD\content31; N:\CAD\content32.
 - Working directories are: W:\Wbmff_Works\workA; W:\Wbmff_Works\workB; W:\Wbmff_Works\workC.
-

```
:: Parallel A
java -jar WbmFf.jar -o GETATTRIBUTES -content N:\CAD\content11 -workdir W:\Wbmff_Works\workA
java -jar WbmFf.jar -o LISTFILES -content N:\CAD\content11 -workdir W:\Wbmff_Works\workA
java -jar WbmFf.jar -o PROCESSFILES -content N:\CAD\content11 -workdir W:\Wbmff_Works\workA
java -jar WbmFf.jar -o GETATTRIBUTES -content N:\CAD\content12 -workdir W:\Wbmff_Works\
```

```

workA
java -jar WbmFf.jar -o LISTFILES -content N:\CAD\content12 -workdir W:\Wbmff_Works\workA
java -jar WbmFf.jar -o PROCESSFILES -content N:\CAD\content12 -workdir W:\Wbmff_Works\
workA
:: Parallel B
java -jar WbmFf.jar -o GETATTRIBUTES -content N:\CAD\content21 -workdir W:\Wbmff_Works\
workB
java -jar WbmFf.jar -o LISTFILES -content N:\CAD\content21 -workdir W:\Wbmff_Works\workB
java -jar WbmFf.jar -o PROCESSFILES -content N:\CAD\content21 -workdir W:\Wbmff_Works\
workB
:: Parallel C
java -jar WbmFf.jar -o GETATTRIBUTES -content N:\CAD\content31 -workdir W:\Wbmff_Works\
workC
java -jar WbmFf.jar -o LISTFILES -content N:\CAD\content31 -workdir W:\Wbmff_Works\workC
java -jar WbmFf.jar -o PROCESSFILES -content N:\CAD\content31 -workdir W:\Wbmff_Works\
workC
java -jar WbmFf.jar -o GETATTRIBUTES -content N:\CAD\content32 -workdir W:\Wbmff_Works\
workC
java -jar WbmFf.jar -o LISTFILES -content N:\CAD\content32 -workdir W:\Wbmff_Works\workC
java -jar WbmFf.jar -o PROCESSFILES -content N:\CAD\content32 -workdir W:\Wbmff_Works\
workC

:: After all parallels are completed: PRELOAD
java -jar WbmFf.jar -o PRELOAD -workdir W:\Wbmff_Works\workA -mergedir
W:\Wbmff_Works\workB -mergedir
W:\Wbmff_Works\workC

:: STAGE
java -jar WbmFf.jar -o STAGE -workdir W:\Wbmff_Works\workA

```

Custom Mapper Interface

The custom mapper interface allows you to programmatically provide system and custom attributes for a specific CAD file. The Java class implementing the interface must be declared in the settings file. Once declared, the interface methods invoke at the appropriate time during the WBMFF operation LISTFILES. First, the method `getHeader` is called. The method returns an array of attribute names. Then, for each encountered CAD file method, `getDataStringsForFile` is called. This method returns an array of attribute values in the same order as the attribute names.

Technical Details

The custom mapper is a class implementing interface com.ptc.windchill.migration.wbm.wbmff.cad.CustomMapper:

```
public interface CustomMapper {  
    /**  
     * Gets list of attribute column names  
     * @return String array containing attributes column names  
     */  
    String[] getHeader();  
  
    /**  
     * Gets attribute values for a give CAD file  
     * @param fileLocation - CAD file directory path relative to the content root  
     * @param fileName  
     * @param sysAttributeDefaults - String array of the attribute default values in the order of column names provided by getHeader()  
     * @return String array of attribute values in the order of column names provided by getHeader()  
     */  
    String[] getDataStringsForFile(String originId, String fileLocation,  
        String fileName, String[] sysAttributeDefaults);  
  
    /**  
     * Invoked before start of the default CAD processing to allow implementor's configuration  
     * @param properties - properties collection of the WBMFF settings file  
     */  
    void configure(java.util.Properties properties);  
}
```

The interface implementation is provided through the `FilesAnalyzer.CustomMapper.Implementor=<full-class-name>` settings file property. The interface methods are invoked during the `listfiles` command. Methods `configuration` and `getHeader` are invoked once before preparing the CAD file attributes data, and `getDataStringsForFile` is invoked for each CAD file before attribute values are parsed from the CAD-neutral data file and merged with default attribute values.

When the above mentioned settings property is not specified, the default implementation of the custom mapper is used.

For Creo processing, the default custom mapper implementation is: `com.ptc.windchill.migration.wbm.wbmff.cad.FileExtensionToIterationMapper`. This class converts Creo files numeric extension to

EPM Document iteration. For example, files `a.asm.1`, `a.asm.3`, and `b.prt.5` are migrated to `a.asm`, iteration 1; `a.asm`, iteration 3; and `b.prt`, iteration 5, respectively.

For non-Creo processing, the default custom mapper implementation is:
`com.ptc.windchill.migration.wbm.wbmff.cad.catia.DefaultCustomMapper`. For more information, see [Setting Revision and Iteration Using DefaultCustomMapper on page 192](#).

Setting Revision and Iteration Using DefaultCustomMapper

`DefaultCustomMapper` is used for non-Creo processing when `CustomMapper` implementation is not provided through the `FilesAnalyzer.CustomMapper.Implementor=<full-class-name>` settings file property.

The `DefaultCustomMapper` is also used when an implementation class is explicitly specified:

```
FilesAnalyzer.CustomMapper.Implementor=
com.ptc.windchill.migration.wbm.wbmff.cad.catia.DefaultCustomMapper
```

`DefaultCustomMapper` implements the extraction of the CAD file system attributes from the file directory path. For example, if the file path is `MyAssembly\top`, version `B.0\main.asm`, then if properly configured, the following system attributes are provided:

- Revision: B
- Iteration: 0
- Life-cycle-state: RELEASED

Settings file properties configuring above behavior:

- `DefaultCustomMapper.VersionMatchKeyword=version`
- `DefaultCustomMapper.IfIteration.Number=0`
- `DefaultCustomMapper.IfIteration.State=RELEASED`

The `MyAssembly\top, ver. B.0\main.asm` version pattern can be parsed using the following setting:

```
DefaultCustomMapper.VersionMatchKeyword=ver.
```

The `MyAssembly\top (B.0)\main.asm` version pattern can be parsed using the following setting:

```
DefaultCustomMapper.VersionMatchPattern=\ \ ((\\w+)\ \ .
(\\d+)\ \ )
```

Example: Calling Windchill Bulk Migrator From Files Using a Class Implementing CustomMapper

For the following example, we will use a settings property named `FilesAnalyzer.CustomMapper.Implementor=wbmff.MapParams`.

The following class file is located in the Windchill Bulk Migrator From File run/install location:

```
FilesAnalyzer.CustomMapper.Implementor=wbmff.MapParams
```

If run from the `<wbmff-install>` directory, the `listfiles` command using `MapParams` may look like this:

```
java -cp "custom;Wbmff.jar"  
com.ptc.windchill.migration.wbm.wbmff.WbmFromFiles -o  
listfiles [the rest of LISTFILES options as usual]
```

Custom System Attributes

Custom system attributes (CSA) are attribute values associated with EPM objects that are not migrated as global attributes or as predefined attributes of the object such as EPM Document Revision or Iteration. EPM document non-file based (NFB) attributes are the object creation and modification dates, and creator and modifier user. These NFB attributes are examples of CSAs and can be handled during WBMFF migration using CSA framework.

CSA framework allows the following features:

- Definition of a CSA: CSA label, attributes file column name, and default
- Per document value modification through CustomMapper or attributes file
- Loading of CSA to a temporary table of the staging area database
- Definition of custom SQL queries to modify target staging area database objects using CSA values

CSA framework is designed to be used when you must define necessary elements. CSA values are loaded during the `preload` command into the `ffItemAttribute` staging area database table. A row in the table is associated with CSA through the `paramName` column, and with EPM Document through the `originId` and `sourcId` columns. Handling of EPM document NFB attributes is implemented in WBMFF out-of-the-box to serve as an example of CSA attributes.

EPM Document Non-File Based Attributes Migration

Before migration, perform the following actions:

- In the settings file, uncomment lines defining CSAs from the sample settings file:

```
sysparam.SYS_CREATED_BY=*,creator,Default Creator
sysparam.SYS_CREATED_AT=*,created-at,2014-01-16T21:00:00-07:00
sysparam.SYS_MODIFIED_BY=*,modifier,Default Modifier
sysparam.SYS_MODIFIED_AT=*,modified-at,2014-01-16T21:00:00-07:00
The property name after sysparam. defines the CSA label; the label used as the attribute name in database.
```

In the three-field CSV value, the second value (for example, creator) defines the column name for this attribute in `fileAttributes--*` file, and the third column defines the default value for the attribute.

- Modify the property values (for example, default values) as necessary.
- In the settings file, add the following line:

```
customSqlCodeFile=customCode.prop.xml
```

During the migration, after the `listfiles` command, modify the attributes values in `fileAttributes--*` file, if necessary.

CAD Metadata Extraction Log Files

The CAD metadata extraction process creates log files that record process results, and can be used for troubleshooting. The log files are located at: `<CADME_Work>logs`.

The following log file is generated: `wbmFromFiles<timestamp`. This log file records everything from the `-o` step.

CAD Metadata Extractor Settings File Properties

Property	Required	Description	Default
<code>excludeFTwithDuplicateMemberNames</code>	true/false	Enables automatic exclusion of FTs with duplicate member names.	—
<code>ignoreDuplicateFTColumns</code>	true/false	Allows the migration to continue despite finding duplicate FT columns.	—

Property	Required	Description	Default
stagingDbConnectionString	Yes	Staging database connection string.	—
allowMissingLinks	No	Allows migration to continue if missing links are found. Possible values are ALL, NON-REQUIRED, or NONE.	NONE
cadAdapterDllPath	Yes	Path to Non-Creo CAD adapter DLL.	—
cadAdapterProcessingLimit	No	Number of CAD files to be processed before CAD adapter restart. If 0 or -1: never restart.	0
cadJniDllPath	Yes	Path to JNI adapter DLL (CadJniAdapter.dll).	—
creoAdapterDllPath	Yes	Path to Creo adapter DLL (wppndmerge.dll).	—
customSqlCodeFile	No	SQL code resource XML file for custom SQL scripts. For example, custom system attributes handling. The file path can be absolute or relative to the install directory.	—
dbBatchExecutionBatchSize	No	Batch size for the database update or insert operations.	10000
dbPassword	Yes	Oracle connection password.	—

Property	Required	Description	Default
dbUserName	Yes	Oracle connection user name.	—
enableAttributeValueTypeCast	No	Enables change of the CAD document parameters type to the type defined in attribute definitions file.	false
isReadOnlyCADProcessing	No	Enables read-only processing of CAD files. When enabled, no CAD files are touched during the processfiles command, but data provided during the getattributes command is used.	false
jniDebugLog	No	Enables debug logging of CAD adapter API calls.	false
config.AllowNonIntegerIteration	No	Flag to allow non-integer iteration. Disallow: 0, Allow: 1.	0
config.AuthoringAppType	No	Default Authoring application type. For example, PROE.	PROE
config.DefaultContainer	No	Default container name.	CHANGEME-product
config.DefaultContainerType	No	Default container type. This value can be PRODUCT or LIBRARY	PRODUCT
config.DefaultIteration	No	Default iteration.	1
config.DefaultLifeCycle	No	Default life cycle template name.	Basic

Property	Required	Description	Default
config.DefaultLifeCycleState	No	Default life cycle state	INWORK
config.DefaultOrganization	No	Default organization.	CHANGEME-org
config.DefaultPrecision	No	Default precision for floating point attribute.	9
config.DefaultRevision	No	Default revision.	A
config.DefaultUser	No	Default user.	Administrator
config.LibraryTemplate	No	Library container template name.	General Library
config.OrgContainerTemplate	No	Organization template name.	General (PDM)
config.ProductTemplate	No	Product container template name.	General Product
config.SeriesName	No	Revision series name. For example, wt.series.HarvardSeries.	(empty string)
config.SoftTypeEPMDocument	No	Target system EPMDocument soft type.	(empty string)
config.SoftTypeEPMDocumentMaster	No	Target system EPMDocument-Master soft type.	(empty string)
config.SoftTypeEPMMemberLink	No	Target system EPMMemberLink soft type.	(empty string)
config.SoftTypeEPMReferenceLink	No	Target system EPMReference-Link soft type.	(empty string)
config.SoftTypePrefix	Yes	Target system soft type prefix.	-
config.SourceSiteId	Yes	Source site ID.	-
config.SourceSiteUrl	No	Source site URL.	(empty string)

Property	Required	Description	Default
config.TargetSiteUrl	Yes	Site URL of the target Windchill.	—
sysparam.PTC_WM_ITERATION	No	Configuration for the system attribute iteration.	PTC_WM_ITERATION,,iteration,1
sysparam.PTC_WM_LIFECYCLE	No	Configuration for the system attribute life cycle name.	PTC_WM_LIFECYCLE,,Default
sysparam.PTC_WM_LIFECYCLE_STATE	No	Configuration for the system attribute life cycle state.	PTC_WM_LIFECYCLE_STATE,lifecycle_state,INWORK
sysparam.PTC_WM_ORGANIZATION_ID	No	Configuration for the system attribute organization ID.	PTC_WM_ORGANIZATION_ID,,an organization
sysparam.PTC_WM_REVISION	No	Configuration for the system attribute revision.	PTC_WM_REVISION,revision,A

Using Incremental Migration

You can implement the Incremental Migration mode by setting the following property in <WT_HOME>\codebase\com\ptc\windchill\migration\wbm\wbm.properties.xconf to true:

```
<Property name="com.ptc.windchill.migration.wbm.mode.incremental" overridable="true"
targetFile="codebase/com/ptc/windchill/migration/wbm/wbm.properties"
default="false"/>
```

The Incremental Migration Mode property setting, if desired, must be set after the initial Standard Migration event, and before the first incremental migration event.

Note

- It is mandatory to use the same working directory, which was used in the initial standard migration event. Make sure the working directory is not cleaned up.
- Attempting to perform an incremental migration for an object already migrated without setting the above property in `wbm.properties` will result in errors reported by the `DataLoadRouter`. For example, you may encounter the following error:

```
objectType:EPMDocument, stgId:2 =>
wt.util.WTException: wt.pom.UniquenessException: CAD
Part - part-name.prt is not unique
```

By default, this property is set to `false`.

The change must be propagated using the `xconfmanager -p` command.

Using the `flagmigrated` Command

The `flagmigrated` command prepares the Windchill Bulk Migrator for incremental migrations by updating certain internal staging tables after verifying that either the initial standard migration completed successfully or the last incremental migration processed by the tool completed successfully.

The command can be run from the CAD Metadata Extractor client system command window:

```
java -jar Wbmff.jar -operation flagmigrated
```

Note

After the CAD Metadata Extractor staging operation is successful, all normal CAD Metadata Extractor operations remain locked until the `flagmigrated` command has completed successfully. When the lock is enabled, normal CAD Metadata Extractor operation produces the following message in console and log:

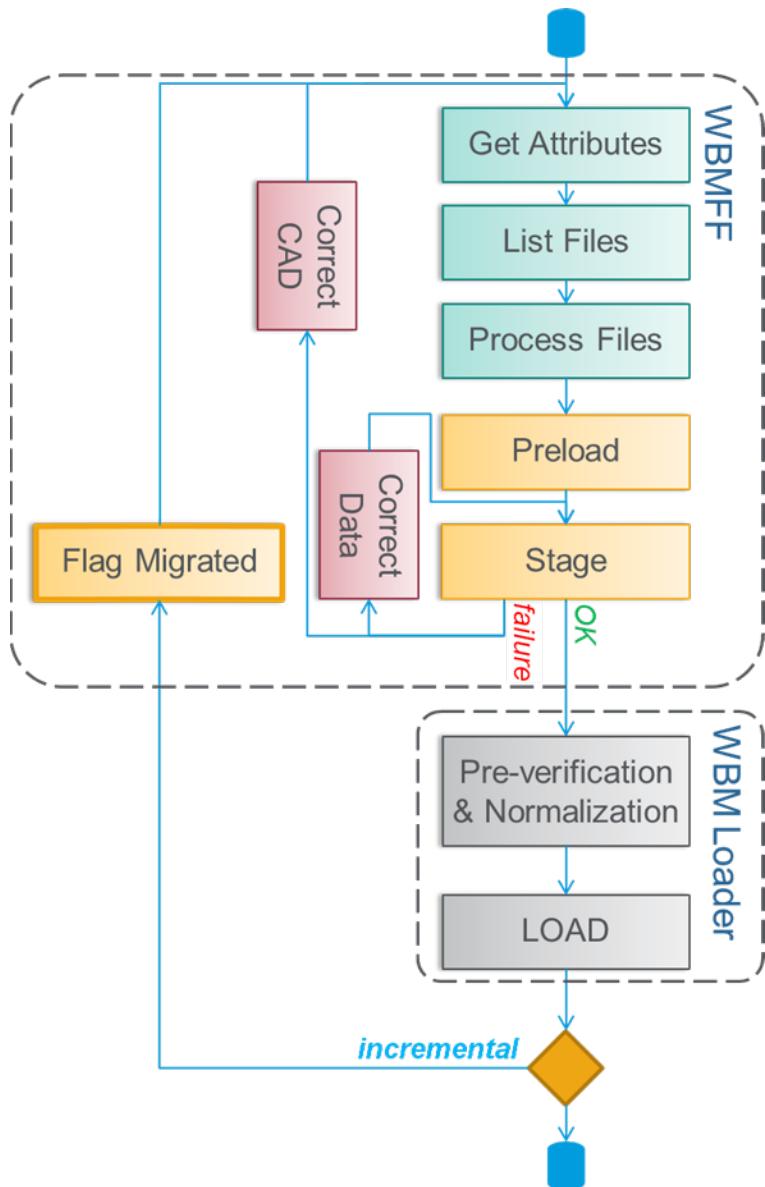
```
*** Normal operations are locked.  
*** To start incremental migration one has to:  
*** 1. Finish all WBM Loader operations, resolve  
encountered issues or  
*** cleanup failed and not-migrated records.  
*** 2. Execute WBMFF operation FLAGMIGRATED that, when  
successful,  
*** should unlock incremental migration operations
```

If the command finds that either the standard migration or the last incremental migration is incomplete, the command will report the tables with not-migrated records and exits with the following message:
** SA tables migrated
status data indicates incomplete WBM Load. ** Current
migration cannot be flagged until it is completed.

Aborting

When successful, this operation prepares the CAD Metadata Extractor for incremental migration and unlocks normal CAD Metadata Extractor operations.

The following graphic shows the process workflow for incremental extraction and the subsequent loading.



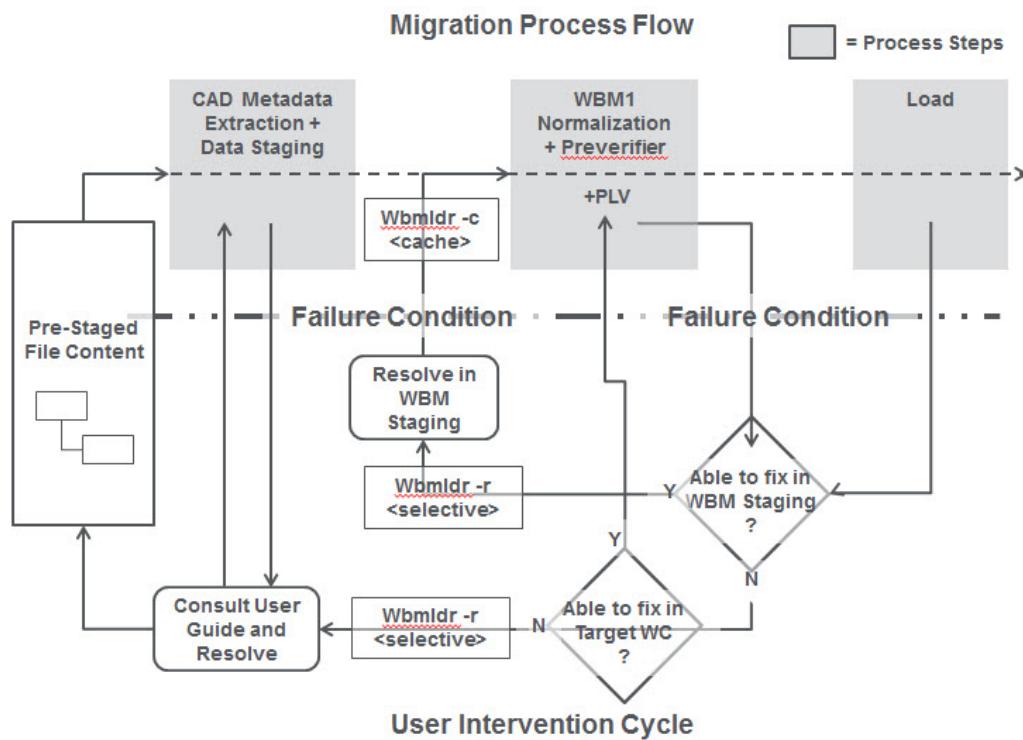
Migration Failure Resolution Options

The migration process consists of a multi-step workflow that acts on and moves the data through a number of unique process steps. Each process step can indicate unique failure conditions, or conflicts, that could be resolved by the migration user in more than one way and in more than one location. For example, the following errors could occur:

- A corrupt CAD file on a disk cannot be opened by the CAD adapter API
- Normalized data in the Windchill Bulk Migrator reports Family Table verification errors
- Data loaders report errors and reject data for any number of reasons

Depending on the error, you must determine the appropriate resolution as well as how and where to apply the resolution, sometimes using the Windchill Bulk Migrator commands.

The following graphic displays the recommended failure resolution workflow if errors are encountered during the overall migration process:



Failure Conditions During CAD Metadata Extraction and Data Staging

The intent of the CAD file processing is to extract the necessary metadata to define the corresponding EPM object in the target Windchill system as well as to ensure that you are migrating a complete data set with referential integrity. Therefore, issues that are discovered or reported by the CAD Metadata Extraction

tool as a result of processing the CAD files should be resolved by taking the appropriate cleansing action on the data stored in the file directories. Once the CAD file metadata is successfully staged (after the –operation stage command is executed), it is still necessary to complete the remaining Windchill Bulk Migrator data normalization and loading steps. The remaining steps themselves can report different types of errors, which in some situations, impact the overall migration workflow and tasks that you can take to resolve the issues.

Failure Conditions During Data Normalization and Pre-load Verification

The data normalization and pre-load verification steps both transform the CAD metadata to a state where it is ready to be loaded, and they perform analysis tasks to check for potential issues that could cause loading to fail. If errors are reported by the tool, there are two options for applying the specific resolution and cleansing steps:

1. Delete/Remove from Staging, Resolve on disk, Re-migrate: You may decide that an error condition needs to be handled by cleansing the data on disk, or by adjusting the migration user inputs, such as mapping definitions. In this case, you can run the `wbmldr` commands to remove the data from the normalized tables, take corrective action on the data set outside of the Windchill Bulk Migrator staging database, and re-execute the CAD Metadata Extraction process steps.
2. Resolve in Staging: Some error conditions can be resolved by cleansing the metadata in the Windchill Bulk Migrator staging database. This means that you are able to determine which files are problematic and also how to uniquely modify the metadata in order to resolve the issue. If this is the case, you can run the `wbmldr` commands to remove the data from the normalized tables, take corrective action, and then run the `wbmldr` command to re-cache the metadata directly to the de-normalized staging tables and re-run normalization and pre-load validation.

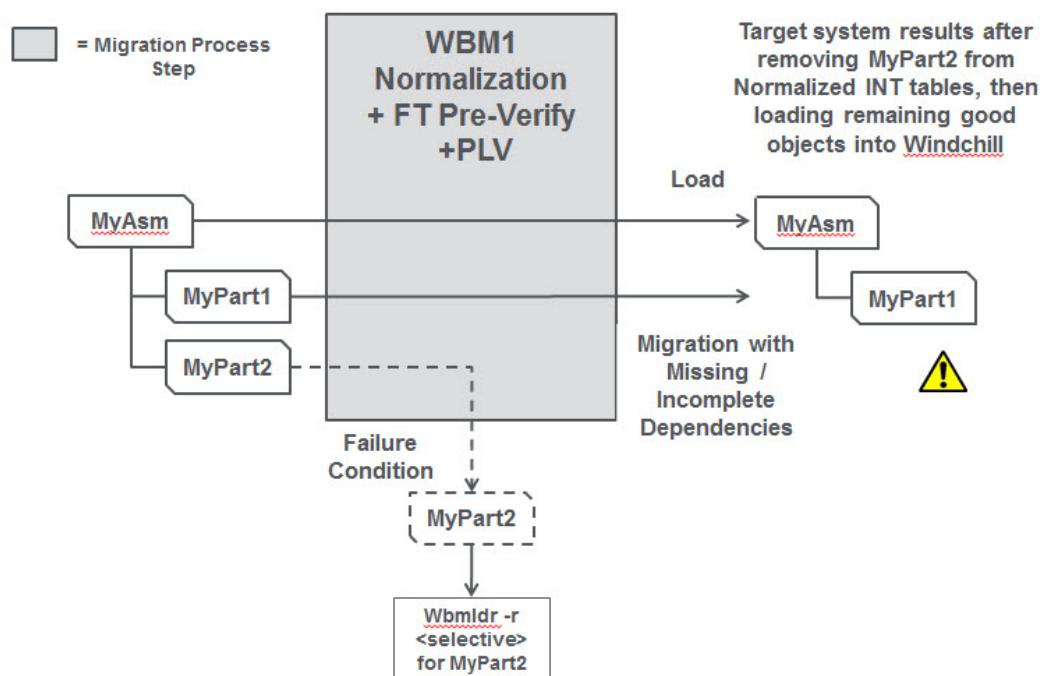
Failure Conditions During Data Loading

Data loading is a transaction-based process whereby groups of objects of one type, such as EPMDocument, are loaded together per transaction. Data loading is not based upon assembly structures or other similar seed ID based approach. Therefore, special care should be taken when considering how to handle errors encountered during loading data.

Note

It is possible to load incomplete CAD assemblies using the Windchill Bulk Migrator. Also, when migrating objects into active production domains, the CAD data structure will not be complete until all EPM and relational object types have finished loading.

For example, consider the following loading scenario illustrated below:



There are two options for resolving various loading errors:

1. Resolve in staging and reload the data. This option is a general best practice recommendation migration default. This process loop involves using the `wbmldr` commands to remove the failed objects from the normalized tables and eventually re-caching the data to the de-normalized tables and reloading

the data. It is not starting over and re-executing the CAD Metadata Extraction process. Using the `wbmldr` re-cache workflow loop retains any original referential link objects to also be migrated and established in the target system.

2. Resolve outside of staging in file system directories and re-run CAD Metadata Extraction and migration process.

 **Note**

This option does not re-establish any parent/child links. If you want to re-establish the links, the parent object must be iterated in the file system directories and migrated as an incremental data set.

15

Workflow Migration

Overview	207
Workflow Migration Scope.....	207
Workflow Migration Sequence	208
Windchill Bulk Migrator Mapping Client	210
Attribute Definition Analyzer Client.....	210
Updating the WBMWFProcessMapping.xml File	210
Validating the Workflow Template Mapping: Pre-Load Validation	214
Extracting, Staging, and Loading All Other Objects to be Migrated Using the Windchill Bulk Migrator	214
Extracting Workflow Data into the Staging Database Using the Windchill Bulk Migrator.....	215
Validating the Workflow Data Prior to Loading.....	217
Loading Workflows into the Target Windchill System	217
Publishing PBO Data From the RMD to the Public Domains	218
Initializing Workflow Tasks for Running Workflow Instances	219
Workflow Migration Staging Tables	220
Supported Workflow Process Variables for Migration.....	221

Overview

This chapter describes how to implement the Windchill Bulk Migrator for the purpose of migrating life cycle-based workflow data and processes for Windchill-to-Windchill migration use cases. The purpose of the workflow migration process is to:

- Collect information related to both completed workflow instances and open running workflow instances from the source Windchill system.
- Migrate and reinstantiate the same workflow processes on the target Windchill system. The reinstated instance accurately reflects the condition (status) of the workflow instance, node location, assigned active and completed tasks, and relation to the primary business object for a given life cycle as feasible with the given Windchill architecture.

During the target solution design process, ensure that the source Windchill workflow instance can be instantiated on the target Windchill system, as well as be resolvable by the underlying Windchill administrative configurations that control such process as team creation and assignment, role resolution, and life cycle definition. This means that just migrating the workflow data may not be enough to reinitiate the workflow instance to a usable state that reflects the condition in the source system. In other words, the successful migration of the workflow instances depends upon a combination of migration tool capabilities, workflow template equivalency, and target solution design.

Workflow Migration Scope

The workflow migration capabilities described in this chapter have boundary conditions that impact the scope of which workflow instances are able to be migrated to the target Windchill system. Workflow instances that fall outside defined boundary conditions are excluded from migration.

Immutable Boundary Conditions on Workflow Extraction:

- Source and target workflow templates must be equivalent; that is, the structure, node order, node type, and node configurations must be the same between the templates. No differences are allowed.
- Only workflow instances created from the latest iteration of workflow templates are considered for migration.
- Only workflow instances that are in a state of OPEN_RUNNING, CLOSED_COMPLETED_EXECUTED, CLOSED_COMPLETED_NOT_EXECUTED, CLOSED_TERMINATED, and CLOSED_ABORTED are considered for migration.
- The source Windchill system is offline during workflow migration; that is, changes are not allowed in the source system that could impact the primary business object (PBO) or the state of the workflow. Incremental migration of workflow instances is not allowed.

- Only workflow instances for a migration administrator-defined PBOs are considered for migration. PBO types must be an Advance life cycle-managed object type and of an object type supported by the Windchill Bulk Migrator.
- Workflow instances that are disabled are not migrated.
- Workflow instances that are suspended are not migrated.
- Workflow instances for PBOs are not be migrated if the PBO is:
 - Checked out
 - A working copy
 - Checked out in a project
 - A one-off version
 - In another application container other than product, library, or project
 - Associated to more than OPEN_RUNNING workflow instance
 - Newly created in a workspace
- Workflow instances are actively in transition; for example, the workflow instances have:
 - Running workflow robot activities
 - Yet to be started: OPEN_NOT_RUNNING_NOT_STARTED
 - Activities that are in a suspended state

Workflow Migration Sequence

The following describes the intended order of migration for the workflow instance and the related Primary Business Object (PBO).

Prerequisites

- The target Windchill solution design is complete and implemented.
- All administration data is available and frozen (for example, workflow templates, life cycle templates, containers, users, and roles).
- The source Windchill and target Windchill workflow templates are equivalent.

Migration Sequence

1. Run the Windchill Bulk Migrator Mapping Client command line utility on the source Windchill system to create the default mapping XML files for source-to-target Windchill mapping. The `mapping.xml` file allows the migration administrator to define which workflow templates are migrated using a true/false flag. In addition, the utility analyzes the workflow templates and reports

on key internal workflow template and workflow node definition keys. These definitions are needed to accurately map and populate the workflow metadata in the Windchill Bulk Migrator staging area upon extraction.

2. Run the Attribute Definition Analyzer Client command line utility on the source Windchill system to generate the data dictionary information about all of the Windchill Bulk Migrator-supported object types, including workflow.
3. Update the `WBMWFProcessMapping.xml` file by setting the appropriate flag to specify which workflow template instances to migrate. Additionally, look for and document the target Windchill system workflow template and workflow node definition keys. These keys correspond to the persistable identifier (`ida2a2`) values that are assigned to the objects in the Windchill database. These definitions are needed to accurately map and populate the workflow metadata in the Windchill Bulk Migrator staging area upon extraction.
4. Complete the `workflow mapping.xml` template by entering the target Windchill `ida2a2` keys for the workflow template and workflow node definitions. This defines and maps the equivalency of the workflow structure between source and target.
5. Validate the workflow template mapping. Launch the Windchill Bulk Migrator Pre-Load Validation (PLV) utility and run the PLV tasks for data mapping and workflow process.
6. Extract, stage, and load all other objects required for migration from the source Windchill system to the target Windchill system using the Windchill Bulk Migrator. It is mandatory that the PBOs that are referenced by the workflow instances be available in the target system to recreate the workflow instance on the target system. During this process, it is recommended to run PLV data validation tasks for all of the object types that are being migrated.
7. Extract workflow data into the staging database using the Windchill Bulk Migrator. Optionally, use the Windchill Bulk Migrator to suspend running workflows on the source system after successful extraction to ensure the workflow does not change after extraction.
8. Validate the workflow data prior to loading. Launch the Windchill Bulk Migrator Pre-Load Validation (PLV) utility and run the PLV tasks for data mapping and workflow process.
9. Load workflow data into the target Windchill system using the Windchill Bulk Migrator to create the workflow instance, associate the workflow to the PBO, create and enqueue the active workflow tasks for open running instances.
10. Publish PBO data as well as all objects referenced by workflow process variables from the restricted migration domain to the public domains using the Windchill Bulk Migrator.
11. Start the workflow queues to distribute and assign the workflow tasks for the open, running workflow instances.

Windchill Bulk Migrator Mapping Client

The Windchill Bulk Migrator mapping client analyzes the source Windchill system and creates the administrative `mapping.xml` templates. Run the client on the source Windchill system by entering the following command from a Windchill shell:

```
windchill  
com.ptc.windchill.migration.wbm.wc2wc.mapping.  
client.WBMMappingClient <wtuser> -p <wtpassword>
```

The relevant output for workflow information will be in `WBMWFProcessMapping.xml` file in the mapping output directory.

Attribute Definition Analyzer Client

The attribute definition analyzer client generates the data dictionary for all of the Windchill Bulk Migrator supported object types. The client utility uses introspection of the source Windchill system on the object types supported by the Windchill Bulk Migrator. The Windchill Bulk Migrator extractors use the data dictionary information and the Windchill Bulk Migrator `ClassMap.csv` file to extract, map, and transform the data, then use it to populate the staging database tables.

Run the client on the source Windchill system by entering the following command from a Windchill shell:

```
windchill  
com.ptc.windchill.migration.wbm.wc2wc.util.AttDefAnal  
izerClient -a
```

Updating the `WBMWFProcessMapping.xml` File

Setting the Migration Flag (True/False) for Workflow Templates

The mapping file lists all workflow templates found in the source system. Each template can be marked with a flag to control whether the workflow instances associated with the template are considered for extraction. The flag in the `mapping.xml` file is: `toBeMigrated=<true/false>`.

The default setting for each template is true.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <WBMExtractorMapping>
  - <WfProcessTemplates>
    + <WfProcessTemplate toBeMigrated="true">
    + <WfProcessTemplate toBeMigrated="true">
```

Finding and Documenting the Target Windchill Workflow and Node ida2a2 Keys

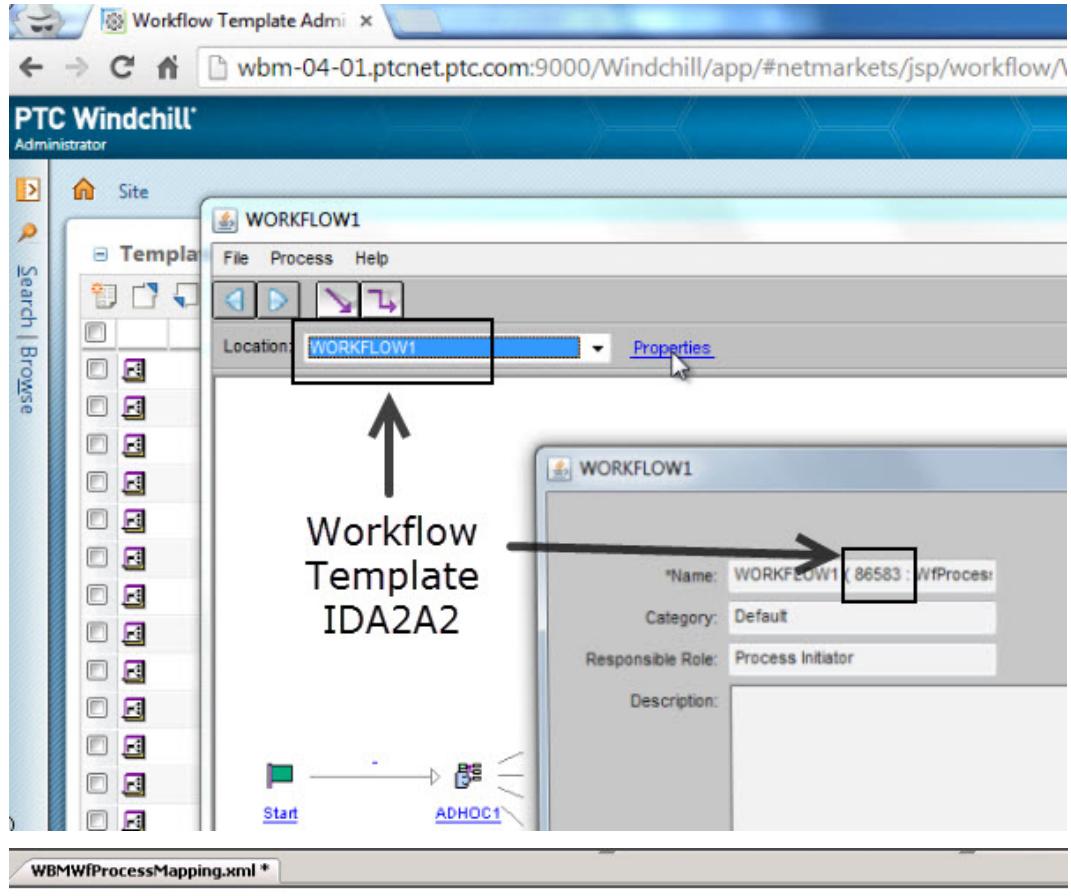
The workflow mapping activity must identify a variety of ida2a2 mappings from source system workflow template to the target system workflow template. The following sections describe the different methods to identify the key mappings.

Workflow Template ida2a2

For the workflow template, the ida2a2 value should be entered for the corresponding `<Target id=" "Name="WorkflowName">` entry in the `WBMWFPProcessMapping.xml` file.

Identification of the workflow template ida2a2 is show by the following graphics.

From the target Windchill Workflow Template Administrator, select the workflow template and review the properties.



```

</WfProcessTemplate>
<WfProcessTemplate toBeMigrated="true">
<Context>
  <Source id="86583" Name="WORKFLOW1">
    <ContainerName>Site</ContainerName>
    <ContainerType>SITE</ContainerType>
    <ContainerOrgName />
  </Source>
  <Target id="7" Name="WORKFLOW1">
    <ContainerName>Site</ContainerName>
    <ContainerType>SITE</ContainerType>
    <ContainerOrgName />
  </Target>
</Context>
<ProcessVariables>
  <ProcessVariable Type="int" SrcName="decisionvar" TargetName="decisionvar" />
  <ProcessVariable Type="wt.part.WTPart" SrcName="wtpart" TargetName="wtpart" />
  <ProcessVariable Type="wt.pdmLink.PDMLinkProduct" SrcName="pdmLink" TargetName="pdmLi
  <ProcessVariable Type="wt.folder.SubFolder" SrcName="folder" TargetName="folder" />
  <ProcessVariable Type="wt.folder.Cabinet" SrcName="cabinet" TargetName="cabinet" />
  <ProcessVariable Type="wt.doc.WTDocument" SrcName="wtdoc" TargetName="wtdoc" />
  <ProcessVariable Type="wt.org.WTUUser" SrcName="wtuser" TargetName="wtuser" />
  <ProcessVariable Type="java.lang.String" SrcName="string1" TargetName="string1" />
  <ProcessVariable Type="boolean" SrcName="boolean1" TargetName="boolean1" />
  <ProcessVariable Type="int" SrcName="integer1" TargetName="integer1" />
  <ProcessVariable Type="float" SrcName="float1" TargetName="float1" />
  <ProcessVariable Type="int" SrcName="r" TargetName="r" />
</ProcessVariables>
<WfBlockRef id="86596" Name="BLOCK1" />
<WfBlockRef id="86621" Name="ROBOT" />
<Activities>
```

Enter Target Workflow IDA2A2 for Target id

Node ida2a2

The target Windchill system workflow and node ida2a2 keys are required to complete the workflow template mapping from the source to target system. These values can be discovered by implementing the following properties in `wt.properties`:

```
wt.clients.workflow.definer.displayTemplateNodeIdIdentifier=All
```

After setting the property it will be possible to see the ida2a2 values from the Workflow Definer Viewer.

The following graphic shows the display of the ida2a2 values after setting the property:

The screenshot shows a portion of a configuration file with several annotations:

- A yellow dashed box highlights the `<Target id="86583" Name="WORKFLOW1">` element.
- A red box highlights the `<Source>` element within the `<Activity>` section.
- An annotation "○ ○" is placed next to the `<Target id="" Name="ADHOC1">` element.
- An annotation "All WF Nodes must have IDA2A2 mapping" is placed next to the `<Type>wt.workflow_definer.WfAdHocActivityTemplate</Type>` element, with an arrow pointing to it.
- A red arrow points from the `<Source>` element to the `<Target id="" Name="ADHOC1">` element.

```
</Source>
<Target id="86583" Name="WORKFLOW1">
  <ContainerName>Site</ContainerName>
  <ContainerType>SITE</ContainerType>
  <ContainerOrgName />
</Target>
</Context>
<ProcessVariables>
  <ProcessVariable Type="int" SrcName="decisionvar" TargetName="decisionvar" />
  <ProcessVariable Type="wt.part.WTPart" SrcName="wtpart" TargetName="wtpart" />
  <ProcessVariable Type="wt.pdmlink.PDMLinkProduct" SrcName="pdmlink" TargetName="pdml" />
  <ProcessVariable Type="wt.folder.SubFolder" SrcName="folder" TargetName="folder" />
  <ProcessVariable Type="wt.folder.Cabinet" SrcName="cabinet" TargetName="cabinet" />
  <ProcessVariable Type="wt.doc.WTDocument" SrcName="wtdoc" TargetName="wtdoc" />
  <ProcessVariable Type="wt.org.WTUser" SrcName="wtuser" TargetName="wtuser" />
  <ProcessVariable Type="java.lang.String" SrcName="string1" TargetName="string1" />
  <ProcessVariable Type="boolean" SrcName="boolean1" TargetName="boolean1" />
  <ProcessVariable Type="int" SrcName="integer1" TargetName="integer1" />
  <ProcessVariable Type="float" SrcName="float1" TargetName="float1" />
  <ProcessVariable Type="int" SrcName="r" TargetName="r" />
</ProcessVariables>
<WfBlockRef id="86596" Name="BLOCK1" />
<WfBlockRef id="86621" Name="ROBOT" />
<Activities>
  <Activity>
    <Source id="86594" Name="ADHOC1">
      <Type>wt.workflow_definer.WfAdHocActivityTemplate</Type>
    </Source>
    <Target id="" Name="ADHOC1">
      <Type>wt.workflow_definer.WfAdHocActivityTemplate</Type>
    </Target>
  </Activity>
  <Activity>
    <ProcessVariables>
      <ProcessVariable Type="wt.part.WTPart" SrcName="wtpart" TargetName="wtpart" />
    </ProcessVariables>
  </Activity>
</Activities>
```

“And” Connector ida2a2

Note

Start and End connector idA2A2 can only be obtained in the following ways:

1. Query the target system database.
2. Use the export dump of the workflow template.

Workflow process/activity variables can also be mapped using the variable name; assuming that variable type in both the source and the target are the same.

Validating the Workflow Template Mapping: Pre-Load Validation

The Windchill Bulk Migrator includes a Pre-Load Validation utility that performs multiple types of data checks. For workflows, the PLV utility helps to validate that the `ida2a2` mappings for the source and target workflow templates are valid.

Launch the PLV utility from a target Windchill shell using the following command:

```
windchill  
com.ptc.windchill.migration.wbm.preloadvalidator.ui.Pre  
LoadValidationManager
```

To validate the workflow template mappings perform the following steps once the PLV utility launches:

1. Select the **Workflow Templates** check box from the Target Administrative Data Snapshot list.
2. Select **Fetch Data**. The PLV utility then contacts the target Windchill server to obtain information about the target system workflow templates. Then, compare the information against the mapping values defined in `WBMWFProcessMapping.xml`. After the data has been fetched, select **Next**. The utility will go back to the start screen.
3. Select **User Defined Mapping** from the Validations (type) options.
4. Select **Choose Folder** for User Defined Mapping. Navigate to, and select the folder that contains the `mapping.xml` files that were created from running the Windchill Bulk Migrator Mapping Client.
5. From the start screen, select **Validate**. When the message window (in the lower left) indicates a condition of “Validated”, the task is complete.
6. Review the PLV report. The report is located at: `<Windchill>\logs\WBM\reports\preloadverification\<Date>\PreLoadVerificationReport_Mapping.html`
7. Once the report shows an error count of 0 for the workflow validation tasks, you can execute the extractor for workflow data.

Extracting, Staging, and Loading All Other Objects to be Migrated Using the Windchill Bulk Migrator

The migration of workflow data should be last in the order of migration. This means that all other object types should be migrated before migrating workflow data. For workflows, it is important that the workflow PBOs exist in the target system.

Migrate all objects into the target Windchill system using the Windchill Bulk Migrator before proceeding onto workflow migration.

Extracting Workflow Data into the Staging Database Using the Windchill Bulk Migrator

A workflow is a business process modeled through a set of objects in Windchill. The Windchill Bulk Migrator extracts workflows related to a user-specified PBO, where the PBO must be a `LifeCycleManaged` object type. For a given workflow process, information about all related objects are collected and extracted in the same transaction populating relevant records and tables in the staging database.

By default, the workflows that are extracted all have one of the following states:

- `CLOSED_COMPLETED_EXECUTED`
- `OPEN_RUNNING`
- `CLOSED_COMPLETED_NOT_EXECUTED` (Skipped)
- `CLOSED_TERMINATED` (Terminated without execution)
- `CLOSED_ABORTED` (Aborted)

The following extractor command should be run from a source Windchill shell:

```
Windchill
com.ptc.windchill.migration.wbm.wc2wc.extractor.client.
WindchillDataExtractor -o <PBO ClassTypeName>-wf -u <admin
user> -p<admin password>
```

For example, if a migration administrator wants to extract workflows that had `WTPart` objects defined as a PBO, they would execute the following command:

```
Windchill
com.ptc.windchill.migration.wbm.wc2wc.extractor.client.
WindchillDataExtractor -o wt.part.WTPart -wf -u wcadmin -p
wcadmin
```

This step should be repeated for each PBO object type.

Note

The workflow gathers all associated objects such as sub-processes, `WfBlock`, various activities, assignments, tasks, and audit information and places them in respective staging tables.

Workflow Extraction Filtering Options

The Windchill Bulk Migrator allows different forms to specify which workflows are extracted. The associated use cases occur if the migration administrator only wishes to extract a certain subset of workflows.

There are three ways to control or filter which workflow instances are extracted:

1. Workflow Template Name—this method is controlled by the `toBeMigrated` flag in the `WBMWFProcessMapping.xml` file.
2. Workflow State—the workflow state depends on the results from fired transition events as different workflow nodes are executed and completed. Examples of the default workflow states that will be migrated are: `OPEN_RUNNING` and `CLOSED_COMPLETED_EXECUTED`.
3. PBO Lifecycle State—the Windchill Bulk Migrator extracts all workflow instances for a given PBO by default. However, it is also possible to configure the extract utility to only consider workflow instances for a given PBO at a specific life cycle state.

Filtering conditions for workflow state and PBO life cycle state can be defined in the `extract.xconf` file. The desired entry form must follow the following format:

```
wt.workflow.engine.WfProcess|<PBO ClassTypeName>=
WFState:<WFState>|LCState:<LCState>
```

Example `extract.xconf` entries:

```
wt.workflow.engine.WfProcess|wt.part.WTPart=
WFState:CLOSED_COMPLETED_EXECUTED|LCState:APPROVED
```

The above entry extracts workflow instances that are closed and completed for WTParts at a life cycle state of APPROVED.

Other Workflow Extraction Options

There are optional source Windchill or Windchill Bulk Migrator configurations that can be set to control system or tool behavior during workflow extraction.

Suspend Workflows

The Suspend Workflows option enables the migration administrator to suspend all open running workflows on the source Windchill system upon a successful extraction. Using this option prevents the workflow from further progressing and prevents further change in the PBO and associated persistent variables.

This option is enabled by setting the following `extract.xconf` property:

```
data.extract.workflow.suspendProcess=<true/false>
```

By default, this property is set to `false`.

If this option is set to `true`, the Windchill Bulk Migrator takes a snapshot of all running activities and records them in the `RUNNINGWFSUSPENSIONAUDIT` staging table.

To resume suspended workflows on the source Windchill system, execute the following command from a Windchill shell:

```
java  
com.ptc.windchill.migration.wbm.wc2wc.extractor.  
client.WfProcessResumeHandler
```

Validating the Workflow Data Prior to Loading

Launch the pre-load validation utility from a target Windchill shell using the following command:

```
windchill  
com.ptc.windchill.migration.wbm.preloadvalidator.ui.Pre  
LoadValidationManager
```

To validate the workflow template mappings, perform the following steps once the pre-load validation utility launches:

1. Select **Staged Data** from the **Validations (type)** options.
2. Select **WFPROCESS** and the **Workflow PBO** class from the list of available object types to validate, and add them to the process list.
3. Click **Validate** and authenticate the target Windchill system when prompted. When the message window appears indicating **Validated**, the task is complete.
4. If no errors are reported for the workflow validation tasks, execute the Windchill Bulk Migrator loader tool for workflow data.

Loading Workflows into the Target Windchill System



Note

Ensure that all PBO types have been loaded into the target Windchill system before continuing.

Launch the `WBMDataloadController` from a target Windchill system shell using the following command:

```
windchill
com.ptc.windchill.migration.wbm.DataLoadRouterImp -u
<user> -p<password>
```

To load the workflows, run the workflow process loader using the following command:

```
windchill
com.ptc.windchill.migration.wbm.DataLoadController -l
WFPROCESS
```

Workflow loading creates the workflow instance on the target system. While preparing a load transaction, the workflow collects all associated objects (sub workflow, blocks, activities, assignment, tasks, and audit information from various tables in staging), and saves them with workflow objects in an integral manner. In addition, it also enqueues the open tasks associated with an active workflow. At this point, the tasks are not yet initialized. The enqueued active tasks must be initialized in order to be picked up by the workflow instance and continue with the current and downstream tasks. Initialization should occur after ensuring that the PBO has been published from the Windchill Bulk Migrator Restricted Migration Domain (RMD) to the Public Domains.

Publishing PBO Data From the RMD to the Public Domains

To move migrated data to Windchill public domains, enter the following command from a target system Windchill shell:

```
java
com.ptc.windchill.migration.wbm.DataPublishController
```

Initializing Workflow Tasks for Running Workflow Instances

Active tasks for the running workflow are enqueued for initiation in special migration queues on the target Windchill system. Three workflow queues are created once workflow data is loaded into Windchill:

WfUserWorkQueue	Pool	Started	Enabled	Default	0
WfScheduleQueue	Schedule	Started	Enabled	Default	0
WfPropagationQueue	Pool	Started	Enabled	Default	0
WBMWfPublishingQueue_2	Process	Stopped	Disabled	Default	0
WBMWfPublishingQueue_1	Process	Stopped	Disabled	Default	2
WBMWfPublishingQueue_0	Process	Stopped	Disabled	Default	5
SubStringIndexSchedule	Schedule	Started	Enabled	Default	1
shadowCacheScheduleQueue	Process	Started	Enabled	Default	0

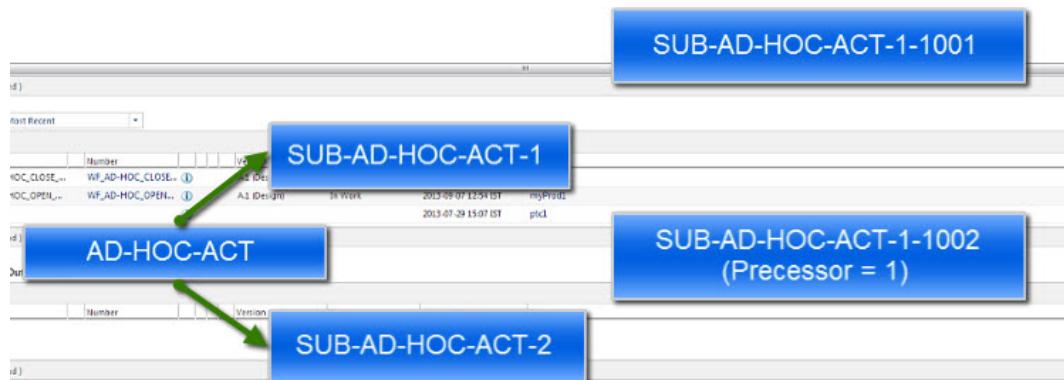
Enable and start the WBMWfPublishingQueue to initialize the workflow tasks. This brings the active workflow to the identical active state as it was in the source, and allows the process to continue from thereon.

Ad-Hoc Task Migration

Active ad-hoc tasks of an open and running workflow in the source system are recreated in the target system. That is, it is possible that ad-hoc tasks which were created on the fly in the source system will not be brought from the source, and you will have to re-execute those tasks on the target after re-initialization.

Example workflow:

In the following graphic, AD-HOC-ACT is the originating task that creates the downstream tasks as shown:



The graphic below highlights the source system tasks:

	Name	Assignee	Role	Vote
✓	SUB-AD-HOC-ACT-OPEN-2	testuser1	Assignee	
	SUB-AD-HOC-ACT-OPEN-1...	testuser1	Assignee	
✓	SUB-AD-HOC-ACT-OPEN-1-1001	testuser1	Assignee	
	SUB-AD-HOC-ACT-OPEN-1	testuser1	Assignee	
	AD-HOC-ACT	testuser1	Assignee	

After the migration, there is an open task on the target system at the initial root:

Process Status				
	Name	Assignee	Role	Vote
	AD-HOC-ACT	testuser1	Assignee	

Workflow Migration Staging Tables

The following staging database tables are used by the Windchill Bulk Migrator to migrate workflow data:

Workflow Object Staging Table	Windchill Bulk Migrator Status Table
WFADHOCACTIVITY	WFADHOCACTIVITYWBMSTATUS
WFASSGMTEVENTAUDIT	WFASSGMTEVENTAUDITWBM-STATUS
WFASSIGNEDACTIVITY	WFASSIGNEDACTIVITYWBMSTA-TUS
WFASSIGNMENT	WFASSIGNMENTWBMSTATUS
WFBASICROBOTACTIVITY • wt.workflow.robots.WfExternalMethod • wt.workflow.robots.WfInternalMethod • wt.workflow.robots.WfExpressionRobot • wt.workflow.robots.WfApplicationRobot	WFBASICROBOTACTIVITYWBM-STATUS
WFBLOCK	WFBLOCKWBMSTATUS
WFPROCESS	WFPROCESSWBMSTATUS

Workflow Object Staging Table	Windchill Bulk Migrator Status Table
WFREQUESTERACTIVITY	WFREQUESTERACTIVITYWBM-STATUS
WFSYNCHROBOT	WFSYNCHROBOTWBMSTATUS
WFTIMERACTIVITY	WFTIMERACTIVITYWBMSTATUS
WFURLROBOT	WFURLROBOTWBMSTATUS
WFVOTINGEVENTAUDIT	WFVOTINGEVENTAUDITWBM-STATUS
WORKITEM	WORKITEMWBMSTATUS

These tables are considered internal tables for use by the Windchill Bulk Migrator only. The Wfprocess loader fetches data from these tables to fully form the Windchill Bulk Migrator loader transactions.

Note

In the current implementation, status tables other than WFPROCESS have been dropped. All internal tables have respective audit tables in target Windchill database.

Supported Workflow Process Variables for Migration

Only the workflow variables that are supported for migration are populated in the WBMWfProcessMapping.xml file during map file generation.

The supported variable types that are extracted are:

- BOOLEAN
- BYTE
- CHAR
- DOUBLE
- FLOAT
- INT
- LONG
- SHORT
- String.class
- URL.class
- TimeStamp.class

-
- Date.class
 - WfDueDate class
 - WfInstruction class
 - WfDuration class
 - WfAssigneeNotification class
 - WTGroup class
 - ObjectReference.class
 - Any administrative data coming from the extractor mapping files
 - Objects supported by the Windchill Bulk Migrator through the loader
 - WTObject.class
 - Any administrative data coming from the extractor mapping files
 - Objects supported by the Windchill Bulk Migrator through loader
 - Persistable.class
 - Any administrative data coming from the extractor mapping files
 - Objects supported by the Windchill Bulk Migrator through the loader
 - Team
 - Team instances that are associated to the TeamManaged Persistable objects supported for migration by the Windchill Bulk Migrator
 - WTPrincipalReference, that will cover following references:
 - WTUser
 - WTOrganziation

16

Windchill Bulk Migrator Utilities Performance Considerations

Extraction Utilities	224
Loader Infrastructure	225

Extraction Utilities

Extraction is multi-threaded and threads are spread evenly between all foreground method servers. The following properties impact performance:

- `data.extract.threadCount`

This property controls the total number of threads for all foreground method servers. The general rule is to have two or three threads per foreground method server. Since extraction threads consume a lot of memory and SQL connections, the higher number of threads does not provide additional performance benefits since threads start to wait for each other. Too many extraction threads run on a method server could cause the method server to throw an out of memory exception.

- `data.extract.threadCountModulus`

Controls the number of items (masters) to be extracted in a single thread. The items are then extracted in batches. By default, this property is set to 2000 in `extract.xconf`. Optimal value of the property depends on individual setup. In general, it should be the value that strikes the balance between method servers used memory that grows with the `threadCountModulus` and network communications that decrease with the `threadCountModulus`.

The `data.extract.threadCountModulus` can also be set to 0. When set to 0, the extractor programmatically calculates the number of items per thread during the extraction. However, the calculation itself may take a long time, so the overall performance is not as good as explicitly setting the optimal value.

- `data.extract.threadCountModulusLimit`

Specifies the maximum number of items to extract in the single thread. If the `data.extract.threadCountModulus` property (either programmatically calculated when it is 0 or user specified) exceeds the `data.extract.threadCountModulusLimit` property, the number of items per thread will use the value specified in `data.extract.threadCountModulusLimit` to avoid running out of memory when `data.extract.threadCountModulus` is too large.

- `data.extract.BatchSize`

Controls the size of an extraction batch. Items in a batch are inflated together, meaning that their iterations, global attributes, standard attributes (if applicable), and references are extracted and inflated in a single call to Windchill. All extracted data in the batch are committed to the staging database in one transaction. If the batch size is too big or too small, the benefits of Windchill's multi-object API will diminish.

- `data.extract.dbBatchSize`
Controls the size of extracted items batch to be inserted into staging area.
- Connection Pooling Settings: Following properties can be located in `extract.properties`
 - `data.extract.commonConnectionPool.maxStagingPoolSize = 25` (default value 5)
 - `data.extract.commonConnectionPool.maxWindchillPoolSize = 25` (default value 5)
 - `data.extract.commonConnectionPool.minStagingPoolSize = 10` (default value 1)
 - `data.extract.commonConnectionPool.minWindchillPoolSize = 10` (default value 1)

For more details, refer to the section '[Extraction Properties of Connection Pooling on page 68](#)

Loader Infrastructure

The following properties are related to the Windchill Bulk Migrator load thread configuration. You can modify these properties in `wbm.properties` to improve performance:

- `com.ptc.windchill.migration.wbm.fetchBufferSize = 5000` (default value 5000)
- `com.ptc.windchill.migration.wbm.numberOfLoadThreadsToStart = 16` (default value 4)
- `com.ptc.windchill.migration.wbm.loadThreadQueue.-max.size = 2` (default value 10)
- `com.ptc.windchill.migration.wbm.loadThreadQueue.-min.size = 1` (default value 5)
- `com.ptc.windchill.migration.wbm.loaders.nonVersionedObjectTransactionSize = 5000` (default value 5000)

The following settings control the number of database connections that the Windchill method server possesses.

- `wt.pom.maxDbConnections = 20` (default value 15)
- `wt.pom.minDbConnections = 5` (default value 5)
- ConnectionPool setting: The following properties impact performance as they handle all the connections to the Staging Database. These properties located in `wbm.properties`.

For more details, refer to section [ConnectionPool Settings on page 65](#).

- `data.loader.commonConnectionPool.maxPoolSize = 30`
(default value 10)
- `data.loader.commonConnectionPool.minPoolSize = 20`
(default value 5)
- Disable indexing while bulk loading
Disable indexing by setting all the properties with suffix `index.enabled` to false.
 - `wt.index.enabled`
 - `com.ptc.windchill.partslink.index.enabled`
 - `com.ptc.windchill.ata.index.enabled`
 - `com.ptc.windchill.rules.index.enabled`

 **Note**

Optimal value of all the properties for performance depends on individual setup configuration.

17

Customizing the Windchill Bulk Migrator

Understanding the classMap.csv File 228

Understanding the classMap.csv File

The `classMap.csv` lists the mapping of an object type (Java class name) to the staging database table and the object properties to the staging database table columns.

Class Row

Example:

```
CLASS:wt.part.WTPart, TABLE:WTPART, IBA_TABLE:ATT_WTPART, VIEW:VEW_WTPART, IBA_VIEW:ATV_WTPART
```

In the example, `wt.part.WTPart` objects are extracted to staging table `WTPART`. The soft attributes are extracted to staging table `ATT_WTPART`. After successful normalization, data in staging table `WTPART` and `ATT_WTPART` is removed. However, you can still see the similar data in the `VEW_WTPART` and `ATV_WTPART` views.

Referenced Classes Combination Row

There are Windchill classes whose object reference can have different object types. Those Windchill objects are extracted to different staging tables based on the types of its object references.

For example, `wt.change2.FormalizedBy` has `roleBObjectRef` that can either be `wt.change2.WTChangeIssue` or `wt.change2.WTVariance`.

Example:

```
CLASS:wt.change2.FormalizedBy REFERENCED_CLASSES_COMBINATION:roleBObjectRef:wt.change2.WTChangeIssue, roleAObjectRef:wt.change2.WTChangeRequest2, TABLE:FORMALIZEDBY_547627207, VIEW:VEW_FORMALIZEDBY_547627207, REFERENCED_CLASSES_COMBINATION:roleBObjectRef:wt.change2.WTVariance, roleAObjectRef:wt.change2.WTChangeRequest2, TABLE:FORMALIZEDBY_1923749937, VIEW:VEW_FORMALIZEDBY_1923749937,
```

This means that the `FormalizedBy` object is extracted to `FORMALIZEDBY_547627207` staging table if `roleBObjectRef` is `wt.change2.WTChangeIssue` and `roleAObjectRef` is `wt.change2.WTChangeRequest2`. The `FormalizedBy` object is extracted to the `FORMALIZEDBY_1923749937` staging table if `roleBObjectRef` is `wt.change2.WTVariance` and `roleAObjectRef` is `wt.change2.WTChangeRequest2`.

There are cases where the staging tables are decided by nested object reference types. Nested object reference means the object reference of the object reference.

Example:

```
CLASS:wt.change2.VarianceBlockEffectivity REFERENCED_
CLASSES_
COMBINATION:effContextReference:wt.part.WTPartMas
ter,targetReference:wt.change2.ReportedAgainst|ro
leAObjectRef:wt.change2.WTChangeIssue,targetReferen
ce:wt.change2.ReportedAgainst|roleBObjectRef:wt.
part.WTPart,TABLE:VARIANCEBLKEFF_614909374,VIEW:VEW_
VARIANCEBLKEFF_614909374,
```

This means that the VarianceBlockEffectivity is extracted to VARIANCEBLKEFF_614909374 staging table if effContextReference is wt.part.WTPartMaster, its targetReference is wt.change2.ReportedAgainst, its targetReference's roleAObjectRef is wt.change2.WTChangeIssue and its targetReference's roleBObjectRef is wt.part.WTPart.

Sometimes the object reference or the nested object reference may be null; this is called optional reference. The following rows:

```
CLASS:wt.change2.VarianceBlockEffectivity REFERENCED_
CLASSES_
COMBINATION:effContextReference:NULL,targetReferen
ce:wt.change2.ReportedAgainst|roleAObjectRef:wt.chan
ge2.WTChangeIssue,targetReference:wt.change2.Reporte
dAgainst|roleBObjectRef:wt.part.WTPart,TABLE:VARIAN
CEBLKEFF_614909374,VIEW:VEW_VARIANCEBLKEFF_
614909374,
```

This means that the VarianceBlockEffectivity is extracted to VARIANCEBLKEFF_614909374 staging table if its effContextReference is NULL, its targetReference is wt.change2.ReportedAgainst, its targetReference's roleAObjectRef is wt.change2.WTChangeIssue and its targetReference's roleBObjectRef is wt.part.WTPart.

Keyword WBM.SOURCECLASS and WBM.SOURCEID

The value of WBM.SOURCECLASS gives the general description of the extracted class. For Windchill extractions, the value is the Java full class name of the extracted class.

The value of WBM.SOURCEID is the identifier of the extracted object. For Windchill extractions, the value is the ida2a2 of the extracted source object. Both attributes are optional.

Examples:

```
CLASS:wt.part.WTPart,TABLE:WTPART,IBA_TABLE:ATT_
WTPART,VIEW:VEW_WTPART,IBA_VIEW:ATV_WTPART COL_
TYPE:NORMAL,WBM.SOURCECLASS,WBMSOURCEDESCRIPTION COL_
TYPE:NORMAL,WBM.SOURCEID,WBMSOURCEIDENTIFIER COL_
```

```
TYPE:OBJREF_
DATA, masterReference | WBM.SOURCEID, MASTERWBMSOURCEI
DENTIFIER
```

Object Reference row

Example:

```
CLASS:wt.part.WTPart, TABLE:WTPART, IBA_TABLE:ATT_
WTPART, VIEW:VIEW_WTPART, IBA_VIEW:ATV_WTPART COL_
TYPE:OBJREF_
DATA, masterReference | WBM.SOURCEID, MASTERWBMSOURCEI
DENTIFIER
```

This means that the `wt.part.WTPart` has the object reference whose property name is `masterReference`. The `ida2a2` of the persistable object corresponding to `masterReference` are extracted to the `MASTERWBMSOURCEIDENTIFIER` column.

IBA Row

Example:

```
CLASS:wt.part.WTPart, TABLE:WTPART, IBA_TABLE:ATT_
WTPART, VIEW:VIEW_WTPART, IBA_VIEW:ATV_WTPART COL_
TYPE:IBA_TABLE_
NORMAL, WBM.SOURCEID, IBAHOLDERWBMSOURCEIDENTIFIER
```

This means that the `ida2a2` of `WTPart` is extracted to the `IBAHOLDERWBMSOURCEIDENTIFIER` column in the `ATT_WTPART` IBA table.

Imbedded Row

Sometimes, Windchill models a couple of persistent fields into one single class as the type for the instance variable of the main object type that is extracted, but the class itself is not the reference class, such as `wt.part.LineNumber` and `wt.part.Quantity`. Those classes are called imbedded classes. The extractor infrastructure hardcodes the handling of those imbedded classes. For example:

```
COL_TYPE:IMBEDDED_DATA,<pqw>,<theSAColumnName>
```

The extractor framework has special handling to get the desired information for the `pqw`. Here, `pqw` represents one of the primitive persistent fields in the imbedded class.

Example:

```
CLASS:wt.part.WTPartUsageLink, TABLE:WTPARTUSAGELINK
K, IBA_TABLE:ATT_WTPARTUSAGELINK, VIEW:VIEW_WTPARTUSAGELINK, IBA_VIEW:ATV_WTPARTUSAGELINKCOL_
TYPE:IMBEDDED_DATA, lineNumber.value, LINENUMBER_COL_
TYPE:IMBEDDED_DATA, quantity.amount, AMOUNT_COL_
TYPE:IMBEDDED_DATA, quantity.unit, UNITOFMEASURE
```

It is possible that you may encounter other imbedded classes which the Windchill Bulk Migrator extractor does not handle out-of-the-box. In this case, you must write your own appender if you want to extract information from those classes. However, you should not use the keyword COL_TYPE:IMBEDDED_DATA. Rather, use COL_TYPE:NORMAL in the classMap.csv.

Normal Row

For other modeled attributes and the attributes inherited from a number of interfaces, the COL_TYPE:NORMAL is used:

```
COL_TYPE:NORMAL,<pqw>,<theSAColumnName>
```

In the above example, pqw can be one of the following:

- The modeled attribute name from the introspection properties on the main object type specified with CLASS:. The pqw should exist in the data dictionary file of the main object type.
- The reserved attribute name coming from the common interfaces the extractor handles. The following table provides a list:

Reserved Attribute Name	Common Interfaces that the Extractor Handles
createStamp	wt.fc.Persistable
modifyStamp	wt.fc.Persistable
updateStamp	wt.fc.Persistable
iterationInfo	wt_vc.Iterated
versionInfo	wt_vc.Versioned
location	wt.folder.Foldered
owner	wt.ownership.Ownable
state.lifeCycleId.name	wt.lifecycle.LifeCycleManaged
state.state	wt.lifecycle.LifeCycleManaged
teamTemplateId.name	wt.team.TeamManaged
typeDefinitionReference	wt.type.Typed
view.name	wt_vc.views.ViewManageable
organizationReference.name	wt.org.OrganizationOwned
containerReference.organizationReference.name	wt.inf.container.WTContained
containerReference.name	wt.inf.container.WTContained
CNTAINER_TYPE	wt.inf.container.WTContained
creator.name	wt.enterprise.FolderResident wt.enterprise.Managed wt.org.OrganizationOwnedVesion

Reserved Attribute Name	Common Interfaces that the Extractor Handles
iterationInfo.modifier.name	wt.vc.Iterated
format	wt.content.FormatContentHolder

Example:

```

CLASS:wt.part.WTPart, TABLE:WTPART, IBA_TABLE:ATT_WTPART, VIEW:VIEW_WTPART, IBA_VIEW:ATV_WTPART, SL_TABLE:SL_WTPART, SL_VIEW:SLV_WTPART COL_TYPE:NORMAL, versionInfo, REVISION
COL_TYPE:NORMAL, view.name, VIEWNAME
COL_TYPE:NORMAL, variation1, VARIATION1
COL_TYPE:NORMAL, variation2, VARIATION2
COL_TYPE:NORMAL, contractNumber, CONTRACTNUMBER
COL_TYPE:NORMAL, expression, EXPRESSION
COL_
TYPE:NORMAL, jobAuthorizationNumber, JOBAUTHORIZATIONNUMBER
COL_TYPE:NORMAL, maximumAllowed, MAXIMUMALLOWED
COL_TYPE:NORMAL, minimumRequired, MINIMUMREQUIRED
COL_TYPE:NORMAL, partType, PARTTYPE
COL_TYPE:NORMAL, phase, PHASE
COL_TYPE:NORMAL, source, SOURCE
COL_TYPE:NORMAL, location, FOLDERPATH
COL_TYPE:NORMAL, iterationInfo, ITERATION
COL_TYPE:NORMAL, iterationInfo.note, ITERATIONNOTE
COL_TYPE:NORMAL, creator.name, CREATEDBY
COL_
TYPE:NORMAL, iterationInfo.modifier.name, MODIFIEDBY
COL_TYPE:NORMAL, state.lifeCycleId.name, LIFECYCLE
COL_TYPE:NORMAL, state.state, LIFECYCLESTATE
COL_TYPE:NORMAL, createStamp, CREATEDDATE
COL_TYPE:NORMAL, modifyStamp, MODIFIEDDATE
COL_TYPE:NORMAL, teamTemplateId.name, TEAM
COL_TYPE:NORMAL, typeDefinitionReference, TYPE
COL_TYPE:NORMAL, WBM.SOURCECLASS, WBMSOURCEDESCRIPTION
COL_TYPE:NORMAL, WBM.SOURCEID, WBMSOURCEIDENTIFIER
COL_TYPE:OBJREF_
DATA, masterReference | WBM.SOURCEID, MASTERWBMSOURCEIDENTIFIER
COL_TYPE:IBA_TABLE_
NORMAL, WBM.SOURCEID, IBAHOLDERWBMSOURCEIDENTIFIER
COL_TYPE:SL_TABLE_
NORMAL, WBM.SOURCEID, SLHOLDERWBMSOURCEIDENTIFIER

```

Secondary Classes Row

Example:

```
CLASS:wt.workflow.engine.WfProcess, TABLE:WFPROCESS  
SECONDARY  
CLASSES:wt.workflow.work.WfAssignedActivity;wt.workflow.engine.WfAdHocActivity;wt.workflow.engine.WfRequesterActivity;wt.workflow.engine.WfBlock;wt.workflow.robots.WfInternalMethod;wt.workflow.robots.WfExternalMethod;wt.workflow.robots.WfExpressionRobot;wt.workflow.robots.WfApplicationRobot;wt.workflow.robots.WfSyncRobot;wt.workflow.robots.WfTimerActivity;wt.workflow.robots.WfURLRobot;wt.workflow.work.WfAssignment;wt.workflow.work.WorkItem;wt.workflow.engine.WfVotingEventAudit;wt.workflow.engine.WfAssignmentEventAudit
```

The above example means that the classes listed in SECONDARY_CLASSES are extracted in the same extraction command as `wt.workflow.engine.WfProcess`. For more information, see [One-to-Many Extraction on page 83](#).

A

Additional Information on Select Use Cases

File Content Migration.....	235
Using the Standard Content Migration Option	235
Using the Enhanced Content Migration Option.....	241
NavigationCriteria - Selective Class Loading	256
Incremental Migration	257
Installing the Staging Schema on Cross-Platform Configurations	261
Setting up Properties for Remote SQL Server Staging Database Tool.....	261
Setting up the Migration Tables in Target Windchill Database.....	262
Setting up the Staging Database	263
EPM Data Loaders	268
Enterprise Systems Integration Data Loaders	273
Classification Attribute Configuration.....	275
Representation Migration	279
Change Object Migration	282
Flexible Change Item Delegate Configuration	284
Managing Life Cycle Services and Life Cycle History During Loading	285
Migrating into a Live Production System.....	287
Using Restricted Domains in the Migration Process	289
Setting the Active Production System to Read Only	291
Understanding the Staging Database Table Attributes	294
Security Labels	295
Supplier Management Data Loaders.....	296
Pre-requisite	297

File Content Migration

This section describes the implementation details for the Windchill Bulk Migrator content migration options. The Windchill Bulk Migrator loads file content and URL data into a unified, single data loader and staging database table for file content and URL data for Windchill content holder types.

Using the Standard Content Migration Option

There are certain Windchill Bulk Migrator and Windchill configurations that are required to implement the Standard content migration option. The following sections provide an overview of the migration procedure workflow involving content migration.

File Content Loaders for Standard Option

The Standard content migration option uses the ContentMetaDataLoader.

The staging tables that the loaders use and look to for inputs is controlled by the selection of the content migration option and is defined in `wbm.properties`.

File Content Staging for Standard Option

The following sections provide information about staging file content metadata, URLData, staging tables, and the actual content.

Metadata Staging

The Standard content migration option requires information to fully define the content item link to the associated Windchill object and to allow the loaders to locate the source content file. The Windchill Bulk Migrator staging database table, `CONTENTMETADATA`, defines this information.

This unified table contains the content metadata for all Windchill content holder types supported by the Windchill Bulk Migrator with the class of the content holder identified by the `HOLDERCLASSNAME` table column.

The `HOLDERCLASSNAME` column contains the fully qualified class name for the content holders to be loaded. Valid values are:

- `wt.epm.EPMDocument`
- `wt.epm.familytable.EPMSepFamilyTable`
- `wt.doc.WTDocument`
- `wt.part.WTPart`
- `wt.change2.WTChangeIssue`
- `wt.change2.WTChangeActivity2`
- `wt.change2.WTChangeOrder2`

-
- wt.change2.WTChangeRequest2
 - wt.change2.WTVariance
 - com.ptc.wpcfg.doc.VariantSpec

Refer to the next section for more information about each of the attribute values that can be defined in this table. The information in this table is populated by the Migration Administrator prior to the content migration.

Staging Table Attribute Definition

The Migration Administrator must provide values for the required attributes in the CONTENTMETADATA staging table to facilitate the intended content transfer option.

CONTENTMETADATA	Required		Comments
	File Transfer	URLData	
SrcSiteURL	Yes	Yes	For URLData yields the Target URL
SrcFilePath	Yes	Yes	For URLData the value yields the URL label
SrcFileName	Yes	Yes	
SrcFileSize			
SrcFileVersion			
UniqueContentIdentifier			Unique value definition of the association to the owning (holder) object version, iteration in Windchill
TargetSiteURL	Yes		
TargetVaultName			
HolderClassName	Yes	Yes	
FileName	Yes	Yes	
ObjectNumber	Yes	Yes	
HoldRevision	Yes	Yes	
HolderIteration	Yes	Yes	
ORGANIZATION_NAME	Yes	Yes	
Role	Yes	Yes	
Category			
Type	Yes	Yes	
Description			Description for the URL or Application Data
Comment			
CreatedBy	Yes	Yes	
CreateDate	Yes	Yes	

CONTENTMETADATA	Required		Comments
	File Transfer	URLData	
ModifiedBy	Yes	Yes	
ModifiedDate	Yes	Yes	
MIGRATED	Yes	Yes	
BATCH_ID	Yes	Yes	
SEQUENCENUMBER	Yes	Yes	
UPLOADEDFROMPATH			
CONTAINER_NAME			
CONTAINER_TYPE			
CONTAINER_ORGANIZATION_NAME			
HOLDERREFERENCEID			
SOURCECLASS			
SOURCEID			

Additional descriptions for selected attributes:

- UniqueContentIdentifier – This attribute can be used to provide unique or common associations between multiple object metadata versions and iterations. For example, in Windchill, a Revise event will in of itself not create a new content file that must be vaulted. Only events which also have a corresponding modification to the content file will require a new vaulted object. The system will treat "0" and "NULL" values the same. It will create new copy of the content in the mounted vault.
- SRCFILEPATH is the absolute file path of the source content file. For URL Data, the value yields the target URL of the URLData.
- SRCFILENAME is actual name of the file on the disk, for example, the file name used in the external vault. For URLData, this value yields the label for the URLData.
- FILENAME is the displayed file name. For example, the name given the primary content for CAD documents or the file name specified when creating a WTDocument or attachment for change objects.
- ABSOLUTEPATH is the absolute file path included the source file name for change object attachments.

Content Staging

File content can be extracted from source systems and staged in operating system directories, or the original source content in its native state and location can be used, so long as the Migration Administrator can provide the association definition to the Windchill object type that owns the content.

However, careful planning is needed to ensure optimized performance of content migration. That is, the location of the files and the migration strategy should take into account the architecture design of the Windchill system, the file server and vault locations, the available disk space, and the backup strategy.

Windchill Configurations for Standard Option

The following sections provide information about the required Windchill configurations.

File Vault Definition

The Standard content migration option does not require that external file vaults be used in the target Windchill system.

Note

Standard content migration always loads data into the cache vault of the selected (preferred) site.

Property Settings

No property settings need to change to enable the Standard content migration option. By default, the following property that controls the selection of the content migration option is provided in `wbm.properties`:

```
com.ptc.windchill.migration.wbm.xferContent=true
```

Performing Content Migration using the Standard Option

Execution of the content migration is performed using the normal Data Load Controller command:

```
com.ptc.windchill.migration.wbm.DataLoadController.
```

For details about the unified approach to content migration, continue reading this section and see [Loading Data Overview on page 130](#).

Standard Content Migration Example

The following high level procedure is for an example use case where file content is to be migrated to two locations: one file server that is local to the Windchill application server and one remote file server.

1. Extract or stage content to make available to the target Windchill system.
2. Populate the staging database `CONTENTMETADATA` table with source object metadata and the required information about the source file content and `URLData`.
3. Place the target Windchill system into Read Only mode by executing the Windchill Bulk Migrator `ROAPController` command with the Read Only option. For more information on executing the command, see [Migrating into a Live Production System on page 287](#).
4. Run the Data Load Controller to create/migrate Windchill object metadata that will own the content attachment. The `ContentMetaDataLoader` takes an optional additional parameter which is the target object type to load content for; for example, to load content data for `WTDocument` types, the command for the load controller would be:

```
windchill  
com.ptc.windchill.migration.wbm.DataLoadController -l  
CONTENTMETADATA WTDocument
```

This optional parameter has the following conditions:

- To load content for specific object types, this additional parameter is required.
 - If no additional parameter is passed, the system will pick up records for all object types for loading from the CONTENTMETADATA table.
5. Run the Data Load Controller for the content.
 6. Place the target Windchill system into Read/Write mode by executing the Windchill Bulk Migrator ROAPController command with the Read/Write option.

Step 1 through Step 5 can be repeated as needed to support incremental migrations prior to the overall bulk migration completing. For information about supported incremental migration use cases, see [Incremental Migration on page 257](#).

Standard Content Migration Usage Notes:

Review the following notes before implementing the Standard option for file content migration:

- The Standard option does not move the source file content from the source location to Windchill. The option uploads a copy of the content file into Windchill. The copy is ultimately stored in the database or external vaults according to the configured rules in Windchill.

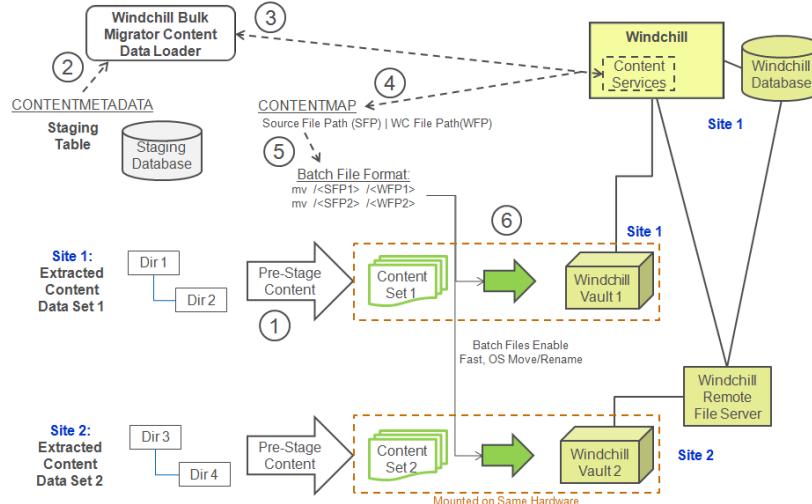
The Migration Administrator should ensure there is appropriate disk space available on the target file server, especially if the hardware is also used to temporarily store the staged or extracted content files.

- Standard Content migration for VariantSpec updates the VariantSpec XML file in Target Windchill with the IDA2A2 information.

Using the Enhanced Content Migration Option

This section describes the Windchill Bulk Migrator and Windchill configurations that are required to implement the Enhanced content migration option. It also provides an overview of the migration procedure workflow involving content migration.

The following diagram provides an overview of the process. The diagram shows two systems (Site 1 and Site 2) where data is to be stored in Windchill:



The following list provides details about the numbered process steps in the diagram:

1. Pre-stage content on both Site 1 and Site 2 so that the file content is available for batch migration scripts.
2. Pre-process and load content.
 - a. Run the content pre-processor, ContentLoaderPreLoadProcessor, to populate the target vault and site where content will be loaded. For Multi-Vault setup, this step is necessary when both the TARGETSITEURL and TARGETVAULTNAME are not specified. For single-vault setup, TARGETSITEURL should not be specified.
 - b. Run content loader to load the metadata for the Windchill objects that have content.
3. As a result of loading the metadata, Windchill Content Services returns information to loader.
4. Loader populates the CONTENTMAP database table.
5. Using the information in the CONTENTMAP table, create batch files for loading content data. Each batch file uniquely identifies the content for a site.
6. Run each batch file to load the content data into a Windchill vault.

File Content Loaders for Enhanced Option

The Enhanced content migration option uses the ContentMetaDataLoader.

The ContentMetaDataLoader takes an optional parameter, which is the target object type to load content, to load content data, for example, for WTDocument types. The command for the load controller is:

```
windchill
com.ptc.windchill.migration.wbm.DataLoadController -l
CONTENTMETADATA WTDocument
```

For VariantSpec content migration, the command would be as follows:

```
windchill
com.ptc.windchill.migration.wbm.DataLoadController -l
CONTENTMETADATA VariantSpec
```

Note

Enhanced content migration does not update the IDs in the VariantSpec content file. You must run the VariantSpecContentModifier utility to update the referenced object IDs from the target system object IDs in the VariantSpec content file.

Where the unified syntax for content loading equals:

```
windchill
com.ptc.windchill.migration.wbm.DataLoadController -l
ContentMetaData <object type>
```

<object type> must be a supported content migration object type and it is optional. For more information about supported object types, see [Supported Object Types on page 298](#).

As noted, the additional parameter is optional with the applicable conditions noted below:

- To load content for specific object types, this additional parameter is required.
- If no additional parameter is passed, the system will pick up records for all object types for loading from the CONTENTMETADATA table.

The staging tables that the loaders use and look to for inputs is controlled by the selection of the content migration option and is defined in `wbm.properties`.

File Content Staging for Enhanced Option

The following sections provide information about staging file content metadata, staging tables, and the actual content. For more information on enhanced content migration, see the “Enhanced Content Migration Usage Notes” section.

Metadata Staging

The Enhanced content migration option needs information to fully define the content item link to the associated Windchill object and to allow the loaders to locate the source content file. This information is defined in the following Windchill Bulk Migrator staging database table:

CONTENTMETADATA

Refer to the next section for more information about each of the attribute values that can be defined in this table. The information in this table is populated by the Migration Administrator prior to content migration.

Staging Table Attribute Definition

The Migration Administrator must provide values for the required attributes in the CONTENTMETADATA staging table to facilitate the intended content transfer option.

The following table shows the required attributes for the Enhanced migration options. The attribute is required if the Required column contains an 'x'. The migration can occur only when each attribute marked as required has a value that is not null.

Column	Required		Description
	File Content	URL Data	
SrcSiteURL	X		Source Site Identity (for example, Main Site, Remote Site)
SrcFilePath	X	X	Explicit file directory path to content file. For URLData, yields the target URL.
SrcFileName	X	X	Source content file name as stored on the disk. For URLData the value yields the URL label.
SrcFileSize	X		Source content file size (bytes)
SrcFileVersion			Version of the source file.
UniqueContentIdentifier			Unique value definition of the association to the owning (holder) object version, iteration in Windchill
TargetSiteURL	X		Target Windchill URL
TargetVaultName	X		Target Windchill file vault. Required for multi-vault configuration.
HolderClassName	X	X	Owning (holder) object class
FileName	X		Displayed file name in Windchill user interface
ObjectNumber	X	X	Owning object number
HolderRevision	X	X	Owning object revision
HolderIteration	X	X	Owning object iteration
Organization_Name	X	X	Owning object organization
Role	X	X	Windchill content association type
Category			CAD authoring content type

The following table shows the required attributes for the Enhanced migration options. The attribute is required if the Required column contains an ‘x’. The migration can occur only when each attribute marked as required has a value that is not null.

Column	Required	Description
	File Content	URL Data
HOLDREFERENCEID		Used as reference key for content holder. In case of FamilyTable migration, this column holds the FamilyTableID from EPMContainln table in the staging database.
Type	X	Vaulting type: either wt.content.ApplicationData (content) or wt.content.URLData
Description		Description of the content file or URLData.
Comments		Created by user designation
CreatedBy	X	Created by date and timestamp
CreatedDate	X	Modified by user designation
ModifiedBy	X	Modified by date and timestamp
ModifiedDate	X	Flag identifier for migration status: 0 or 1
MIGRATED	X	Identifier designating different groups of content to migrate
BATCH_ID	X	Order definition for migration processing
SEQUENCENUMBER	X	UPLOADEDFROMPATH
		CONTAINER_NAME
		CONTAINER_TYPE
CONTAINER_ORGANIZATION_NAME		

The following table shows the required attributes for the Enhanced migration options. The attribute is required if the Required column contains an 'x'. The migration can occur only when each attribute marked as required has a value that is not null.

Column	Required			Description
	File Content	URL Data		
HOLDERREFERENCEID				
SOURCECLASS				
SOURCEID				

Additional descriptions for selected attributes:

- UniqueContentIdentifier – This attribute can be used to provide unique or common associations between multiple object metadata versions and iterations. For example, in Windchill, a Revise event will in of itself not create a new content file that must be vaulted. Only events which also have a corresponding modification to the content file will require a new vaulted object. The system will treat "0" and "NULL" values the same. It will create a new copy of the content in the mounted vault.
- Role – Windchill enables the definition of multiple content association roles where the association is to the owning object and controls how the association is shown in the Windchill user interface. Valid definition options supported by the Windchill Bulk Migrator are Primary and Secondary.
- FileName – The FileName field provides the Windchill User Interface File Name value for the content object. The field is optional in the case of loading content of the Primary Content Role type for EPMDocument Holder Class as the value is obtained by default during loading of the object class. The field is mandatory for the case of loading content of the Secondary Content Role type, or for Primary Content Role where the Holder Class is of a type other than EPMDocument (normally a customization scenario).
- Type – Content can be defined as an object defined by a file (with a file size), or as a URL link to and alternate external location. Valid definitions supported by the Windchill Bulk Migrator are `wt.content.ApplicationData` and `wt.content.URLData`.

Content Staging

PTC recommends that you mount the extracted files from the source system directly to the physical hardware that is hosting the Windchill File Server. Files can be segregated and mounted separately for each local and remote site. To maximize the performance of the resulting batch file move and rename, the supporting architecture and strategy should minimize network traffic as well as hardware I/O.

Windchill Configurations for Enhanced Option

The following sections provide information about the required Windchill configurations.

File Vault Definition

The Enhanced content migration option requires that external file vaulting is used by the target Windchill system. It can be used in two ways:

- Only one vault per Windchill site (local or remote) be used to contain the migrated content files.
- Multiple sites and multiple vaults (local or remote) be used to contain the migrated content files. This is the default.

These configurations should be configured prior to starting the content migration. The Enhanced content migration option does not use the **Vaulting and Replication ▶ Preferred File Server** user preference.

Each target file vault that is used for the content migration should be designated to use the `autoFolderCreate` option. The `autoFolderCreate` option causes a new directory folder to be created automatically after a threshold value has been reached for the number of content files stored in the directory.

Property Settings

Use the `xconfmanager` to set and propagate the following properties.

wbm.properties

```
com.ptc.windchill.migration.wbm.xferContent=false
com.ptc.windchill.migration.wbm.datasource.locale=""
com.ptc.windchill.migration.wbm.processor.postload.CONTENTMETADATA=
com.ptc.windchill.migration.wbm.processor.ContentPostLoadProcessor
com.ptc.windchill.migration.wbm.processor.preload.CONTENTMETADATA=
com.ptc.windchill.migration.wbm.processor.ContentLoaderPreLoadProcessor
com.ptc.windchill.migration.wbm.vaultconfig.single=false
com.ptc.windchill.migration.wbm.loaders.threadpool.max.size=5
```

wt.properties

```
wt.fv.useVaultsForAllContent=False

wt.fv.forceContentToVault=false

wt.fv.useFvFileThreshold=true

wt.fv.fvFileThreshold=4000

wt.fv.uploadtocache.ignoreFileMoveCheck=true
```

To support multiple sites and multiple vaults you must load an additional preference for the preferred vault. This preference does not come out-of-the-box with the Windchill installation. It should be loaded using the following command:

```
windchill wt.load.LoadFromFile -d <Windchill>/loadFiles
/preference.content.incremental.R10M010toR10M020.xml -u
<site admin> -p<site admin password>
```

To use the vault setting for multiple sites with a single master vault per site, change the following property to "true" in the `wbm.properties` file:

```
com.ptc.windchill.migration.wbm.vaultconfig.single=
false
```

Note

The number of load threads used combined with the value set for the folder threshold impacts the resulting number of content files that are assigned to a logical folder directory for the file vault. The number of content files that actually get assigned to a vault will be plus or minus a percentage of the desired threshold value. The variable number of files is a result of the multi-thread batch processing which cannot guarantee a specific number.

Use the values provided in this section as a starting point for migration. The goal of the vault folders is to ultimately contain 15,000 files plus or minus 10%. Typically, after 50,000 objects are in a folder, the performance becomes sluggish. Therefore, you should ensure that the number of files in a logical folder is under 50,000.

The values may need to be adjusted depending upon the hardware sizing and the actual desired number of files you want in the folder. Using more threads generally requires a lowering of the threshold to keep the same plus or minus accuracy percentage for the desired threshold.

After migration, reset the threshold value to a number that is appropriate for normal production operations.

To load content in multiple target site and vaults, both the columns <TARGETSITEURL> and <TARGETVAULTNAME> in the CONTENTMETADATA table should have a value populated. This can be done one of two ways:

1. Populate both the columns with the user specified target site URL and the vault name.
2. Using content pre-load processor. In this case, the system will find out both the site and the vault based on the setup vaulting rule.

Pre-load Processor Functionality

The pre-load processor finds both the target site and the target vault where the content will be migrated. The conditions for finding the target site and vault are:

-
1. If you pre-define both the target site URL (TARGETSITEURL) and the target vault name (TARGETVAULTNAME) in the CONTENTMETADATA table, the system will consider the highest preference for loading the content.
 2. If you do not define both the target site URL (TARGETSITEURL) and the target vault name (TARGETVAULTNAME) in the CONTENTMETADATA table, the system will find the target site and the target vault based on the vaulting rule defined in the target Windchill. If no vaulting rule is found, the system reports the error and logs it in the error log file.
 3. If you define only the target vault name (TARGETVAULTNAME) but not the target site URL (TARGETSITEURL), the system ignores the record as it is invalid for processing.
 4. If you define only the target site URL (TARGETSITEURL), but not target vault name (TARGETVAULTNAME), the vaulting rule defined in the target system will override the pre-defined information (TARGETSITEURL).
 5. If the content holders are not migrated successfully (for example, EPMDocument or WTDocument record that is marked MIGRATED as "2" in their respective staging tables), the system does not pick those records.
 6. If the wbm.properties file contains the single site property com.ptc.windchill.migration.wbm.vaultconfig.single=true, the system expects the TARGETSITEURL column to be pre-populated. In this case, the pre-processing step is redundant.

Note

ContentHolder CLASSNAMEs supported in the current release are:

- wt.epm.EPMDocument
- wt.epm.familytable.EPMSepFamilyTable
- wt.doc.WTDocument
- wt.part.WTPart
- wt.change2.WTChangeIssue
- wt.change2.WTChangeActivity2
- wt.change2.WTChangeOrder2
- wt.change2.WTChangeRequest2
- wt.change2.WTVariance
- com.ptc.wpcfg.doc.VariantSpec

The pre-load processor can be run by executing the following command:

```
windchill
com.ptc.windchill.migration.wbm.DataLoadController -pre
CONTENTMETADATA <object type>
```

Where <object type> is any supported content migration object type. It is an optional parameter. If no object type is given, all object types in the CONTENTMETADATA table are considered.

Performing Content Migration using the Enhanced Option

Before running the content loaders, ensure that the relevant Windchill vaulting queues are disabled. The queues can be manually disabled through the Windchill Queue Administration utility. Alternatively, queues can be automatically disabled by using the ROAPController command. For information on executing the command, see [Migrating into a Live Production System on page 287](#).

Note

Before continuing, perform a visual check in Windchill to ensure that the following queues are disabled:

commonScheduleQueue
ReplicationQueue
RevaultingQueue

Execution of the content migration is performed using the normal Data Load Controller command:

```
com.ptc.windchill.migration.wbm.DataLoadController.
```

The ContentMetaDataTable takes an optional parameter, which is the target object type to load content, to load content data, for example, for WTDocument types. The command for the load controller is:

```
windchill
com.ptc.windchill.migration.wbm.DataLoadController -l
CONTENTMETADATA WTDocument
```

As noted, the additional parameter is optional with the applicable conditions noted below:

- To load content for specific object types, this additional parameter is required.
- If no additional parameter is passed, the system will pick up records for all object types for loading from the CONTENTMETADATA table.

For details, see [Loading Data Overview on page 130](#).

Execution of the `DataLoadController` command for the content loaders populates the `CONTENTMAP` table in the staging database, as well as updates the staging `AUDIT` and `STATUS` tables accordingly.

The `CONTENTMAP` table holds the following information:

Column	Description
<code>SRCSITEURL</code>	The URL of the system where the staged content files reside
<code>SRCFILENAME</code>	The physical file name of the source file content
<code>FILENAME</code>	The object file name metadata value that will be displayed in the Windchill user interface
<code>SRCFILEPATH</code>	The directory location of the physical file content
<code>TARGETSITE</code>	Host URL of the system where the target file vault directories reside
<code>TARGETFILEPATH</code>	The directory location of the target vault and folder path
<code>TARGETFILENAME</code>	The hexadecimal hash code value for the Windchill content item
<code>CONTENTHOLDERCLASSNAME</code>	The class name of the owning object of the content item
<code>MIGRATEDTIMESTAMP</code>	Supports incremental migrations. This column holds the timestamp for the data that was migrated.

 **Note**

`SRCSITEURL`, `FILENAME` and `CONTENTHOLDERCLASSNAME` are provided in the `CONTENTMAP` table for reference purposes only and are not needed to create the Move and Rename batch files.

After running the `DataLoadController` command, it is possible to create Move and Rename batch files by querying the `CONTENTMAP` table and saving the result in an external file.

If the execution of the `DataLoadController` command cannot write to the `CONTENTMAP` table for any reason (such as a network failure), then the Windchill Bulk Migrator attempts to write the content to a file. The Windchill Bulk Migrator leverages the logger infrastructure to create the following file:

```
<windchill>/logs/wbm/errorContentMap-<JVM_start_timestamp>-<JVM_process_id>-log4j.log
```

where:

- <windchill> is the Windchill installation directory.
- <JVM_start_timestamp> is the date formatted as yyMMddHHmm.
- <JVM_process_id> is the process ID of the JVM.

If the file is created, the file content is a comma-separated list of the column data that is intended for the CONTENTMAP table.

Enhanced Content Migration Example

The following high level procedure is for an example use case where file content is to be migrated to two locations: one file server that is local to the Windchill application server and one remote file server.

1. Extract or stage content to make the content available to the target Windchill system. Mount content to the Windchill File Server for each local and remote site.
2. Populate the staging database tables with source object metadata and populate the CONTENTMETADATA table that defines the required information about the source file content.
3. Place the target Windchill system into Read Only mode by executing the Windchill Bulk Migrator ROAPController command with the Read Only option. For information on executing the command, see [Migrating into a Live Production System on page 287](#).
4. Run the pre-load processor commands.
5. Run the DataLoadController command to create or migrate Windchill object metadata that will own the content attachment.
6. Run the DataLoadController command for the content loaders to populate the source and target content mapping information in the Windchill Bulk Migrator CONTENTMAP staging database table.

Note

The loader processes the information for each Windchill site sequentially.

7. Create the output Move and Rename batch script by querying the CONTENTMAP staging table.
8. Divide the batch script created in step 6 into separate batch scripts: one for the local site and one for the remote site. Each batch file is for a respective Site/ File Server.

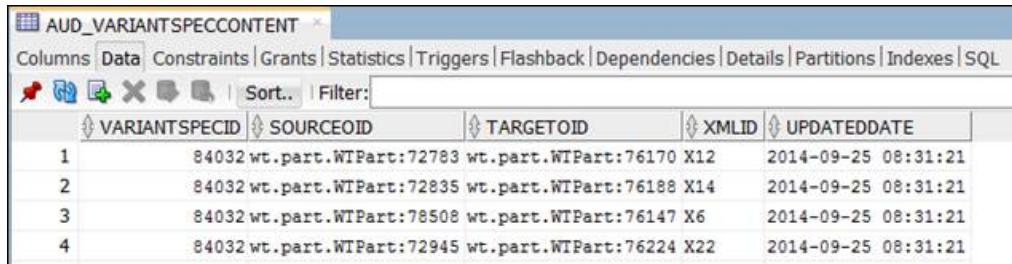
9. Copy each batch script to the corresponding file server (local and remote file server) and execute each script to perform content move and rename directly to the Windchill vault directory mounts on the local Windchill server and remote file server.
10. Place the target Windchill system into Read/Write mode by executing the Windchill Bulk Migrator ROAPController command with the Read/Write option.
11. Execute the VariantSpecContentModifier command to update the XML content file that has been migrated:

```
windchill
com.ptc.windchill.migration.wbm.VariantSpecContentMo
difier -u <user> -p <password>
```

Note

VariantSpec content must be migrated to target Windchill before executing the VariantSpecContentModifier. The VariantSpecContentModifier command will update the referenced object ID in the VariantSpec content file with the target system object ID.

After the VariantSpec content migration to target Windchill, target Windchill database AUD_VARIANTSPECCONTENT will contain information for VariantSpecID, Source ID, Target ID, and XMLID (XMLID information is present in the XML content of VARIANTSPEC).



VARIANTSPECID	SOURCEOID	TARGETOID	XMLID	UPDATEDDATE
1	84032 wt.part.WTPart:72783 wt.part.WTPart:76170	X12	2014-09-25 08:31:21	
2	84032 wt.part.WTPart:72835 wt.part.WTPart:76188	X14	2014-09-25 08:31:21	
3	84032 wt.part.WTPart:78508 wt.part.WTPart:76147	X6	2014-09-25 08:31:21	
4	84032 wt.part.WTPart:72945 wt.part.WTPart:76224	X22	2014-09-25 08:31:21	

Step 1 through Step 8 can be repeated as needed to support incremental migrations prior to the overall bulk migration completing. For information about supported incremental migration use cases, see [Incremental Migration on page 257](#).

Enhanced Content Migration Usage Notes:

Review the following notes before implementing the Enhanced option for file content migration:

- Enhanced migration requires that external vaulting be implemented in the target Windchill system. The Windchill Bulk Migrator is applicable only for physical files on disc to migrate to Windchill external vaults.
- Enhanced migration works with Windchill master vaults. The rootfolders are the part of the External File Vaults and do not need to be identified.
- Enhanced migration assumes that adequate disk space is available for the content migration in the target Windchill vault rootfolders. The Migration Administrator must ensure that adequate disk space is available.
- The Windchill object that is to own the content attachment must be migrated and exist in the target system before implementing the content migration tools and manual steps.
- Content migration must occur prior to the migrated object metadata being transitioned to public domains and while the migrated metadata is associated to the restricted migration domain.
- Before the execution of the content migration, Windchill will be placed in Read Only mode by the tools and vaulting and replication queues will be suspended. Read Only mode will be lifted and the queues will be re-enabled after the content migration.
- The vaults must be configured (with AutoFolder Creation enabled) in advance of content migration from the Windchill External Storage Administrator.
- The Enhanced migration enables the transfer of all content to an external vault on the target system. PTC recommends using the pre-processor to compute the target vault and site according to vaulting rules after the holder metadata (such as EPMDocument and WTDocument) is migrated.
- Enhanced migration does not apply to Windchill file replication or replication rules configured in Windchill.

NavigationCriteria - Selective Class Loading

This section describes the NavigationCriteria object loading for specific class type objects.

It supports the following objects:

- DERIVEDIMAGE
- WTPart
- WTUser

To load only the derived image, use the following command in the load controller:

```
windchill  
com.ptc.windchill.migration.wbm.DataLoadController -l  
NAVIGATIONCRITERIA DERIVEDIMAGE
```

To load WTPart and WTUser, use the following command in the load controller:

```
windchill  
com.ptc.windchill.migration.wbm.DataLoadController -l  
NAVIGATIONCRITERIA WTPart,WTUser
```

The mapping of the classes is maintained in the wbm.properties. This approach is supported and in use for the content loader.

Incremental Migration

Incremental migration (also known as Delta migration) is defined as the migration of a subset of data from the source system where the data subset represents only the data that has changed (delta data set) since the start of the bulk migration procedure. Incremental migration increases the source system uptime during migration and can be considered when the data volume is very large and requires long running migration times. The use of incremental migration assumes that the delta data set will be much smaller than the original data set and the resulting incremental migration would ultimately fit within a manageable time window.

The incremental migration of delta data sets can occur multiple times within the overall migration procedure, as necessary, before the final production go-live event.

Enabling Incremental Migration

Implementation of the Incremental Migration mode is enabled by setting the following property in wbm.properties to true:

```
com.ptc.windchill.migration.wbm.mode.incremental
```

Note

Ensure that this property is set to false for the migration of new data and object masters into Windchill. By default, this property is set to false.

Support of Incremental Migration

The Windchill Bulk Migrator object loaders enable the incremental loading of the following object types:

- WTPart
- WTManufacturerPart
- WTVendorPart
- WTDocument

-
- EPMDocument
 - EPMFamilyTable
 - WTChangeIssue
 - WTVariance
 - WTChangeRequest2
 - WTChangeOrder2
 - WTChangeActivity2

Incremental migration can be implemented to either append objects to the version history in the target or incremental migration can be set to insert mode to specify object types that should be added within the existing target version history.

PTC supports the use of incremental migration only when migrating into a restricted migration domain and within the context of an overall bulk migration event.

PTC does not support using incremental migration to update existing objects in public domains in a populated Windchill system.

After data is moved to public domains, new data object masters can be migrated into Windchill, but updates to iteration histories of public objects are not allowed.

Using Incremental Migration

Incremental Migration mode only migrates new iterations (iterations that had not been previously migrated from the source system).

The incremental mode does not enable the migration of data where changes did not result in a new object iteration being created in the source system. For example, objects that have been renamed in the source system since the bulk migration has started are not be migrated or synched with the staging database or target Windchill system.

Be aware that objects that have been deleted in the source system after the bulk migration was started are not deleted in the staging database or target Windchill system.

The following example migration process flow includes incremental migration:

1. Extract source system object metadata and file content at Time=T1.
2. Stage, map, and transform data as needed in the staging database.
3. Load data into a restricted migration domain in Windchill and using the Windchill Bulk Migrator. At completion of this step, the Time=T2.
4. Identify and extract the delta data set created in the time period: T2-T1.
5. Stage, map, and transform data as needed in the staging database.
6. Load data into a restricted migration domain in Windchill and using the Windchill Bulk Migrator. At completion of this step, the Time=T3.

-
7. Repeat steps 4-6, if necessary for delta data set: T3-T2.
 8. Repeat steps 4-6, if necessary for (Tn)-(Tn-1).
 9. Transition data from the restricted migration domain to the active public domains.
 10. Complete post migration tasks.
 11. Go Live.

Incremental Migration Use Case

Use the incremental migration after the initial bulk migration and before the transition of data to public domains.

Note

Updates to objects in public domains are considered to be part of a federation or data sharing use case. For these types of use cases, investigate other Windchill capabilities such as Packages or Import/Export.

The following is an overview of the migration process in which users continue working while the migration proceeds:

1. Bulk migrate metadata.
2. Bulk migrate content.
3. Migrate delta metadata and content.
4. Repeat as necessary.
5. Transition to public domains.

Enabling Incremental Insert Mode

Implementation of the incremental insert mode is enabled by setting the following properties in wbm.properties:

```
wbm.versioned.objects.support.incremental.insert=
wt.part.WTPart,wt.doc.WTDocument,wt.epm.EPMDocument
```

Set the property by using a comma separated list of fully qualified versioned object types. By default, this property is left blank which causes the incremental migration to be in append mode.

```
com.ptc.windchill.migration.wbm.mode.incremental=true
```

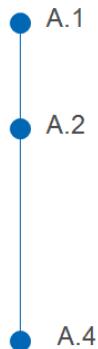
By default, this property is set to false.

Note

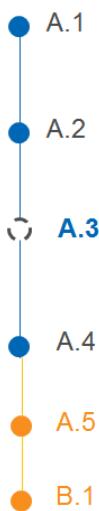
The append and insert modes only apply to objects loaded using legacy loaders. They do not apply to objects loaded using generic loaders and the behavior observed works as expected. For a list of object types that belong to both categories, see the output of `windchill`

```
`com.ptc.windchill.migration.wbm.DataLoadController  
-l?
```

For example, in the initial bulk migration, the user migrated the following objects for WTPart, WTDocument or EPMDocument:



With the incremental migration in insert mode, you can migrate any of the objects as long as they do not conflict with existing objects (You cannot migrate A.1, A.2, or A.4 as they are already available in the target system).



In this example, A.3 can be inserted. Additionally, A.5 or B.1 can be appended using insert mode.

Installing the Staging Schema on Cross-Platform Configurations

The following sections provide information about installing staging schema on cross-platform configurations.

Setting up Properties for Remote SQL Server Staging Database Tool

The Windchill Bulk Migrator SQL Command Tool allows you to connect Oracle and SQL Server databases that use different operating systems. This tool connects across operating system platforms and executes the SQL commands on the remote SQL Server database for the user. If you are connecting to a remote SQL Server staging database or connecting Oracle databases across platforms, continue with the steps in this section.

There is no need to setup the properties file for a local or native setup, where the SQLServer Database and Windchill Bulk Migrator are installed on the same local machine.

To set up the properties files for the SQL Command Tool for a remote setup, perform the following steps:

1. In a text editor, create a file and name it `SQLConfig.properties`. Later in this guide, the variable <File having the DB connectivity credential details> is used to refer to this file.
2. Modify the file to match your site's information (for connecting to SQL Server):

```
com.ptc.windchill.migration.wbm.util.db.sqlcommandtool.sqldbType=sqlserver  
com.ptc.windchill.migration.wbm.util.db.sqlcommandtool.sqldbUser=<SQL Server DB name>  
com.ptc.windchill.migration.wbm.util.db.sqlcommandtool.sqldbPasswd=<SQL Server DB password>  
com.ptc.windchill.migration.wbm.util.db.sqlcommandtool.sqldbName=<server name>  
(by default the machine name)>\<instance_name>:Port:<SQL Server DB name>  
com.ptc.windchill.migration.wbm.util.db.sqlcommandtool.sqldbMode=<native> or <java>
```

Local: If the SQL Server Database and the Windchill Bulk Migrator are installed on the same local machine and the command will use the `sqlcmd` utility for SQL Server.

Remote: If the SQL Server Database and the Windchill Bulk Migrator are installed on different machines, a Java utility will be run from the Windchill Bulk Migrator machine that works using JDBC to connect to the database.

If no instance exists, leave it blank.

3. Save the file.

Setting up the Migration Tables in Target Windchill Database

Complete the following steps to set up the migration tables in the target Windchill database:

1. Create audit tables:

Oracle Database

- a. From a Windchill prompt, navigate to the following directory:
 - i. For single-byte English locale data sets: `<Windchill>/db/sql/wbm`
 - ii. For multi-byte English or non-English locale data sets: `<Windchill>/db/sql3/wbm` folder.
- b. Start `sqlplus` and log in as Windchill database user.
- c. At the SQL prompt, enter:

```
SQL>@create_audit_schema.sql
```

Caution

The `create_audit_schema.sql` or any individual SQL scripts that create audit tables will do not try to drop the existing tables.

SQL Server Database

- a. From a windchill prompt, navigate to the following directory:
`<Windchill>\db\sqlServer\wbm`
- b. At the prompt, enter the command appropriate for your system:

For Remote Setup:

Windows

```
make_staging.bat <File having the DB connectivity credential details>
create_audit_schema.sql
```

UNIX

```
./ make_staging.csh <File having the DB connectivity credential details>
create_audit_schema.sql
```

For Local Setup (Native Setup):

Use `sqlcmd` directly with parameter as given below instead of taking input via a properties file.

```
sqlcmd -S %sqlserverhostname% -U %dbUser% -P %dbPasswd% -d %database% -i %sql_file(create_audit_schema.sql)% -o "sqlserver_output.log"
```

Note

If the <File having the DB connectivity credential details> file is not in the current directory, provide the full path.

The following is a sample row from EPMDocumentWBM Audit:

MIGRATION SITE	PLMSILOBJECTID	TARGETOBJECTID	CLASS
54150	320000	95704	ut.epm.EPMDocument org.rnd.DefaultEPMDocument

Setting up the Staging Database

Complete the following steps to set up the staging database for the out-of-the-box schema:

1. Create a staging database user:

Oracle Database

- a. From a Windchill prompt, navigate to the following directory:

- For single-byte English locale data sets: `<Windchill>/db/sql`.
- For multi-byte English or non-English locale data sets:
`<Windchill>/db/sql3`.

- b. Log in to Oracle database as the system database user.

- c. At the SQL prompt, enter:

```
SQL>@create_user.sql
```

SQL Server Database

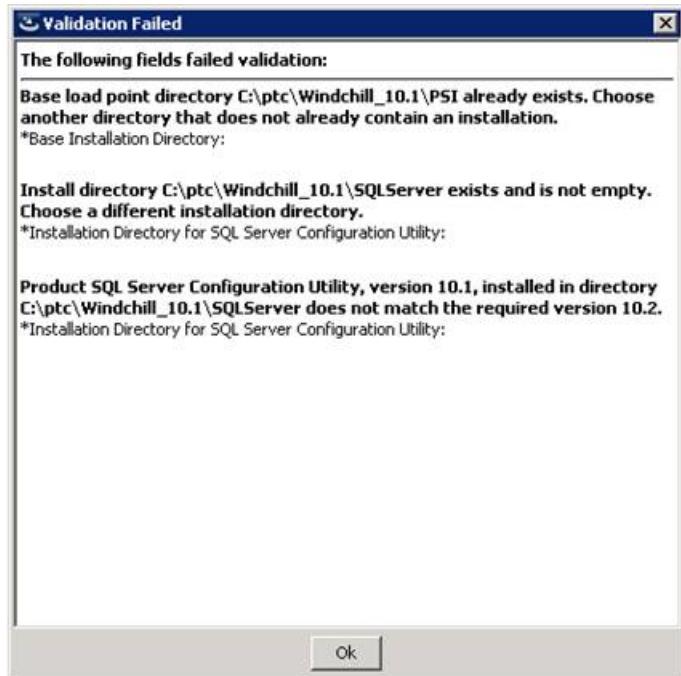
Creating a staging database user remotely is not supported for SQL Server using PSI. To accomplish this option for SQL server, PSI must run on the SQL Server host and the SQL Server Configuration option must be selected to create a database and user.

 **Note**

This section assumes you have installed the database as SQL Server and it must be running.

- a. Launch the PTC Solution Installer.
- b. Choose the language for this installation session and click **OK**.
- c. On the **Before You Begin** window, click **Next**.
- d. On the **License Agreement** window, read the PTC Customer Agreement, and accept the license agreement to confirm that you have legal authority to install the software.
- e. Select the installation type and click **Next**.
- f. Select **Standalone Product or Component** and click **Next**.
- g. Select **SQL Server Configuration**.
- h. Under **SQL Server Configuration**, select **Create PTC Windchill Database and Installation User**.
- i. Select the base installation directory for your staging. The base installation directory is the parent directory in which subdirectories are created for your staging.

- j. Ensure that the directory you selected does not have any other PSI data. If the folder does have other PSI data, the following message appears:



- k. Define the database size and click **Next**.
l. Follow the prompts and provide the necessary information for creating a staging user:

Field	Description
SQL Server Installation Directory	The same directory you used when installing SQL Server Home: <SQL_SERVER_HOME>. For example, E:\Program Files\Microsoft SQL Server.
SQL Server Client Installation Directory	The same directory you used when installing SQL Server Client: <SQL_CLIENT_HOME>. For example, C:\Program Files\Microsoft SQL Server\100\Tools.
SQL Server DNS Registered Host Name	The same name you used when installing SQL Server. For example, i4527.ptcnet.ptc.com.
Installed SQL Server Instance Name (Named Instance only)	The name you used when installing SQL Server. If you used the default instance during installation, this can be left empty.

Field	Description
TCP Port Number for SQL Server Instance	The same port number you used when installing SQL Server.
Password for User	The password for the master administrator for SQL Server.
SQL Server User Name for Windchill	The username Windchill needs to access the database.
SQL Server User Password for Windchill	The password Windchill needs to access the database.
Confirm SQL Server User Password for Windchill	Enter the password again for verification.

- m. In **Staging Area**, browse to your staging directory location. If you have downloaded or copied the CDs to the staging directory, navigate to the CD drive where you placed the CD and provide this location. Click **Next**.
 - n. On the **Review Settings** window, save the `summary.html` file if required, by clicking **Save**. Click **Install**.
 - o. After completing the creation of the database user, data files are found in the following location: `<base installation directory>\SQLServer\datafiles`.
2. Connect as the staging database user and create staging database objects:

Oracle Database

- a. From a Windchill prompt, navigate to the following directory:
 - i. For single-byte English locale data sets: `<Windchill>/db/sql/wbm`
 - ii. For multi-byte English or non-English locale data sets: `<Windchill>/db/sql3/wbm` folder.
- b. Log in to Oracle database as the staging database user (which was created in step 1).
- c. At the SQL prompt, enter:

```
SQL>@create_staging_schema.sql
```

SQL Server Database

- a. From a windchill prompt, navigate to the following directory: `<Windchill>\db\sqlServer\wbm`
- b. At the prompt, enter the commands:

For Remote Setup:

Windows

```
make_staging.bat <File having the DB connectivity credential details>
create_staging_schema.sql
```

UNIX

```
./ make_staging.csh <File having the DB connectivity credential details>
create_staging_schema.sql
```

For Local Setup (Native Setup):

Use `sqlcmd` directly with parameter as given below instead of taking input via a properties file.

```
sqlcmd -S %sqlserverhostname% -U %dbUser% -P %dbPasswd% -d %database% -i %sql_file(create_staging_schema.sql)% -o "sqlserver_output.log"
```

Note

If the <File having the DB connectivity credential details> is not in the current directory, provide the full path.

3. Load staging tables with data using commands in the following format:

```
SQLLDR userid=<oracle_user/password@SID>
control=<table.ctl_file>
log=<table.log> bad=table.bad
```

where:

- <*oracle_user/password@SID*> is the user created in step 1.
- <*table.ctl_file*> is an Oracle bulk loader (SQLLDR) control file, which specifies how data should be loaded into the staging database.
- <*table.log*> is the name of a log table that you want to use.

Note

Avoid setting up tables or creating/modifying tables if you are re-installing Windchill Bulk Migrator.

Note

If the staging database has been created using multi-byte SQL3 scripts, you must use UTF-16 or Unicode for CSV files.

EPM Data Loaders

The Windchill Bulk Migrator includes loaders for some Windchill EPM object types. The loaders in this release correspond to only a subset of the total Windchill EPM object types and do not fully support the migration of all EPM data, relationships, and modeled attributes. Support of the existing EPM loaders is limited to specific use cases and data types as described in this section.

EPMDocument Loader Details

This versioned object loader enables the creation of EPMDocument objects (or subtypes) and version/iteration history in Windchill. In addition, the loader supports the definition of key Windchill attributes such as:

- Name
- Number
- Authoring Application
- Created By
- Created Date
- Sub Type
- Revision
- Iteration
- Life cycle
- Life cycle state

To review a complete list of supported attribute definition, refer to the EPMDocument tables that are created by the installation procedure.

The loader can be used to load standalone objects or files as an wt.epm. EPMDocument type. Use cases supported with this release include loading Arbortext dynamic documents as EPMDocuments.

EPMMemberLink Loader Details

The binary link loader is used to define various associations of type wt.epm. structure.EPMMemberLink between EPMDocuments in Windchill, including Child/Parent relationship structure. The links defined are intended to be maintained over the iteration history of a parent structure unless explicitly removed.

The loader supports the definition of key Windchill attributes such as:

- Role A (EPMDocument)/Role B (EPMDocument)
- Dependency Type
- Child Name
- Required

- Placed
- Suppressed

The following use case is supported:

Load a Parent/Child relationship link between Arbortext Dynamic Documents, where the documents are loaded as type EPMDocument or EPMDocument subtype.

The following are unsupported use cases:

- The loader does not fully support or is tested for Creo or Pro/ENGINEER assembly or membership relationships.
- The loader does not support CATIA V4 or CATIA V5 assembly or model to model relationships.
- The loader does not support spatial view states or transformation references.

EPMP paramMapForDoc Loader Details

This loader creates the EPMPParameterMap entries for EPMDocuments. EPMPParameterMap holds the mapping information when designated parameters or dimensions map to Windchill global attributes of a different name. It sources the data from EPMP paramMapForDoc staging database table.

EPMP paramMapForML Loader Details

This loader creates the EPMPParameterMap entries for EPMMemberlinks. EPMPParameterMap holds the mapping information when designated parameters or dimensions map to Windchill global attributes of a different name. It sources the data from EPMP paramMapForML staging database table.

EPMDocumentUnitInfo Loader Details

This loader creates the unit information and scale factor entries for EPMDocuments. EPMDocumentUnitInfo holds the scale factor (conversion factor between Creo and Windchill units) and Unit system names of Windchill. It sources the data from EPMDocumentUnitInfo staging database table.

EPMMemLinkUnitInfo Loader Details

This loader creates the unit information and scale factor entries for EPMMemberLinks. EPMMemLinkUnitInfo holds the scale factor (conversion factor between Creo and Windchill units) and Unit system names of Windchill. It sources the data from EPMMemLinkUnitInfo staging database table.

EPMReferenceLink Loader Details

This binary link loader is used to define non-structural associations of type wt. `epm.structure.EPMReferenceLink` between EPMDocuments in Windchill including CAD model to Drawing references, or for CATIA data, MML and/or Kinematic links.

The loader supports the definition of key Windchill attributes such as:

- Role A (EPMDocument)/Role B (EPMDocument)
- Dependency Type
- Child Name
- Required
- Type Definition Reference
- Unique Link ID

The following are unsupported use cases:

- The loader does not fully support or has not been tested for Creo or Pro/ ENGINEER CAD model to drawing relationships or other non-structural model to model dependencies.
- The loader does not support CATIA V4 or CATIA V5 MML and Kinematic links.

EPMFamilyTable Loader Details

This loader creates the composition of a family table in the target Windchill system. This includes connecting the family table members (EPMDocuments) to the EPMSepFamilyTable object via EPMContainedIn links and defining the generic-instance hierarchy via EPMVariantLink. It sources the data from EPMCONTAINEDIN and EPMVARIANTLINK staging database tables.

EPMFTProp Loader Details

This loader creates the family table matrix for family tables successfully created by the EPMFamilyTable loader. This includes creating family table (a) columns for parameters, features, designated parameters / dimensions and components [applicable to assembly family tables], (b) cells and (c) cell values. It sources the data from EPMFTPROP staging database table.

EPMAssStored Loader Details

This loader creates EPMAssStoredConfig and EPMAssStoredMember objects in Windchill and enables collection of as-stored dependencies for a seed parent EPMDocument iteration in target Windchill. It sources the data from EPMASSTORED staging database table.

EPMDescribeLink Loader Details

This binary link loader enables the creation of relationships of type wt.epm.structure.EPMDescribeLink that are used in Windchill to define the WTPart to EPMDocument link. The usage of this link is intended to provide a passive association between the part and CAD document and not intended to drive CAD or BOM structure. The link is shown as type Content in Windchill 10.0.

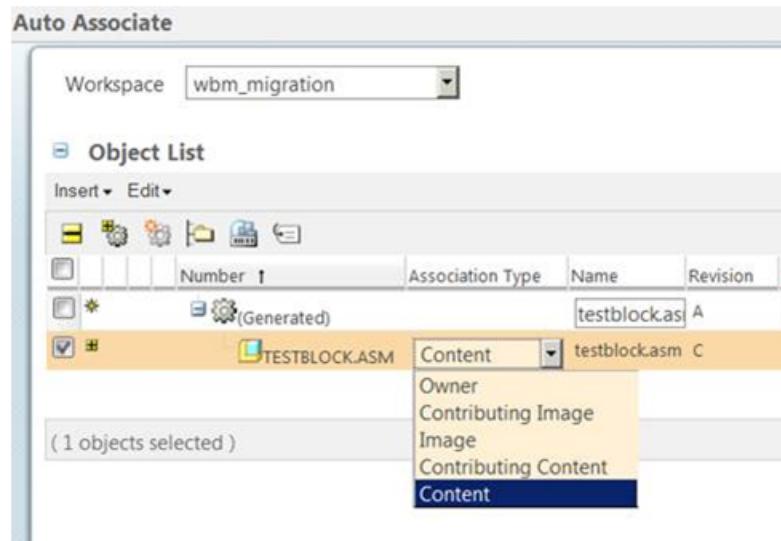
The loader supports the definition of key Windchill attributes such as:

- Role A (Describes: WTPart)/Role B (Described By: EPMDocument), where cardinality is many-to-many
- Part Number, Revision and Iteration
- EPMDocument Number, Revision and Iteration

The following is a supported use case:

Create a Describe Link between an existing WTPart and EPMDocument. The link is uniquely defined and may be created to a specific Part or EPMDocument Number, Revision, and Iteration.

The following shows Windchill 10.0 displaying the different types of WTPart to EPMDocument association types for the implementation of the EPMDescribeLink and EPMBuildRule:



EPMBuildRule Loader Details

This binary link loader is used to create an active association of type wt.epm.build.EPMBuildRule between a WTPart and an EPMDocument where the intent is to have the EPMDocument drive or pass information to the WTPart, such as attribute, structure information, or representation.

The loader supports the definition of key Windchill attributes such as:

- Role A (Build Source: EPMDocument)/Role B (Build Target: WTPart)
- Build Type
- Unique Link ID (where this ID is generated automatically by the Windchill Bulk Migrator for every EPMBuildRule that is created in the staging area by the Migration Administrator)

The following is a supported use case:

Create a Build Rule between a WTPart and an EPMDocument. The Build Type (owner, contributing image, image, contributing content) can be set by the migration administrator and is defined in staging for the build rule object type.

Note

There can be only one owner link between a WTPart and an EPMDocument. If use cases exist where additional relationships are needed between a single WTPart and multiple EPMDocuments, then the additional link definitions should be created as EPMDescribeLinks.

The implementation of the Build Type, and thus the behavior of the Build Rule, is controlled by the following characteristics:

- Build Structure: Indicates that the EPMDocument drives the WTPart structure.
- Build Attributes: Indicates that attributes will be published or contributed from the EPMDocument to the associated WTPart
- CAD Representation: Indicates that when the corresponding CAD document is added to an assembly, the build process will add its associated part to the generated product structure. The process will also transfer the model's viewable/representation file from the CAD Document to the associated WTPart.

The Build Type of the WTPart and EPMDocument relationship is defined in the staging database table for the Build Rule by providing a numerical value for the Build Type column. The value assigned is additive for each of the different characteristics or behavior options that control the passing of information from the EPMDocument to the WTPart. The options, implemented as internal flags with assigned values are:

- BUILD_STRUCTURE=1
- BUILD_ATTRIBUTES=2
- CAD_REPRESENTATION=4

The corresponding values for each of the supported Build Types are therefore:

- Owner: BUILD_STRUCTURE|BUILD_ATTRIBUTES|CAD REPRESENTATION = (1|2|4 = 7)
- Contributing Image: CAD REPRESENTATION|BUILD_ATTRIBUTES = (4|2 = 6)
- Image: CAD REPRESENTATION = 4
- Contributing Content: BUILD_ATTRIBUTES = 2

EPM Build History Migration

The EPM Build History Loader allows you to migrate build history relations between WTPartStructure and the related CAD Structure. The following loaders contribute to loading the details for the EPM Build History:

1. WTPartUsageLinkLoader
2. PartUsesOccurrenceLoader
3. EPMBuildHistoryLoader

The EPMBuildHistory Loader depends on EPMBuildRule. EPMBuildRule must be loaded prior to EPMBuildHistory. The order of execution is very important with the enhancement of WTPartUsageLinkLoader and the creation of PartUsesOccurrenceLoader. By default, attributes are extracted on both WTPart and EPMDocument and migrated. They are loaded with the respective primary objects.

The use cases supported with this functionality include:

In a CAD-driven BOM

- The WTPartUsageLink schema and PartUsesOccurrence Schema contains CAD information. EPM Build History Migration makes it possible to migrate all CAD related data like EPMDocument, EPMMemberLink, EPMReferenceLink and EPMBuildRule prior to WTPartUsageLink and PartUsesOccurrence.
- WTPartUsageLink data must be migrated prior to PartUsesOccurrence as PartUsesOccurrence also contains references to WTPartUsageLink.

Other uses

- WTPartUsageLink must be loaded prior to PartUsesOccurrence as PartUsesOccurrence has references to WTPartUsageLink.

Enterprise Systems Integration Data Loaders

The Windchill Bulk Migrator provides support for the loading of the following Enterprise Systems Integration (ESI) object types:

-
- com.ptc.windchill.esi.tgt.ESITargetAssignmentLink
 - com.ptc.windchill.esi.txn.ESIRelease
 - com.ptc.windchill.esi.txn.ReleaseActivity
 - com.ptc.windchill.esi.txn.ReleaseActivityMessage
 - com.ptc.windchill.esi.txn.ESIReleaseObjectLink
 - com.ptc.windchill.esi.txn.ESITransactionRelease
 - com.ptc.windchill.esi.txn.ESIReleaseTransactionLink
 - com.ptc.windchill.esi.txn.ESITransaction
 - com.ptc.windchill.esi.txn.ESITransactionMessage
 - com.ptc.windchill.esi.txn.ESITransactionTargetLink
 - com.ptc.windchill.esi.txn.ESITransactionObjectLink
 - com.ptc.windchill.esi.bomAlternateItemGroup

In addition to associating Windchill objects to an ESI distribution target, the strategy for how the object and its related objects are published and released to manufacturing are determined during the installation and configuration of the ESI module. The release to manufacturing process generally is initiated when a release activity is created for the given object and an associated target. The initiation of the release to manufacturing process is dependent on the ESI configuration, in particular, settings of the following two ESI preferences:

- com.ptc.windchill.esi.part.configSpec.state

This preference defines the life cycle state values of a part at which the ESI workflow is to be auto-launched.

- com.ptc.windchill.esi.wf.autoLaunch

This preference must be true for the ESI Release to Manufacturing process to be initiated. By default, this is set to true.

The Windchill Bulk Migrator ESIReleaseActivityLoader has been implemented with special behaviors that govern the initiation of the ESI release to manufacturing process. This loader takes into consideration the previous two ESI preferences.

If the life cycle state of the subject part is a member of the values defined within the com.ptc.windchill.esi.part.configSpec.state preference, then it is considered valid to initiate the ESI release to manufacturing process. By default, this is set to RELEASED.

If the subject part state is not in the defined list the process will not be initiated. In addition, The com.ptc.windchill.esi.wf.autoLaunch preference defines whether or not the ESI workflow is launched automatically. This preference must be true for the ESI release to manufacturing process to be initiated. By default, this is set to true. Setting the value of this preference to false disables the initiation of the

workflow during data loading. In this manner, it is possible to segregate a small batch of release activities where it is necessary to initiate the release to manufacturing process.

Classification Attribute Configuration

You can extract and load data related to classified parts for Windchill to Windchill migrations. Loading is also supported for third party, non-Windchill source system to Windchill migrations.

Note

Windchill Bulk Migrator does not support the creation of the classification structure or the system attribute holders associated with the classification nodes. Creation and management of the structure must be done through Windchill CounterPart for Windchill 10.1 and earlier. Beginning with Windchill 10.2, classification structures can be managed from within Windchill, and CounterPart is no longer required to manage classification structures. See [Supported Object Types on page 298](#) for the name of the loader and loader alias.

Classification Data Loading Order

Note

Before loading classification data, the target Windchill system Classification cnode structure must be defined in the Classification Administrator and the corresponding classification attribute definitions must be created.

The following describes the intended order of migration for parts and part classification attribute data:

1. Run the Windchill Bulk Migrator PartsLink Classification Attribute Map Generator command line utility on the target Windchill system. The utility analyzes the target Windchill system and reports on key internal classification attribute definitions. These definitions accurately map and populate the classification attribute data in the Windchill Bulk Migrator staging area to the correct Windchill part and classification attribute.
2. For Windchill-to-Windchill migrations, run the `com.ptc.windchill.migration.wbm.wc2wc.mapping.client.WBMMappingClient` command line utility on the source Windchill system. The utility analyzes the source Windchill system and

creates the `WBMClassificationMapping.xml` and `WBMTTypeDefMapping.xml` files by default, that properly maps classification attribute data to the correct Windchill Bulk Migrator staging table. The `WBMClassificationMapping.xml` file should be edited to provide the correct `targetAttrName` and `targetNodeName` for the target Windchill attributes. The `WBMTTypeDefMapping.xml` file should be edited to provide the correct `targetAttrName` for the target Windchill attributes.

WBMClassificationMapping.xml	
Source (Windchill 10.1, Windchill 10.0, or Windchill 9.0)	Target (Windchill 10.2)
<code>NodePath srcNodePath</code> Example: Electronic Parts / CHIPSET	<code>targetNodeName</code> Example: CHIPSET
<code>Attribute srcAttrGlobalDefName</code> Example: RoHS_Compliant	<code>targetAttrName</code> Example: IBA CC8CF9B8C115
WBMTTypeDefMapping.xml	
Source (Windchill 10.1, Windchill 10.0, or Windchill 9.0)	Target (Windchill 10.2)
<code>srcAttrGlobalDefName</code> Example: IBA Part	<code>targetAttrName</code> Example: partclassify

3. Populate the Windchill Bulk Migrator staging database tables with the relevant part and classification attribute data that is to be loaded.
 - For Windchill-to-Windchill migrations, use the WBM Extractor utility to perform this step automatically.
 - For third party, non-Windchill migrations to Windchill, you must extract the data from the source system and create load files that conform to the Windchill Bulk Migrator staging database table design for part and classification.
4. Load the parts to the target Windchill system. Along with WTPart, classification attributes also get migrated from the IAT_WTPart table (normalized table).
5. Validate the migrated data.

Attribute Map Generator

The attribute map generator utility analyzes the target Windchill system and reports on key internal classification attribute definitions. These definitions accurately map and populate the classification attribute data in the Windchill Bulk Migrator staging area to the correct Windchill part and classification attribute.

To run the attribute map generator, enter the following command in a Windchill shell:

```
windchill com.ptc.windchill.migration.wbm.util.PartsLinkAttrMapGenerator
```

The utility generates the output report in the Windchill logs directory with the name `ClassificationAttrMap.csv`.

The output report lists each classification attribute created in the target Windchill system along with information that is needed to help map the attributes correctly in the Windchill Bulk Migrator staging database to the classification data that is staged and migrated.

The following map file columns are available:

Column	Description
Node Name	Classification structure node name.
Internal Name	Internal name of the classification attribute
Name Space	Name of the classification structure.
Path	The node structure path to the node name. This path is mapped to the STRINGVALUE in the ATT_WTPart column in the staging table.
IBA Name	This value must be unique. Attribute Name corresponds with the global attribute name. The unique value must be mapped to IBANAME in the Windchill Bulk Migrator staging area. For Windchill-to-Windchill migrations, this value should populate the targetAttrName for the given attribute in the WBMClassificationMapping.xml file.
Display Name	The Windchill attribute name that is displayed in the user interface. This value does not have to be unique in

Column	Description
	<p>Windchill.</p> <p> Note Do not map the display name to the IBANAME in the Windchill Bulk Migrator staging database.</p>
Logical Identifier	The classification attribute logical identifier.
Data Type	The attribute type. The entry must be an allowed value for a Windchill attribute type such as: String, Timestamp, URL, Real, or Boolean.

windchill com.ptc.windchill.migration.wbm.wc2wc.mapping.client.WBMMappingClient Command

The `ClassificationMappingXMLHandler.java` command cross-references the source Windchill system and generates a default mapping XML file for the classified attributes. This mapping file is used as input for the Windchill Bulk Migrator extractors for the extraction of classification data. Before extraction, you must edit the mapping XML file and update the entries for `targetAttrName` which correspond with the unique attribute values configured for the target Windchill system.

Staging Tables

Classification Attribute and Node Name are extracted to the `ATT_Part` table in staging the database along with `WTPart`.

Other Usage Notes

- Windchill Bulk Migrator 4.0 supports loading to multiple classification nodes, as well as true, multi-value attribute values for each attribute associated with the classified part.
- If there is no classification attribute (null values) provided in the staging pertaining to a classification node, upon loading, it will result in a null value (for example, no values assigned in Windchill). However, there is an option to populate such classification attributes with default target system values. If you want to populate with default target system values, set the following value in `wbm.properties`:
`com.ptc.windchill.migration.wbm.classifyPart.keepDefaultAttributeValue=true`
By default, this is set to false. If set to false, no value is assigned to the attribute and does not exist in staging.

Representation Migration

You can extract and load data related to Windchill Visualization Representations for Windchill to Windchill migrations.

Note

Windchill Bulk Migrator does not support the actual creation of the representations by Windchill visualization services or the verification of the representation against the owning CAD or part structure in the source and target system. Windchill Bulk Migrator implements the Windchill APIs that are used to extract and create data in Windchill. See [Supported Object Types on page 298](#) for the name of the loader and loader alias.

Representation Migration Order

Note

The source and target Windchill systems must be available.

The following describes the intended order of migration for representation (derived image) objects.

1. Populate the Windchill Bulk Migrator staging database tables with the relevant data to be loaded, including:
 - part
 - document
 - CAD data
 - all related link and relationship information

Tip

For Windchill to Windchill migrations, you can use the Windchill Bulk Migrator extractor utility to perform this step automatically.

2. Enter the shared folder path (viewable directory) in the source system Windchill Bulk Migrator extractor .xconf property. Run the `xconfmanager -p` command in a Windchill shell and restart the Windchill method server to register the property update.

-
3. Enter the shared folder (viewable directory) in the source system Windchill Bulk Migrator extractor.xconf property. Stop Windchill by running the xconfmanager -p command in a Windchill shell and restart the source system Windchill method server to register the property update.
 4. Extract the representation objects to the Windchill Bulk Migrator staging database tables using the Windchill Bulk Migrator extraction capabilities.

 **Note**

The object class used for the command line is:

`wt.viewmarkup.DerivedImage`. This step creates subfolders in the viewable directory and extracts all representation file types that represent the Windchill visualization object types.

5. Load the parts, documents, CAD data, and all of their related link, relationship information, and ContentMetadata to the target Windchill system using the WBM loading capabilities.
6. Publish the migrated object types in the target Windchill system from the restricted migration domain to the public domains.

 **Note**

This step must be completed prior to loading the representation objects.

-
7. Load the representation object data to the target Windchill system. This step will pre-process the representation data and update the Windchill Bulk Migrator staging tables for the object type. It also initiates the loading process.

Note

The process creates additional file system directories that are used by Windchill Visualization Services during loading. Representations that are to be loaded are either copied or moved into the directories automatically during the loading process. The action to copy or move is controlled by the com.ptc.windchill.migration.wbm.moveStagedService dImageContent=false/true property in wbm.properties. If set to false, the tool copies representation files from the extraction folders to the Windchill Visualization Services folders during loading. If set to true, the tool moves representation files from the extraction folders to the Windchill Visualization Services folders during loading. By default, this property is set to false.

8. Validate the migrated data.

Source System Extractor.xconf Properties

The shared directory that holds the extracted representation files must be defined in the Windchill Bulk Migrator extractor.xconf file.

1. Navigate to the extractor.xconf file on the source Windchill system at <Windchill>/codebase/com/ptc/windchill/migration/wbm/wc2wc/extractor/extract.xconf.
2. Search for the following property:

```
<Property default=" " name="com.ptc.windchill.migration.wbm.util.wc2wc.viewablesDirectory" overrideable="true" targetFile="codebase/wt.properties"/>
```

The shared directory location which is accessible by network should be entered inside of the quotations for Property default=" ". For example:

```
<Property default="\server\name\ViewablesDirectory" name="com.ptc.windchill.migration.wbm.util.wc2wc.viewablesDirectory" overrideable="true" targetFile="codebase/wt.properties"/>
```

Staging Tables

DERIVEDIMAGE – This table includes information about the representation objects that are to be loaded into the target Windchill system.

Change Object Migration

When loading change objects using the Windchill Bulk Migrator, the migration consultant must consider whether the change objects to be loaded need to be versioned. If the change objects must be versioned, the following actions must be completed:

- The associated change object life cycles must be configured to allow **Revise** actions for the life cycle states that are applicable.
- The change objects listed in the following table and their out-of-the-box life cycle templates by default deny **Revise** actions and must be updated to support revisions.

Business Class	Appropriate Life Cycle Templates
wt.change2.WTChangeIssue	Problem Report Life Cycle
wt.change2.WTChangeOrder2	Change Notice Life Cycle
wt.change2.WTChangeRequest2	Change Request Life Cycle

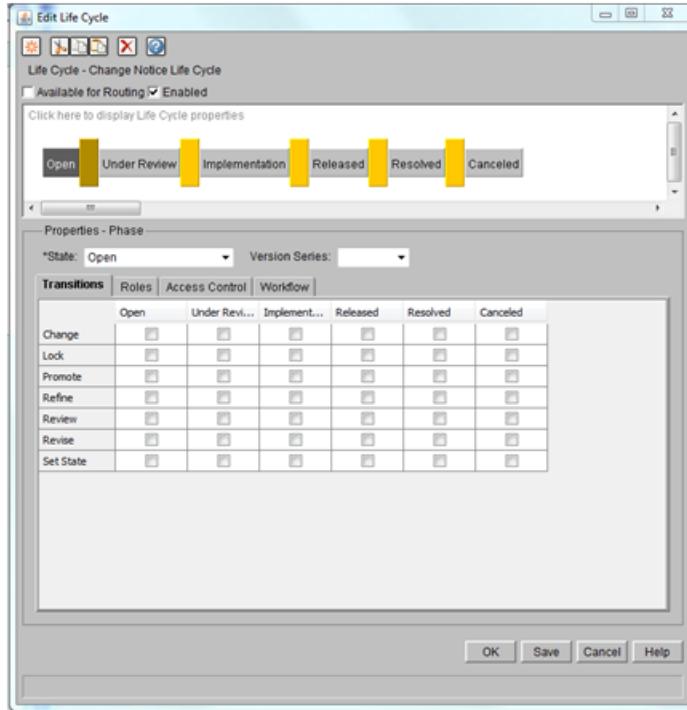
- If the out-of-the-box templates are used with change versions, then the templates must be modified and the **Revise** option checked at appropriate transition levels.

Editing Life Cycle Templates to allow Revise

For the initial change object creation with the Windchill Bulk Migrator, the first version of a change object uses the standard revise APIs to set the initial revision of the change object. Subsequent object versions use the ChangeHelper APIs and, as such, the object is bound by the change management business services.

Therefore, to revise a change object the **Revise** action must be enabled for the state transition in the associated life cycle.

By default, the **Revise** action is disabled in the out-of-the-box change management life cycles as illustrated in the following figure:

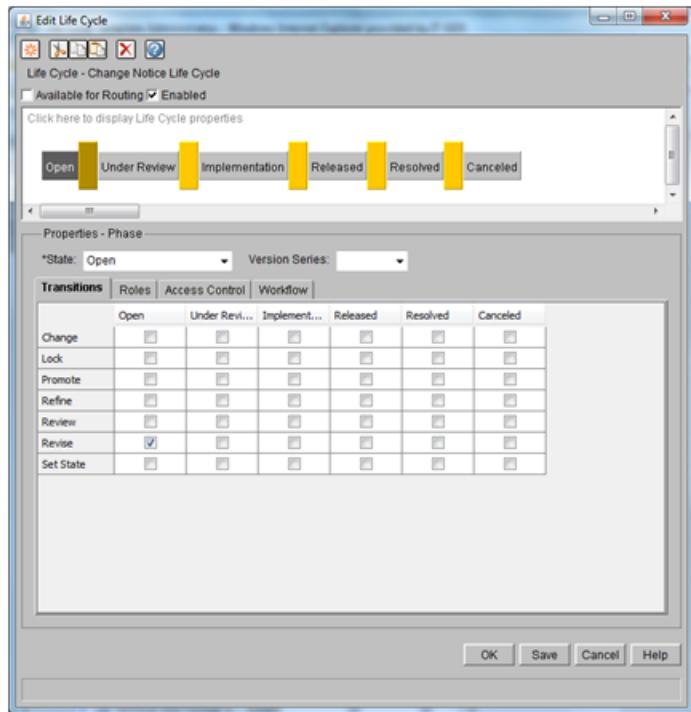


The figure displays the out-of-the-box Change Notice Life Cycle. Notice that in the **Open** state, there is no **Revise** Transition defined. Therefore, the object cannot be revised in this state.

Attempting to load change order versions with the default Change Notice Life Cycle results in the object not being created and the following exception being thrown:

```
(wt.lifecycle.lifecycleResource/223) wt.util.WTException: Revise transitions not  
defined for current version in life cycle template.
```

Update the Change Notice Life Cycle as illustrated in the following figure:



The figure shows the **Revise** Transition selected in the **Open** state. Therefore, the object can now be revised in this state.

Flexible Change Item Delegate Configuration

Note

Change in Association Modes for Change Management Objects

For Windchill 11.1 F000 or later, the default change association mode is Flexible.

In order to migrate the **Legacy** links data using Windchill Bulk Migrator, the system should be set back to **Mixedmode**, using the Windchill `FlexibleChangeConverter` conversion utility.

You can convert a change items from legacy to flexible when it meets readiness criteria. For change items migrated through Windchill Bulk Migrator, delegate – `WBMChangeItemReadinessDelegate` has been introduced to check the readiness criteria.

To register the WBMChangeItemReadinessDelegate delegate, add the following property in the ChangeMgmt-service.properties.xconf file and propagate it:

```
<Option serviceClass="wt.change2.flexible.WorkflowReadinessDelegate"
        selector="20"
        requestor="wt.change2.FlexibleChangeItem"
        cardinality="singleton"/>
Out of the box delegate registration:
<!--Delegates for computing readiness state for converting legacy change items.
The selector attribute must be unique and must be an integer as the delegates
are called in order based on the selector value
-->
<Service context="default" name="wt.change2.flexible.FlexibleChangeItemReadinessDelegate">
<Option serviceClass="wt.change2.flexible.ResolutionDateReadinessDelegate"
        selector="10"
        requestor="wt.change2.FlexibleChangeItem"
        cardinality="singleton"/>
<Option serviceClass="wt.change2.flexible.WorkflowReadinessDelegate"
        selector="20"
        requestor="wt.change2.FlexibleChangeItem"
        cardinality="singleton"/>

</Service>
```

Note

- Multiple delegates can be registered, and delegates can be added or removed to configure the conversion process according to the requirement.
- The delegates are executed in ascending order of their selector number. The selector number should be unique.
- WBMChangeItemReadinessDelegate should be given highest priority (selector=1).

Managing Life Cycle Services and Life Cycle History During Loading

The Windchill StandardLifeCycleService runs various listeners that perform key actions as new business objects and versions/iterations are created. For example, the service is responsible for creating Windchill team instances and associating them to corresponding business objects, along with several other activities which are important and should be controlled during bulk migration, such as:

-
1. Setting access controls depending on the life cycle state
 2. Optionally, Saving life cycle history
 3. Initiating workflow execution events based on the workflow configuration for the state of the object associating to advanced life cycle templates

Setting Access Controls Depending on the Life Cycle State

During past migration experience it was observed that the access controls were being applied by the Windchill listeners only when creating the first iteration of a given revision, and were skipped for any subsequent iteration of a given branch. While this behavior caters appropriately to while executing through normal business operations where other events were responsible for taking care of setting appropriate access, but have been a problem during loading during migration.

Beginning with 2.0 and continuing with newer Windchill Bulk Migrator releases, the Windchill services were enhanced to ensure the services set the appropriate ACLs while performing a load operation through Windchill Bulk Migrator.

Saving Life Cycle History (Optional)

Optionally, while saving the object the life cycle services listener can also create corresponding life cycle history objects that record information about the data management events. If you want to migrate an object's life cycle history information explicitly for a given life cycle managed object from a legacy source system, set the following property in `wbm.properties.xconf` to true before starting the migration:

```
com.ptc.windchill.migration.wbm.load.lifecyclehistory.explicit
```

Setting

```
com.ptc.windchill.migration.wbm.load.lifecyclehistory.explicit =true
```

prevents the automatic creation of a life cycle history object for migrated objects. You can then load LifecycleHistory objects for each object as defined in the Windchill Bulk Migrator staging area. The active StandardLifeCycleService ensures that all migrated WTPart or WTDocument objects are initialized correctly with the correct Windchill team instances and life cycle history objects.

Note that setting `wt.lifecycle.enablehistory=false` in `wt.properties` will override the settings done for `com.ptc.windchill.migration.wbm.load.lifecyclehistory.explicit` globally at the Windchill level. The scope of the setting `com.ptc.windchill.migration.wbm.load.lifecyclehistory.explicit` is limited to the load transaction through Windchill Bulk Migrator and does not affect the entire Windchill instance; thus, it will not impact the other active users on the target.

Initiating Workflow Execution Events Based on Workflow Configuration

Workflows are not initiated upon loading data into Windchill using the Windchill Bulk Migrator. You can initiate a workflow using the **Reassign Life Cycle** or **Set State** action. For more information, see [Workflow Migration on page 206](#).

LifecycleHistory Data Loader

This section describes implementation considerations when loading life cycle history objects for product data types such as WTParts or WTDocuments.

When operating normally, the Windchill StandardLifeCycleService runs as a listener and performs key actions as new business objects and iterations are created. For example, the service is responsible for creating Windchill Team instances and associating them to corresponding business objects. Optionally, it can also create corresponding life cycle history objects that record information about the data management events.

Migrating into a Live Production System

PTC strongly recommends fully testing and rehearsing the bulk migration on test hardware prior to migrating to production hardware. Functional testing of migrated data should also occur after each test migration and prior to production migration. Functional testing includes executing the business use cases to validate the target system configurations (such as object initialization, life cycle and workflows) as well as validating the data mappings and transformations that may occur as part of the migration. Testing the migration tools and procedure ensures that production migration can be smoothly executed and functional testing ensures that the migrated data can be accessed and used as intended in the target system.

Note

Migration creates new data in the target system. It is not possible to undo the resulting transactions or remove new objects except by completely restoring the system using a data snapshot of the system before migration occurred. For this reason, it is critical to have a rigorous validation plan as part of the project and to execute that plan before the production migration.

For a list of supported versioned, non-versioned, and link objects that can be created in the migration, see [Supported Object Types on page 298](#).

The Windchill Bulk Migrator does not require that the target Windchill system be offline to users during the execution of the migration. That is, there are no restrictions on user Read/Write access to the target Windchill system during migration. However, PTC recommends as a general good practice that the production migration occur while the system is offline to users.

Migration to an offline system eliminates the risk that administrators or users can create data or make system changes that cause any of the following issues:

- New data could conflict with the data that is migrated.
- Changes to the system could invalidate data mappings.
- Changes to the system could invalidate object behavior.

If Read access to the data is needed during migration, then consider using one of the following options:

- Create a clone of the production system that then would be available to users while the migration to the production system is occurring.
- Execute the ROAPController command that puts your production Windchill solution in read-only mode for the duration of the migration. PTC includes an ant script for this purpose as part the Windchill Bulk Migrator code that is installed.

For details on using the ROAPController command, see [Setting the Active Production System to Read Only on page 291](#).

If Read/Write access to the data is needed during migration, you should carefully implement a plan that mitigates the risks described in this section. The risk mitigation plan should incorporate a backup and restore procedure in the event that a rollback is needed.

 **Note**

For migrations where the target system is made available to users in a Read/Write state, a full restore (recovery) of the pre-migrated state will eliminate any additions or changes to the production data that occurred during the migration. This is another reason why migrating in an offline mode is a lower risk migration option.

Iterable foldered objects that are migrated can be cordoned off from active user access to avoid conflicts. This is done by loading the data into restricted domains. You can use restricted domains for migrated data when allowing users read-only access or when allowing users read-write access to the production data. PTC recommends that you always use restricted domains to ensure that the migrated data can be inspected before it is available on the production system. The default migrator settings use restricted domains as part of the migration process.

Using Restricted Domains in the Migration Process

The Windchill Bulk Migrator uses properties in the `wbm.properties` file to control the use of restricted domains and to control which object types can be loaded into a restricted domain. For property details, see [Restricted Domain Configurations in `wbm.properties` on page 58](#).

As part of a migration that uses restricted domains, the Windchill Bulk Migrator creates the following:

- New Windchill domains at the site and organization levels.
- Associated access control rules.
- Migration-specific user groups that restrict access to objects associated with the new domains.

Access to a restricted domain can be enabled by identifying specific users and adding them to migration groups. Data that is migrated into Windchill is assigned to this restricted access domain until a command is given to the migration utilities to transition the data to active target domains. Using migration groups allows control of user visibility to the migrated data, even when the target system is in normal Read/Write mode during migration. This also enables performing simple validations on the migrated data while in the restricted domain prior to go-live.



Note

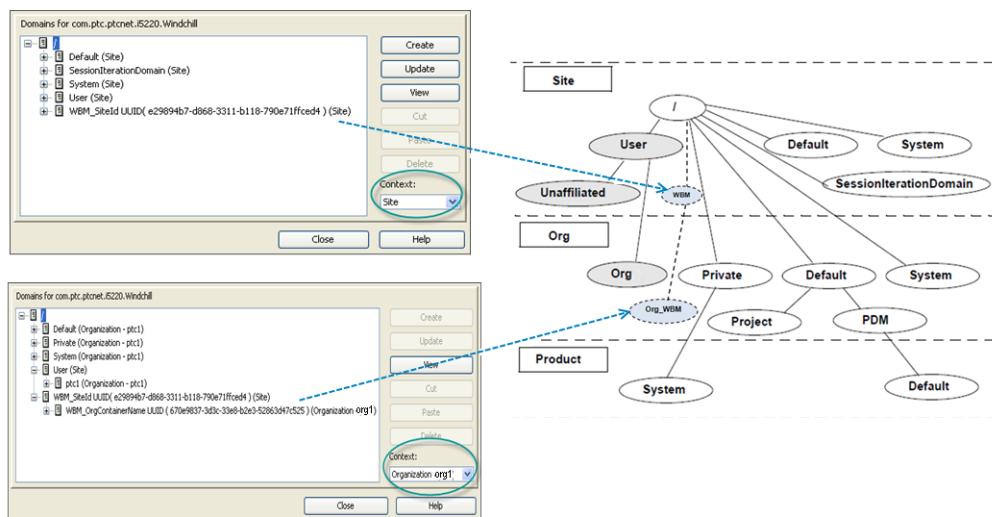
PTC recommends that simple validations involve only those types of checks that involve Read and Download. It is not recommended to perform functional validations of the type that would Revise or create new objects, since those types of activities could create conflicts with data that is yet to be migrated, or create new object iterations that are not meant to be in the production life cycle history. Any functional validations involving changes to the data should occur after the transition to the active domains.

The data that is bound by the access control rules set in the restricted domains is versioned product data (specifically, foldered Iteratable objects which are `DomainAdministered`). Any non-versioned objects that are created by the migration are not subject to the restricted domain. Examples of non-versioned data are administrative data (such as containers, folders and ACLs) and some link data that are used to define relationships to the versioned objects. The links will not be user accessible if the objects to which they link are not accessible.

Restricted Domain Architecture

The names of the restricted domains that are created are determined by values that are set in MigrationSourceSite table in the staging database. For details on this table, see [Installing the Staging Schema on Cross-Platform Configurations on page 261](#).

The following diagram shows a sample of domains created under the **Site** and **org1** contexts:



System Participants for Restricted Domains

The Windchill Bulk Migrator creates the following system participants that, if needed, are granted access to a restricted domain:

- **MigrationAdministrator**

The **MigrationAdministrator** group initially includes Windchill users who are either designated as Windchill site administrators (those users in the Windchill Administrators group) or designated as Windchill organization administrators (those users listed in the **Organization Administrators** table for each organization).

A site administrator can load data and access data in the restricted domain across all organizations. An organization administrator can load data and access data in the restricted domain only in his or her own organization context.

- **MigrationTestUser**

You can add users to MigrationTestUser group through the Participant Administration utility. Users in this group are allowed Read and Download access to data associated with the restricted domain. Use of this group is intended to enable simple verification activities during migration.

Access Control Policies for Restricted Domains

The following diagram shows an example of the access control policies that the Windchill Bulk Migrator created for the restricted domain under the org1 context:

Domain: /WBM_OrgContainerName UUID (d448b959-3670-3db7-9092-3122172fb13c)						
Context: Site, Organization - org1						
Properties Access Control Indexing Notification						
Domain	Context	Type	State	Principal	Permissions	Applies To
WBM_OrgContainerName UUID (d448b959-3670-3db7-9092-3122172fb13c) org1	WTObject	All	MigrationTestUser	+Read +Download	Principal	
WBM_OrgContainerName UUID (d448b959-3670-3db7-9092-3122172fb13c) org1	WTObject	All	MigrationAdministrator	+Full Control (All)	Principal	
WBM_OrgContainerName UUID (d448b959-3670-3db7-9092-3122172fb13c) org1	WTObject	All	Organization Administrat...	+Full Control (All)	Principal	
Root	Site	DeliverableDefinition	All	ALL	+Read +Create	Principal
Root	Site	Team	All	OWNER	+Read	Principal
Root	Site	EPMDocConfigSpec	All	ALL	+Full Control (All)	Principal
Root	Site	Meeting	All	OWNER	+Full Control (All)	Principal
Root	Site	DataMonitor	All	ALL	+Read +Modify +Cr...	Principal
Root	Site	ESITarget	All	ESI Administrators	+Full Control (All)	Principal
Root	Site	ESITarget	All	ESI Authors	+Read	Principal
Root	Site	DerivedImage	All	OWNER	+Full Control (All)	Principal
Root	Site	DerivedImage	All	ALL	+Read +Create +D...	Principal
Root	Site	DerivedImage	All	Team Members	+Full Control (All)	Principal
Root	Site	ESITransaction	All	ESI Administrators	+Full Control (All)	Principal

The **Principal** column shows the MigrationTestUser, Organization Administrator, and MigrationAdministrator participants for the domain and the **Grant Permissions** shows the permissions set for those participants. Notice that the MigrationTestUser group has only Read and Download permissions while the other groups have all permissions.

Setting the Active Production System to Read Only

If, while you are migrating data into an active production system, users must be able to read objects and download content from the system, a site administrator can run a command that sets up the system in a read-only mode. When the migration is complete, the command can also be used to restore the system to its original settings.

Running the command that sets the system to a read-only mode performs the following tasks:

- Saves the states of all queues before setting the system to a read-only mode.

The states are saved in the <Windchill>/codebase/wbm/wbmROAPWtQueuesBkp.bin file, where <Windchill> is the Windchill installation directory.

- Changes the states of all queues to stopped.

The task that changes queue states uses the following default property values from properties in wbm.properties:

Property	Description	Default Value
wbm.queue.max.retry.count	Number of attempts made to change the queue state.	3
wbm.queue.retry.interval	Number of milliseconds between retries.	1000

As a site administrator, check to ensure that all queues have been stopped before migrating data. If all are not stopped, you can change the property values listed and rerun the command, or stop the queues manually.

- Creates the WBM_ROAP_ML group in the site context and includes the site administrators and organization administrators as members of the group.
- Creates an access control rule at site level that sets absolute deny. The following screen shot highlights the rule that is created in the site context root domain for WTObject:

The screenshot shows the 'Administrative Domain' interface with the 'Access Control' tab selected. The 'Context' is set to 'Site'. A specific row in the table is highlighted, showing a rule for 'WTObject' with 'All' permissions. This row is part of a larger table listing various objects and their permissions across different domains and contexts.

Domain	Context	Type	State	Principal	Applies To	Grant ...	Deny ...	Absol...
Root	Site	DerivedImage	All	teamMembers	Principal	Full Cont...		
Root	Site	LifeCycleMa...	All	teamMembers	Principal	Modify D...		
Root	Site	WTObject	All	[WBM_ROAP_ML]	All except principal	Read Do...	Modify ...	
Root	Site	Team	All	OWNER	Principal	Read		
Root	Site	Meeting	All	OWNER	Principal	Full Cont...		
Root	Site	DerivedImage	All	OWNER	Principal	Full Cont...		
Root	Site	WTMarkUp	All	OWNER	Principal	Modify D...		
Root	Site	Notification...	All	OWNER	Principal	Full Cont...		
Root	Site	ESITarget	All	ESI Authors	Principal	Read		
Root	Site	ESITransact...	All	ESI Authors	Principal	Read		
Root	Site	ESIRelease	All	ESI Authors	Principal	Read		
Root	Site	ESITarget	All	ESI Administrators	Principal	Full Cont...		
Root	Site	ESITransact...	All	ESI Administrators	Principal	Full Cont...		
Root	Site	ESIRelease	All	ESI Administrators	Principal	Full Cont...		
Root	Site	WTObject	All	Administrators	Principal	Full Cont...		

The current command does not:

- Reset the properties, such as wt.queue.executeQueues, in wt.properties.
- Disable background method servers.
- Disable the polling for queue tasks.

Using ROAPController Command to Set Read-Only Mode

To set the system to a read-only mode, open a Windchill Shell and enter the following command (on one line):

```
windchill <heap_size_args> com.ptc.windchill.migration.wbm.ROAPController -r  
-u <username> -p <password>
```

In the command, the `-r` argument is required. Specifying heap size values (`<heap_size_args>`) and user credential arguments (`-u` and `-p`) are optional:

- `-r` specifies read-only mode.
- `-u <username>` specifies the user under which the command is run. The user name specified in `<username>` must be a site administrator (who is a member of the Windchill Administrators group).
- `-p <password>` specifies the password of the user specified in `-u <username>`.
- `<heap_size_args>` indicates one or more arguments for managing heap size values. Using the Windchill Bulk Migrator on some platforms requires additional java arguments.

If you omit the user credential arguments on the command, the **Authorization Request** window opens; you must enter the credentials in this window.

Using ROAPController Command to Restore System Settings

To restore the system to the settings that were in effect before read-only mode, open a Windchill Shell and enter the following command (on one line):

```
windchill <heap_size_args> com.ptc.windchill.migration.wbm.ROAPController -rw  
-u <username> -p <password>
```

In the command, the `-rw` argument is required and indicates that you want the original settings restored. Specifying arguments for heap size (`<heap_size_args>`) and user credential arguments (`-u` and `-p`) are optional and can be specified as was done when setting the system to read-only mode.

Entering the `com.ptc.windchill.migration.wbm.ROAPController -rw` command deletes the `<Windchill>/codebase/wbm/wbmROAPWtQueuesBkp.bin` file that was created when setting the Windchill system to read-only mode.

Restricted Domain Configurations in wbm.properties

To use restricted domains when migrating data, the following property in `wbm.properties` must be set to true:

```
com.ptc.windchill.migration.wbm.domain.restricted
```

By default, this property is set to true. Therefore, unless you change the default, restricted domains are used in the migration process.

Use the following property to identify the list of object types that can be migrated into a restricted domain:

```
wbm.objectType.supported.restricted.domain=<obj_type_list>
```

In the object type list, separate each object type in the list using a comma. For individual object types in the list, use the last part of internal name of the top-level object type. For example, the internal name for the Part type is wt.part.WTPart. Therefore, include WTPart in <obj_type_list>. If additional types are added using modeled subclasses, those types must be added to the list.

The wbm.objectType.supported.restricted.domain property is also used to identify top-level object types when moving objects from a restricted domain to a public domain.

Understanding the Staging Database Table Attributes

When migrating data from non-Windchill systems into Windchill, it is necessary to populate the ATT_<object> tables with the object attribute definitions. The format of the attribute definition will vary depending upon the type of attribute. The following table describes the different types of attributes and important information for the definition:

IBA Type	Externalized Form	Sample Value	Oracle Data Type Definition	SQL Server Data Type Definition
STRINGVAL-UE	<Value> in string	“An Attribute value”	VARCHAR2 (500)	NVARCHAR (500)
INTEGER-VALUE	<Value> in INTEGER	100	NUMBER	NUMERIC
FLOATVAL-UE	<Value> <Precision>	3.14 3	VARCHAR2 (500)	NVARCHAR (500)
FLOATVA-LUEWITHU-NITS	<Value> <Precision> <Unit>	9.75 3 m	VARCHAR2 (500)	NVARCHAR (500)
BOOLVALUE	0 1 (0:FALSE, 1: TRUE)	1	NUMBER (1,0) CONSTRAINT CHECK (BOOLVALUE IN (1,0))	BIT CONSTRAINT CHECK (BOOLVALUE IN (1,0))
TIMEVALUE	TIMESTAMP WITH TIME	03-06-15	TIMESTAMP WITH TIME	DATETIMEOFFSET

IBA Type	Externalized Form	Sample Value	Oracle Data Type Definition	SQL Server Data Type Definition
	ZONE	06:30:00.000-000000 PM GMT	ZONE	
URLVALUE	<URL> <label>	www.ptc.com PTC	VARCHAR2 (500)	NVARCHAR (500)
REFERENCE-VALUE	<ClassName>:<WBMSourceIdentifier>	wt.doc.WTDocument:56576	VARCHAR2 (400)	NVARCHAR (400)

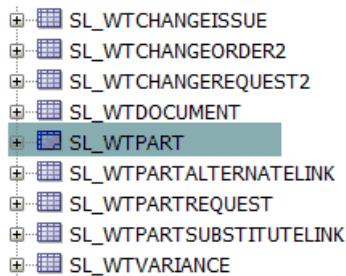
Note

For REFERENCEVALUE attributes, the fully qualified class name provided is considered the referenceable object. WBMSourceIdentifier is an identifier through which the referenceable can be uniquely identified in the source system. For loading, it is necessary to first load the referenceable object before loading any other IBAHolder object using the reference.

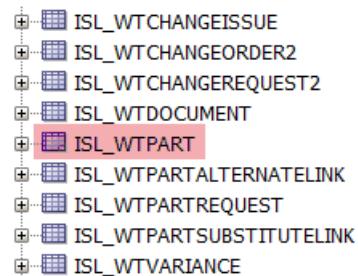
Security Labels

The following are some of the security labels object types tables structure from the Staging Database:

- De-normalized Tables:



- Normalized Tables:



The following is an example of Security labels on Source Windchill Database for WTPART Object type:

The screenshot shows the WTPART table in Oracle SQL Developer. The table has columns: OWNERSHIP, PARTTYPE, PHASE, SECURITYLABELS, SOURCE, and ATGATESTATE. The data row is: 1, separable, (null), 1, SL1=v3,SL6=v1,SL10=v1, make, 0.

OWNERSHIP	PARTTYPE	PHASE	SECURITYLABELS	SOURCE	ATGATESTATE
1	separable	(null)	1, SL1=v3,SL6=v1,SL10=v1	make	0

The following are security label tables in the Staging Database (for example, WTPart object type):

- De-normalized table of SL_WTPART (before Normalization of WTPart):

The screenshot shows the SL_WTPART table in Oracle SQL Developer. The table has columns: SLHOLDERWBMSOURCEIDENTIFIER, SLNAME, and STRINGVALUE. The data rows are: 1, SL1, v3; 2, SL6, v1; 3, SL10, v1.

SLHOLDERWBMSOURCEIDENTIFIER	SLNAME	STRINGVALUE
1	SL1	v3
2	SL6	v1
3	SL10	v1

- Normalized table of ISL_WTPART (after Normalization of WTPart):

The screenshot shows the ISL_WTPART table in Oracle SQL Developer. The table has columns: SLHOLDERREFERENCE, SLNAME, STRINGVALUE, and MFD. The data rows are: 1, 19, SL1, v3, 0; 2, 19, SL6, v1, 0; 3, 19, SL10, v1, 0.

SLHOLDERREFERENCE	SLNAME	STRINGVALUE	MFD
1	19	SL1	v3
2	19	SL6	v1
3	19	SL10	v1

Supplier Management Data Loaders

The Windchill Bulk Migrator provides support for the loading of the following supplier management (SuMa) object types:

- com.ptc.windchill.suma.part.VendorPart
- com.ptc.windchill.suma.part.ManufacturerPart
- com.ptc.windchill.suma.supplier.SupplierChangeEvent
- com.ptc.windchill.suma.supplier.SupplierDocumentLink
- com.ptc.windchill.suma.axl.AXLEntry
- com.ptc.windchill.suma.axl.AXLChangeEvent
- com.ptc.windchill.suma.axl.OEMPartAXLContextLink
- com.ptc.windchill.suma.axlrule.AMLPolicyRuleSet

- com.ptc.windchill.suma.axlrule.AMLPolicyRule
- com.ptc.windchill.suma.axlrule.AVLPolicyRuleSet
- com.ptc.windchill.suma.axlrule.AVLPolicyRule
- com.ptc.windchill.suma.axlrule.AXLRuleToEntryLink
- com.ptc.windchill.suma.npi.WTPartRequest (Content Holder)
- com.ptc.windchill.suma.npi.PartRequestRelatedPartLink

Pre-requisite

- Source Windchill prior to Extraction

Prior to extraction the WBMOrganizationMapping.xml (Mapping file) has to be updated with AXL context information correctly

- Target Windchill prior to Loading
 - WTOrganization/CAGE organization has to be migrated.
 - Supplier administrator has to be created.
 - Supplier organization (Manufacturer and Vendor Organization) should be configured on the target system by supplier administrator.
 - Sourcing context / AXL context should be configured.

B

Supported Object Types

The tables in this appendix identify the supported object types for loaders on versioned, binary link, and non-versioned object types.

Versioned Object Types

The following table has information about the out-of-the-box extract and load object support provided with the Windchill Bulk Migrator which is identified by the versioned business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Content Holder	Binary Link	Category	IBA Holder	Security Labelled
Yes	AuthorizationAgreement	Yes	wt.access.agreement.AuthorizationAgreement	FALSE	FALSE	Generic Loadable	TRUE	TRUE
Yes	Choice	Yes	com.ptc.windchill.option.model.Choice	FALSE	FALSE	Generic Loadable	TRUE	TRUE
Yes	ChoiceConstraint	Yes	com.ptc.windchill.option.model.ChoiceConstraint	FALSE	FALSE	Generic Loadable	FALSE	TRUE
Yes	ChoiceEnable	Yes	com.ptc.windchill.option.model.ChoiceEnable	FALSE	FALSE	Generic Loadable	FALSE	TRUE
Yes	ChoiceMapping	Yes	com.ptc.windchill.option.model.ChoiceMapping	FALSE	FALSE	Generic Loadable	FALSE	TRUE
Yes	ChoiceRule	Yes	com.ptc.windchill.option.model.ChoiceRule	FALSE	FALSE	Generic Loadable	FALSE	TRUE
Yes	EPMDocument	Yes	wt.epm.EPMDocument	TRUE	FALSE	Generic Loadable	TRUE	FALSE
Yes	EnterpriseData	Yes	com.ptc.windchill.enterprise.data.EnterpriseData	FALSE	FALSE	Generic Loadable	TRUE	FALSE
Yes	ExpressionAlias	Yes	com.ptc.windchill.option.model.ExpressionAlias	FALSE	FALSE	Generic Loadable	TRUE	TRUE
Yes	IndependentAssignedExpression	Yes	com.ptc.windchill.option.model.IndependentAssignedExpression	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	ManufacturerPart	Yes	com.ptc.windchill.suma.part.ManufacturerPart	TRUE	FALSE	Generic Loadable	TRUE	TRUE
Yes	MPMControlCharacteristic	Yes	com.ptc.windchill.mpmi.MPMControlCharacteristic	FALSE	FALSE	Generic Loadable	TRUE	TRUE
Yes	MPMMfgProcess	Yes	com.ptc.windchill.mfgprocess.MPMMfgProcess	TRUE	FALSE	Generic Loadable	TRUE	TRUE
Yes		Yes	com.ptc.windchill.mpmi.mfgprocess.	TRUE	FALSE	Generic	TRUE	TRUE

Versioned Object Types

The following table has information about the out-of-the-box extract and load object support provided with the Windchill Bulk Migrator which is identified by the versioned business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Content Holder	Binary Link	Category	IBA Holder	Security Labelled
	MPMMfgStandardGroup		MPMMfgStandardGroup			Loadable		
Yes	MPMOperation	Yes	com.ptc.windchill.mpmpl.processplan.operation.MPMOperation	TRUE	FALSE	Generic Loadable	TRUE	TRUE
Yes	MPMPlant	Yes	com.ptc.windchill.mpmpl.resource.MPMPlant	TRUE	FALSE	Generic Loadable	TRUE	TRUE
Yes	MPMPProcessMaterial	Yes	com.ptc.windchill.mpmpl.resource.MPMPProcessMaterial	TRUE	FALSE	Generic Loadable	TRUE	TRUE
Yes	MPMProcessPlan	Yes	com.ptc.windchill.mpmpl.processplan.MPMProcessPlan	TRUE	FALSE	Generic Loadable	TRUE	TRUE
Yes	MPMResourceGroup	Yes	com.ptc.windchill.mpmpl.resource.MPMResourceGroup	TRUE	FALSE	Generic Loadable	TRUE	TRUE
Yes	MPMStandardCC	Yes	com.ptc.windchill.mpmpl.pmi.MPMStandardCC	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMStandardOperation	Yes	com.ptc.windchill.mpmpl.processplan.operation.MPMStandardOperation	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMSequence	Yes	com.ptc.windchill.mpmpl.sequence.MPMSequence	TRUE	FALSE	Generic Loadable	TRUE	TRUE
Yes	MPMSkill	Yes	com.ptc.windchill.mpmpl.resource.MPMSkill	TRUE	FALSE	Generic Loadable	TRUE	TRUE
Yes	MPMTooling	Yes	com.ptc.windchill.mpmpl.resource.MPMTooling	TRUE	FALSE	Generic Loadable	TRUE	TRUE
Yes	MPMWorkCenter	Yes	com.ptc.windchill.mpmpl.resource.MPMWorkCenter	TRUE	FALSE	Generic Loadable	TRUE	TRUE
Yes	Option	Yes	com.ptc.windchill.option.model.Option	FALSE	FALSE	Generic Loadable	TRUE	TRUE
Yes	OptionSet	Yes	com.ptc.windchill.option.model.OptionSet	FALSE	FALSE	Generic	TRUE	TRUE

Versioned Object Types

The following table has information about the out-of-the-box extract and load object support provided with the Windchill Bulk Migrator which is identified by the versioned business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Content Holder	Binary Link	Category	IBA Holder	Security Labelled
Yes	PartList	Yes	OptionSet	com.ptc.arbortext.windchill.partlist.PartList	TRUE	FALSE	Loadable	TRUE
Yes	PartPathOccurrence	Yes		wt.part.PartPathOccurrence	FALSE	FALSE	Generic Loadable	FALSE
Yes	PlantFunctionalData	Yes		com.ptc.windchill.enterprise.data.PlantFunctionalData	FALSE	FALSE	Generic Loadable	FALSE
Yes	ServiceEffectivity	Yes		com.ptc.arbortext.windchill.siscore.serviceeff.ServiceEffectivity	FALSE	FALSE	Generic Loadable	TRUE
Yes	VariantSpec	Yes		com.ptc.wpcfg.doc.VariantSpec	TRUE	FALSE	Generic Loadable	TRUE
Yes	VendorPart	Yes		com.ptc.windchill.suma.part.VendorPart	TRUE	FALSE	Generic Loadable	TRUE
Yes	WTChangeActivity2	Yes		wt.change2.WTChangeActivity2	TRUE	FALSE	Generic Loadable	TRUE
Yes	WTChangeIssue	Yes		wt.change2.WTChangeIssue	TRUE	FALSE	Generic Loadable	TRUE
Yes	WTChangeOrder2	Yes		wt.change2.WTChangeOrder2	TRUE	FALSE	Generic Loadable	TRUE
Yes	WTChangeRequest2	Yes		wt.change2.WTChangeRequest2	TRUE	FALSE	Generic Loadable	TRUE
Yes	WTDocument	Yes		wt.doc.WTDocument	TRUE	FALSE	Generic Loadable	TRUE
Yes	WTPart	Yes		wt.part.WTPart	TRUE	FALSE	Generic Loadable	FALSE
Yes	WTPartRequest	Yes		com.ptc.windchill.suma.npi.WTPartRequest	TRUE	FALSE	Generic Loadable	TRUE

Versioned Object Types

The following table has information about the out-of-the-box extract and load object support provided with the Windchill Bulk Migrator which is identified by the versioned business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Content Holder	Binary Link	Category	IBA Holder	Security Labelled
Yes	WTVariance	Yes	wt.change2.WTVariance	TRUE	FALSE	Generic Loadable	TRUE	TRUE
Yes	WTWorkSet	Yes	wt.change2.workset.WTWorkSet	FALSE	FALSE	Generic Loadable	TRUE	TRUE

-
1. The migration of EPMDocument object types is supported for Windchill-to-Windchill migrations and also for Creo and CATIA V5 data from file system directories. The loading of other CAD types from file system directories are not supported with this release of the Windchill Bulk Migrator.
 2. Extraction of EPMParameterUnitInfo is only supported for Windchill 10.0 M030 and up.

Binary Link Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the binary link business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Versioned	Category	IBA Holder	Security Labelled
Yes	AddressedBy2	Yes	wt.change2.AddressedBy2	FALSE	Generic Loadable	TRUE	FALSE
Yes	AffectedActivityData	Yes	wt.change2.AffectedActivityData	FALSE	Generic Loadable	TRUE	TRUE
Yes	AgreementPrincipalLink	Yes	wt.access.agreement. AgreementPrincipalLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	AlternateItemGroup	Yes	com.ptc.windchill.esi.bom. AlternateItemGroup	FALSE	Generic Loadable	TRUE	FALSE
Yes	AssociativeToSC-CLink	Yes	com.ptc.windchill.mpmi.pmi. AssociativeToSCCLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	AXLRuleToEntry-Link	Yes	com.ptc.windchill.suma. axlrule.AXLRuleToEntryLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	BaselineMember	Yes	wt.vc.baseline. BaselineMember	FALSE	Generic Loadable	TRUE	TRUE
Yes	ChangeActionCandidates	Yes	wt.changeAction. ChangeActionCandidates	FALSE	Generic Loadable	TRUE	FALSE
Yes	ChangeProcessLink	Yes	wt.change2. ChangeProcessLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	ChangeRecord2	Yes	wt.change2.ChangeRecord2	FALSE	Generic Loadable	TRUE	FALSE
Yes	ChangeReference-Link	Yes	wt.change2. ChangeReferenceLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	ChoiceMappable-ChoiceLink	Yes	com.ptc.windchill.option. model. ChoiceMappableChoiceLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	ConfigurableDescri-bleLink	Yes	wt.configurablelink. ConfigurableDescribeLink	FALSE	Generic Loadable	FALSE	FALSE
Yes		Yes	wt.configurablelink.	FALSE	Generic Loadable	FALSE	FALSE

Binary Link Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the binary link business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Versioned	Category	IBA Holder	Security Labelled
	ConfigurableMastersLink		ConfigurableMastersLink				
Yes	ConfigurableReferenceLink	Yes	wt.configurablelink.ConfigurableReferenceLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	ConsumptionLink	Yes	wt.associativity.ConsumptionLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	EnterpriseDataLink	Yes	com.ptc.windchill.enterprise.data.EnterpriseDataLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	EnterpriseDataMasterLink	Yes	com.ptc.windchill.enterprise.data.EnterpriseDataMasterLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	EPMBuildHistory	Yes	wt.epm.build.EPMBuildHistory	FALSE	Generic Loadable	FALSE	FALSE
Yes	EPMBuildRule	Yes	wt.epm.build.EPMBuildRule	FALSE	Generic Loadable	FALSE	FALSE
Yes	EPMBuildRuleAssociationLink	Yes	wt.epm.build.EPMBuildRuleAssociationLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	EPMDescribeLink	Yes	wt.epm.structure.EPMDescribeLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	EPMMemberLink	Yes	wt.epm.structure.EPMMemberLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	EPMReferenceLink	Yes	wt.epm.structure.EPMReferenceLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	EquivalenceLink	Yes	wt.associativity.EquivalenceLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	ESIReleaseObjectLink	Yes	com.ptc.windchill.esi.txn.ESIReleaseObjectLink	FALSE	Generic Loadable	TRUE	FALSE

Binary Link Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the binary link business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Versioned	Category	IBA Holder	Security Labelled
Yes	ESIReleaseTransactionLink	Yes	com.ptc.windchill.esi.txn.ESIReleaseTransactionLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	ESTargetAssignmentLink	Yes	com.ptc.windchill.esi.tgt.ESITargetAssignmentLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	ESTransactionObjectLink	Yes	com.ptc.windchill.esi.txn.ESITransactionObjectLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	ESTransactionRelease	Yes	com.ptc.windchill.esi.txn.ESITransactionRelease	FALSE	Generic Loadable	TRUE	FALSE
Yes	ESTransactionTargetLink	Yes	com.ptc.windchill.esi.txn.ESITransactionTargetLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	ExpressionMemberLink	Yes	wt.option.ExpressionMemberLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	FormalizedBy	Yes	wt.change2.FormalizedBy	FALSE	Generic Loadable	TRUE	FALSE
Yes	GenericizableTemplateLink	Yes	wt.generic.GenericizableTemplateLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	GenericVariantLink	Yes	com.ptc.wpcfg.variants.GenericVariantLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	HangingChangeLink	Yes	wt.change2.HangingChangeLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	Impacts	Yes	wt.change2.Impacts	FALSE	Generic Loadable	TRUE	FALSE
Yes	IncludedIn2	Yes	wt.change2.IncludedIn2	FALSE	Generic Loadable	TRUE	FALSE
Yes	IssueImplementedBy	Yes	wt.change2.IssueImplementedBy	FALSE	Generic Loadable	TRUE	FALSE
Yes	MadeFromLink	Yes	wt.enterprise.MadeFromLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	ModelItemContainedIn	Yes	wt.epm.model.items.ModelItemContainedIn	FALSE	Generic Loadable	TRUE	FALSE

Binary Link Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the binary link business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Versioned	Category	IBA Holder	Security Labelled
Yes	ModelItemLink	Yes	wt.epm.modelitems. ModelItemLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	ModelItemPartLink	Yes	wt.epm.modelitems.link. ModelItemPartLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	MPMCompatibility-Link	Yes	com.ptc.windchill.mpmml. MPMCompatibilityLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	MPMDescribeMfg-ProcessLink	Yes	com.ptc.windchill.mpmml. mfgprocess. MPMDescribeMfgProces- sLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	MPMDocumentDe- scribeLink	Yes	com.ptc.windchill.mpmml. MPMDocumentDescribeLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	MPMDocumentRe- ferenceLink	Yes	com.ptc.windchill.mpmml. MPMDocumentReference- Link	FALSE	Generic Loadable	TRUE	FALSE
Yes	MPMEPMDDocu- mentDescribeLink	Yes	com.ptc.windchill.mpmml. MPMEPMDDocumentDescri- beLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMMfgStandar- dUsageLink	Yes	com.ptc.windchill.mpmml. mfgprocess. MPMMfgStandardUsageLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	MPMOperationPro- cessLink	Yes	com.ptc.windchill.mpmml. processplan.operation. MPMOperationProcessLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMOperation- QualityLink	Yes	com.ptc.windchill.mpmml. MPMOperationQualityLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMOperationTo- ConsumableLink	Yes	com.ptc.windchill.mpmml. processplan.operation.	FALSE	Generic Loadable	FALSE	FALSE

Binary Link Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the binary link business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Versioned	Category	IBA Holder	Security Labelled
			MPMOperationToConsumableLink				
Yes	MPMOperation-ToOperatedPartLink	Yes	com.ptc.windchill.mpmml.processplan.operation.MPMOperationToOperatedPartLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMOperationToPartLink	Yes	com.ptc.windchill.mpmml.processplan.operation.MPMOperationToPartLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMOperationToWorkCenterLink	Yes	com.ptc.windchill.mpmml.processplan.operation.MPMOperationToWorkCenterLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMOperationUsageLink	Yes	com.ptc.windchill.mpmml.processplan.operation.MPMOperationUsageLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	MPMPartPathOccurrenceLink	Yes	com.ptc.windchill.mpmml.processplan.operation.MPMPartPathOccurrenceLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMPartQualityLink	Yes	com.ptc.windchill.mpmml.pmi.MPMPartQualityLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	MPMPartToPlanAssignmentLink	Yes	com.ptc.windchill.mpmml.MPMPartToPlanAssignmentLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	MPMPartToProcessPlanLink	Yes	com.ptc.windchill.mpmml.processplan.MPMPartToProcessPlanLink	FALSE	Generic Loadable	TRUE	FALSE

Binary Link Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the binary link business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Versioned	Category	IBA Holder	Security Labelled
Yes	MPMPlantAssignmentLink	Yes	com.ptc.windchill.mpmi.MPMPlantAssignmentLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	MPMPlantLocalizationLink	Yes	com.ptc.windchill.mpmi.MPMPlantLocalizationLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMProcessPlanLocalizationLink	Yes	com.ptc.windchill.mpmi.MPMProcessPlanLocalizationLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	MPMProcessQualityLink	Yes	com.ptc.windchill.mpmi.pmi.MPMProcessQualityLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	MPMQualityLink	Yes	com.ptc.windchill.mpmi.pmi.MPMQualityLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMSequenceUsageLink	Yes	com.ptc.windchill.mpmi.sequence.MPMSequenceUsageLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	MPMStandardCC>ToProcessPlanLink	Yes	com.ptc.windchill.mpmi.pmi.MPMStandardCC>ToProcessPlanLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMStandardCC>ToResourceLink	Yes	com.ptc.windchill.mpmi.pmi.MPMStandardCC>ToResourceLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMOperationToStandardCCLink	Yes	com.ptc.windchill.mpmi.pmi.MPMOperationToStandardCCLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMStandardCCUsageToDDLLink	Yes	com.ptc.windchill.mpmi.MPMStandardCCUsageToDDLLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMSdtCCUUsageToProcessPlanLink	Yes	com.ptc.windchill.mpmi.pmi.MPMSdtCCUUsageToProcessPlanLink	FALSE	Generic Loadable	FALSE	FALSE

Binary Link Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the binary link business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Versioned	Category	IBA Holder	Security Labelled
			MPMStdCCUsageToProcess-PlanLink				
Yes	MPMStdCCUsage-ToResourceLink	Yes	com.ptc.windchill.mpmpl.pmi.MPMStdCCUsageToResourceLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMStandardProcedureLink	Yes	com.ptc.windchill.mpmpl.MPMStandardProcedureLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	MPMWTPart-ToEPMDocumentLink	Yes	com.ptc.windchill.mpmpl.pmi.MPMWTPartToEPMDocumentLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	NoteHolderNote-Link	Yes	wt.note.NoteHolderNoteLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	ObjectHistory	Yes	wt.lifecycle.ObjectHistory	FALSE	Generic Loadable	FALSE	FALSE
Yes	OptionChoiceMasterLink	Yes	com.ptc.windchill.option.model.OptionChoiceMasterLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	OptionSetChoiceRuleLink	Yes	com.ptc.windchill.option.model.OptionSetChoiceRuleLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	OptionSetChoiceRuleMasterLink	Yes	com.ptc.windchill.option.model.OptionSetChoiceRuleMasterLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	OptionSetIgnoreChoiceRuleLink	Yes	com.ptc.windchill.option.model.OptionSetIgnoreChoiceRuleLink	FALSE	Generic Loadable	FALSE	FALSE
Yes		Yes	com.ptc.windchill.option.	FALSE	Generic Loadable	FALSE	FALSE

Binary Link Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the binary link business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Versioned	Category	IBA Holder	Security Labelled
	OptionSetMemberLink		model.OptionSetMemberLink				
Yes	OEMPartAXLContextLink	Yes	com.ptc.windchill.sumaxl.OEMPartAXLContextLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	OverrideRegisteredOptionSetLink	Yes	com.ptc.windchill.option.model.OverrideRegisteredOptionSetLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	PartListItem	Yes	com.ptc.arbor текст.windchill.partlist.PartListitem	FALSE	Generic Loadable	TRUE	FALSE
Yes	PartListItemSubstituteLink	Yes	com.ptc.arbor текст.windchill.partlist.PartListitemSubstituteLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	PartListToEPMDocumentLink	Yes	com.ptc.arbor текст.windchill.partlist.PartListToEPMDocumentLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	PartListToPartLink	Yes	com.ptc.arbor текст.windchill.partlist.PartListToPartLink	FALSE	Generic Loadable	TRUE	TRUE
Yes	PartRequestRelatedPartLink	Yes	com.ptc.windchill.sumanpi.PartRequestRelatedPartLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	PartToPartListLink	Yes	com.ptc.arbor текст.windchill.partlist.PartToPartListLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	PlantTagLink	Yes	com.ptc.windchill.mpmml.resource.PlantTagLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	PlantFunctionalDataLink	Yes	com.ptc.windchill.enterprise.data.PlantFunctionalIDataLink	FALSE	Generic Loadable	FALSE	FALSE

Binary Link Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the binary link business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Versioned	Category	IBA Holder	Security Labelled
Yes	ProblemProduct	Yes	wt.change2.ProblemProduct	FALSE	Generic Loadable	FALSE	FALSE
Yes	REGISTEREDOPTIONSETLINK	No	com.ptc.windchill.option.model.RegisteredOptionSetLink	FALSE	Legacy Loadable	FALSE	FALSE
Yes	ReleaseActivity	Yes	com.ptc.windchill.esi.txn.ReleaseActivity	FALSE	Generic Loadable	FALSE	FALSE
Yes	RelevantRequestData2	Yes	wt.change2.RelevantRequestData2	FALSE	Generic Loadable	FALSE	FALSE
Yes	ReportedAgainst	Yes	wt.change2.ReportedAgainst	FALSE	Generic Loadable	FALSE	FALSE
Yes	RuleMemberActionLink	Yes	com.ptc.windchill.option.model.RuleMemberActionLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	SubjectProduct	Yes	wt.change2.SubjectProduct	FALSE	Generic Loadable	TRUE	FALSE
Yes	SupersedeLink	Yes	wt.supersede.SupersedeLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	SupplementaryReplacementLink	Yes	com.ptc.arbortext.windchill.partlist.SupplementaryReplacementLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	SupplierDocumentLink	Yes	com.ptc.windchill.suma.supplier.SupplierDocumentLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	TranslationLink	Yes	com.ptc.arbortext.windchill.translation.TranslationLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	VariantSpecDeliverableLink	Yes	com.ptc.wpcfg.deliverables.VariantSpecDeliverableLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	VariantSpecVariantLink	Yes	com.ptc.wpcfg.deliverables.	FALSE	Generic Loadable	FALSE	FALSE

Binary Link Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the binary link business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Versioned	Category	IBA Holder	Security Labelled
	WTIDocumentDependencyLink	Yes	VariantSpecVariantLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	WTIDocumentUsageLink	Yes	wt.doc. WTDocumentDependency-Link	FALSE	Generic Loadable	FALSE	FALSE
Yes	WTIDocumentUsageLink	Yes	wt.doc. WTDocumentUsageLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	WTPPartAlternateLink	Yes	wt.part.WTPPartAlternateLink	FALSE	Generic Loadable	TRUE	TRUE
Yes	WTPPartDescribeLink	Yes	wt.part.WTPPartDescribeLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	WTPPartReferenceLink	Yes	wt.part.WTPPartReferenceLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	WTPPartSubstituteLink	Yes	wt.part.WTPPartSubstituteLink	FALSE	Generic Loadable	TRUE	TRUE
Yes	WTPPartUsageLink	Yes	wt.part.WTPPartUsageLink	FALSE	Generic Loadable	TRUE	FALSE
Yes	WTWorkSetComponentLink	Yes	wt.change2.workset. WTWorkSetComponentLink	FALSE	Generic Loadable	FALSE	FALSE
Yes	WTWorkSetReferenceLink	Yes	wt.change2.workset. WTWorkSetReferenceLink	FALSE	Generic Loadable	FALSE	FALSE

Non-Versioned Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the non-versioned business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Content Holder	Binary Link	Category	IBA Holder	Security Labelled
Yes	AgreementAuthorizedObject	Yes	wt.access.agreement. AgreementAuthorizedObject	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	AgreementAuthorizedProxy	Yes	wt.access.agreement. AgreementAuthorizedProxy	FALSE	FALSE	Generic Loadable	FALSE	TRUE
Yes	AMLPolicyRule	Yes	com.ptc.windchill.suma. axrule.AMLPolicyRule	FALSE	FALSE	Generic Loadable	FALSE	TRUE
Yes	AMLPolicyRuleSet	Yes	com.ptc.windchill.suma. axrule.AMLPolicyRuleSet	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	ATONavigationFilter	Yes	com.ptc.windchill.option. model. ATONavigationFilter	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	AVLPolicyRule	Yes	com.ptc.windchill.suma. axrule.AVLPolicyRule	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	AVLPolicyRuleSet	Yes	com.ptc.windchill.suma. axrule.AVLPolicyRuleSet	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	AXLChangeEvent	Yes	com.ptc.windchill.suma. axl.AXLChangeEvent	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	AXLEntry	Yes	com.ptc.windchill.suma. axl.AXLEntry	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	ChangeAction	Yes	wt.change2.ChangeAction	FALSE	FALSE	Generic Loadable	TRUE	FALSE
Yes	ConfigurationItem	Yes	wt.effectivity. ConfigurationItem	FALSE	FALSE	Generic Loadable	FALSE	TRUE
Yes	CONTENTMETADATA	No		FALSE	FALSE	Legacy Loadable	FALSE	TRUE

Non-Versioned Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the non-versioned business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Content Holder	Binary Link	Category	IBA Holder	Security Labelled
Yes	DERIVEDIMAGE	No		FALSE	FALSE	Legacy Loadable	FALSE	TRUE
Yes	EffChangeAudit	Yes	wt.eff.EffChangeAudit	FALSE	FALSE	Generic Loadable	FALSE	TRUE
Yes	EPMASSTORED	Yes	wt.epm.workspaces. EPMASStoredConfig wt. epm.workspaces. EPMASStoredMember	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	EPMDOCUMENTUNITINFO	Yes	wt.epm.attributes. EPMParameterUnitInfo	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	EPMFamilyTable	Yes	wt.epm.familytable. EPMSepFamilyTable wt. epm.structure. EPMContainedIn wt.epm. structure.EPMVariantLink	TRUE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	EPMTProp	Yes	wt.epm.familytable. EPMParameterDefinition wt.epm.familytable. EPMFamilyTableParameter wt.epm.familytable. EPMFeatureDefinition wt. epm.familytable. EPMFamilyTableFeature wt.epm.familytable. EPMFamilyTableMember wt.epm.familytable. EPMFamilyTableReference wt.epm.familytable. EPMFamilyTableAttribute	FALSE	FALSE	Legacy Loadable	FALSE	FALSE

Non-Versioned Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the non-versioned business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Content Holder	Binary Link	Category	IBA Holder	Security Labelled
			wt.epm.familytable. EPMFamilyTableCell wt. epm.familytable. EPMFamilyTableCellDependency wt.epm. familytable. EPMParameterValue wt. epm.familytable. EPMFeatureValue					
Yes	EPMIDGroup	Yes	wt.epm.structure. EPMIDGroup	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	EPMIDSpaceRecord	Yes	wt.epm.structure. EPMIDSpaceRecord	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	EPMMEMLINKUNI-TINFO	Yes	wt.epm.attributes. EPMParameterUnitInfo	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	EPMPARAMMAP-FORDOC	Yes	wt.epm.attributes. EPMParameterMap	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	EPMPARAMMAP-FORML	Yes	wt.epm.attributes. EPMParameterMap	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	EPMPARAMMAP-FORRL	Yes	wt.epm.attributes. EPMParameterMap	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	EPMREFLINKUNI-TINFO	Yes	wt.epm.attributes. EPMParameterUnitInfo	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	ESIRelease	Yes	com.ptc.windchill.esi.txn. ESIRelease	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	ESTransaction	Yes	com.ptc.windchill.esi.txn. ESTransaction	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes		Yes	com.ptc.windchill.esi.txn.	FALSE	FALSE	Generic	FALSE	FALSE

Non-Versioned Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the non-versioned business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Content Holder	Binary Link	Category	IBA Holder	Security Labelled
	ESITransactionMessage		ESITransactionMessage			Loadable		
Yes	Folder	No		FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	Group	Yes	com.ptc.windchill.option.model.Group	FALSE	TRUE	Generic Loadable	FALSE	FALSE
Yes	LibraryContainer	No		FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	LifeCycleHistory	Yes	wt.lifecycle.LifeCycleHistory	FALSE	FALSE	Generic Loadable	TRUE	FALSE
Yes	NAVIGATIONCRIPTERIA	No	wt.filter.NavigationCriteria	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	ManagedBaseline	Yes	wt.vc.baseline.ManagedBaseline	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	ModelItem	Yes	wt.epm.modelitems.ModelItem	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	ModelItemMaster	Yes	wt.epm.modelitems.ModelItemMaster	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	OrgContainer	Yes		FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	PartUsesOccurrence	Yes	wt.part.PartUsesOccurrence	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	PDMLinkProduct	No		FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	PendingBlockEffectivity	Yes	wt.eff.PendingBlockEffectivity	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes			wt.eff.	FALSE	FALSE	Generic	FALSE	FALSE

Non-Versioned Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the non-versioned business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Content Holder	Binary Link	Category	IBA Holder	Security Labelled
	PendingDateEffectivity		PendingDateEffectivity			Loadable		
Yes	PendingLotNumberEffectivity	Yes	wt.eff. PendingLotNumberEffectivity	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	PendingMSNEffectivity	Yes	wt.eff. PendingMSNEffectivity	FALSE	FALSE	Generic Loadable	TRUE	FALSE
Yes	PendingSerialNumberEffectivity	Yes	wt.eff. PendingSerialNumberEffectivity	FALSE	FALSE	Generic Loadable	TRUE	TRUE
Yes	ProductBlockEffectivity	Yes	wt.part. ProductBlockEffectivity	FALSE	FALSE	Generic Loadable	FALSE	FALSE
Yes	ProductLotNumberEffectivity	Yes	wt.part. ProductLotNumberEffectivity	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	ProductMSNEffectivity	Yes	wt.part. ProductMSNEffectivity	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	ProductSerialNumberEffectivity	Yes	wt.part. ProductSerialNumberEffectivity	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	REGISTEREDOPTIONSETLINK	No	com.ptc.windchill.option. model. RegisteredOptionSetLink	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	ReleaseActivityMessage	Yes	com.ptc.windchill.est.txn. ReleaseActivityMessage	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	SHAREDCONTAINERMAP	Yes		FALSE	FALSE	Legacy Loadable	FALSE	FALSE

Non-Versioned Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the non-versioned business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Content Holder	Binary Link	Category	IBA Holder	Security Labelled
Yes	SupersedeGroup	Yes	wt.supersede. SupersedeGroup	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	SupplierChangeEvent	Yes	com.ptc.windchill.suma. supplier. SupplierChangeEvent	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	TranslationDictionaryEntry	Yes	com.ptc.core.td.server. dictionary. TranslationDictionaryEntry	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	VarianceBlockEffectivity	Yes	wt.change2. VarianceBlockEffectivity	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	VarianceDateEffectivity	Yes	wt.change2. VarianceDateEffectivity	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	VarianceLotNumberEffectivity	Yes	wt.change2. VarianceLotNumberEffectivity	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	VarianceMSNEffectivity	Yes	wt.change2. VarianceMSNEffectivity	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	VarianceSerialNumberEffectivity	Yes	wt.change2. VarianceSerialNumberEffectivity	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	View	Yes	wt.vc.views.View	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	WFPROCESS	No		FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	WTChangeDirective	Yes	wt.change2. WTChangeDirective	FALSE	FALSE	Legacy Loadable	FALSE	FALSE

Non-Versioned Object Types

The following table has information about the out-of-the-box loaders provided with the Windchill Bulk Migrator for administrative data which is identified by the non-versioned business class.

Extraction Supported	Loader Alias	Normalized	Business Class	Content Holder	Binary Link	Category	IBA Holder	Security Labelled
Yes	WTIDatedEffectivity	Yes	wtEffectivity WTDatedEffectivity	FALSE	FALSE	Legacy Loadable	FALSE	FALSE
Yes	WTOrganization	No		FALSE	FALSE	Legacy Loadable	FALSE	FALSE

 **Note**

The extraction of the Windchill business classes tagged after "CLASS:" in the <Windchill>/codebase/com/ptc/windchill/migration/wbm/wc2wc/extractor/classMap.csv file is supported out-of-box.
