



HOME

User Manual

DELMIA Process Engineer[®]

IPD Server



Foreword

This manual provides an introduction to the basic operations and functions of the IPD Server.

While developing these functions we have made every effort to create a clearly organized, easy-to-understand program structure.

A user-friendly interface as well as a clear menu guide will enable you to quickly learn how to operate the program and to get familiar with its functions so that you can carry out your planning tasks in a quick and reliable way.

No Liability or Guarantee

Our programs and manuals have been compiled with great care and to the best of our knowledge. They have also been tested in a production setting. However, we assume no liability and provide no guarantee that the software and related descriptions are free of error or are suitable for special purposes.

DELMIA assumes no liability for any damage that may arise from the use of this software. By using this software, the user acknowledges this exclusion from liability and shall hold DELMIA exempt from all claims.

Copyright

The information in our documents may be copied and distributed for internal purposes provided it is done free of charge and the contents are not altered or distorted.

Any other form of usage, especially the sale on CD-ROM or in any other publication in whole or in part is only permitted after prior written consent by DELMIA.

Some parts of this software are owned by Unigraphics Solutions Inc. and are copyrighted © 2010. All rights reserved.

Some parts of this software are owned by combit® GmbH and are copyrighted. Report-/Print module List and Label® Version 8.0: Copyright combit® GmbH 1991-2010.

Modifications

Moreover, DELMIA retains the right to make modifications and improvements to the product described in this manual at any time without prior notification.

DELMIA and the 3DS logo are registered trademarks of Dassault Systèmes or its subsidiaries, in the United States or other countries.

© 2001-2010 Dassault Systèmes - All rights reserved

Table of Contents

1. Introduction	1
1.1 How to Use this Manual	1
1.2 Documentation Conventions and Symbols	2
2. Customization	3
2.1 Customizing Plantypes	3
2.2 Customizing Parent Child Relations	5
2.2.1 Examples	5
List of Figures	10
Index	11

1.Introduction

This manual explains how to use the Process Engineer IPD- Server for your planning purposes.

It shows how to install and set up the IPD-Server so you can use it when working with DELMIA software.

Change Management Controlled (CMC) deals with constraints regarding data consistency. The functionality is necessary to mark objects being under the CMC control and to ensure, that the data consistency according to the Change Management is never broken.

The versions can represent the evolution of an object through successive stages (revisions) or parallel alternatives (variants). The versioning services capture the history of modifications by tracking different versions as affected objects to actions managed by ENOVIA. The configuration services evaluate version selection rules over modification history, in order to control the visibility of the versioned objects.



Note

The Change Management Control options are enabled only when the Manufacturing Hub project is Change Management (CMC) controlled. Otherwise, these options are shown in a disabled state.

The availability of the Change Management Control options functionality is controlled by a separate license. You must have this license installed to open a project which is Change Management (CMC) controlled.

The configuration services allow to:

- Label projects to be CMC versioned
- Work with CMC in detached mode
- Perform CMC related consistency checks
- Perform filtering necessary for displaying of versioned objects

1.1 How to Use this Manual

This manual enables you to get familiar with the operation and functions of the Process Engineer. This manual briefly describes:

- IPD- Server Functions



Note

When handling the IPD- Server functions, please also refer to the general introduction to Process Engineer in the General Introduction Manual.



Click [General Introduction](#) to access the manual.

1.2 Documentation Conventions and Symbols

The symbols used in this manual are intended to provide you with keys to the contents in an immediately understandable manner.



This symbol is used to introduce key concepts that are covered in the sections immediately following this symbol. As a result, this symbol most frequently appears at the beginning of chapters or sections.



Note

*This symbol is used to mark notes, which provide you with additional information you need to have for further work. You will either find the Note sign at the beginning of a chapter or in a particular text passage in the chapter. Texts bearing this sign are additionally marked with **Note**. The text is always in italics.*



Caution

*This symbol indicates that the text that follows describes particular circumstances that you must avoid to avoid potential errors with the operation of the program or harm to data. You will either find the Caution sign at the beginning of a chapter or near a particular text passage in the chapter. Texts that are introduced by this sign are additionally marked with **Caution**. The text is always in italics.*

Example

This symbol marks examples which serve to illustrate a certain situation.



This symbol marks the individual operational steps involved in a particular operating instruction. Operating instructions describe operational steps, for example, how to open a menu or execute a function.



This symbol marks listed subjects. The symbol for listed subjects can be either used to structure a continuous text or to list main subject keywords.



This symbol marks list inside a bulleted or numbered list.



This symbol marks cross reference information that is available in another manual.

1.3 New Functions in IPD Server

No new functionality has been added for this release.

2. Customization

This chapter provides you with information necessary to customize the Manufacturing Hub to properly support the Change Management functionality.

The entire customization is done within a plantype set. Furthermore, the paragraph will include additional information about how Default-PRO-DPM-IGRIP_V5_CMC has been prepared.

2.1 Customizing Plantypes

This paragraph provides you with information necessary to customize the Manufacturing Hub to properly support the Change Management functionality. The entire customization is done within a plantype set. Furthermore, the paragraph will include additional information about how Default-PRO-DPM-IGRIP_V5_CMC has been prepared.

For each plantype that is in scope of Change Management the property '**Has versions**' has to be set to '**Yes**'. Only versionable plantypes are tracked by Change Management; in order to modify them, an action and a modification statement are required. Non-versionable plantypes are not in scope of Change Management, and for them the property '**Has versions**' has to be explicitly set to 'No'. It is especially important for product-plantypes not to be versionable in order to allow their import through the Bridge.



Note

For plantypes in R15 this property is usually set to 'Yes', therefore the focus should also be on those plantypes, which are not in scope of Change Management, especially products.

In Default-PRO-DPM-IGRIP_V5_CMC only processes are versionable.

The property '**Is Configured**' describes the kind of effectivity the object obtains.

By Category Alphabetical	
<input type="checkbox"/> Basics	
Base class type name	Process Component (ergocompprocessdefault)
Caption singular	Operation
Caption plural	
Plantype UUID	2793bcb4-ac27-4d19-9d95-23b00f5f314f
Description	
<input type="checkbox"/> Display	
Bitmap	process
Attributes per row	1
Dialog Type	Property Sheet
Special attribute	
Depends on values	No
<input type="checkbox"/> Flags	
Has versions	Yes
Is searchable	No
Is abstract	No
Has attachments	No
Is relevant	No
Own rights	Inherited from parent type
Display parent/child relations in dbeditor	Inherited from parent type
Is configured	Max. Configured
Defined by	Customer

Figure 1: Configuration

We distinguish:

- No (Not configured)
- Yes (Standard configuration mode)
- Max. Configured (maximum configuration mode)
- Notes to Configurable



Note

In Default-PRO-DPM-IGRIP_V5_CMC processes are Max. Configured and the plantype Manufacturing Assembly (non-versionable) applies the standard configuration.

- If plantypes are not configurable, no effectivity is applied on them. In case of standard configuration we distinguish whether plantypes are versionable or not.
- For versionable plantypes, the newly created objects or checked-out versions obtain the effectivity from the modification statement.
- While checking-in a version, the effectivity of any version with a planning state equal to that is cut with the effectivity from the selected modification statement. There is a special handling for non-versionable plantypes that are configurable. This requires a modification statement to be selected when these objects are created. Their effectivity is then again obtained from the selected modification statement.
- Maximum configuration means plantypes hold either “always” or “never valid” effectivity.

- When these objects are created or checked-out no effectivity is applied on them; when they are checked-in, the effectivity of any predecessor version, which has a planning state equal to that, is set to '*never valid*'. The maximum configured plantypes don't support parallel versioning.

2.2 Customizing Parent Child Relations

Only the parent child relations that have an owner, which is a versionable plantype, are in scope of Change Management.

Before the parent child relation is customized for Change Management, it should be first overwritten in the plantype set. This has to be done for each plantype which participates in the relation. It also has to be done for the so called *reverse* relation.

2.2.1 Examples

An example on how to customize the parent child relation "process_implementation_requirements" is described here.

The relation is customized such that the operation is the owner.

First step is to overwrite the relation, which is described in the figures below. For this purpose, the 'overwrite' on the source plantype has to be selected in the context menu, and then the target plantype is selected.

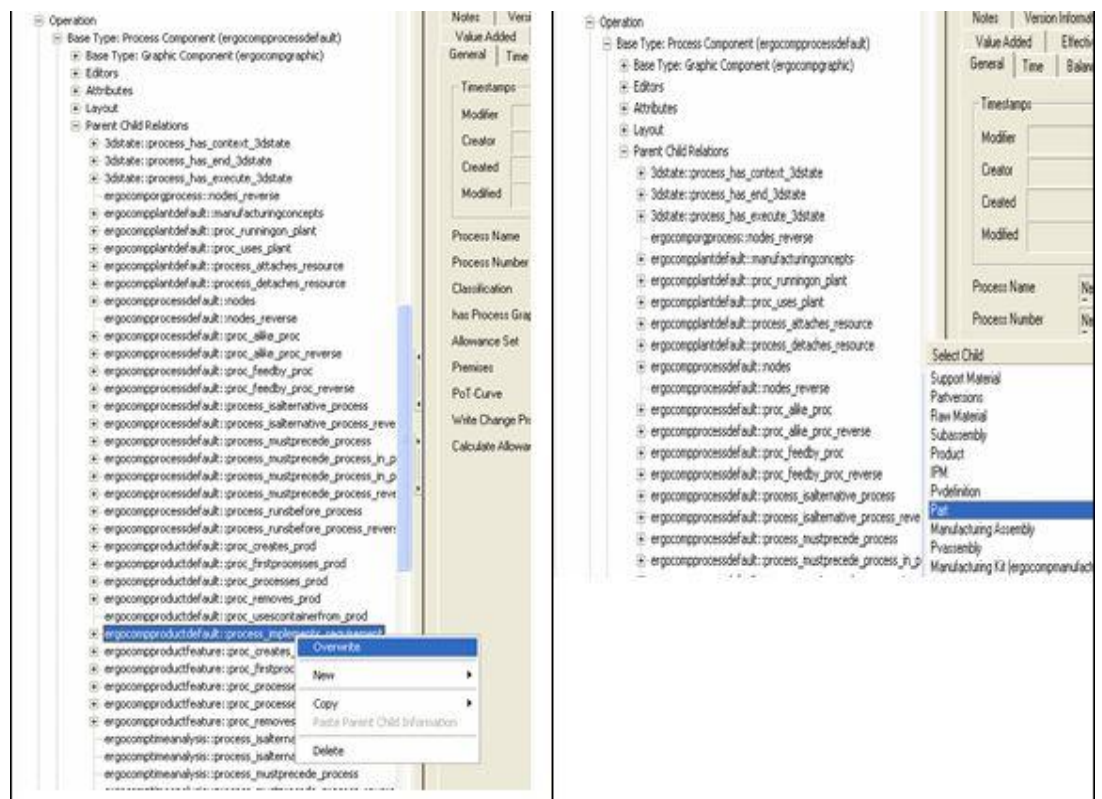


Figure 2: Example - process_implementation_requirements

We now have to customize the relation for Change Management. Properties marked red are important. The first three of them turn off some functionality that conflict with Change Management. We have to set them always to these values; otherwise Change Management won't work properly.

- The property '**Copy link to child (Change Management Controlled)**' has to be set to 'Yes' for all relations that are copied when a new version is created.
- For relations that are in scope of Change Management, 'Yes' is mandatory.

The property '**Owner type**' specifies how the owner plantype is related to the relation.

We distinguish among:

- No Owner
 - Source Owner
 - Target Owner
 - Explicit Owner
 - Common Parent Owner
- If the relation does not have an owner, it's not in scope of Change Management.
- 'Source owner' and 'Target owner' declare source or target plantype respectively to be owner of the relation. This information can be deduced at the runtime, and the owner doesn't have to be specified explicitly, if the relation is created using API.
- Explicit owner can be an object of any valid plantype. DPE is not able to deduce this information, so the reference to the explicit owner has to be passed to API, when the relation is created.
- 'Common Parent owner' refers to the direct common parent of source and target according to the relation *nodes* (BOM). DPE is not able to deduce this information, so the reference to the common parent owner has to be passed to API, when the relation is created.

Copy Link to Child

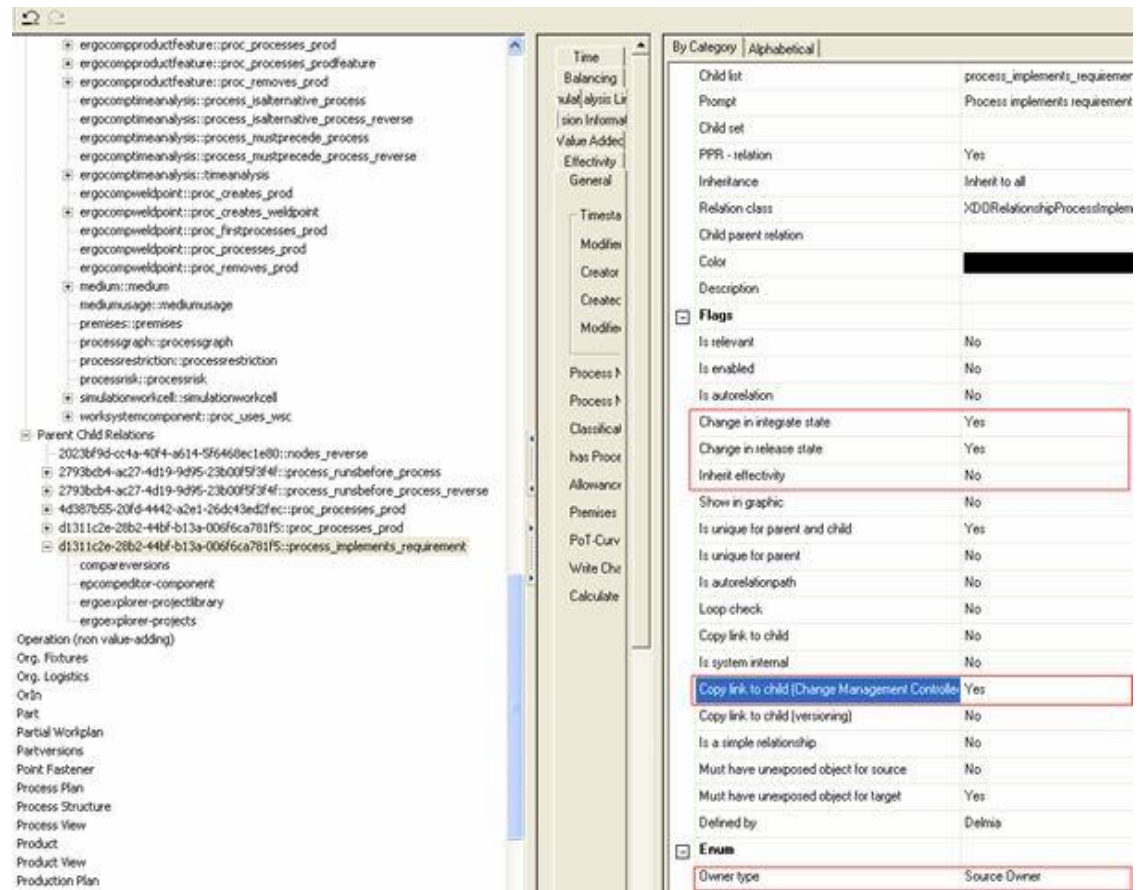


Figure 3: Example Copy Link to Child

We have to repeat this procedure for the reverse relation.

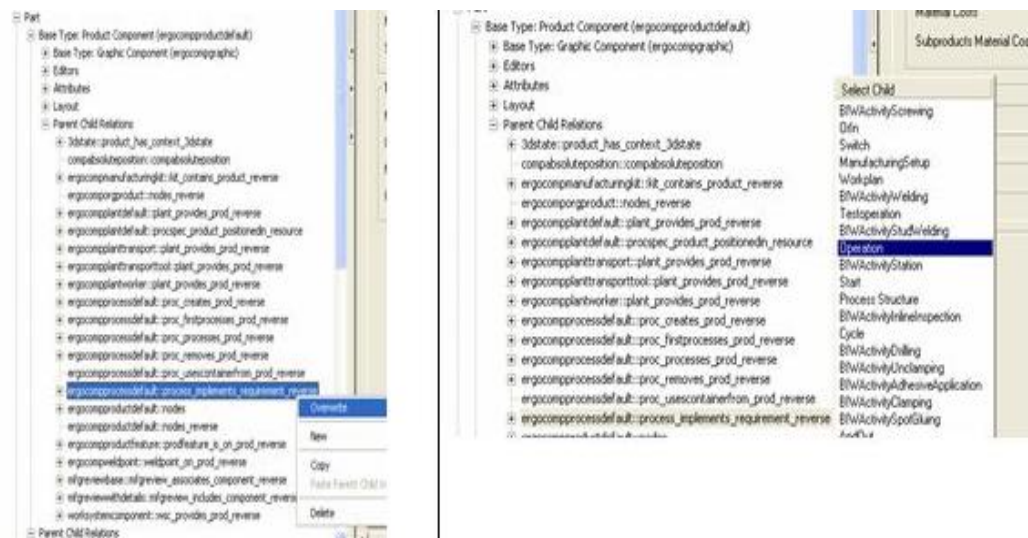


Figure 4: Example Reverse Relation

We use the same settings except for 'Owner type' in the case that the owner is the source or target plantype. If so, we have to use the opposite value, as the Figure 5 shows.

By Category | Alphabetical

Child list	process_implements_requirement_reverse
Prompt	Requirement is implemented by process
Child set	
PPR - relation	PPR Reverse Relation
Inheritance	Inherit to all
Relation class	XDRRelationshipProcessImplementsRequirement
Child parent relation	
Color	
Description	
Flags	
Is relevant	No
Is enabled	Yes
Is autorelation	No
Change in integrate state	Yes
Change in release state	Yes
Inherit effectivity	No
Show in graphic	No
Is unique for parent and child	Yes
Is unique for parent	No
Is autorelationpath	No
Loop check	No
Copy link to child	No
Is system internal	No
Copy link to child (Change Management Controlled)	Yes
Copy link to child (versioning)	No
Is a simple relationship	No
Must have unexposed object for source	No
Must have unexposed object for target	Yes
Defined by	Delta
Enum	
Owner type	Target Owner

Figure 5: Example Owner type

In Default-PRO-DPM-IGRIP_V5_CMC following relations are customized to be in scope of Change Management:

Parent Child Relation	Involved Plantypes	Owner
<i>nodes¹</i>	<i>Process View - Process Plan</i>	<i>No Owner</i>
<i>Nodes</i>	<i>Process Plan - Workplan</i>	<i>Target Owner</i>
<i>Nodes</i>	<i>Workplan - Operation</i>	<i>Target Owner</i>
<i>process_runsbefore_process</i>	<i>Operation - Operation</i>	<i>CommonParent Owner</i>
<i>proc_processes_prod</i>	<i>Operation - Part</i>	<i>Source Owner</i>
<i>proc_processes_prod</i>	<i>Operation - Subassembly</i>	<i>Source Owner</i>
<i>process_implements_requirement</i>	<i>Operation - Part</i>	<i>Source Owner</i>
<i>process_implements_requirement</i>	<i>Operation - Subassembly</i>	<i>Source Owner</i>
<i>process_implements_requirement</i>	<i>Workplan - Part</i>	<i>Source Owner</i>
<i>process_implements_requirement</i>	<i>Workplan - Subassembly</i>	<i>Source Owner</i>

Figure 6: Table of Change Management

Limitation 1

Mixed Mode (Interaction of objects created under Detached Mode and Non-Detached Mode) is not supported.

Limitation 2

Versioning of Ergoitems is not supported by CMC.

Limitation 3

Only non-versionable components (products and resources) can be transferred through the Bridge.

List of Figures

Figure 1:Configuration	4
Figure 2: Example - process_implements_requirements.....	5
Figure 3: Example Copy Link to Child	7
Figure 4: Example Reverse Relation.....	7
Figure 5: Example Owner type.....	8
Figure 6: Table of Change Management.....	8

Index

Nonliabilityii