



HOME

User Manual

DELMIA Process Engineer<sup>®</sup>

# Manufacturing Change Management (MCM)



# Foreword

This manual provides an introduction to the Process Engineer Manufacturing Change Management operations and functions.

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 © 2011. All rights reserved.

Some parts of this software are owned by combit® GmbH and are copyrighted. Report-/Print module List and Label® Version 15.0: Copyright combit® GmbH 1991-2011.

## 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.

This clause applies to all acquisitions of DASSAULT SYSTÈMES commercial computer software by or for the United States federal government, or by any prime contractor or subcontractor (at any tier) under any contract, grant, cooperative agreement, or other activity with the federal government. By accepting delivery of this software, the United States government hereby agrees that this software qualifies as “commercial” computer software within the meaning of the acquisition regulation(s) applicable to this procurement. The terms and conditions of the DASSAULT SYSTÈMES standard commercial end user license agreement shall pertain to the United States government's use and disclosure of this software, and shall supersede any conflicting contractual terms and conditions. If the DASSAULT SYSTÈMES standard commercial license

fails to meet the United States government's needs or is inconsistent in any respect with United States Federal law, the United States government agrees to return this software, unused, to DASSAULT SYSTÈMES. The following additional statement applies only to acquisitions governed by DFARS Subpart 227.4 (October 1988): "Restricted Rights – use, duplication, and disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252-227-7013 (Oct. 1988)."

© 2001-2011 Dassault Systèmes - All Rights Reserved

# Table of Contents

<b>1. Introduction</b>	<b>1</b>
1.1 How to Use this Manual	1
1.2 Documentation Conventions and Symbols	1
1.3 New Functions in MCM	2
<b>2. Rules for Planning MCM-Projects</b>	<b>3</b>
2.1 MCM-Projects Planning Criteria	3
2.2 Version Compatibility	5
2.3 Rules for Editing MCM-Projects	6
2.3.1 Exclude Attributes from MCM-Project	8
<b>3. Working with MCM-Project</b>	<b>9</b>
3.1 Creating MCM-Project	9
3.1.1 Mark Project as MCM-Project	9
3.1.2 Creating Actions	12
3.1.3 Creating Modification Statements	15
3.1.4 Displaying Selected Action and Modification Statement in Status Bar	16
3.1.5 ENOVIA Change Order	17
3.1.6 Display ENOVIA Change Orders Sorted by Name	18
3.2 Opening a MCM-Project	19
3.2.1 Historical View of the Planning State	24
3.2.2 Selecting Actions in the Open Project	24
3.2.3 Searching ModStatement	27
3.3 Calculating Absolute Effectivity, Position, and Bounding Boxes	28
3.4 Filtering MCM-Projects	31
3.4.1 Using Planning State as a Filter	31
3.4.2 Using the Product Structure as a Filter	34
3.4.3 Improving Performance in GetChildren() Server API with Implicit Filtering	39
<b>4. Versioning</b>	<b>41</b>
4.1 Versioning in PPR Components	41
4.1.1 Versioning for Maximum Configuration of the PPR-Components	42
4.1.2 Versioning for Standard Configuration of the PPR-Components	46
4.2 Working with Versions	48
4.2.1 Creating Versions with Help of Properties Dialog	48
4.2.2 Deleting Versions	50
4.2.3 Versioning the PPR-Components – Copying Relations	51
4.2.4 Use of Option on Demand	52
4.3 Using Relations for Versions	53

4.3.1	Creating and Editing Relations for Versions	54
4.3.2	Cloning Relations for Versions	58
4.3.3	Deleting Relations	62
4.3.4	Use of Options Owner and CCZ Member for PPR-Components	64
4.3.5	Delete Options for MCM Components	67
4.3.6	Creating NewChild on Old Version	68
	<b>List of Figures</b>	<b>71</b>
	<b>List of Tables</b>	<b>75</b>
	<b>Index</b>	<b>76</b>

# 1.Introduction

This manual explains how to use the Process Engineer Manufacturing Change Management (MCM) for your planning purposes.

## 1.1 How to Use this Manual

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

- How to start and exit the MCM process
- Menus and views required for MCM
- How to start and exit the program
- How to execute menu functions and how to navigate in views



### Note

*When handling the MCM 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.

1

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 MCM

### Creating NewChild on Old Version

You can add new child to a version having successor, without creating reverse relation to successor version, with the help of property **Disable auto creation of link to successor**.

### New CO-Part-Relations

The **Filtering by Change Order** is enhanced with five simple relations (and reverse relations) between Base Type “Base Change Order” and “ErgoComponent” (ergocompbase).

### Improving Performance in GetChildren() Server API with Implicit Filtering

The performance of GetChildren() server API is improved with Implicit Filtering. The query is optimized to get all the relations from configuration database having the property **ConsiderforActivationbyLink** as **Yes**.

### Displaying Selected Action and Modification Statement in Status Bar

You can display names of selected action and modification statement in the status bar. Also when there is a change in the action and mod statement during re-loading of project; then corresponding name gets updated in the status bar. This helps to see current action and modification statement names without any additional action in title bar of loaded project.

### Excluding Absolute Position Calculation

The Server methods for calculation of absolute effectivity, position, and bounding boxes can be run on the configured objects without selecting a specific action/mod statement even if some calculated objects are under MCM control.

## 2. Rules for Planning MCM-Projects

The availability of the MCM functionalities is controlled by a separate license. This license must be installed to open and edit projects related to the MCM.

You can control the actions of projects by using the MCM. The actions can be defined both in ENOVIA and in the DELMIA Process Engineer, DPE (with the corresponding modification statements).

The following fields must be activated when opening a project that is supervised by MCM (**Is CMC Mode**):

- Action
- Modification Statement
- Planning States of Versions Owned



### Note

*A MCM-project could contain PPR components subject to MCM as well as PPR components not subject to MCM. This is defined via the respective plantypes of the PPR components. PPR components subject to the MCM may be changed only if an action with a modification statement was selected when the MCM-project was opened. The actions available for opening a MCM-project are defined via the access rights of the action.*



For more information, please refer to the *MCM Installation and Preparation* in the [Administration Manual](#).

### 2.1 MCM-Projects Planning Criteria

This section helps you to get familiar with the MCM procedure. It helps to know how to plan MCM-projects with the help of **not** versionable PPR-components and temporary modification statement. The important steps are displayed which you have to consider in MCM planning. If you have planned MCM-projects you may know most of the functions i.e. **Delete Components** or **Relations**.

Up to Release PE 5.17 SP2 it was possible to plan a MCM-project with **versionable** PPR-components which were fixed assigned to a function and modification statement. Starting from this release the function range to plan MCM-projects is extended. With the help of **not versionable** PPR-components and temporary modification statements you can plan MCM-projects which are not fixed assigned to a concrete function.

The planning of MCM-projects must be considered under the following three criterions:

- Planning of MCM-projects with the help of **versionable** PPR-components which are fixed assigned to a function and modification statement.
- Planning of MCM-projects with the help of **not versionable** PPR-components which are fixed assigned to a function and modification statement.



- Planning of MCM-Projects with the help of **not versionable** PPR-components and temporary modification statement which are fixed assigned to a function.

### Planning MCM-Projects with Temporary Modification Statement

The temporary modification statements help in the modifications on PPR-components via extended effectivities. In order to plan MCM-projects with the temporary modification statements, make sure for the following requirements:

- In order to indicate extended effectivities for PPR-components make sure that for the type **ergoproject** the transient attribute **tempmodstatement** is created and displayed in **Properties** dialog of the project.
- The attribute **tempmodstatement** is displayed under the **Filter** tab with the notation **Temp MOD**.
- This attribute helps to indicate extended effectivities. After you have given this data in properties dialog all afterwards created PPR-components receive this effectivity.
- The value mentioned in this attribute is temporary, i.e. the mentioned value remains as long as you do not close the project.
- During a session the value can be changed arbitrarily. The new values are considered accordingly with creating new components or deleting of components.

General	Filter	Notes
Temp MOD		
		{1-100}
Coderule Mode		
		Standard Mode with operators { } - + /
Coderule Type		
		Relative
Category-Label Separator		
		.

Figure 1: Extended Effectivity

- For planning of MCM-projects you can use plantypes for which the option **Is configured** is set to **Yes** or **Max Configuration**. Please refer to the [Figure 2](#).
- For **not versionable** plantypes you have to select **No** or **Only Planning State Support** for option **Has versions**.

[-] <b>Flags</b>	
Has versions	No
Is searchable	No
Is abstract	No
Has attachmer	No
Is relevant	No
Own rights	Inherited from parent type
Display parent	Inherited from parent type
CCZ Membersh	Inherited from parent type
Is configured	Yes
Defined by	Customer

Figure 2: Configure Options in Configuration Tools

### Planning Persistent Modification Statements and Not Versionable PPR-Components

Plan MCM-projects with the help of **not versionable** PPR-components which are fixed assigned to a function and modification statement. For planning MCM-projects with the help of versionable plantypes create functions with modification statements. When opening the project you have to select a modification statement respective to a function.

### Planning Temporary Modification Statements and Not Versionable PPR-Components

When planning MCM-projects with the help of temporary modification statements and non-versionable PPR-components, temporarily give extended effectivities in **Properties** dialog of the object. When opening the project do not select a modification statement respective to a function. All created PPR-components contain the value which is indicated in the temporary modification statement. *Please refer to the Table 1.*

**Table 1: Behavior of Extended Effectivities – Standard Configuration**

Modifications - PPR-Components	Rule to apply the extended effectivity of the selected modification statement (standard configuration)
Create	The extended effectivity of the selected modification statement is set at a newly created component.
Delete	The extended effectivity of a deleted component is the previous effectivity cut back by the effectivity of the selected modification statement.
Copy	When a component is copied, the extended effectivity of the copy is equal to that of the selected modification statement.

**Table 2: Behavior of Extended Effectivities – Max Configuration**

Modifications - PPR-Components	Rule to apply the extended effectivity of the selected modification statement (standard configuration)
Create	An empty extended effectivity – means always valid – is set at a newly created component.
Delete	The extended effectivity of a deleted component is set to the value NULL, means never valid.
Copy	When a component is copied, the extended effectivity of the copy is an empty extended effectivity, means always valid.

## 2.2 Version Compatibility

Version compatibility of the relevant plantypes is a mandatory prerequisite for working with an MCM-project.

MCM-projects can be considered as the historical presentation and tracing of all executed planning steps in the project. MCM-projects are continuously checked by various consistency tests. Projects marked as an MCM-project are subject to Change Management Controlled (CMC). The components and relations can be edited through the defined tasks.

- Versions are marked in the project by planning states that display the respective life cycle stage of versionable (new versions can be created) components.



- Several versions of PPR-components can exist in a project.
- The planning state can be used as a filter setting for opening projects.

Versionable components can be edited in the following two ways:

- Versioning for maximum configuration of the PPR-components, *Please refer to the [Versioning for Maximum Configuration of the PPR-Components](#).*
- Versioning for standard configuration of the PPR-components, *Please refer to the [Versioning for Standard Configuration of the PPR-Components](#).*



For more information on the configuration of MCM-projects, planning, configuration of the plantype set, and versioning, *Please refer to the [IPD Server Manual](#) and [PPR Navigator Manual](#).*

## 2.3 Rules for Editing MCM-Projects

A MCM-project can be edited only if you have, previously, executed the steps described below:

### Preparing Plantype Sets

In the first step you need to check which PPR-components and relations are subject to the MCM-project in the plantype set.

### Marking PPR-Components for MCM-Project

All plantypes of the product, resource, and process structure can be used for an MCM-project. Only if you set the attribute **Has versions** to **Yes** in the configuration tool is a plantype subject to the consistency tests in the MCM-project.

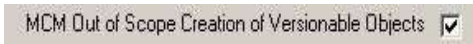
### Marking Relations for MCM-Projects

In order for relations to undergo a consistency test in the MCM-project, these relations must have an owner. Owners can only be PPR-components for which versions are possible.

The PPR-components that are owners of a relation are set in the configuration tool under the attribute **Owner type** of the relation.

### Preparing Versionable PPR-Components for the MCM-Project

As of version PE 5.16 SP4 MCM-projects can be created in which you can create, edit, or delete versionable PPR-components without selecting a task and modification statement. The prerequisite for this is you have to select the option **MCM Out of Scope Creation of Versionable Objects** under the tab **Filter** in the properties dialog of the project.



**Figure 3: Edit MCM-project without Selected Task**

With the help of this procedure you can prepare an MCM-project without setting with which task and modification statement these PPR-components are to be edited. No consistency checks can be made for PPR-components that can be created without a selected action.

Ensure that no versions are created for PPR-components that have been created without an action. After the project preparation for the MCM-project has been completed, you can create and edit versions for these PPR-components with a selected task. All the rules for an MCM-project then apply here. *Please refer to the [Mark Project as MCM-Project](#).*

### Create Planning State in the Project

Various planning states are required in order to edit tasks in the MCM-project. Planning states are created in the project library. As many planning states are required for editing, the tasks should be created in the project. In order to edit tasks in an MCM-project, the option **Corresponding action state** is of special importance. Please refer to the [Figure 4](#).

For more information on **effectivity** and **promotion behavior**, Please refer to the [PPR Navigator Manual](#).

### Note

*When creating a MCM-project, the number and valence of planning states is set according to the edited tasks. It is up to you to define how many and which planning states should have previously been created in the project. The reference to a task is created via the planning state. In the Process Engineer three valence ratings of planning states are specified, whereby the planning state Unlock has the highest priority and Edit has the lowest. The sorting index specifies the priority of a planning state within a valence rating. The sorting index plays a major role in the the switching of planning states.*

**Figure 4: Planning State for MCM-Project**

### Corresponding Action State

You can use the **corresponding action state** to create reference of the planning state to the state of a task. Please refer to the [Figure 4](#).

The reference of a planning state to a task is created via the following options:

- Under planning state, the option **corresponding action state**
- Under a task, the option **action state**

When the action state is changed, another planning state is switched to the following:

- Both options (**corresponding action state** and **action state**) must correspond to each other, i.e. they must receive the same mark. This applies to all planning states that are required for editing actions.

- The corresponding planning state of a PPR-component is assigned when the action state is changed. This assignment ensures that the PPR-components that are assigned to this action have the same planning state. *Please refer to the [Creating Actions](#).*

### Create New Version is Forbidden

If this field is activated for a planning state you cannot create any new versions for the PPR-components whose previous version has been assigned this planning state. *Please refer to the [Figure 4](#).*

## 2.3.1 Exclude Attributes from MCM-Project

You want to use attributes, which are not subject for MCM – Project usually accomplished consistency checks in your MCM-project. With the help of the option **Exclude from MCM** you can mark these attributes. The option **Exclude from MCM** can be used for all attributes.



### Note

*This setting should be accomplished only by an administrator or an equal employee.*

You can execute this setting on the types and plantypes level. On default the option is set to **No**. Attributes with which the option **Exclude from MCM** is set to **Yes** are not subject for MCM-Projects usually accomplished consistency checks any longer. For example, if the option is set to **Yes** on attribute Process Name, then you can accomplish changes at any time, independent of which task the process component was produced for this attribute.

But if the option is set to **No** with the attribute, changes can be only accomplished, if the task and the range of validity are selected, with which the component were produced.

Double min.	0.0000000000000000
Exclude from MCM	No
Extended Copy	No
Group	Group:1240 (Calculation)-Page:1000 (General)

Figure 5: Exclude from MCM

## 3. Working with MCM-Project

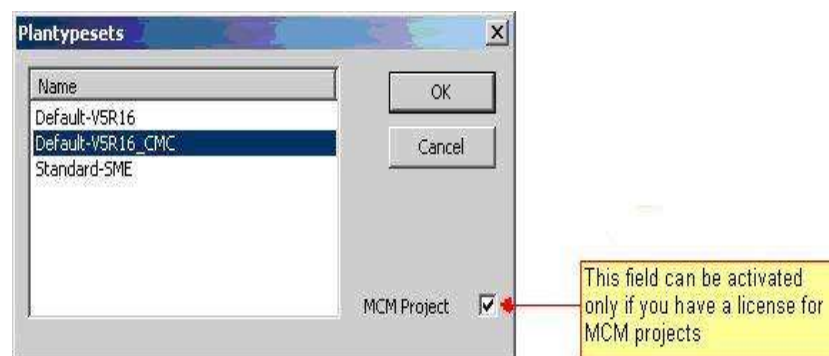
### 3.1 Creating MCM-Project

#### 3.1.1 Mark Project as MCM-Project

When creating a new project you can decide whether the project is subject to the MCM supervision.

##### Note

*An MCM-project can be created, opened, or copied only if you have a license. If you do not have a license, the field is deactivated for MCM-project in the standard configuration.*



**Figure 6: Create MCM-Project**

By using the menu item **Regular Project < - - > MCM-Project** you can mark existing projects as MCM-projects, and you can also undo the MCM-project mark.

You cannot set the time to mark a project as an MCM-project without restrictions. A basic rule is that the project must be marked as an MCM-project when an action is created for a project. As soon as an action is created, a MCM-project can no longer be undone in retrospect.

- In order to be able to assign actions to a project in ENOVIA, a MCM-project must be created.
- A project can be marked as a MCM-project in retrospect only if the PPR-components in the project have only one version.

##### Note

*For projects that are marked retrospectively as MCM-projects, the planning state for previously created PPR-components should have the highest priority. If this condition is not met, the planning state must allow the creation of further versions.*

##### Note

*This menu item can be executed only by the administrator or by an employee who have the function permission regular project < - - > MCM - projects.*



Figure 7: Function Permission MCM-Project

### To Mark Projects as MCM-Projects

- Open **Tools** menu and select the **Regular Project <--> MCM-Project**. The menu item is active only if a license is available.

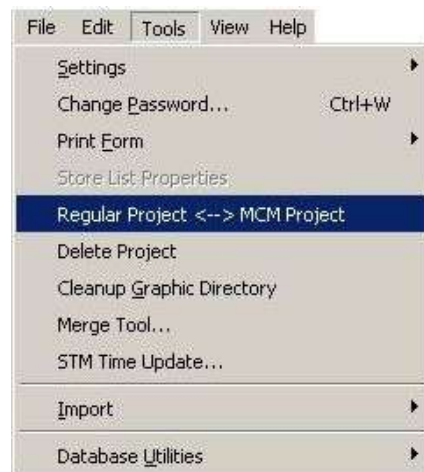


Figure 8: Mark MCM-Project

- 1) Check **MCM-Project** checkbox to activate and deactivate the change management. The project is subject to the MCM only if this field is checked.
- 2) Only projects that are not open are displayed.



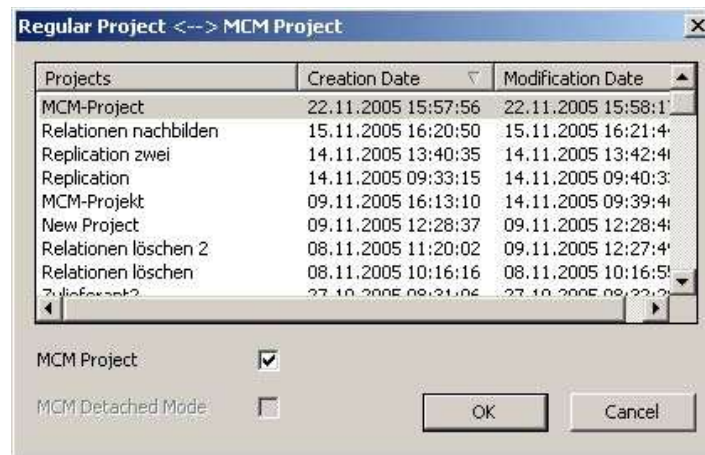
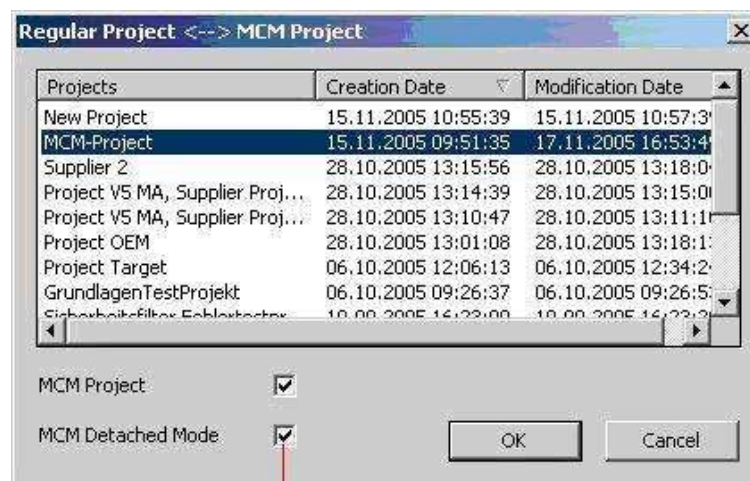


Figure 9: Mark MCM-Project

### Activating MCM Detached Mode

In projects subject to MCM, the editing in the project is carried out via defined actions. The actions can be defined in ENOVIA and in the Process Engineer. If actions are defined in the Process Engineer, the field **MCM Detached Mode** must be activated.

- 3) The field **MCM Detached Mode** can be activated if the project has previously been marked as a MCM-project. Open the dialog via the menu item **Project < - - > MCM - Project**.



Where the actions are created is managed with the help of MCM Detached Mode – ENOVA or Process Engineer

Figure 10: Defining Actions in the Process Engineer

- If the checkbox **MCM Detached Mode** is active, the MCM is controlled by the DPE. The actions are defined in the DELMIA Process Engineer®.
- If the checkbox **MCM Detached Mode** is not active, the MCM is controlled by ENOVIA. Actions and modification statements can be defined only in ENOVIA.





### Caution

*If the checkbox MCM Detached Mode is not active, you cannot define any new actions in the DELMIA Process Engineer® and cannot change any action states.*



For more information, Please refer to the chapter on MCM Installation and Preparation in the [Administration Manual](#).

## 3.1.2 Creating Actions

MCM-projects are edited only for defined actions. Actions are available in ENOVIA or are created in the Process Engineer. You can only edit and create actions if the project is marked as a MCM-project. MCM-projects can be edited only by those who have the corresponding access rights.

- In order to create actions in the Process Engineer you must have previously activated the option **MCM Detached Mode** in the properties dialog of the project. Please refer to the [Figure 10](#).
- In order to create actions for a project in ENOVIA, the MCM-project must have already been created.



### Note

*Actions that are created in ENOVIA must also be completed in ENOVIA. Creating and changing the action state is executed exclusively in ENOVIA. The required planning states used for editing the actions created in ENOVIA must be created in the Process Engineer.*

*Immediately after the actions are created in ENOVIA, these tasks get created via the Web services in the Process Engineer. The options in the properties dialog cannot be changed.*

Actions are created in the Process Engineer in the project library. You must create effectivity ranges for actions. Please refer to the [Creating Modification Statements](#).



### Note

*Actions and effectivity ranges can be defined in the Process Engineer if you have activated the field CMC detached mode in the properties dialog of an MCM-project.*

### To Create New Actions

- 1) Open project library.
- 2) Select the menu entry **New < Action** in the context menu.

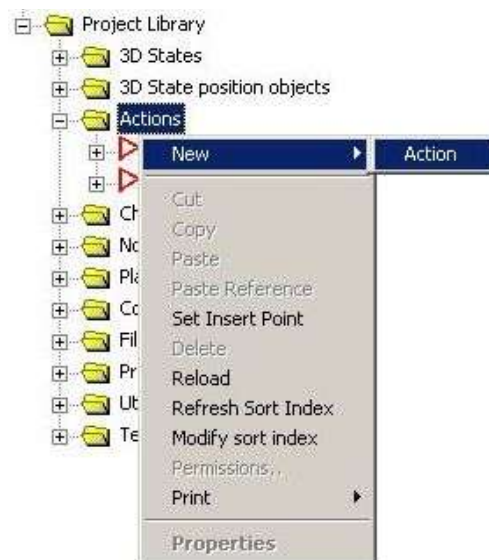


Figure 11: Create New Action

The two options **Action state** and **Is usable** are of particular importance in the creating and editing of actions. All further options such as name, action type, and group number can be named according to the action. With the options action type and group number you can further specify the action, such as information on production areas or the users who are authorized to work with actions in the project.

### Note

*When switching the action state is recommended to create a so-called value list in the configuration. With the help of the created value list you can select the action state from a combo box.*

For more information on the configuration of value lists, [Please refer to the Administration Manual](#).

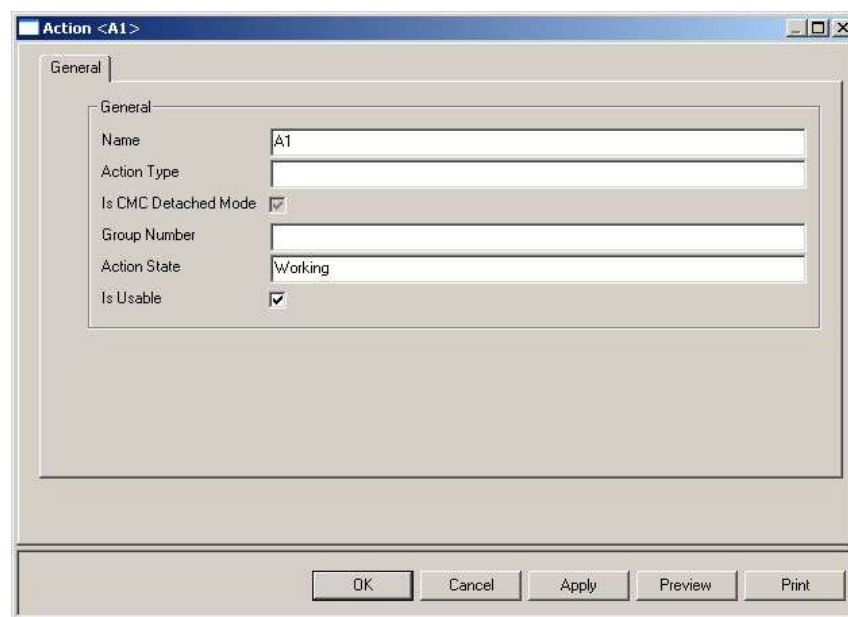


Figure 12: Define New Action Dialog

### Note

*These fields cannot be edited for actions created in ENOVIA.*

### Option Is Usable

You can select and edit actions only if you have activated this field. This field remains activated until the action is completed. When creating actions that are edited simultaneously in the context of further actions, activate this field after all actions have been created. Release the actions for editing only if the action volume has been definitely set.

### Using Action State

Use **Action State** to create reference to the **Corresponding action state** of the respective planning state. This field must be marked unambiguously. When the action state is changed, another planning state is switched to. Please refer to the [Create Planning State in the Project](#) and [Corresponding Action State](#).

### Change Action State

When changing the action state, only the PPR-components that are assigned to the action are taken into consideration. Another planning state is switched to one which corresponds to the new action state.

### Note

As of version PE 5.16 SP4 you can use your own programs via the interface *User Exit* in the IPD-Server. With the help of these programs you can check the consistency of the planning in an MCM-project before you approve to a higher action status. The action status is changed only if the check is successful.

For more information, Please refer to the [Administration Manual](#).

### Example of Action State Mark

The action state is to be changed – from the planning state **Working** to the higher planning state **Integrate**.

### Initial Situation

The assigned action state corresponds to the actually created name of the planning state – **Working**.

Figure 13: Assignment of Planning State to the Action

The change of action state is created via the two options **action state** and **corresponding action state**. This example should show that the assignment is ensured even with a freely selected mark, and that it does not depend on the priority of a planning state.

Figure 14: Mark of the Options can be freely Selected

### Reset Action State

It is also possible to reset the action state as long as the components belonging to the action do not have any successive versions.

## 3.1.3 Creating Modification Statements

At least one effectivity range with effectivity must be created for an action. The extended effectivity filters are used for the effectivities. Any number of effectivity ranges can in principle be created for every action.



### Note

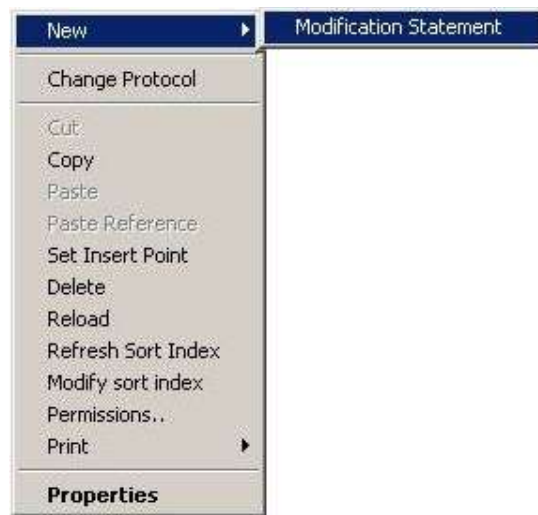
*Extended effectivities play a special role in MCM-projects. Effectivity ranges for an action are specified exclusively with extended effectivities (i.e. production numbers). The extended effectivities are used in an MCM-project to ensure the display and effectivity of the versions – for example, when creating versions with different effectivity ranges.*



For more information, Please refer to the [Project Library](#).

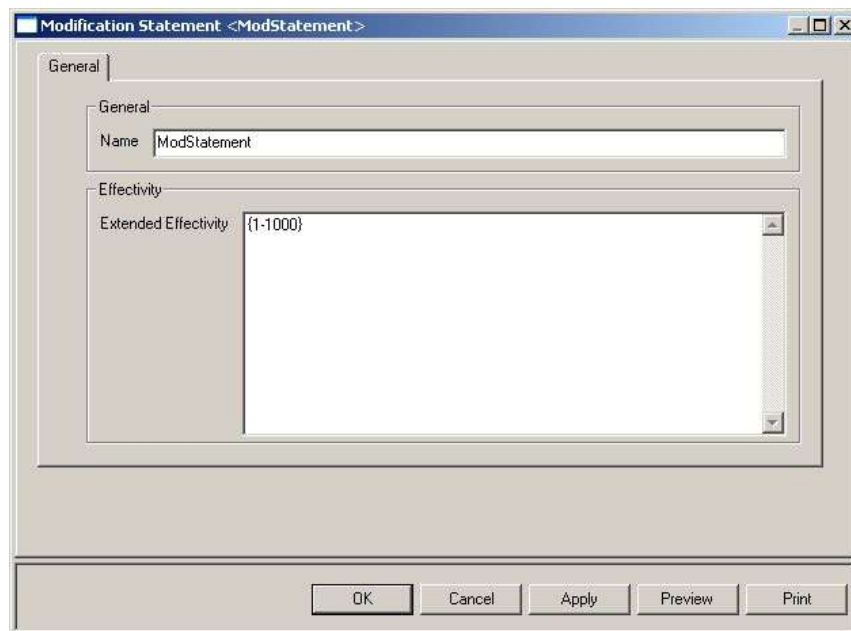
### To Create Effectivity Range for Actions

- 1) Open the project library in order to create effectivities ranges.
- 2) In the context menu of an **action** select the entry **New < Modifaction Statement**.



**Figure 15: Create New Modification Statement**

Mark the effectivity range in the dialog. When entering the effectivity you must pay attention to the spelling; both the syntax in the Process Engineer and that of ENOVIA.



**Figure 16: Dialog for a New Modification Statement**



### Note

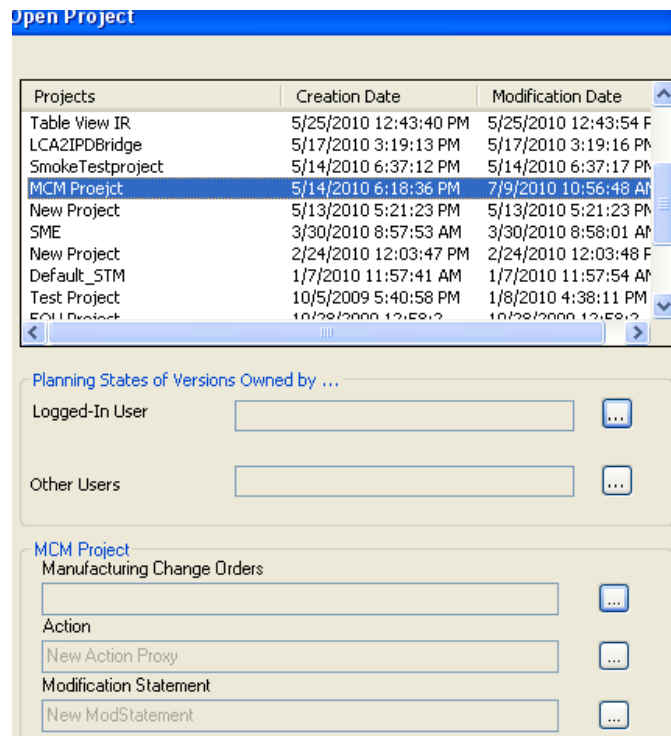
*In the field Extended Effectivity there must be at least one effectivity expression entered – in the example it is the production number {1-100}. Ensure that the spelling is correct.*

## 3.1.4 Displaying Selected Action and Modification Statement in Status Bar

You can display the name of selected Action and Modification statement in the status bar. Also when there is a change in the action and mod statement during re-loading of project, the corresponding name gets updated in the status bar. If two instance of same MCM project is opened with different Action and Modification statement then Action and Modification statement names in the

status bar are updated according to the selection of current project. The maximum limit to display the text in status bar is 127 characters.

- 1) Go to the **Tools > Settings > Change > Miscellaneous** tab.  
Select **Show Action and MOD statement in status bar** check box to display selected Action and MOD statement in the status bar.
- 2) Open a MCM project having predefined Action and Modification Statement using **File > Open Project**.



**Figure 17: Open MCM Project with predefined Action and Mod Statements**

The Action and Modification statement name appears in the status bar. The Modification statement name is displayed prior to the Action name.



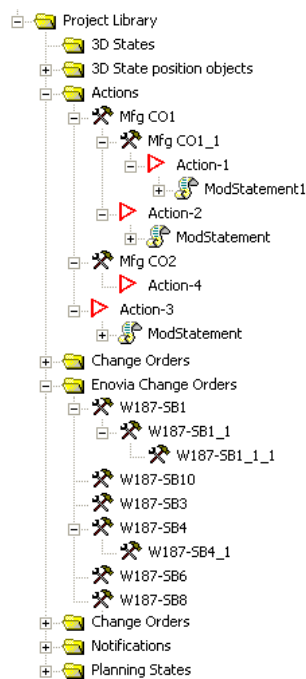
**Figure 18: Action and Mod Statements Name in Status Bar**

### 3.1.5 ENOVIA Change Order

A new folder “ENOVIA Change Orders” is available in Project library for MCM projects. The ENOVIA Change Orders (ECO's) are created inside the folder “ENOVIA Change Orders” in the Project Library and also under another ECO.

The ECO's created inside the “ENOVIA Change Orders” folder are linked to each other and with other ECO in order to create a hierarchy.

An ECO can be moved under another ECO in ENOVIA Change Orders folder by drag and drop.



**Figure 19: ENOVIA Change Orders Folder in Project Library**

Relation can be created between ECO's with calculation models and parts.

The folder "ENOVIA Change Orders" has no impact on filtering behavior with existing Mfg CO's.

The ECO's takes part in filtering using CO. *Please refer to the [Using the Product Structure as a Filter](#).*

By default the CO Loader loads the Manufacturing Change Order into "ENOVIA Change Orders" folder.

The old behavior of creating the Manufacturing Change Orders under the Action folder can be invoked through the following registry setting:

[HKEY\_CURRENT\_USER\\SOFTWARE\\DELMIA\\ergoplan\\PPRLoader]  
CreateCOOldWay = 1|true|yes.



### Note

*It is not possible to create Action Proxies under ECO's.*


*The link between Mfg CO's and ECO's is not supported.*

*The searching of ECO's in "Mod-Statement Selection dialog" is not supported.*


### Advantage of ECO

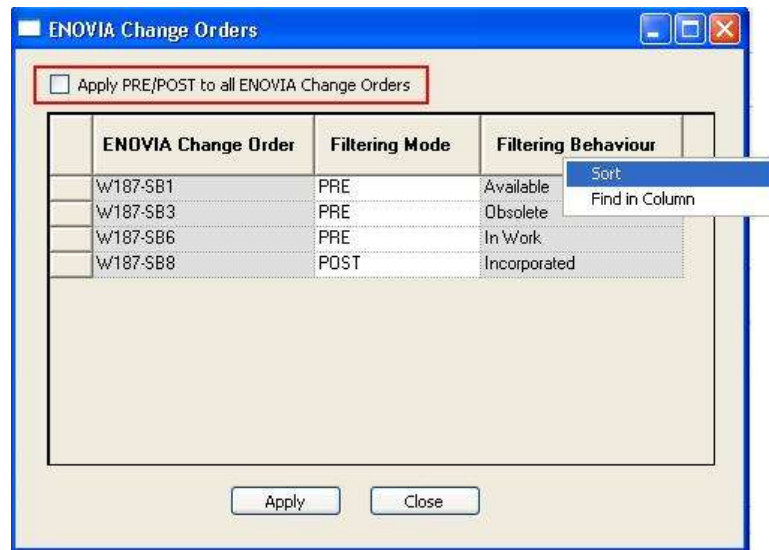
It reduces the amount of data in the Action root folder to avoid having to load all Mfg Change Order each time in order to get an Action Proxy/Mod Statement.

## 3.1.6 Display ENOVIA Change Orders Sorted by Name

- 1) Create a MCM Project.  
Create ECO's, Actions, and Mod-Statements under action folder.  
Create link between ECO's with Calculation Model.
- 2) Select CM in **Filter** combo-box in the **Product** tab of OPD. *Please refer to the [Figure 44](#) and [Using the Product Structure as a Filter](#).*
- 3) Select **Apply Filtering by Change Order** in OPD.  
The  button gets enabled.



- 4) Click  button  
 The ECOs linked to the selected calculation model are displayed in **ENOVIA Change Orders with Behaviour** dialog box.  
 The ENOVIA Change Orders appear sorted by name.



**Figure 20: ENOVIA Change Orders with Behavior**

- 5) Select **Apply PRE/POST to all ENOVIA Change Orders** to override all ECOs to PRE/POST.
- 6) Unselect **Apply PRE/POST to all ENOVIA Change Orders** to revert the ECOs status back to the default Filtering Mode based on Incorporation Status.
- 7) You can sort the grid entries by name by right mouse click on any of the column headers and clicking **Sort** option in the context menu. You can search for entries in a particular column by clicking **Find in Column** option in context menu.
- 8) Click **Apply** to save changes and **Close** to close the dialog box.



### Note

*The number of columns in ENOVIA Change Orders with Behavior dialog box is fixed. Extra columns cannot be added. Columns (ENOVIA Change Order, Filtering Mode, and Filtering Behavior) in ENOVIA Change Orders dialog box cannot be reordered.*

## 3.2 Opening a MCM-Project

In order to open a MCM-project it is necessary to select an action and effectivity range in order to plan and edit versionable PPR-components. For creating a MCM-project, *Please refer to the [Mark Project as MCM-Project](#).*

When opening a project you can only edit the versions of the PPR-components that have previously been created for the selected action or that you have created for the action. Only if the planning state permits it can versions be edited with this action. *Please refer to the [Versioning](#).*

Select the checkbox **Disable MCM Warning Message**, **Tools < Settings < Change < Miscellaneous < Disable MCM Warning Message** to not get the warning message while opening the MCM project.



This warning message comes only if you have not selected a modification statement.

All further versionable PPR-components that are assigned to another task are not available for editing even if they have the same planning state. When actions are edited a consistency test is executed; it ensures that only the PPR-components assigned to the action can be edited.



### Note

*If there is no license for MCM-projects, you receive the License information message when opening the MCM-project.*



Figure 21: License Information



### Note

*If the setting - expand last open project in the browser – is activated in the DPE, the DPE is always opened with the last saved project. If the last saved project is an MCM-project and the license for MCM-projects has expired, the MCM-project is not opened; instead the DPE is opened with the dialog Open project.*



### Note

*Only the actions of the checkbox Is Usable is activated can be selected when ever opening a project.*

### To Open a MCM-Project

- 1) Select **File < Project**.
- 2) Select MCM-Project.

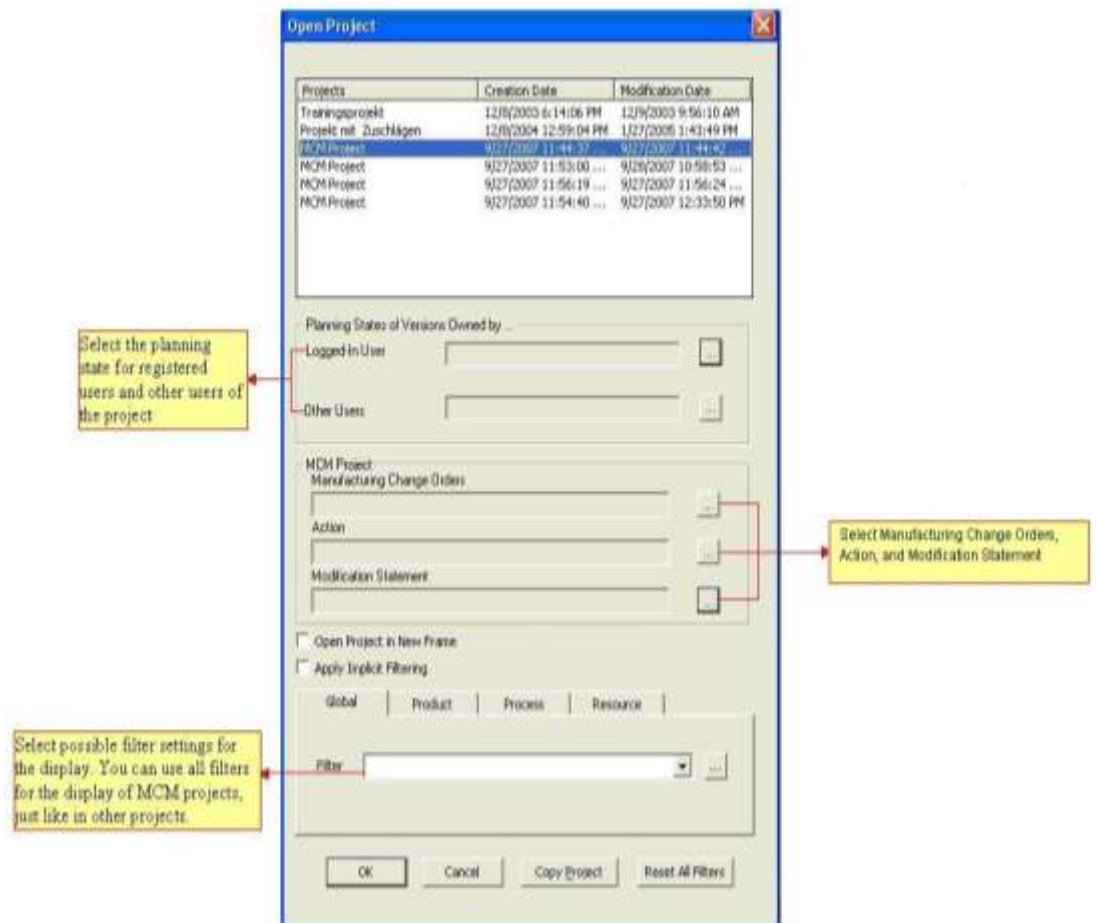


Figure 22: Open MCM-Project

### Customization

- 3) Click **Tools < Settings < Global** and set the **configmanagementfilter\_by\_co** global key to value 1.

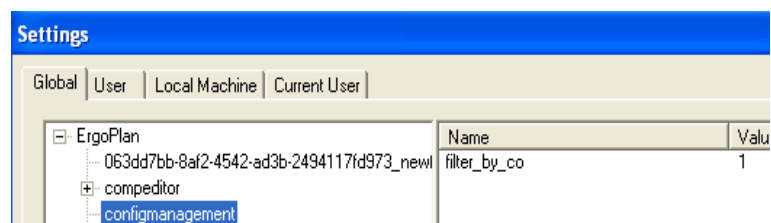


Figure 23: Global Key

- 4) Click **Tools < Settings < Local Machine < ErgoPlan < dialogs** and set the **activatesearchinactiontree** to value 1.

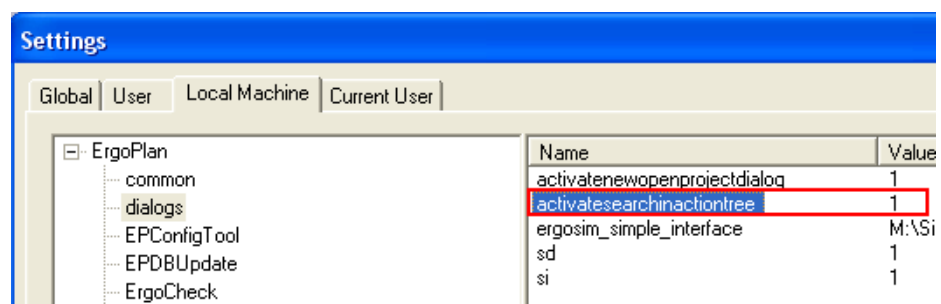


Figure 24: Local Machine - Dialogs

- 5) Click **Tools < Settings < Local Machine < ErgoPlan < common** and set the **expandactiontree** to value 1.

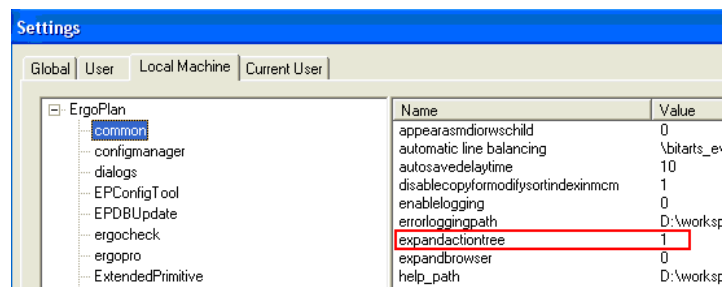



Figure 25: Local Machine - Ergo Plan

### Note

From R19 onwards, the client, search functionality, and expand tree options are activated by default.

- 6) Click Push button with the three dots , Please refer to the [Figure 22](#), and select **Manufacturing Change Orders, Action, or Modification Statements**.
- 7) A **Select Modification Statement** dialog displaying the Manufacturing Change Orders, Actions, and Modstatements in a tree like structure appears.

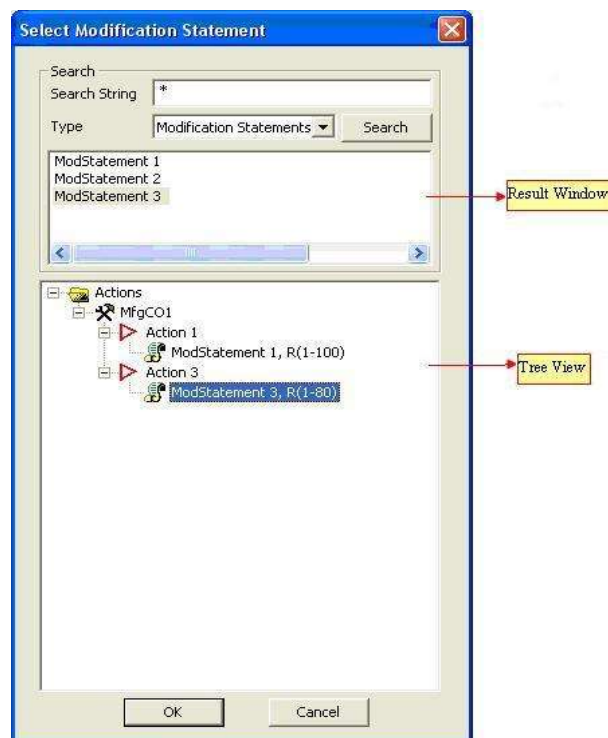


Figure 26: Select Modification Statement

- 8) Select **Type** that you want to search in the tree structure or enter the search strings and click **Search**.
- The search is performed based on your inputs. Entire database is searched for matching objects, and the results are displayed in the result window. If two or more items of the same name exist in tree, the result lists out all the matching entries.

- 9) If you used search string as "\*" and in **Type** <All> is selected, then the search result includes all the three types (Manufacturing Change Orders, Actions, and Modification Statements) and all existing items of all three type. If you used search string as "\*" with any one type selected in **Type** combo-box, the search result includes all components of selected type.
- 10) You can select the entries either from the result window or tree view.
  - Double click the items in result list (result window) expand the tree structure and highlight the result item in tree. You have to manually expand and collapse the highlighted item in tree.
  - Double click the **Actions** folder in the Tree view, the Actions folder expands and corresponding items related to that folder get displayed. Double click the selected action (▶), and the corresponding modification statements (🔧) gets displayed. Select the modification statement and click **OK**. To deselect a Modification Statement, click **Actions** folder and confirm with **OK**.



### Note

*The displayed name for Modification Statements can be configured in Configuration Tool. The selected modification statement name is shown in Modification Statement field in Open Project dialog as well as Edit Action Proxy/Mod Statement dialog with configured name. The tree view of Select Modification Statement also displays the modification statement with configured name.*

At least one modification statement must be defined for every action. Please refer to the [Creating Modification Statements](#). You cannot edit any versionable components without the selected action and effectivity range.



**Figure 27: Message: No Effectivity Range Selected**



### Note

*In addition to this information, you should use the Extended validity filter when opening an MCM-project. When stating the extended validity filter, consider that the value must be within the validity range - for instance, the value should range between 1 and 100.*

*To edit MCM-projects, select a task and a validity range. To ensure that all functions i.e. deleting of relations – are represented in accordance with the functions performed, the Extended validity filter should be used.*

*Only by opening an MCM-project with the aid of the Extended validity filter you can ensure that the views contain the data that are actually assigned to the validity range.*

### 3.2.1 Historical View of the Planning State

All PPR-components and versions that have been created for an action in the MCM-project are displayed in the project library in the directory Actions under the respective effectivity range. This type of display makes it possible to comprehend the historical course of the changes to the planning state of an action for every time period.

In this example, a process structure for the action **A3** has been created. In the directory of the action, the generated relations are shown along with the PPR-components and versions.

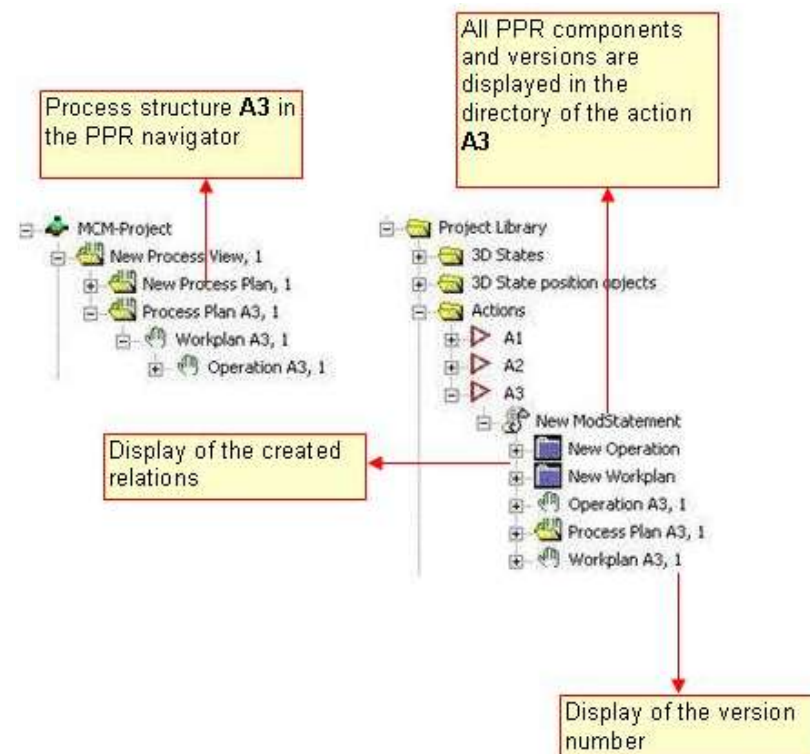


Figure 28: Historicized Display of PPR-Components and Relations of an Action

### 3.2.2 Selecting Actions in the Open Project

Actions and effectivity ranges can also be selected in the open MCM-project.

- 1) In order to select a new action for editing in an open MCM-project, open the context menu on the project node.
- 2) Select the menu item **Edit Action Proxy and Mod Statement**.

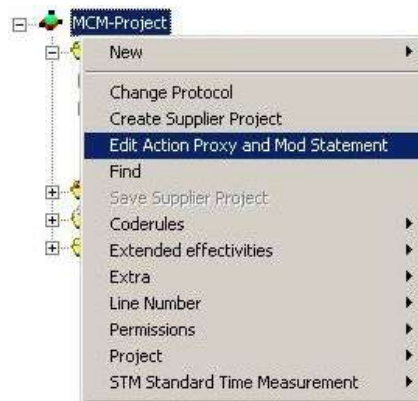



Figure 29: Open the Context Menu on the Project Node

- 3) Click Push button , Please refer to the [Figure 30](#), and select **Manufacturing Change Orders, Action, or Modification Statements..**

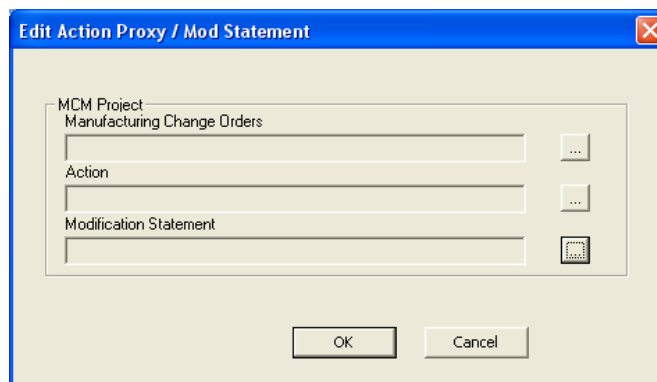
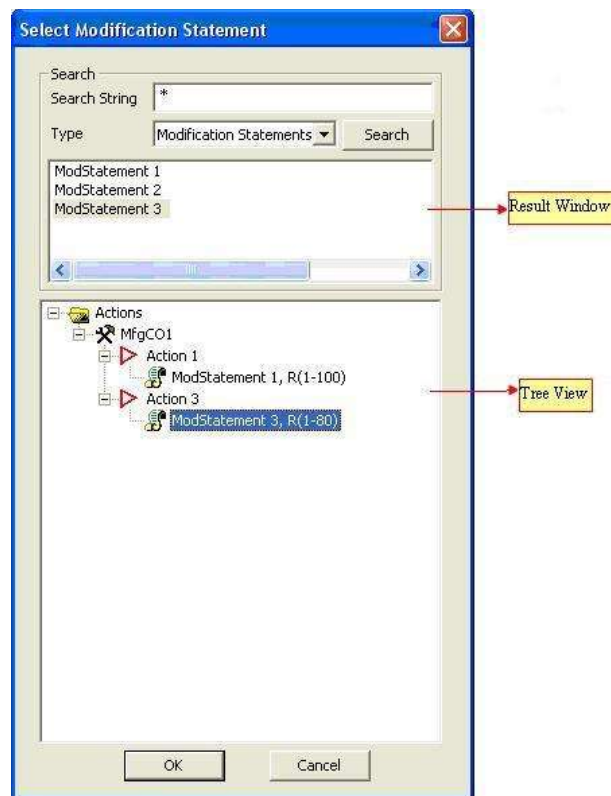


Figure 30: Edit Action Proxy/Mod Statement

- 4) A **Select Modification Statement** dialog displaying the Manufacturing Change Orders, Actions, and Modstatements in a tree like structure appears. Please refer to the [Opening a MCM-Project](#).



**Figure 31: Select Modification Statement**

- 5) Confirm the selection with **OK**.
- 6) To deselect a Modification Statement, click Actions folder and confirm with **OK**.



### Note


*Only the actions for which the option Is Usable is activated can be selected.*

### Selecting Modstatements, Manufacturing change orders, and Actions

You can select only those modstatements, manufacturing change orders, and actions for which you have set access rights on parent till root node. For example,

If you have created the MCM project structure like the below one with your own user rights:

Mfg CO1: No Access  
 Mfg CO1.1: Read Access  
 Action 1: No Access  
 ModStatement1  
 Action 2: Read and Execute Access Rights  
 ModStatement2

When you open project dialog for MCM project and click Push button , Please refer to the [Figure 22](#), and select **Manufacturing Change Orders**, **Action**, or **Modification Statements**. A **Select Modification Statement** dialog appears. Please refer to the [Figure 26](#).

If you select \* in **Search String** and <All> in **Type** and start Search, you get Action 2, Mfg Co 1.1, ModStatement2 in list. If you try to select Action 2 and manufacturing change order (Mfg Co1.1), you get the following message.

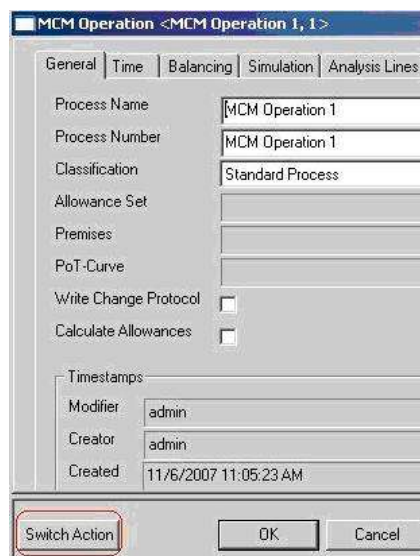


**Figure 32: Access Rights Message**

This message appears because you set No access for parent Mfg CO1. If you change the access permissions then you can select the action, manufacturing change order, and modstatements.

### Improved Usability via Switch Action Button

The **Switch Action** button appears on **Properties** dialog for the component.



**Figure 33: Switch Action Button on Property Dialog**

- 1) Click **Switch Action** button and **OK** on Properties dialog.
- 2) Right click project node and select **Edit action proxy and mod statement**.
- 3) The Action and ModStatement associated to the component on project get set.

### 3.2.3 Searching ModStatement

Create a manufacturing change order in the project library and organize actions, modification statements in the structures. *Please refer to the [Figure 34](#).*



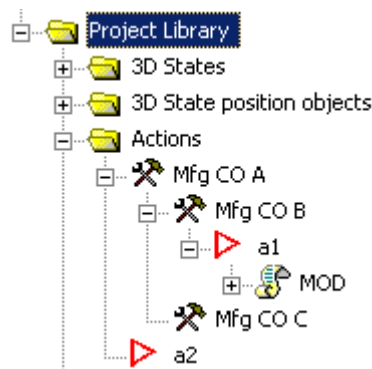


Figure 34: Actions Organized into Manufacturing Change Orders

### Find Usage Functionality

- 1) Right-click MCM Workplan and select **Find Usage < General** command. Please refer to the [Figure 35](#).
- 2) Find Usage is used to find out by which modification statement an MCM object was created or modified. Find Usage functionality shows the action to which a modification statement belongs. It also display manufacturing change orders for the found usage of an action.

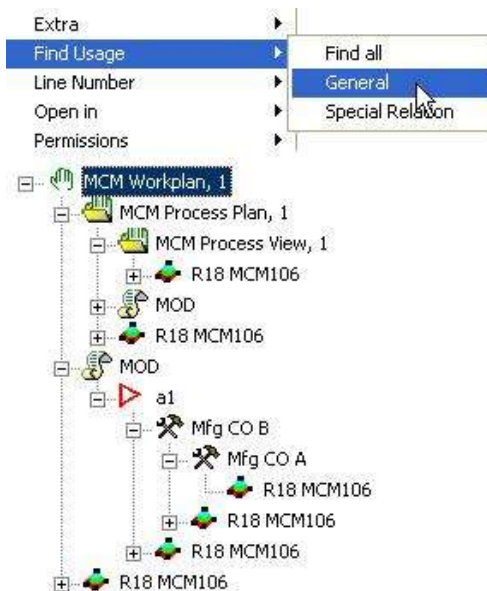


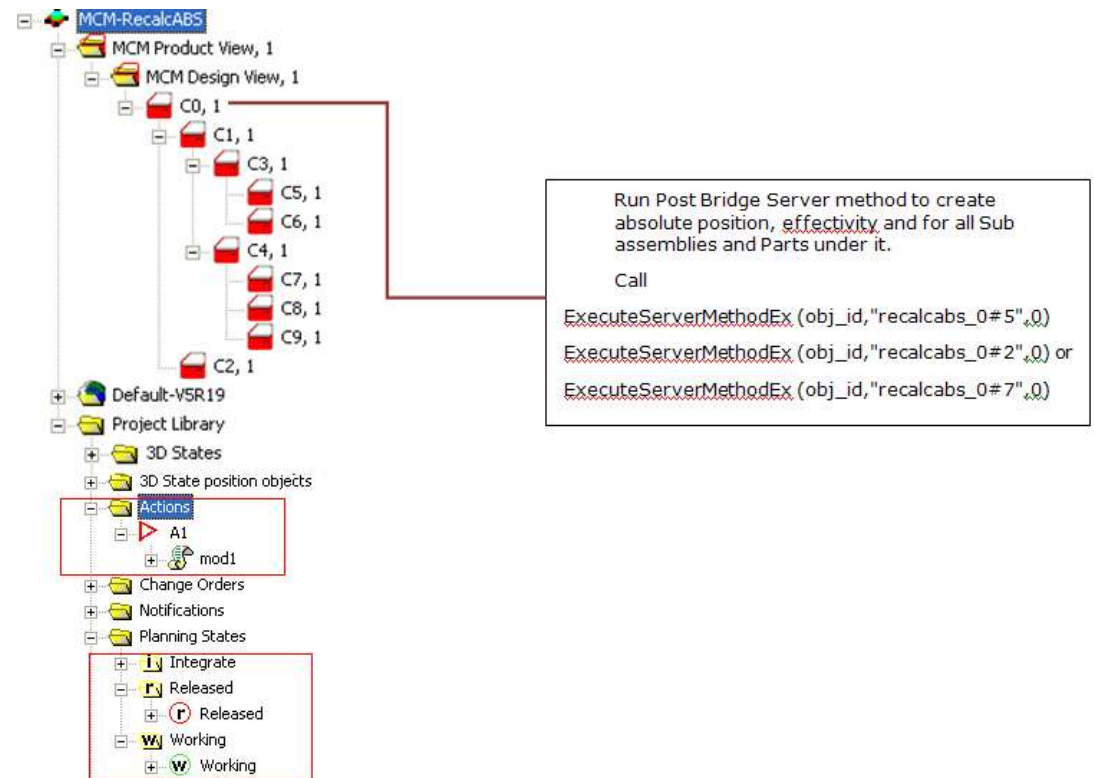
Figure 35: Find Usage Command

## 3.3 Calculating Absolute Effectivity, Position, and Bounding Boxes

If attribute property **“Exclude from MCM”** is set to **Yes** in the configuration tool, the server call method for calculation of absolute effectivity, position, and bounding boxes can be run on MCM configured objects without selecting a specific action and mod statement even if some calculated objects are under MCM control. The calculations are carried out using the server call method without giving the MCM consistency check error.

If the attribute property **“Exclude from MCM”** is not set then the default behavior prevails; server call method for calculation of position, effectivity and bounding boxes respect MCM. If mod and action is not selected then MCM consistency check error is displayed.

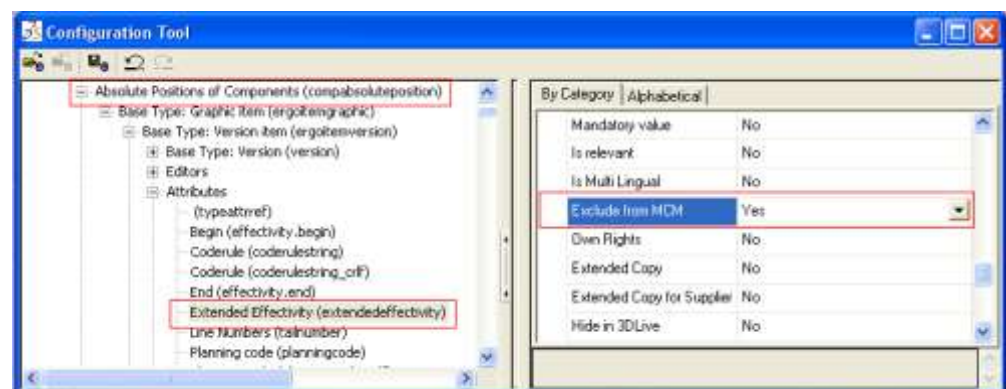
- 1) Create a MCM project **“MCM-RecalcABS”** in **Detached** mode.



**Figure 36: MCM Project “MCM-RecalcABS” in Detached Mode**

- 2) Open the Plan type editor on MCM sub Assembly and set **“Has versions”** to **Yes** and **“Is configured”** to **Yes**.
- 3) Create an Action and modification statement as **A1** and **mod1**. Also make sure that the correct planning states have been created.
- 4) Run the following Post Bridge Server method on **MCM Design View** to create the absolute position, effectivity, and boundary boxes for all Sub assemblies and Parts under it.  
Call  

```
ExecuteServerMethodEx (obj_id,\"recalcabs_0#5\",0)
ExecuteServerMethodEx (obj_id,\"recalcabs_0#2\",0) or
ExecuteServerMethodEx (obj_id,\"recalcabs_0#7\",0)
```
- 5) Set the property **“Exclude from MCM”** to **Yes** for all the required attributes like **“alternativenam”**, **“extended effectivity”** under type **compabsoluteposition** and **“extendedeffectivity”** under **“ergocompbase”**.
  - **“Extendedeffectivity”** to **Yes** under type **“compabsoluteposition”**



**Figure 37: Extendedeffectivity” to Yes under type “compabsoluteposition”**

- **“alternativenam”** to **Yes** under type **“compabsoluteposition”**

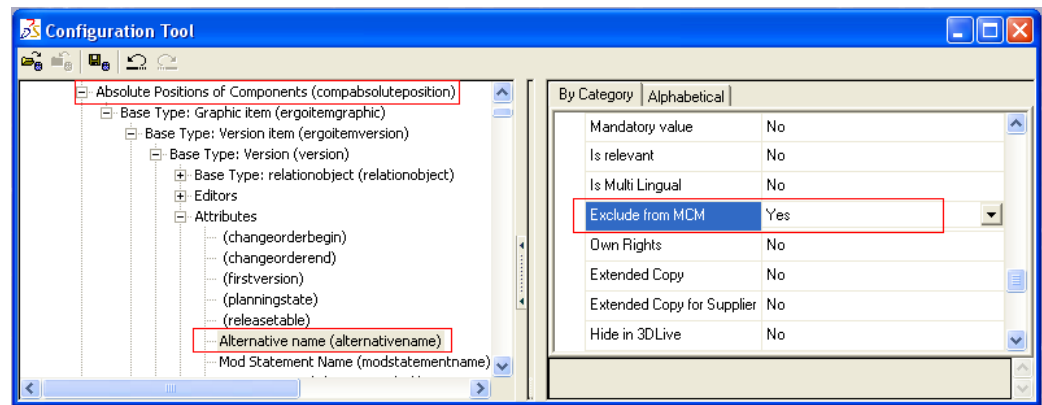


Figure 38: “Alternativenam” to Yes under type “compabsoluteposition”

- “Extended Effectivity” to Yes under type “ergocompbase”

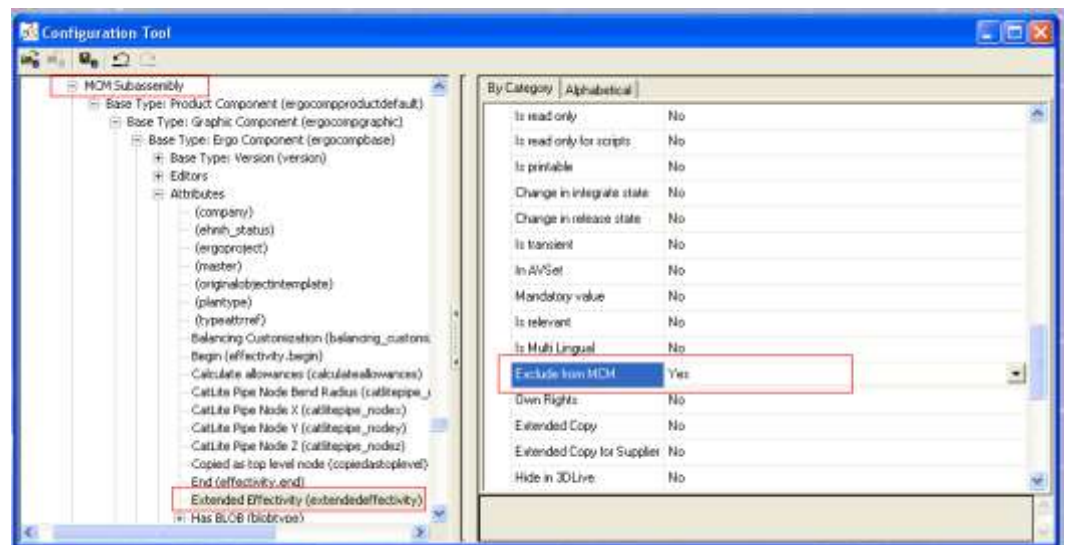


Figure 39: “Extended Effectivity” to Yes under type “ergocompbase”

If recal abs has to be calculated in released state, then “change in released state” property of “compabsoluteposition” relation must be set to **Yes**.

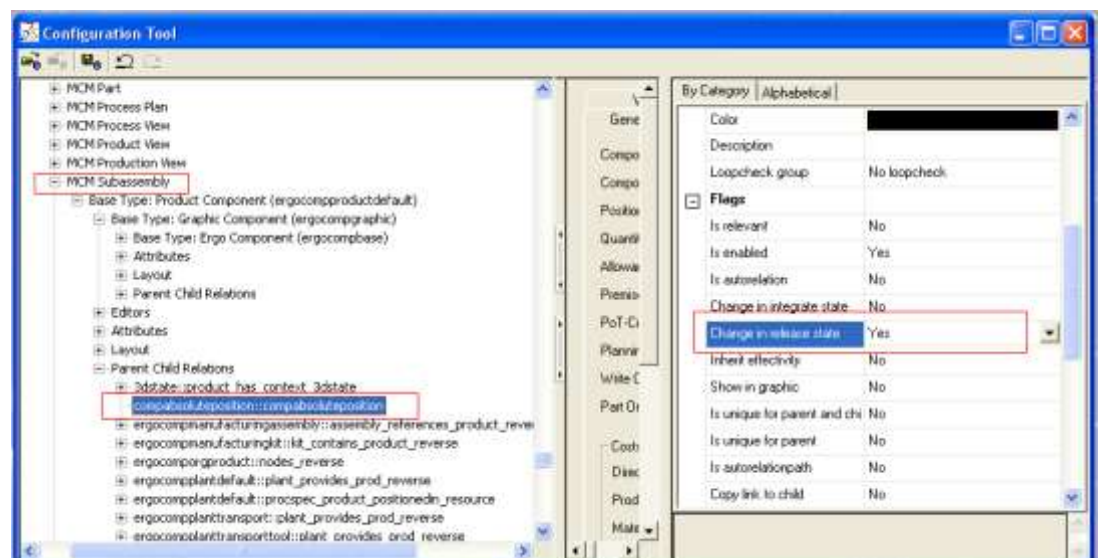


Figure 40: “Change in released state” property of “compabsoluteposition” relation to Yes

- 6) Run the following Recalculation script on the Subassembly C0.

```

sub main(id)
obj_id =Data.GetAttributebyId(id, "ergocompbase")
outArray = data.ExecuteServerMethodEx
(obj_id,"recalcabs_0#15",0)
end sub

```

The calculations are carried out without giving the MCM consistency check error.

## 3.4 Filtering MCM-Projects

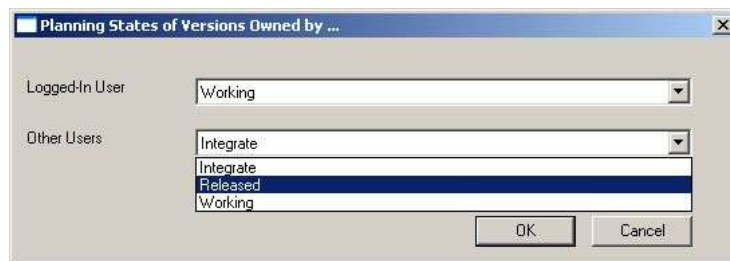
### 3.4.1 Using Planning State as a Filter

MCM-projects can be filtered according to the planning state when opened.

The selection is based on the planning state used in the project. After the planning state has been selected under both options **Logged-In User** and **Other Users**, the most current version is determined in the project and then displayed after opening the project. No more than one version of a PPR component can be displayed.

#### To Open a MCM-Project Filtered according the Planning State

- In order to open a MCM-project filtered according the planning state; enter the respective planning state in both fields.



**Figure 41: Select Planning State**

The filter settings for the planning state can also be used for projects that are not subject to CMC. These filter settings are developed specifically for MCM-projects in order to display different processing states of the actions on the basis of the planning state. In an MCM-project, a so-called version filter ensures that the most current version is displayed after the filter settings are entered. The version filter is available only in MCM-projects.

A MCM-project can be edited by an unlimited number of users. A prerequisite for this is that these users must have rights for accessing the MCM-project.

Two states are checked for these filter settings:

- Planning state for the logged-in user under the option **Logged-In User**.
- Planning state of the other users that have access to the project under the option **Other Users**.

After entering the planning state the program determines the version that corresponds to these filter criteria on the basis of the specified planning state. In order to determine the version you have to specify a planning state under both options.

### Sample Cases of Opening MCM-projects – Planning State Filter Setting

The following example shows the filtering of the project on the basis of the planning state; all further filter settings are not taken into consideration.

The basic procedure for filtering projects is discussed below:

- All set filters are checked in the first step. For calculation this means that additional filter settings, apart from those under the planning state, could limit the selection from the very start.
- The specifications are checked under the planning state in the second step.

The planning state, creation date, and user are decisive for the display of a version. The entered planning states and higher are taken into consideration for the calculation of the version. In the following example, two users have access to the project:

- JNH as a logged-in user
- DUF as another user

The versions available in the project have different planning states.

The following planning states are created in the project:

- The planning state **Released**, with the highest priority
- The planning state **Integrate**, with the second-highest priority
- The planning state **Working**, with the lowest priority

Input of the planning state for JNH (**Logged-In User**) and DUF (**Other Users**).

**Figure 42: Planning States of Versions**

The following versions are available in the project and are assigned to the respective users (owners). The calculation is executed on the basis of this filter setting. All versions are checked.

**Version 4** has the planning state **Working**. This planning state has the lowest priority and is therefore excluded from the start. Only the three versions 1 to 3 that have the same or a higher planning states are considered for the display of the version.

Version 1 - Planning state Released Owner DUF	Version 2 - Planning state Released Owner JNH	Version 3 - Planning state Integrate Owner JNH	Version 4 - Planning state Working Owner DUF
---	---	--	--

**Figure 43: Versions with Planning State in the Project**

The most current version is displayed. Since all three versions correspond to the filter settings, the version with the most current version number is dis-

played (with the most recent creation date); this is **Version 3**. The planning state of this version has a low priority in comparison to other versions 1 and 2.

When the filter settings are changed for both users to the planning state **Working**, all versions are integrated in the calculation.

**RESULT:** The most current version is displayed, and this is **Version 4**, that of the other user, DUF.

### Table with Examples

Six planning states are created in the project. The numbering of the planning states and the versions corresponds to the priority or the recent version. P1 has the lowest value and P6 the highest. The version numbers 1 to 4 indicates the recent version. The logged-in user JNH selects in all cases the planning state P3 as **Logged-In User** and the other user DUF selects the planning state P5 as **Other Users**. Please refer to the [Table 3](#)

The [Table 3](#) shows some possible cases for the display of versions when the filter settings for the planning state are being used.

In the first three columns of the table, the properties of the version are described. Column four indicates the essential filter settings selected on opening. Column five shows which versions fulfill the selected filter settings.

### Case Criteria

- Under which conditions, versions in the project fulfill the filter settings (column filter settings when opening the project).
- If several versions fulfill the filter settings: which versions are displayed on opening the MCM-project on the basis of the version filter. The displayed versions are marked green in the table.

**Table 3: Display Versions – Important Cases**

Version number	Owner	Planning State	Filter Settings when Opening the Project	Filter Settings Fulfilled
<b>Case 1</b>				
Even though V2 is more current, version V1 is displayed. V2 does not fulfill the filter settings.				
V1	DUF	P6	>= P5 (Others)	fulfilled
V2	DUF	P1	>= P5 (Others)	not fulfilled
<b>Case 2</b>				
Even though V2 is more current, version V1 is displayed. V2 does not fulfill the filter settings.				
V1	DUF	P6	>= P5 (Others)	fulfilled
V2	JNH	P1	>= P3 (Logged-In User)	not fulfilled
<b>Case 3</b>				
Both versions correspond to the filter conditions. In this case the most current version, V2, is displayed.				
V1	DUF	P6	>= P5 (Others)	fulfilled
V2	JNH	P3	>= P3 (Logged-In User)	fulfilled
<b>Case 4</b>				
Three of the versions (V1, V2, and V3) correspond to the filter settings. In this case again the most current version, version 3, is displayed.				
V1	DUF	P6	>= P5 (Others)	fulfilled
V2	JNH	P5	>= P3 (Logged-In User)	fulfilled
V3	DUF	P5	>= P5 (Others)	fulfilled




Version number	Owner	Planning State	Filter Settings when Opening the Project	Filter Settings Fulfilled
V4	JNH	P2	>= P3 (Logged-In User)	not fulfilled
<b>Case 5</b> Two versions correspond to the filter settings. In this case again the most current version, version 2, is displayed.				
V1	DUF	P6	>= P5 (Others)	fulfilled
V2	JNH	P5	>= P3 (Logged-In User)	fulfilled
V3	DUF	P4	>= P5 (Others)	not fulfilled
V4	JNH	P2	>= P3 (Logged-In User)	not fulfilled

### 3.4.2 Using the Product Structure as a Filter

MCM-projects can be filtered according to the product structure.



**Figure 44: Product Structure as Filter Criteria**

In the **Product** tab, **Filter** combo-box contains all the previously defined calculation models and filters.  push button is used to create new filters/calculation models or modify existing ones. The check box **Apply Filtering by Change Orders** activates filtering by change orders and by default this option is un-checked.

When **Apply Filtering by Change Orders** option is checked, only the top level Manufacturing Change Orders (CO) and ECO's that are related to the selected Calculation Model (CM) are considered and the ECO names along with it is the filtering mode and the relation that exists with the CM is displayed, *Please refer to the [Figure 45](#) and [Figure 46](#)*. Move the scroll bars to access all the ECO's and their filtering behavior are shown, if in case there are more number of ECO's.

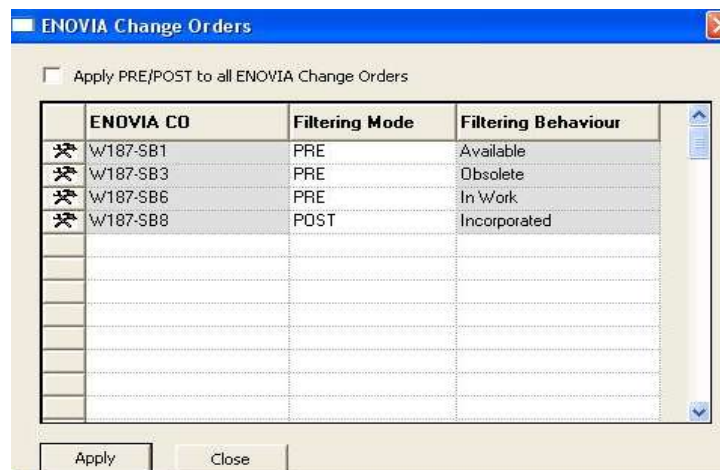


Figure 45: Filtering by ENOVIA Change Orders

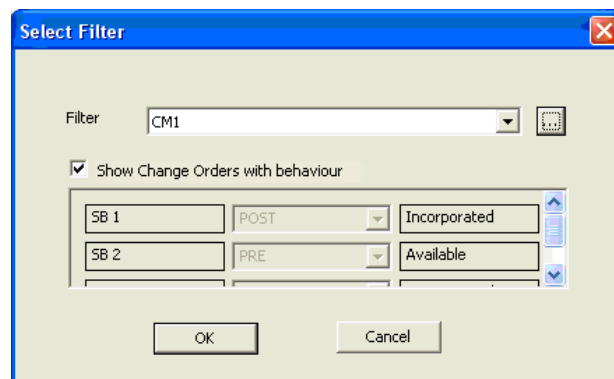


Figure 46: ENOVIA Change Orders with Behaviors

The possible relations that exist between ECO and CM are In Work, Available, Incorporated, Non-Incorporated, Obsolete, and Removed. The filtering mode being applied is dependent on the type of the relation between a CM and ECO. The [Table 4](#) shows the default filtering behavior.

Table 4: Default Filtering Behavior

Relation between ECO and CM	Default Filtering Mode
In Work	PRE
Available	PRE
Incorporated	POST
Non-Incorporated	PRE
Obsolete	PRE
Removed	PRE

Depending on the type of relation that exists between CM and CO, the filtering mode value is set in the combo box adjacent to ECO". You can override this value by selecting a different filtering mode value. You can apply PRE/POST override to all ENOVIA Change Orders by selecting the checkbox "Apply PRE/POST to all ENOVIA Change Orders" in **ENOVIA Change Orders** dialog box.

Once you selects the **Apply Filtering by Change Orders** and opens the project in E5 navigator, filtered product structure is loaded in the E5 Navigator.





The **Apply Filtering by Change Orders** is supported only for MCM-projects, for non MCM-projects **Apply Filtering by Change Orders** is disabled and you cannot check this. In project property dialog you can see only selected filter and filter models you cannot change the filter or filter mode.

### Note

*The last used filter for a project is saved in the manufacturing hub using the session data manager. When you open a DPE project the last used filter is selected.*

*When filter is selected for global, product, process, or resource, it will be stored in registry as last used filter, when you click **OK** and open the project. If you select a filter before clicking **OK**, and then switch to another project in OPD and re-select the previous project with the same filter, it will not be stored in registry as last used filter.*



### Note

*MCO relationship with calculation model for delete relationship type is not displayed in the MCO properties page.*



### Note

*Multiple links between MCM parts and MCOs/ECO's are not considered when using ECO filter dialog. Error message comes if multiple relationship types are displayed.*



### Note

*The relation between a CM and ECO changes over a period of time, this filtering info is not persisted as part of the filter/calculation model. The criteria used for filtering by ECO are remembered in SDM and is restored on subsequent opening of the same project.*

### Enhancement in Apply Filtering by Change Order: New Change Order-Part-Relations

The **Apply Filtering by Change Order** is enhanced by providing the below mentioned five simple relations (and reverse relations) between Base Type “Base Change Order” and “ErgoComponent” (ergocompbase). The **Apply Filtering by Change Order** is enabled for Manufacturing Change Order and ENOVIA Change Order. The filtering behavior is based on the type of these relations. The filtering by change orders is primarily based on the type of the relation between a change order and a part.

- delmia\_add
- delmia\_cut
- delmia\_extend
- add\_rework
- cut\_rework

The filtering mechanism applies the filtering by change orders before any other filtering by effectivity has been applied. The filtering by change orders is primarily based on the type of the relation between a change order and a part. Three different filtering modes are distinguished with five new simple relations.

**Table 5: Filtering Modes**

	PRE	POST	PRE/POST
delmia_add	OUT	IN	IN
delmia_cut	UNKNOWN	OUT	UNKNOWN
delmia_extend	UNKNOWN	IN	IN
add_rework	OUT	IN	IN
cut_rework	UNKNOWN	OUT	UNKNOWN

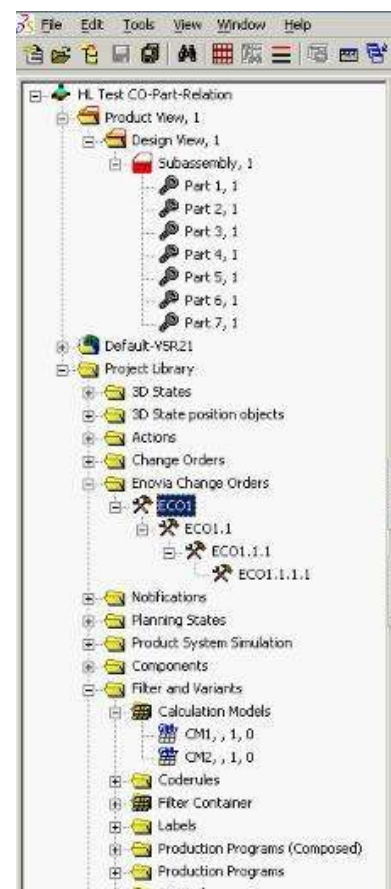
The change orders are considered for the filtering independent of each other, even if they are connected into a deep structure containing several levels.

The filtering behavior and the filtering mode being applied depend on the type of the relation between a calculation model and a change order. The following table shows the default filtering behavior.

**Table 6: Filtering Behavior**

In Work	Available	Incorporated	Non-Incorporated	Obsolete	Removed
PRE	PRE	POST	PRE	PRE	PRE

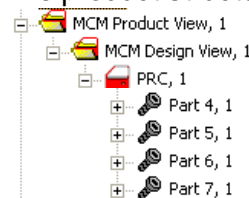
1) Create a product structure in the MCM project as shown below.

**Figure 47: Product Structure for Change Order-Part-Relations**

2) Create extended effectivity (NULL) for Part 1, 2, and 3, and create extended effectivity R(1-9999) for Part 4, 5, 6, and 7.

- 3) Create two Calculation Models as CM1 and CM2 and enter the filtering criteria R(1-oo) for extended effectivity.
- 4) Create ENOVIA Change Orders and link CM1 to ECO1 and ECO1.1.1.1 with InWork relation and CM2 to ECO1 and ECO1.1.1.1 with Incorporated relation.
  - Link Part 1 to ECO1 and ECO1.1.1 with the new “delmia\_add” relation.
  - Link Part 2 to ECO1 and ECO1.1.1 with the new “delmia\_extend” relation.
  - Link Part 3 to ECO1 and ECO1.1.1 with the add\_rework relation.
  - Link Part 4 to ECO1 and ECO1.1.1 with the new cut\_rework relation.
  - Link Part 5 to ECO1 and ECO1.1.1 with the new delmia\_cut relation.
- 5) Save and close the project
- 6) Open the project by select the Calculation Model **CM1** as product filter.
- 7) Toggle the **Filtering by Change Orders** on and apply default incorporation relation (PRE) to all change orders.

The product structure is filtered

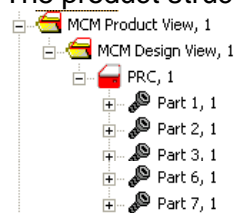


**Figure 48: Product Structure with PRE Filtering Mode**

The Part 1, 2, and 3 are filtered out because the filtering mode is PRE and Part1 has relation delmia\_add; Part2 has relation delmia\_extend and effectivity (null).

- 8) Reopen project under Calculation Model **CM2** as product filter.
- 9) Toggle the **Filtering by Change Orders** on with default incorporation relation (POST).

The product structure is filtered



**Figure 49: Product Structure with POST Filtering Mode**

The Part 4 and 5 are filtered out because the filtering mode is POST and Part4 has relation cut\_rework and effectivity R(1-9999), Part 5 has relation delmia\_cut and effectivity R(1-9999).

- 10) Reopen project under Calculation Model CM1 as product filter.
- 11) Toggle the Filtering by Change Orders on with default incorporation relation (PRE/POST).

The product structure is filtered displaying all the parts.



**Figure 50: Product Structure with PRE/POST Filtering Mode**

The Part1 has relation delmia\_add; Part2 has relation delmia\_extend; Part3 has relation add\_rework; Part4 has relation cut\_rework and effectivity R(1-9999); Part5 has relation delmia\_cut and effectivity R(1-9999).

### 3.4.3 Improving Performance in GetChildren() Server API with Implicit Filtering

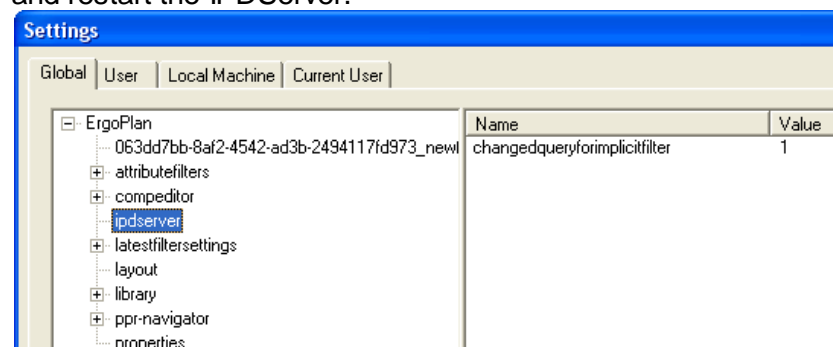
The performance of GetChildren() server API is improved with Implicit Filtering. Previously GetChildren() server API takes more time with implicit filtering as all relation types were queried irrespective of the property on relation types to validate the data object against implicit filtering by Link Activated. But now only those relations are considered which has the property **Consider for Activation by Link** set to **Yes**.

- 1) Create the MCM project using ACTION A1 and MOD M1.
- 2) For all the Workplan and Operation set the attribute considerforactivationbylink to **Yes** in Configuration tool. The relation proc\_firstprocesses\_prod and nodes are configured for the property **Consider for Activation by Link** to **Yes**.

Is configured	No
Is a simple relationship	No
Must have unexposed object for source	No
<b>Consider for Activation by Link</b>	<b>Yes</b>
Exclude from Manufacturing Change	No
Must have unexposed object for target	No
Defined by	Delmia

**Figure 51: Set Property “Consider for Activation by Link” to Yes**

- 3) Set the global key “IPDServer\changedqueryforimplicitfilter” to the value 1 and restart the IPDServer.



**Figure 52: Set Global Key**

- 4) Open the project with implicate filter and expand the “Process Plan” after opening the project.

The Process Plan expansion displays around 40 to 50% performance improvement for implicit filtering.

## 4. Versioning

### 4.1 Versioning in PPR Components

Versions can be created only if the planning state that is assigned to a versionable PPR component is permitted.

In order to create versions of PPR-components, this option may not be activated for a planning state. *Please refer to the [Figure 4](#).*

Versionable components can be edited in two ways:

- Versioning for maximum configuration of the PPR-components, *Please refer to the [Versioning for Maximum Configuration of the PPR-Components](#).*
- Versioning for standard configuration of the PPR-components, *Please refer to the [Versioning for Standard Configuration of the PPR-Components](#).*



#### Caution

*You cannot switch between maximum configuration and standard configuration in the project.*

You can create versions for PPR-components only if the selected action has the action state that corresponds to the planning state **Working**.



For information on versioning, *please refer to the [PPR Navigator Manual](#).*



#### Note

*In the planning state you can decide whether further versions can be created for PPR-components that have this planning state. You can generally allow versions to be created for every planning state.*

*A sensible method would be to allow a versioning only for PPR-components which are assigned the planning state with the highest priority – for example, in the planning state Released.*

*When creating versions you should pay attention to the consistent transparency of the planning. If there are too many versions with the same PPR component and different planning states at the same time, this could lead to the planning losing transparency.*

When creating versions of the PPR-components, the following rules must be observed:

- A dedicated action is required for every version of a PPR component. You can use previously existing actions or create new ones for this purpose.
- You can create versions only if the selected action has the action state Working. All PPR-components that are assigned to this action have the planning state **Working**.
- The planning state must allow the versioning of PPR-components. In order to maintain clarity, versions should be made possible for PPR-components only in the planning state **Release**.

**Applicable to the MCM-project:** Only the versionable PPR-components that are assigned to the selected action can be edited. Versions of the same PPR-components can be assigned to different actions.

The following scheme shows the principle functioning when creating versions. Please refer to the [Figure 53](#).

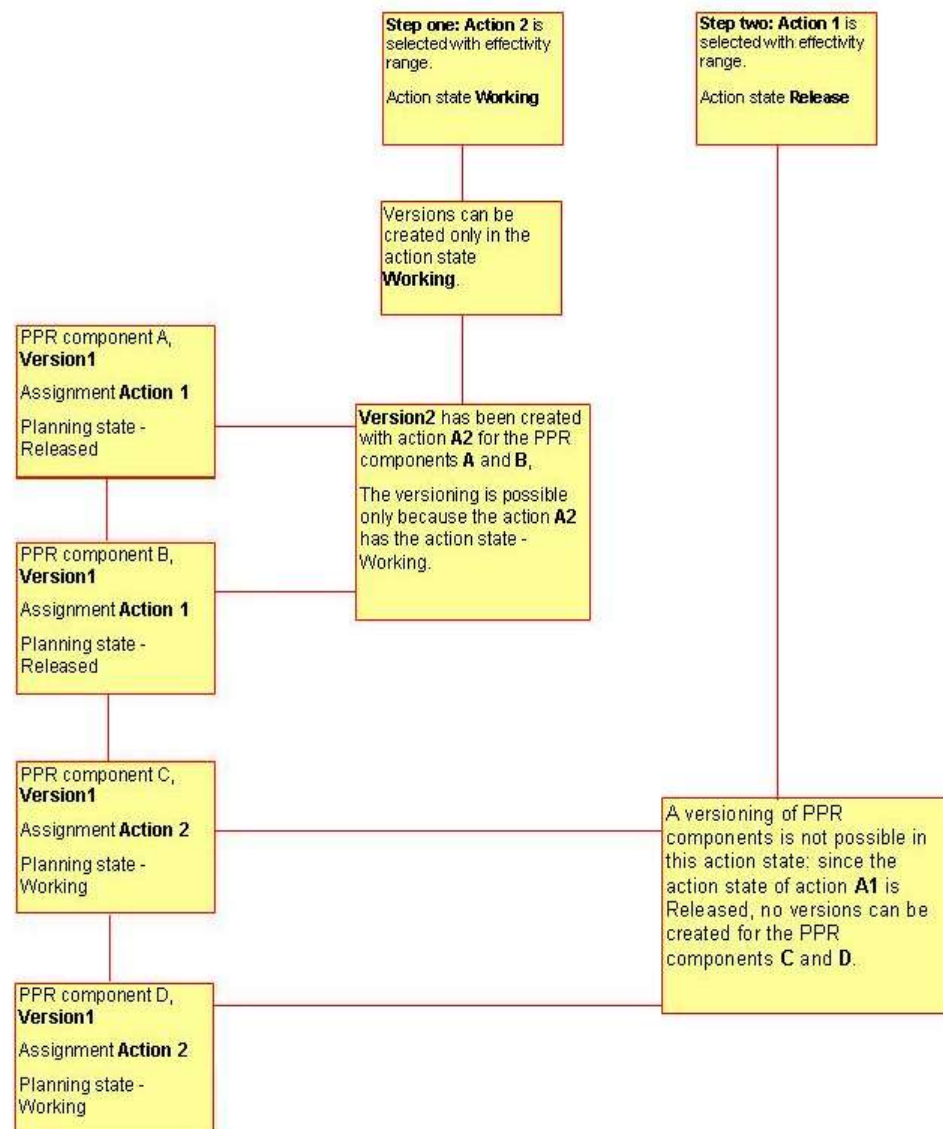


Figure 53: Scheme – Functional Mode when Creating Version

#### 4.1.1 Versioning for Maximum Configuration of the PPR-Components

The versioning of PPR-components under maximum configuration must be taken into consideration especially considering the effectivity of versions. Please refer to the [IPD Server Manual](#).



##### Note

*Parallel versions cannot be created for PPR-components for the editing of MCM-projects with the maximum configuration.*

##### Effectivity of the Versions

The following principle applies to the following:

- A version is always or never valid



- Only one version can be valid for the same planning state, all other versions with the same planning state are set as invalid. This is displayed in the properties dialog under the tab **Effectivity** < **Extended Effectivity**.

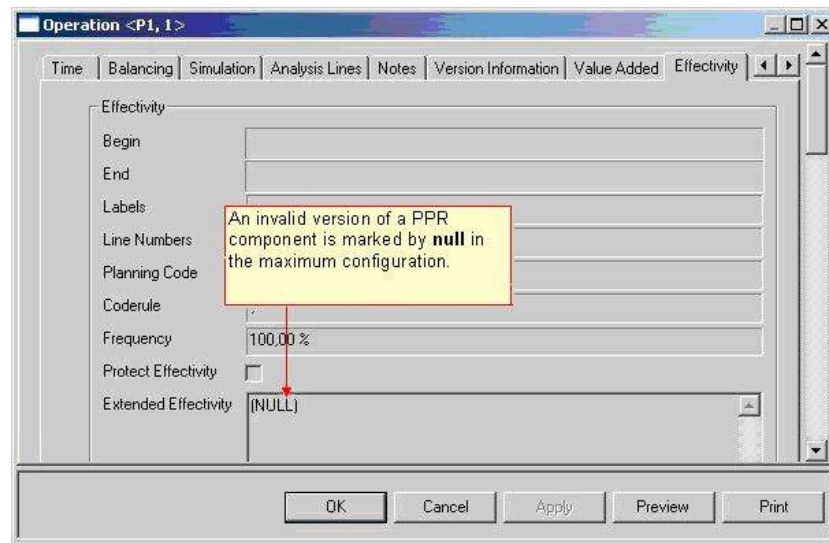


Figure 54: Version is Invalid

### Example

#### Only the most current version of the same planning state is valid

The four versions of the components P1, 1, P1, 2, P1, 3, and P1, 4 have all been created with their own actions. In order to demonstrate this behavior with a simple example, other versions can be created for both planning states **Released** and **Working**. Switching to a higher planning state should be seen as related to switching the action state. The action state corresponds to the respective planning state of the four versions shown in the example.

#### Initial Situation



Figure 55: Planning State Released

Planning state priority:

- Highest priority – Planning state Released
- Second priority – Planning state Integrate
- Third priority – Planning state Checking
- Fourth priority – Planning state Working



In the example, the four versions of the PPR component P1 are marked by four different planning states. The four versions should still be in the stage of the initial situation.

If you promote **Version 4** of the PPR component P1 directly to the planning state **Release**, you will arrive at the same result as if you executed the promotion step by step. Please refer to the [The Program Logic with an Example](#).

### The Program Logic with an Example

The program internally checks all three available previous versions and their planning state. While checking, all priority levels of the individual planning states are taken into consideration, and they are successively and consecutively checked according to their priority. Only the intermediate planning states are taken into account: **Checking** and **Integrate** in the example.

- 1) The direct **previous version 3** has the next higher planning state with the planning state **Checking** in the example. The program recognizes that the planning state was promoted for the successor version 4, and it sets Version 3 as invalid.

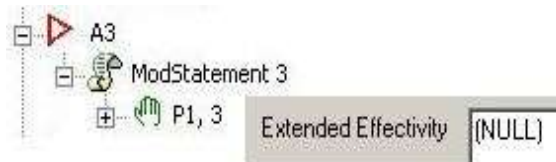


Figure 58: Version 3 is Invalid

- 2) In the next step the planning state **Integrate** (which in terms of priority lies between the two planning states **Checking** and **Release**) of Version 2 is checked. Since it is lower, Version 2 is set as invalid.

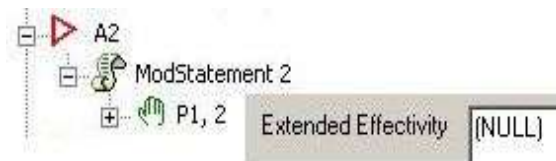


Figure 59: Version 2 is Invalid

The internal check in our example is now complete since all of the intermediate planning states have been checked. Versions **one** and **four** of the PPR component P1 have the same planning state **Release**. Now the rule that only one version of the same planning state can be valid becomes effective here. The most current version of the same planning state is always valid.

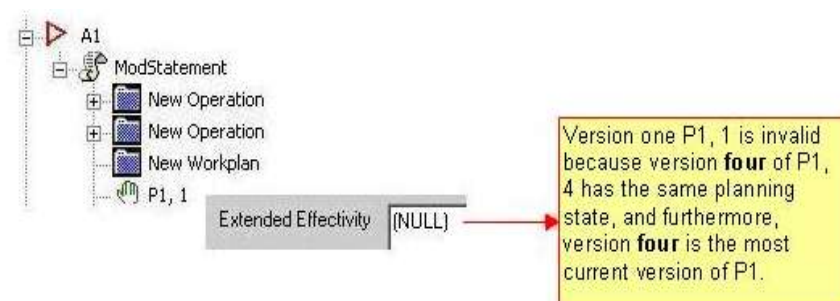


Figure 60: Version 1 is Invalid

### Result

Only version 4 is valid. The planning state of the three previous versions is not changed; the three previous versions are merely marked as invalid versions.



### Important

Thus the rule always applies: If the most recent version has the planning state with the highest priority, all previous versions are invalid.

	Process Name	Version Number	Planning State
	P1	1	Released
	P1	2	Integrate
	P1	3	Checking
	P1	4	Released

The planning state remains the same for version 1-3. Only version 4 is valid since this is the most recent version of P1.

Figure 61: Overview of Valid Version 4

## 4.1.2 Versioning for Standard Configuration of the PPR-Components

The versioning of PPR-components under standard configuration must be taken into consideration especially considering the defined effectivity ranges of versions.

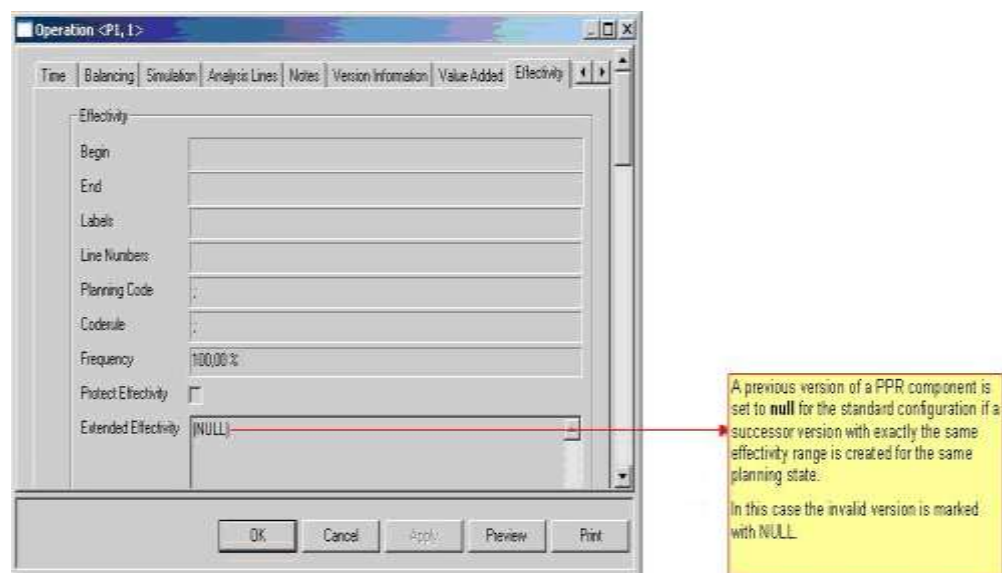


Figure 62: Version is Invalid

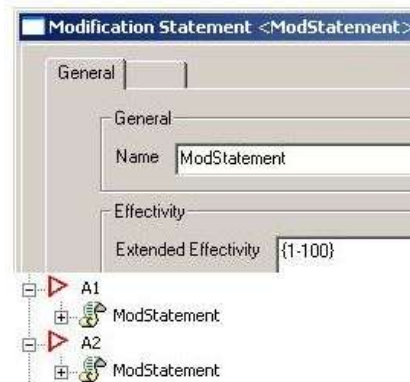
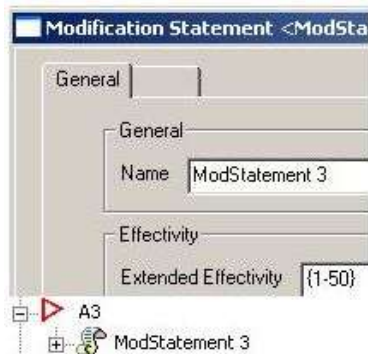
### Effectivity of the Versions

The following principle applies to the following:

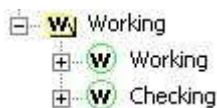
- A new version is created with the effectivity of the effectivity range.
- There can be several versions of the same PPR component for the same planning state if different effectivities are set for the individual versions.

Only one version can be valid for an effectivity range of the same planning state. The definition of an effectivity of versions is shown with an example of a PPR component.

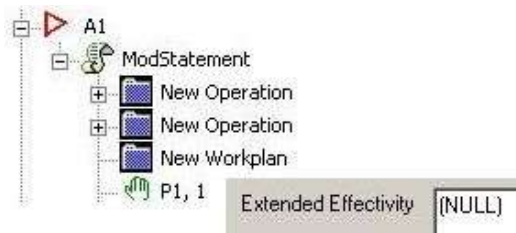
In the first step, the three actions in the MCM-project are set with different effectivity ranges: Actions A1 and A2 should have an effectivity range of 1–100 in the example. Action A3 is created with the effectivity range of 1–50.

**Example****Effectivity Range for Actions A1 and A2****Figure 63: Effectivity Range 1-100 – A1, A2****Effectivity Range of Action A3****Figure 64: Effectivity Range 1-50 – A3****Example**

In the example, both planning states **Working** and **Checking** are used.

**Figure 65: Planning State for the Example**

- After the actions are defined, version1 of process P1, 1 is created with the help of action A1.
- In the next step, the action state of A1 is set to Checking. Version 1 of P1 now also has the planning state Checking.
- In order to create a second version of P1, select action A2.
- After the second version of P1 is created, promote the action state of A2 to Checking. The second version of P1 now also has the planning state Checking.
- The result of promoting the action state of A2 is that version **one** and **two** have the same effectivity and the same planning state. Thus the original effectivity of version 1 is limited by the effectivity of version 2, and for this reason it is set to null: version 2 of P1 is valid.



**Figure 66: Version 1 is invalid**

- In order to create a third version of P1, select action A3.
- After the third version of P1 is created, promote the action state of A3 to Checking. The third version of P1 now also has the planning state Checking.

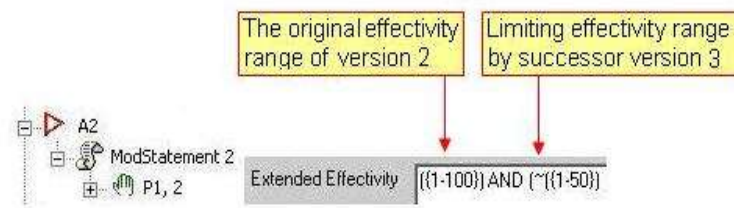
### Valid Versions for Planning State

Here the rule applies that several versions can be valid for the same planning state as long as different effectivities are set for the individual versions.

Version 2 and 3 has a different Effectivity Range. Version 3 has an effectivity range 1-50 and version 2 has an effectivity range of 1-100. In this case the two effectivity ranges overlap.

Both Versions are Valid. The original effectivity range of version 2 is limited by effectivity range of version 3.

- Effectivity of version 2 is henceforth = 51-100.



**Figure 67: Version 2 – Valid for Effectivity Range 51-100**

- Effectivity of version 3 stays the same = 1-50



**Figure 68: Version 3 – Valid for Effectivity Range 1-50**

## 4.2 Working with Versions

### 4.2.1 Creating Versions with Help of Properties Dialog



With help of button **New Version** you can create directly from the properties dialog a new version. New created versions are shown immediately.

- 1) In order to display the button New version in properties dialog you set the value to 1 in menu **Tools < Settings < Maintenance Tool < Local Machine < Versioning < activate notification mechanism**.



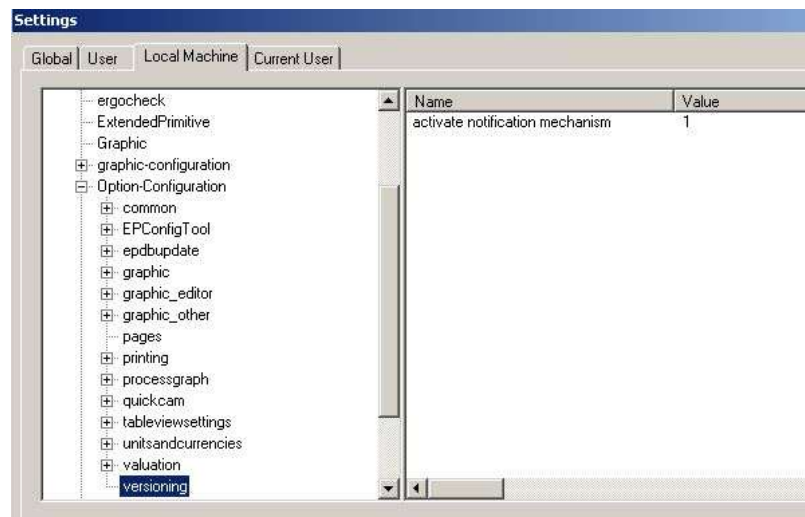


Figure 69: Set Value on 1

### Prerequisites

- The PPR-component for which you want to create a new version has to be in a planning state which allows versioning of PPR-components. In the normal case it is planning state **Released**.
- The function with that the versioning should take, has to be in action state **Working**.

In the example new versions are created for both process components with the help of this button:

- 2) Create a function with PPR-component and increase action state to **Released**.
- 3) Select a new function which is in action state **Working**.
- 4) Create versions with the function and with help of button **New Version** – in the example there are two versions.

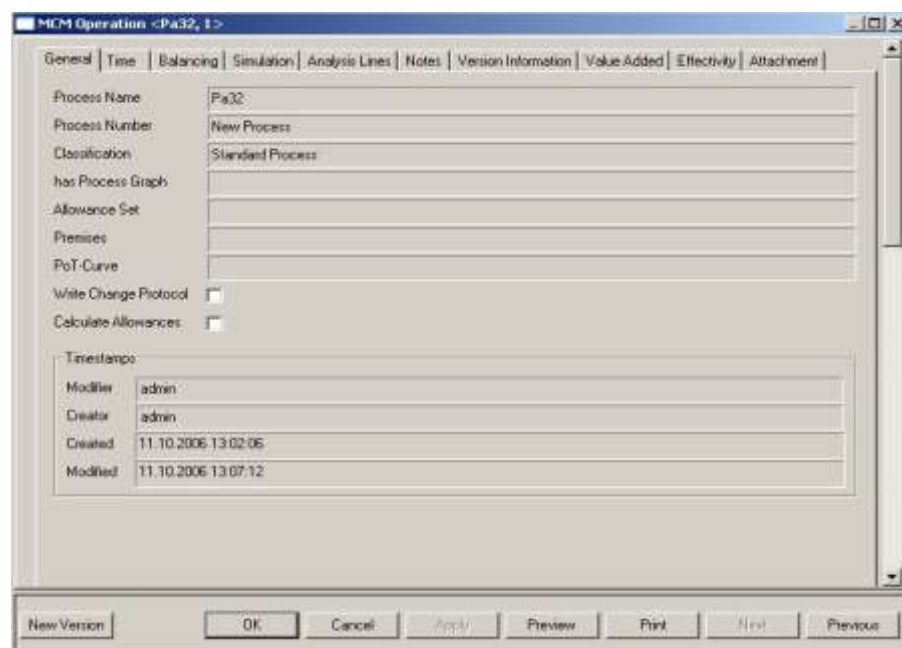


Figure 70: Create New Version

- 5) After you have created the versions they are displayed immediately.



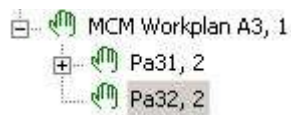


Figure 71: New Versions are Displayed Immediately



### Note

*sa\_newversion script action is now supported for “New Version” button in properties dialog.*

*BEFORE – when script action property is set to Before, script executes before invoking new version property page.*

*AFTER – Script executes when new version property page is closed by “OK” click.*

## 4.2.2 Deleting Versions

Only versions that do not exist for successor versions can be deleted. Versions can be deleted in the PPR-Navigator and in the Versions dialog.



### Note

*Deleting versions is only allowed if the version to be deleted has the planning state Working. Pay attention to the fact that you can delete only versions that belong to the selected action.*

There are two basic situations concerning the deleting of versions:

- [Deleting in the PPR-Navigator](#)
- [Deleting in the Versions dialog](#)

When deleting versions in the **Versions** dialog, the deleted version is completely deleted from the database. There is no need to differentiate between maximum configuration and standard configuration when deleting in the PPR-Navigator. When deleting versions in the PPR-Navigator, only their effectivity ranges are invalid.

### Deleting in the PPR-Navigator

You can delete versions via the menu item **Delete** in the context menu.

- 1) In the PPR-Navigator open the context menu of the selected version.
- 2) Select **Delete**.

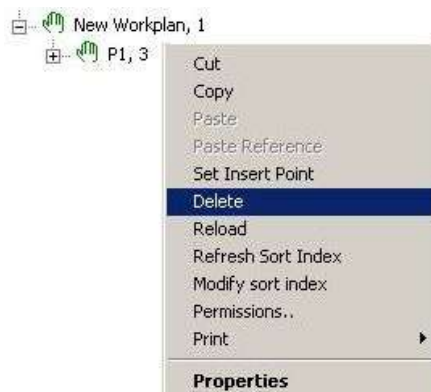


Figure 72: Deleting in the PPR-Navigator

### First Case – No Further Versions Available

The first case could be considered a special case. If only one version of the PPR component that should be deleted is available, this PPR component gets completely deleted.

### Second Case – Other Versions are Available

The assumption here is that there are other versions available. Versions to be deleted are marked as invalid, but they continue to exist in the database. The deleted versions continue to be displayed in the PPR-Navigator if the MCM-project is not opened with a filter.

The deleted version is always displayed in the project library - under the action to which the version belongs and under the plantype.

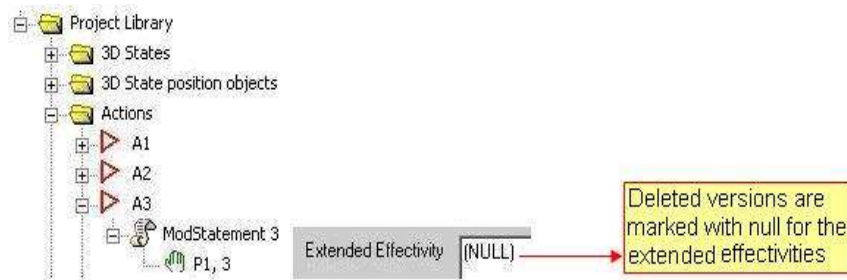


Figure 73: Display of the Deleted Version

### Deleting in the Versions Dialog

When deleting versions in the Versions dialog, the deleted version is completely deleted from the database.

- 1) Open Versions dialog on the selected version in the PPR-Navigator. Select the menu entry **Versions** in the open context menu.
- 2) Select the most current version and open the context menu in the Versions dialog. Select the menu item **Delete**.

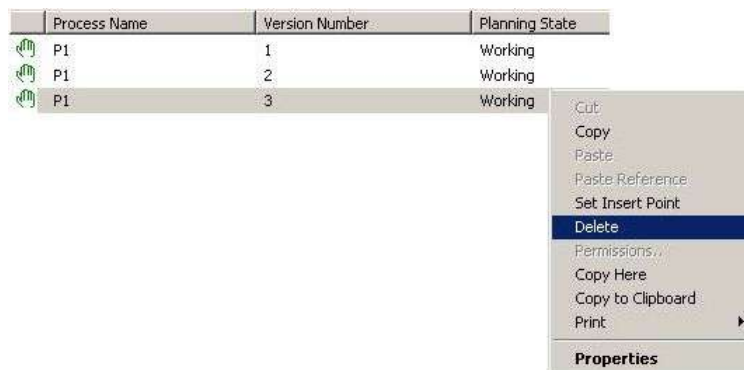


Figure 74: Deleting in the Versions Dialog



For more information on Delete Versions for CCZ Members, please refer to the `DeleteVersionForcedEx()` in [Scripting Manual](#).

## 4.2.3 Versioning the PPR-Components – Copying Relations

When creating new versions for PPR-components, relations can be copied. The attribute **Copy link to child object (change management)** must be set to **Yes** for these relations.

### Relation is Not Copied – Case Example 1

Relation is not copied for the creation of the version

### Example

In this case example, the relation **Process processes Product** is **not** copied for the creation of version 2. The example process is of the plantype Operation.

For the plantype Operation, the attribute **Copy link to the child object (change management)** must be set to **No** for the relation **Process processes Product**.

- 1) In the first step the product **Part one** was linked to the process **P1** (Version1). The link is displayed under the relation **Process processes Product** in the listview of the selected process **P1, 1**.

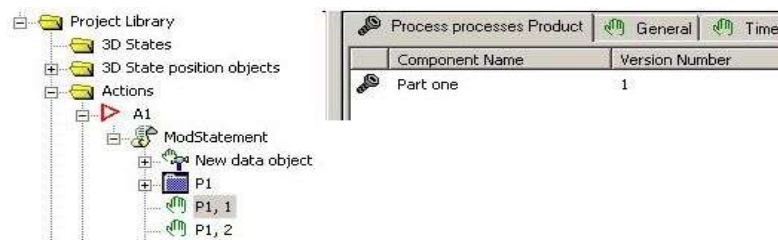


Figure 75: Version 1 of P1 – Display of the Link

- 2) **Version 2** was created for process **P1** after linking. The result is that the relation is **not** copied for **Version 2** of process **P1**. The result is then shown in the listview of the selected process **P1, 2**.

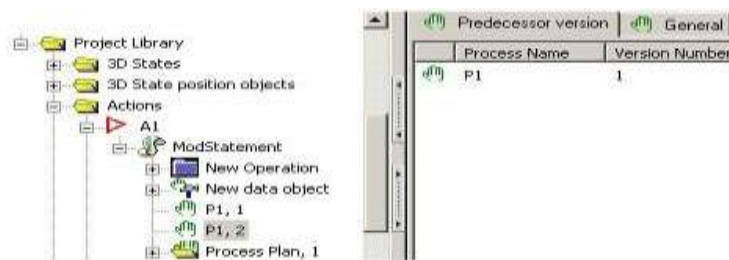


Figure 76: Version 2 of P1 – Relation is not Copied

### Example

Relation is copied for the creation of the version

#### Relation is Copied – Case Example 1

In this case example, the relation **Process processes Product** is copied for the creation of version 2.

**P1, 1 (Version1)** should be linked to the product **Part one** with the relation **Process processes Product**. The attribute **Copy link to child object (change management)** is set to **Yes** in this example.

The relation is copied when **Version 2 of P1** is created. The result is that the link is displayed under the relation in the listview of the selected process **P1, 2**.

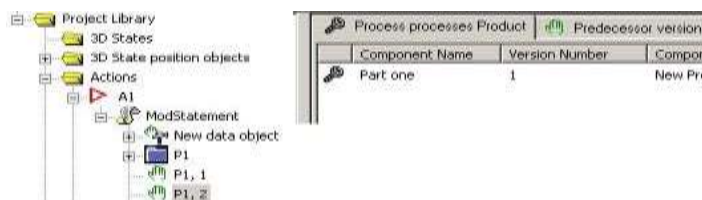


Figure 77: Version 2 of P1 – Relation is Copied

## 4.2.4 Use of Option on Demand

With help of option **On Demand** you can mark relations, which are built in **V5** at a time desired by you.

**Example**

This option is only meant for DPE projects, which are further processed in **V5**. With help of this option different versions of a PPR-component can be edited independently without disturbing each other.

To build the relation between PPR-components you have to execute function **Update Relations in Modification Table** of **V5**.

In configuration tool you activate option **On Demand** directly in the properties editor of the relation: **Copy link to child (Change Management Controlled)**.

For more information, *Please refer to the [Administration Manual](#).*



By Category   Alphabetical	
<input type="checkbox"/> <b>Flags</b>	
Is relevant	No
Is enabled	Yes
Is autorelation	No
Change in integrate state	No
Change in release state	No
Inherit effectivity	Yes
Show in graphic	No
Is unique for parent and child	Yes
Is unique for parent	No
Is autorelationpath	No
Copy link to child	No
Is system internal	No
<b>Copy link to child (Change Management Controlled)</b>	<b>On Demand</b>
Copy link to child (versioning)	No

**Figure 78: Relation – Use of Option on Demand**

## 4.3 Using Relations for Versions

Relations are taken into consideration in a MCM-project only if an owner is specified for the relation under the attribute **Owner type** and if this owner is a versionable component. That is, unless the relation is explicitly excluded by MCM.



For more information on the configuration of relations, *please refer to the [IPD Server Manual](#) and [Administration Manual](#).*

The [Table 7](#) shows which owners can be defined for a relation. Relations are created for every plantype configured for the MCM-project in the directory for parent-child relations in the configuration tool. In a simplified manner one could say that these relations reflect the relations that can exist between PPR-components, such as for father-child relations between hierarchical structure levels or between process and product components. The owner type for a relation must be set on the basis of this relationship. This means nothing more than that if a relation exists between PPR-components, which PPR-components are set as **source components** and **target components**.

You can set the owner type for the relation according to the basic possibilities of these relationships.

**Table 7: Setting Owner for the Relation**

OWNER TYPE	DESCRIPTION
Source Owner	Source PPR component is the owner of the relation.
Target Owner	Target PPR component is the owner of the relation.

OWNER TYPE	DESCRIPTION
Explicit Owner	The owner of the PPR component can be defined individually without restrictions.
CommonParent Owner	The owner is always the direct parent of the linked source and target components.

This property, that the owner can control how the relations are changed, is specified in the general **Configuration Control Zone (CCZ)**. The CCZ is used to ensure that the relation can be edited only if the planning state of the owner allows the editing.

### 4.3.1 Creating and Editing Relations for Versions

Relations can be created and edited for versions of PPR-components.

The **CCZ** is to be taken into consideration for this procedure:

- A relation can have only one owner. An owner of the relation can only be one versionable PPR component.
- The owner type is used to set the relationship between the relation and the owner.
- A relation can be changed only if the owner of the relation (versionable PPR component) has the planning state **Working**.

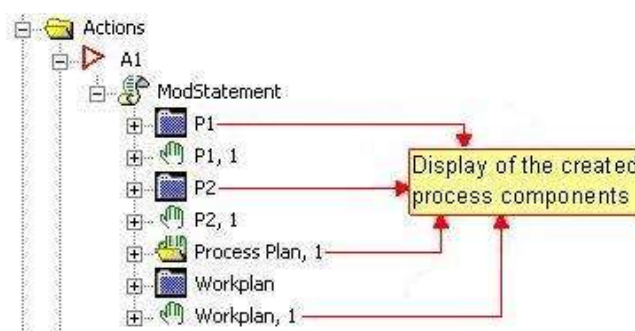
The following examples explain the procedure.

#### Example

The planning state of the **Owner – PPR component** must be **Working** in order for relations to be created

**Sample Case 1 – the owner is the parent node – the owner type CommonParent Owner is specified for the relation**

- 1) An action is required for every version of a PPR component. At least two actions are required in order to clarify the procedure for the example:
  - The process structure is created with the help of action A1.
- 2) After the process structure is created, the action state of **Working** is promoted to **Release**. All process components assigned to action A1 then have the planning state Released. The versioning of PPR-components is allowed for this planning state.



**Figure 79: Example of Process Components – Planning State Released**

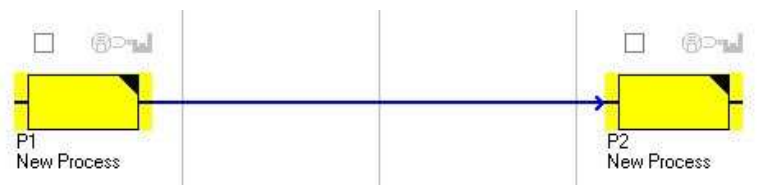
- 3) You must select action A2 in order to create a version. Action A2 must have the action state **Working**.
  - Action A2 is used to create a version and generate a relation.

- In the example the relation **process\_runsbefore\_process** is used; the owner type is **CommonParent Owner**. This relation is assigned to the plantype **Operation**; in our example, the two processes **P1** and **P2**.
  - For a relation for which **CommonParent Owner** is specified as an owner type, the direct parent node comprises the linked PPR-components of the owner, in our example the process **Workplan**.
- 4) To create links between the two processes P1 and P2, the planning state of the owner must be **Working**. The owner is the process **Workplan**.
  - It is therefore completely sufficient if a new version is created for the process **Workplan**.

	Process Name	Version Number	Planning State
	Workplan	1	Released
	Workplan	2	Working

**Figure 80: Version 2 – Planning State Working**

- 5) After the version for the process **Workplan (Version2)** is created, you can create the relation between the two processes P1 and P2. The example also shows that the planning state of the two processes does not play a role. Both processes still have the planning state **Release**.
- 6) In order to create the relation it is necessary to create a process graph for the process **Workplan**. *Please refer to the [Process Graph Manual](#).*
- 7) Switch to the to the process view for the process **Workplan**.
- 8) Open the context menu and select the menu entry **Open Process Graph** in the process view.
- 9) Create the relation between the two processes in the process graph.



**Figure 81: Process\_Runsbefore\_Process**

### Upshot for Sample Case 1

This example shows that if the owner type **CommonParent Owner** is specified for a relation the PPR component that is the owner of the relation must always have the planning state **Working** in order to edit relations.

This example also shows that for the editing of relation in the MCM-project it is sufficient if a version is created for the PPR component that is the owner of the relation. This means that if the versions had been created for both processes you would not be able to create the relation because the planning state of the owner, **Workplan**, would still be released.

### Example

#### Sample Case 2 – Owner is Source – Source Owner is specified as the Owner Type for the Relation

In this example a link between process component **P2** and product **Part one** is created. In order to create a relation, the relation must have the planning state **Working** – in the example, process P2.

The relation **process\_prosesses\_product** is used for the link. **Source Owner** is specified for the attribute **Owner type**.

P1 and P2 have the planning state Released. Only the planning state of the **Owner PPR component** is decisive for the creation of the relation. That is, the planning state Working of the parent node Workplan.



This example also shows that it is possible to create relation between versionable and not versionable PPR-components: In the example, the process components of the plantype **Operation**, **Workplan**, and **Processplan** are versionable components.



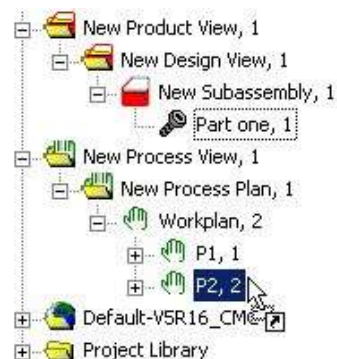
**Figure 82: Process Components**

- 10) At least two actions must be created; in example the actions A1 and A2 are used. Action A1 was used to create the process structure for the previous example. The product structure is not assigned to any action since its PPR-components are not versionable.
- 11) Select action A2 in order to create a version. Action A2 must have the action state **Working**.
- 12) After version two has been created for process P2, 2, it is possible to create the relation between **P2** and **Part one**. No relations can be created between P1, 1 and **Part one** since version one of P1 still has the planning state **Release**.

	Process Name	Version Number	Planning State
	P2	1	Released
	P2	2	Working

**Figure 83: Version 2 of P2**

- 13) Drag the relation between **Part one** and **P2, 2**.



**Figure 84: Relation between Part one and P2, 2**

- 14) Select the relation **Process processes product** in the dialog.



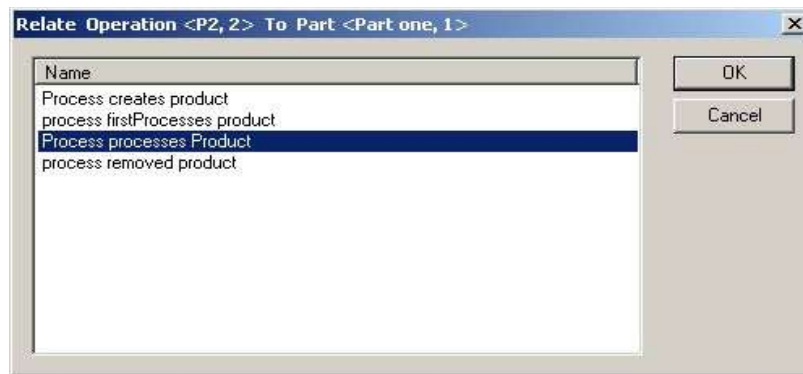


Figure 85: Selecting the Relation

15) Open action **A2** in the project library. The link is displayed in the listview of P2.



Figure 86: Display Link

### Link of P1 and Part one is not Possible

To be brief: the link is not necessary since P1 has the planning state **Released**.

16) If you want to try it anyway, drag the link between Part one and P1.

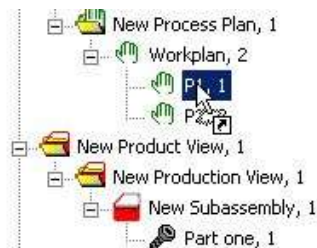


Figure 87: Link between Part one and P1

17) Select the relation **Process processes product**. Confirm the selection with **OK**. You get a notification of the problem instead of a link being created.



Figure 88: Message: Link cannot be Created

### Upshot for Sample Case 2

This example shows on the one hand that if the relation is of the owner type **Source Owner**, relations can be created only if the owner of the relation has the planning state **Working**. And on the other hand that processing with a higher planning state is possible for the owner type **Source Owner** – process P1 has the planning state **Released**.

## 4.3.2 Cloning Relations for Versions

Relations for versions can be cloned when processing MCM-projects to which several users have access. This is the case if the user creates a relation between PPR-components and a previous version, while another user is already processing a successor version. You must use the filter settings for displaying previous versions. *Please refer to the [Using Planning State as a Filter](#).*

- The attribute **Copy link to child object** (change management) must be set to **Yes** for these relations. *Please refer to the [Versioning the PPR-Components – Copying Relations](#) and [Using Relations](#).*
- Only relations that have the owner type **CommonParent Owner** are cloned – for example for the relation **process\_runsbefore\_process** linked to the processes (plantype operation/work sequence) in the process graph.

### Procedure – Sample Case

The cloning of relation for successor versions means that relations that are created between PPR-components are cloned as long as successor versions exist. *Please refer to the [Sample Case](#).*

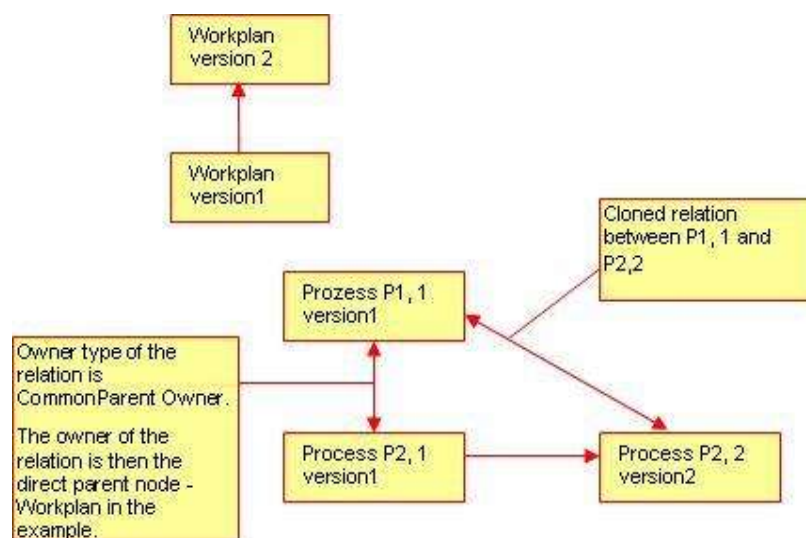


Figure 89: Scheme – Cloning Relations for Successor Versions

### Sample Case

The following sample case shows under which conditions relations are cloned, if, as in the example, two users have access to the MCM-project. In our exam-

ple two users – U1 (administrator) and U2 – are used. In order to demonstrate the procedure with a simple example, user U2 should also have administrator rights.

### Prerequisites

- At least the planning states Working and Released or planning states of the same priority must be created in the MCM-project.
- Two actions (A1 and A2) are created for user U1 in the project.

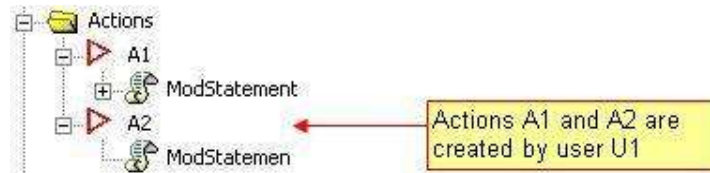


Figure 90: Actions A1 and A2 for user U1

- Action A3 is created for user U2.

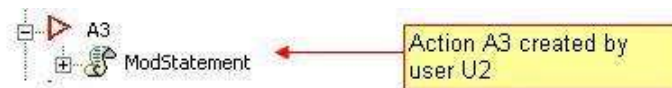


Figure 91: Action A3 for User U2

- Both planning states Working and Released are used as action states. You probably remember that the reference to the respective action state is created via the corresponding action state specified for planning state.

### Step One

User U1 creates the three PPR-components with the help of action A1 in the planning state Working – Workplan, processes P1, and P2.

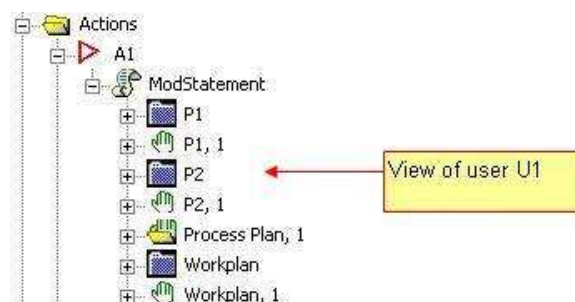


Figure 92: PPR-Components Created with Action A1

### Step Two

In the next step, the action state Working of action A1 must be changed to a higher action state. The valence of the action state changes with the switch to a higher action state; the potential action states must correspond at least to the priority levels Integrate or Released. The action state therefore cannot correspond to the valence of the planning state Working.

In our example the action state of Working is promoted to **Released**.

Action <A1>	
General	
Name	A1
Action Type	
Is CMC Detached Mode	<input checked="" type="checkbox"/>
Group Number	
Action State	Released
Is Usable	<input checked="" type="checkbox"/>

Figure 93: Action State Promoted

### Step Three

In this step, user U1 creates a successor version. A new action must be selected in order to create a version. In our example, action A2 is selected and a version is created for process P2. The action state of action A2 does not need to be changed and it continues to have the action state **Working**.



P2 Version: 2			
	Process Name	Version Number	Planning State
	P2	1	Released
	P2	2	Working

Figure 94: Version 2 of P2 is Created

### Step Four

By following steps **one** to **three** user **U1** has created the structure and version 2 of P2 with actions A1 and A2.

While user **U1** is still working with version **two** of P2, user U2 opens the project in order to continue to work with version one of the processes P1 and P2:

- User U2 wants to create a relation between the two processes P1 and P2.
- Furthermore, user U2 wants to ensure that this relation is formed in the successor version. In our example this would be between P1, 1 and P2, 2.

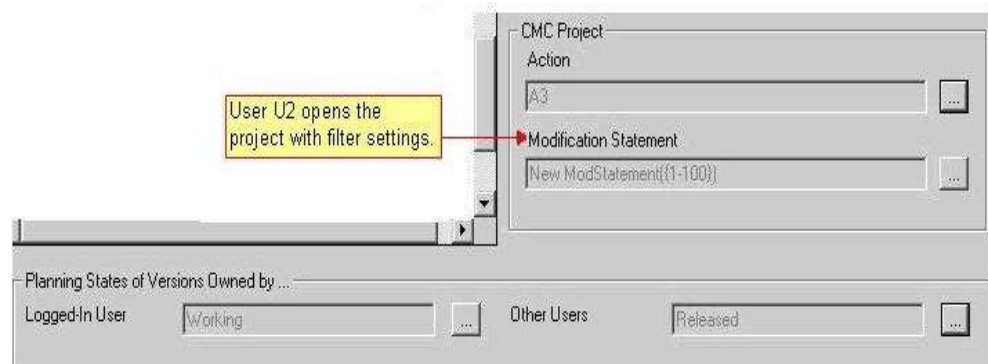


### Note

*Relations can be formed only if they are of the owner type CommonParent Owner.*

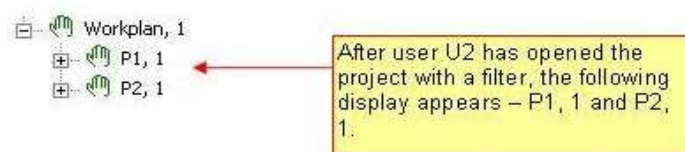
User U2 must pay attention to the following steps:

- The project must be opened with a filter and action **A3** must be selected in our example. Action **A3** must have the action state **Working**.
- The filter settings **Working** and **Released** need to be set in our example.



**Figure 95: Filter Settings – Select Action**

- It is possible to display projects specific to previous versions only if the project is opened with a filter. In our example, this would apply to version 1 of P1 and P2.



**Figure 96: Display of Version One**

The direct parent node must have the planning state Working for the owner type **CommonParent Owner**. In our example it is the parent node Workplan.

- In order to create the relation, user U2 creates version 2 of the parent node with the help of action A3.

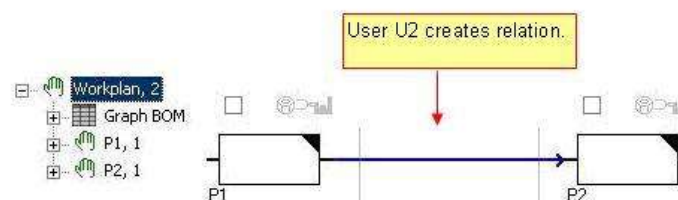
Workplan Version: 2			
	Process Name	Version Number	Planning State
	Workplan	1	Released
	Workplan	2	Working

User U2 creates version two of the parent node (Workplan).

**Figure 97: Workplan Version Two**

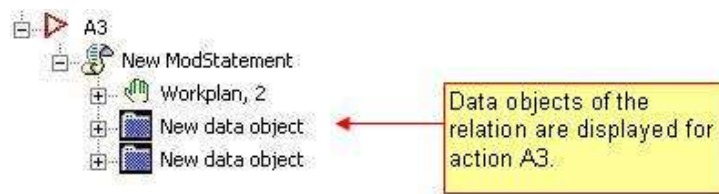
The relation **process\_runsbefore\_process** is used for the example. You can create this relation only in the process graph for the parent node Workplan. Please refer to the [Figure 81](#).

- After user U2 has created the relation between P1, 1 and P2, 1, the relation of P1, 1 to P1, 2 is immediately cloned.



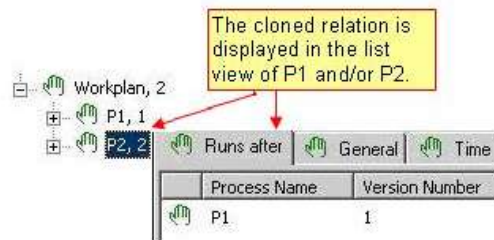
**Figure 98: Relation between P1, 1 and P2, 2 is Cloned**

Display of the cloned relation under action A3:



**Figure 99: Data Objects of the Relation**

- 3) In order to display the cloned relation in the PPR-Navigator, open the project without a filter. The most current version of a PPR component is shown when the unfiltered project is opened – in our example P2, 2 and Workplan, 2.



**Figure 100: Display of the Cloned Relation in the PPR-Navigator**

### 4.3.3 Deleting Relations

Relations are created between PPR-components. Relations can be copied when versions of PPR-components are created.

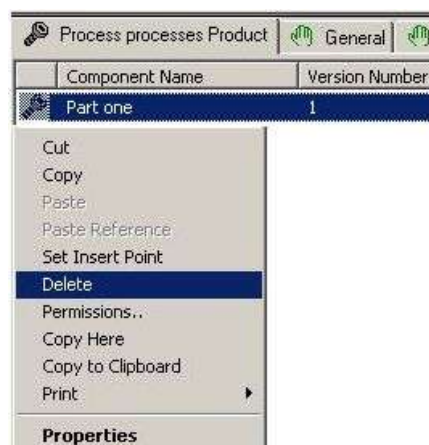


#### Note

*Relations can be deleted only if the user of the relation, i.e. the PPR component, is assigned to the selected action and the action is in the action state Working – all PPR-components assigned to this action also have the planning state Working.*

#### Deleting Relations

Relations are deleted in the listview of the selected PPR-components. You can delete in the PPR-Navigator and in the project library.



**Figure 101: Deleting Relations in the Listview**

There are two basic situations concerning the deleting of relations:

- **Relation is Created with the Selected Action**

If a relation is created with the selected action, this relation will be completely removed from the database upon deletion. The action must always have the action state Working.

#### ▪ Relation is Copied

With regard to all relations copied upon creation of versions: the effectivity range of the relation is restricted when the copied relation is deleted; therefore the relation is invalid for this version.

### Deleting a Copied Relation

#### Note

*Ensure that you select the action with which the owner of the relation is linked. You can only delete relations in the action state Working.*

In the following example, version **one** of the PPR component **P1** and the relation to **Part one** are created with **action A1**.

Version two is created with action **A2**. The relation is copied when version two is created.

Both actions have different effectivity ranges:

- Effectivity range A1 is 1 to 100
- Effectivity range A2 is 50 to 100

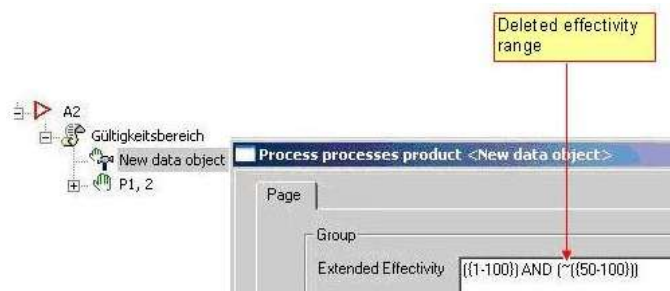
You must select action two when deleting the relation of version **two**. Action A2 must have the action state Working.

#### To Delete a Copied Relation

- 1) Select version two of the PPR component P1.
  - 2) Select the linked object in the listview under the tab of the relation – in the example Part one. *Please refer to the [Figure 101](#).*
  - 3) Open the context menu and select the menu entry **Delete**.
  - 4) A new object is created under action **A2** for the deleted relation after the relation is deleted – properties dialog of the relation.
  - 5) Open the properties dialog. The deleted effectivity range is shown in the dialog.
- In order to display the deleted effectivity range in the dialog, this page must be configured for the dialog.



For more information, *Please refer to the [Administration Manual](#).*



**Figure 102: Properties Dialog of the Relation**

### Completely Deleting the Relation from the Database



The relation to **Part one** is deleted for version **two** of the PPR component. Action A1 must be selected in order to completely delete the relation. The action must have the action state Working.

- 6) After you have selected action **A1**, the relation to **Part one** can be deleted. The relation is completely deleted from the database.
- The complete deleting of the relation is possible only because this relation has been created with action **A1** (see description of first case) and also because no further copied relations to **Part one** exist for successor versions of the PPR component P1.

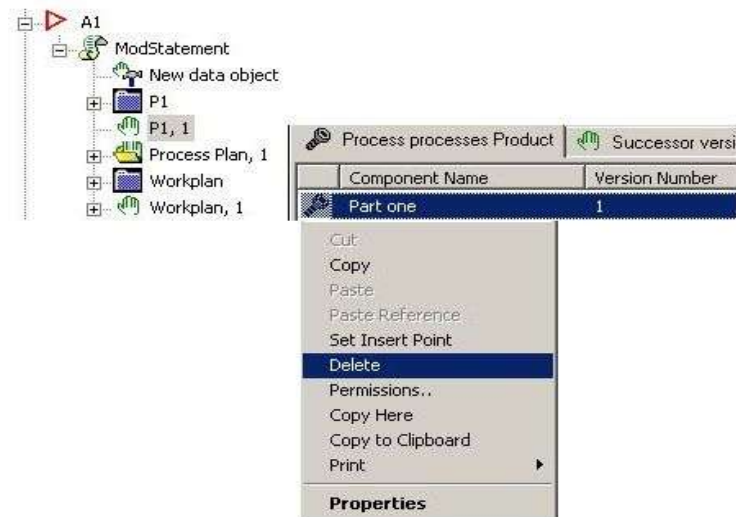


Figure 103: Completely Delete Relation to Part One

#### 4.3.4 Use of Options Owner and CCZ Member for PPR-Components

You can indicate versionable PPR-components in configuration tools as **CCZ Owner**, **CCZ Member** and **NO CCZ Member**. These new options are based on existing versioning mechanism and are an extension of CCZ, in order to be able to control modifications in a structure of individual PPR-components, in which option Owner is set better. Both options can only be configured for types and plantypes (PPR-components).



For more information, *Please refer to the* [Administration Manual](#).

With help of these options you can divide PPR-components in Owner (CCZ Owner) and Members (CCZ Member), and No CCZ Member.

The classification into the groups plays an important role for versioning:

- You can only create versions for PPR-components, which are indicated with option **CCZ Owner** or **No CCZ Member**. With creating a new version of a CCZ Owner, new versions of its **CCZ Members** are created automatically.
- PPR-components which are indicated as **(CCZ Member)** are explicitly linked with a PPR-component **(CCZ Owner)** using the relation nodes.

The [Figure 104](#) points out the relation structure between owner and member by a simple example.

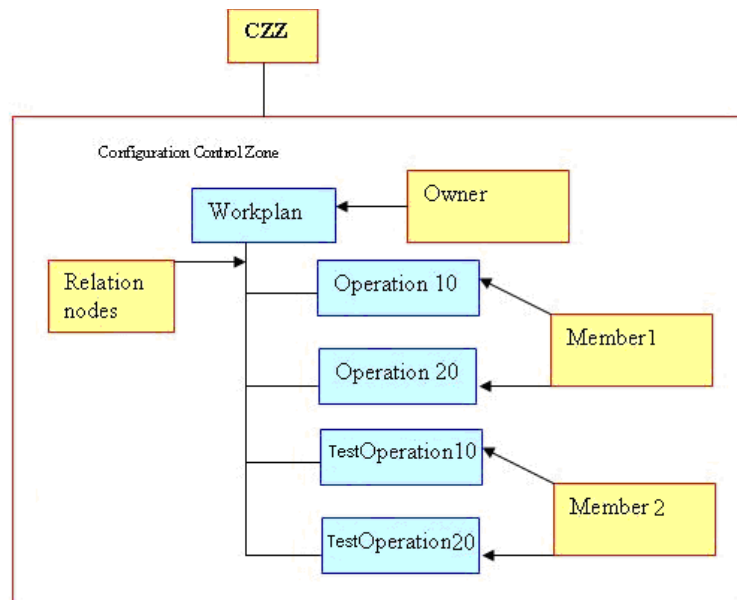


Figure 104: Diagram CCZ

### Activate Options in Configuration Tool

- 1) Activate option **CCZ Owner**, when PPR-component is to be indicated as owner, i.e. Workplan.
- 2) Activate option **CCZ Member**, when PPR-component is to be indicated as member 1, i.e. Operation 10 and 20.
- 3) Activate option **No CCZ Member** for member 2 i.e. TestOperation 10 and 20, when PPR-component is to be excluded from CCZ.



Figure 105: Select Options in Configuration Tool

### Context Menus for Versioning of Indicated PPR-Components

The planttype workplan has two types of children planttypes:

- Operations which are CCZ member
- TestOperations which are not CCZ Members

For PPR-component Operation the option **CCZ Member** is activated. For this PPR-component **no** version can be created (see context menu Operation 10).



**Figure 106: Context Menu – Versioning not Possible**

For PPR-component Workplan option **CCZ Owner** is activated. For this PPR-component versions can be created.

For PPR-component TestOperation, option **No CCZ Member** is activated. For this PPR-component versions can be created.

- You can create versions with help of the options **Check Out**, **Create**, **Check Out (Deep)** or **Create (deep)**.  
 With **Check Out** and **Create** only those children are versioned that are customized as CCZ Member (Operation).  
 With **Check Out (deep)** and **Create (deep)** all children are versioned (Both Operation and TestOperation). This creates versions for the whole structure (independent of CCZ). Please refer to the [Figure 107](#).



**Figure 107: Context menu – Versioning Possible**

For more information, Please refer to the [PPR-Navigator Manual](#).

### Example for Versioning

In this example the behavior is shown, when you create versions of PPR-components.

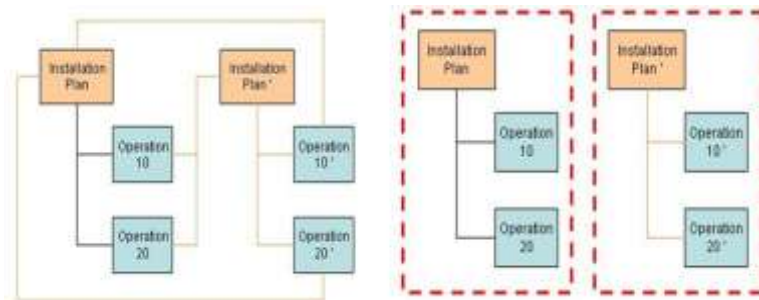


Figure 108: Versioning Behavior

#### Left Side of Figure

In this example the used plantypes are not subject to the CCZ. The PPR-components are also not indicated accordingly with the help of the two options.

For each of the three PPR-components a version is created individually. In sum six new relations are created in this example.

#### Right Side of Figure

In this example plantype **Workplan** (Installation plan) is indicated with option **CCZ Owner** and plantype **Operation** is indicated with option **CCZ Member**.

Versioning is executed on plantype Workplan which is indicated as Owner. For the two children (plantype Operation) versions are created automatically. In sum two new relations are created in this example.

## 4.3.5 Delete Options for MCM Components

- 1) Delete invoked on component which is created in Action/Mod and whose type in configuration has property "CCZ Membership" set to CCZ Owner or CCZ Member.

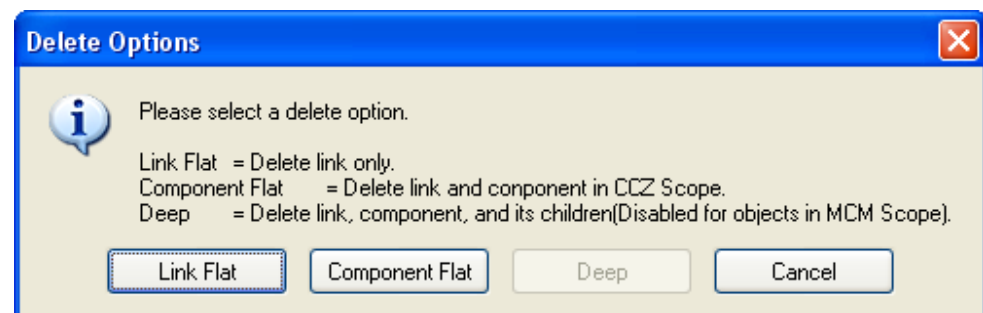


Figure 109: Delete Option1 for MCM Components

- 2) Delete invoked on component which is created and versioned in Action/mod whose type in configuration has property "CCZ Membership" set to CCZ Owner or CCZ Member.

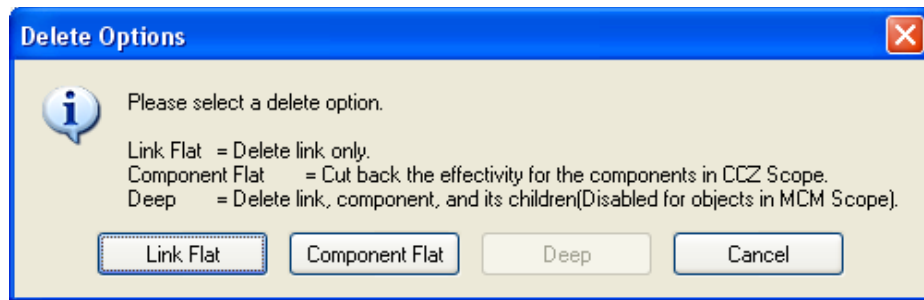


Figure 110: Delete Option2 for MCM Components

- 3) Delete invoked on component which is created in Action/Mod and whose type in the configuration has property “CCZ Membership” set to NO.

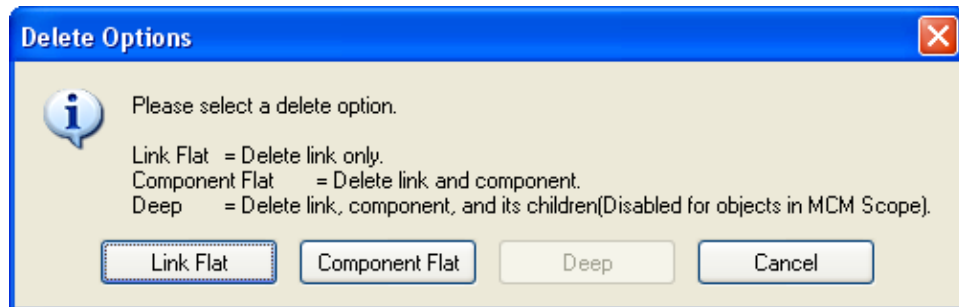


Figure 111: Delete Option3 for MCM Components

- 4) Delete invoked on component which is created and versioned in Action/Mod and whose type in the configuration has property “CCZ Membership” set to NO.

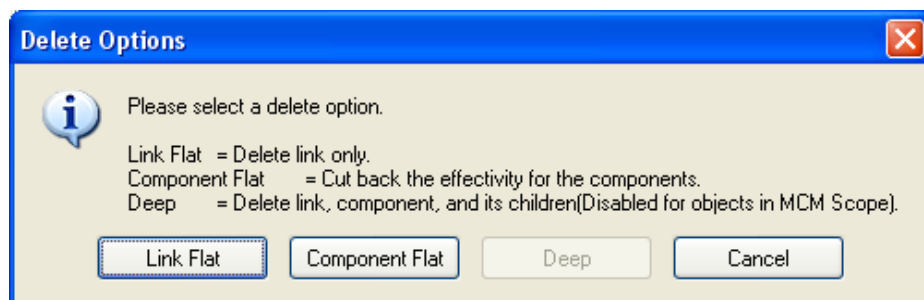


Figure 112: Delete Option4 for MCM Components

### 4.3.6 Creating NewChild on Old Version

A new child under version having successor version can be created with the help of property **Disable auto creation of link to successor** provided on the parent child relation.

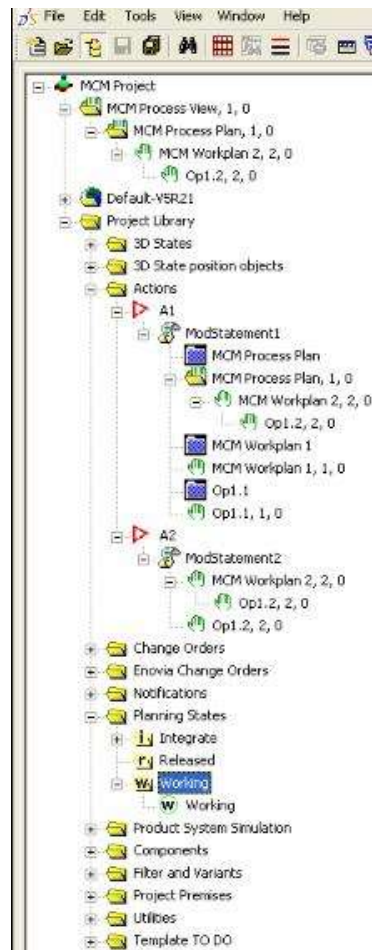
If **Disable auto creation of link to successor** = true then reverse relation to successor is not created by adding a new child to version having successor version.

If **Disable auto creation of link to successor** = false, default behavior prevail. The customization is same for both forward and reverse relation.

- 1) Open the plantype editor on MCMWorkplan and modify the customization for the relation nodes under the type MCMWorkplan and nodes\_reverse under the type MCMOperation as below.  
The owner type = “target owner”  
(for forward “nodes” relationship)

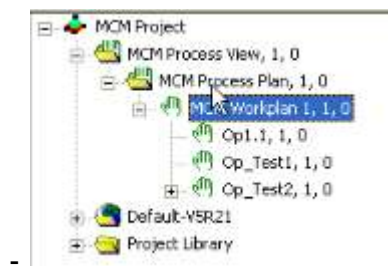
The owner type = “source owner”  
(for reverse “nodes\_reverse” relationship)

- 2) Make sure the types MCMWorkplan and MCMOperation have the property “**Is Configured**” to “**Yes**”.  
Set property **Copy link to child** to **Yes** for both reverse and forward relation.  
Set property **Disable auto creation of link to successor** = **true**
- 3) Create MCM Project using default PTS in detached mode as shown below.



**Figure 113: Product Structure to Create NewChild on Old Version**

- 4) Create Planning State “Working”.
- 5) Create two Actions and Mod-Statements  
A1/ModStatement1 (D (2010-08-10-oo)) and A2/ModStatement2 (D (2010-08-20-oo)).
- 6) Create the second version for MCMWorkplan and MCMOperation using action A2/ModStatement2 (Deep checkout on MCMWorkplan).
- 7) Create a process filter Filter\_PROC with Extended Effectivity Filter String “D (2010-08-10)”.
- 8) Launch DPE and open MCMProject by selecting action A1/ModStatement1 and project filter Filter\_PROC.
- 9) Create new Operation “OP\_Test” under MCMWorkplan.



**Figure 114: New Operation in MCM Workplan**

The new operation “OP\_Test” is created under MCMWorkplan and reverse relation to the successor is not created.



# List of Figures

Figure 1: Extended Effectivity .....	4
Figure 2: Configure Options in Configuration Tools .....	4
Figure 3: Edit MCM-project without Selected Task .....	6
Figure 4: Planning State for MCM-Project.....	7
Figure 5: Exclude from MCM.....	8
Figure 6: Create MCM-Project.....	9
Figure 7: Function Permission MCM-Project .....	10
Figure 8: Mark MCM-Project.....	10
Figure 9: Mark MCM-Project.....	11
Figure 10: Defining Actions in the Process Engineer.....	11
Figure 11: Create New Action.....	13
Figure 12: Define New Action Dialog .....	13
Figure 13: Assignment of Planning State to the Action .....	14
Figure 14: Mark of the Options can be freely Selected.....	15
Figure 15: Create New Modification Statement .....	16
Figure 16: Dialog for a New Modification Statement.....	16
Figure 17: Open MCM Project with predefined Action and Mod Statements .....	17
Figure 18: Action and Mod Statements Name in Status Bar .....	17
Figure 19: ENOVIA Change Orders Folder in Project Library .....	18
Figure 20: ENOVIA Change Orders with Behavior .....	19
Figure 21: License Information .....	20
Figure 22: Open MCM-Project.....	21
Figure 23: Global Key .....	21
Figure 24: Local Machine - Dialogs .....	21
Figure 25: Local Machine - Ergo Plan .....	22
Figure 26: Select Modification Statement.....	22
Figure 27: Message: No Effectivity Range Selected.....	23
Figure 28: Historicized Display of PPR-Components and Relations of an Action.....	24
Figure 29: Open the Context Menu on the Project Node.....	25
Figure 30: Edit Action Proxy/Mod Statement .....	25
Figure 31: Select Modification Statement.....	26
Figure 32: Access Rights Message.....	27
Figure 33: Switch Action Button on Property Dialog .....	27
Figure 34: Actions Organized into Manufacturing Change Orders .....	28
Figure 35: Find Usage Command .....	28

Figure 36: MCM Project “MCM-RecalcABS” in Detached Mode .....	29
Figure 37: “Extendedeffectivity” to Yes under type “compabsoluteposition” .....	29
Figure 38: “Alternativenam” to Yes under type “compabsoluteposition” .....	30
Figure 39: “Extended Effectivity” to Yes under type “ergocompbse” .....	30
Figure 40: “Change in released state” property of “compabsoluteposition” relation to Yes .....	30
Figure 41: Select Planning State .....	31
Figure 42: Planning States of Versions .....	32
Figure 43: Versions with Planning State in the Project .....	32
Figure 44: Product Structure as Filter Criteria.....	34
Figure 45: Filtering by ENOVIA Change Orders .....	35
Figure 46: ENOVIA Change Orders with Behaviors .....	35
Figure 47: Product Structure for Change Order-Part-Relations.....	37
Figure 48: Product Structure with PRE Filtering Mode .....	38
Figure 49: Product Structure with POST Filtering Mode.....	38
Figure 50: Product Structure with PRE/POST Filtering Mode .....	39
Figure 51: Set Property “Consider for Activation by Link” to Yes .....	39
Figure 52: Set Global Key.....	39
Figure 53: Scheme – Functional Mode when Creating Version .....	42
Figure 54: Version is Invalid .....	43
Figure 55: Planning State Released.....	43
Figure 56: Example of Valid Versions for Changing the Planning State .....	44
Figure 57: Display of the Results for Versions of the PPR Component P1 .....	44
Figure 58: Version 3 is Invalid .....	45
Figure 59: Version 2 is Invalid .....	45
Figure 60: Version 1 is Invalid .....	45
Figure 61: Overview of Valid Version 4 .....	46
Figure 62: Version is Invalid .....	46
Figure 63: Effectivity Range 1-100 – A1, A2 .....	47
Figure 64: Effectivity Range 1-50 – A3.....	47
Figure 65: Planning State for the Example.....	47
Figure 66: Version 1 is invalid.....	48
Figure 67: Version 2 – Valid for Effectivity Range 51-100 .....	48
Figure 68: Version 3 – Valid for Effectivity Range 1-50 .....	48
Figure 69: Set Value on 1 .....	49
Figure 70: Create New Version .....	49
Figure 71: New Versions are Displayed Immediately .....	50
Figure 72: Deleting in the PPR-Navigator .....	50
Figure 73: Display of the Deleted Version.....	51

Figure 74: Deleting in the Versions Dialog .....	51
Figure 75: Version 1 of P1 – Display of the Link .....	52
Figure 76: Version 2 of P1 – Relation is not Copied .....	52
Figure 77: Version 2 of P1 – Relation is Copied .....	52
Figure 78: Relation – Use of Option on Demand .....	53
Figure 79: Example of Process Components – Planning State Released .....	54
Figure 80: Version 2 – Planning State Working .....	55
Figure 81: Process_Runsbefore_Process.....	55
Figure 82: Process Components .....	56
Figure 83: Version 2 of P2 .....	56
Figure 84: Relation between Part one and P2, 2 .....	56
Figure 85: Selecting the Relation .....	57
Figure 86: Display Link .....	57
Figure 87: Link between Part one and P1 .....	57
Figure 88: Message: Link cannot be Created .....	58
Figure 89: Scheme – Cloning Relations for Successor Versions .....	58
Figure 90: Actions A1 and A2 for user U1 .....	59
Figure 91: Action A3 for User U2.....	59
Figure 92: PPR-Components Created with Action A1 .....	59
Figure 93: Action State Promoted .....	60
Figure 94: Version 2 of P2 is Created .....	60
Figure 95: Filter Settings – Select Action .....	61
Figure 96: Display of Version One.....	61
Figure 97: Workplan Version Two .....	61
Figure 98: Relation between P1, 1 and P2, 2 is Cloned .....	61
Figure 99: Data Objects of the Relation .....	62
Figure 100: Display of the Cloned Relation in the PPR-Navigator .....	62
Figure 101: Deleting Relations in the Listview .....	62
Figure 102: Properties Dialog of the Relation .....	63
Figure 103: Completely Delete Relation to Part One.....	64
Figure 104: Diagram CCZ.....	65
Figure 105: Select Options in Configuration Tool .....	65
Figure 106: Context Menu – Versioning not Possible.....	66
Figure 107: Context menu – Versioning Possible .....	66
Figure 108: Versioning Behavior .....	67
Figure 109: Delete Option1 for MCM Components.....	67
Figure 110: Delete Option2 for MCM Components.....	68
Figure 111: Delete Option3 for MCM Components.....	68
Figure 112: Delete Option4 for MCM Components.....	68

Figure 113: Product Structure to Create NewChild on Old Version .....	69
Figure 114: New Operation in MCM Workplan .....	70

# List of Tables

Table 1: Behavior of Extended Effectivities – Standard Configuration .....	5
Table 2: Behavior of Extended Effectivities – Max Configuration .....	5
Table 3: Display Versions – Important Cases .....	33
Table 4: Default Filtering Behavior .....	35
Table 5: Setting Owner for the Relation .....	53

# Index

## A

Attributes Exclude ..... 8

## M

### Manufacturing Change Management

Cloning Relations.....58  
 Copying Relations.....51  
 Create Planning State.....7  
 Creating Actions.....12  
 Deleting Relations.....62  
 Deleting Versions.....50  
 Editing MCM-Project.....12  
 Example for Filtering with Planningstate.....33  
 Filter Planningstate for MCM-Project.....31, 34  
 General Information.....3  
 Historical View of the Plannings State .....24  
 Mark MCM-Project.....9  
 Max Configuration.....42

Open MCM-Project .....22, 25  
 Opening a MCM-Project ..... 19  
 Option On Demand.....52  
 Option Owner and Member.....64  
 Planning MCM-Projects ..... 5  
 Preparing MCM-Project.....6  
 Standard Configuration .....46  
 Using Relation ..... 53  
 Using Versions ..... 41

## N

Nonliability ..... ii

## R

### Rules Planning of MCM-Projects

General Information..... 3  
 Temporary Modification Statement..... 4