



HOME

User Manual

DELMIA Process Engineer®

## Administrative Tasks



# Foreword

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.

Nevertheless, there will certainly be some things that we could do even better. If you have any suggestions for improving our software, please be sure to let us know.

We look forward to receiving your constructive feedback. It helps us to make it even easier for you to work with the Process Engineer functions.

The same holds true for the manual that you are now reading. If, at any point when using these instructions, you feel you are not being provided with the clear, unambiguous, and proper guidance necessary to work with this application, please be sure to let us know. We look forward to receiving your comments and tips.

Please feel free to call, send us an E-mail, or contact our user hotline.

## **Please Send your Suggestions to:**

**DELMIA GmbH**

Raiffeisenplatz 4

D-70736 Fellbach

**Phone:** +49/711/27 300-0

**Fax:** +49/711/27 300-599

**E-mail:** [delmia.de.info@3ds.com](mailto:delmia.de.info@3ds.com)

## **User Hotline:**

If you have problems when using DELMIA products, please contact our user hotline at:

**Phone:** +49/711/27 300-400

**Fax:** +49/711/27 300-599

**E-mail:** [delmia.de.support@3ds.com](mailto:delmia.de.support@3ds.com)

## **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 © 2002. All rights reserved.

Some parts of this software are owned by combit® GmbH and are copyrighted. Report-/Print module List and Label® Version 8.0: Copyright combit® GmbH 1991-2001.

**Modifications**

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

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

© 2001-2009 Dassault Systèmes - All rights reserved

Thank you for your interest in our products

**DELMIA GmbH**

Raiffeisenplatz 4

D-70736 Fellbach, Germany

**Phone:** +49 (-400)711/27 300-0

**Fax:** 49/711/27 300-599

# 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 Overview   | 2         |
| 1.4 New Administrator Functions                              | 2         |
| <b>2. General Information</b>                                | <b>3</b>  |
| 2.1 ERGO Components Basic Rules                              | 3         |
| 2.2 Language Settings  | 9         |
| 2.3 Minimum Necessary Access Rights for Users                | 16        |
| 2.4 Saving User Profiles in a CITRIX Environment             | 17        |
| 2.5 Basic Rules for the Configuration of Types and Plantypes | 21        |
| <b>3. Creating Plantype Set and Plantype Structures</b>      | <b>25</b> |
| 3.1 General  | 25        |
| 3.2 Creating Plantype Set Structure                          | 26        |
| 3.3 Creating Attributes, Groups, and Pages                   | 35        |
| 3.4 Configuration Examples                                   | 38        |
| <b>4. The Configuration Manager</b>                          | <b>46</b> |
| 4.1 General Information                                      | 46        |
| 4.2 Opening the Configuration Manager                        | 47        |
| 4.3 Working with the Browser-based Configuration Manager     | 48        |
| 4.4 List of Types  | 50        |
| 4.5 Editing Types  | 55        |
| 4.6 Layout Editing for Pages, Groups and Attributes          | 87        |
| 4.7 Configuration of Parent-Child Relations                  | 92        |
| 4.8 Configuration Samples                                    | 104       |



|   |            |
|---|------------|
| <b>5. Configuration of Work Load Balancing</b>            | <b>118</b> |
| 5.1 General information                                   | 118        |
| 5.2 Opening the Configuration for Work Load Balancing     | 119        |
| 5.3 Adjusting the Plantype Set                            | 122        |
| 5.4 Configuring Work Load Balancing                       | 126        |
| <b>6. Configuration of Automatic Line Balancing</b>       | <b>151</b> |
| 6.1 Individual Configuration (only Version PE 5.10, 5.11) | 151        |
| 6.2 New Installation                                      | 151        |
| 6.3 Preparing ALB plantype Sets                           | 158        |
| 6.4 Update  | 164        |
| 6.5 Opening the Configuration of Automatic Line Balancing | 166        |
| 6.6 The General Settings Tab                              | 167        |
| 6.7 Plantype Forms Tab                                    | 170        |
| <b>7. Settings</b>  | <b>177</b> |
| 7.1 Recommended Settings for the PPR Server               | 177        |
| 7.2 Recommended Settings for the PPR Client               | 177        |
| 7.3 Logging   | 177        |
| 7.4 Expand Action Tree                                    | 184        |
| 7.5 Automatic Termination for Lock and Update Managers    | 184        |
| 7.6 Global Emergency Mode                                 | 185        |
| <b>8. Applying Time Analyses from an Oracle Database</b>  | <b>186</b> |
| 8.1 General Information                                   | 186        |
| 8.2 Extending the Plantype Set                            | 188        |
| 8.3 Attributes of the Process Analysis                    | 188        |
| <b>9. General Information</b>                             | <b>197</b> |
| <b>10. Conversion of ERGOPlan 4 Databases</b>             | <b>198</b> |
| 10.1 Convert Data   | 198        |

|  |            |
|--|------------|
| 10.2 Database Settings                     | 201        |
| <b>11. Cleanup Graphic Directory</b>       | <b>202</b> |
| 11.1 General Comments                      | 202        |
| 11.2 Starting Cleanup Graphic Directory    | 204        |
| 11.3 Server Tools                          | 207        |
| <b>12. Recurring Tasks</b>                 | <b>208</b> |
| 12.1 Database Tools                        | 208        |
| 12.2 Terminating the PPR Server Processes  | 208        |
| 12.3 Terminating the PPR Master            | 209        |
| <b>Appendix</b>                            | <b>210</b> |
| 1. General Information                     | 210        |
| 1.2 Data Types                             | 210        |
| 2. Control Type                            | 211        |
| 3. Data model for Automatic Line Balancing | 213        |
| 4. Description of Relation with Attributes | 219        |
| 5. Release and Location in Documentation   | 223        |
| <b>List of Figures</b>                     | <b>226</b> |
| <b>List of Tables</b>                      | <b>232</b> |
| <b>Index</b>                               | <b>234</b> |

# 1. Introduction

This manual explains how to configure the Process Engineer for your planning purposes. The operations, functionality, and menu layout described in the manual helps system administrators to configure the Process Engineer without any hassles.

## 1.1 How to Use this Manual

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

- The simplified data model of the DELMIA Process Engineer
- Adapting the Software
- Operating the Software



### Note

*When handling the Administrative functions, please remember that there is a general introduction to the Process Engineer in the Basic 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 Overview

The description of the functions is based on a **Standard Configuration of Process Engineer 5.19** The customer-specific application which allows the customer to configure the functions according to his own specific needs is one of the main advantages of the Process Engineer.

### Effects of different configurations of version 5.19

Possible consequences of a customised configuration are, for example, that you deactivate functions which you do not need for your work or that you want to create a completely different user interface for the Process Engineer. The overall functionality and the individual functions of **version 5.19** will be explained in detail based on the standard configuration. This will help you to apply any other configuration quickly and efficiently.

### When should this manual be read?

All functions described below should be, but must not be known to every user. Most of the functions described below have to be executed before starting the actual planning process. Other functions accompany the planning work.

Before you start reading the manual and you are not familiar with older versions of the DELMIA Process Engineer, you should first read the [General Introduction Manual](#), the [PPR Navigator](#), the [Project](#) and [System Library Manuals](#) and the [Settings Manual](#). It is also an advantage to read all the other manuals.

## 1.4 New Administrator Functions

No new functionality has been added for this release.

# A. Adapting the Software

## 2. General Information

This section contains all the information required for adaptation of the software to your running processes (business processes).

This includes the configuration and adaptation of the data model with the aid of the types and plantypes, as well as the settings that the user can make in Process Engineer. Additionally, you will find information for user management.



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

### 2.1 ERGO Components Basic Rules

#### 2.1.1 Base Objects

##### 2.1.1.1 Ergoproject

All objects in a database belong to a project. An exception are those objects which you can find in the library (for example the system items). Against this background it could be assumed that each object knows to which projects it belongs.

This is not always possible, however, since many objects are not assigned an attribute containing a direct reference to the project. This can be remedied however using a script. Using the call



```
Data.GetAttributebyId(object_id, "ergoproject")
```

you will obtain the desired information about every object in the project and the Project Library area.

Using the script from the example below you can verify the aforementioned information.



```
sub main(id)
  project_id = Data.GetAttributebyId(id, "ergoproject")
  name = Data.GetAttributebyId(project_id, "name")
  MsgBox("The project name reads: " + name)
end sub
```

##### 2.1.1.2 Ergocompbase

The **ergocompbase** (XDOErgoCompBase) class is the base class for all ERGOComponents.

Each ERGOComponent

- Assigned to a planning type (ergoplantype)
- Contains a list of ergoitem (XDOErgoItem), for example, ShowGraphics, Cost analysis, Ergonomic analysis etc.
- Contains a list of attributes (XDOAttributeValue)

The attribute list contains for each attribute defined in the configuration for the assigned planning type the current valid value for the ERGOComponent (for example for the attribute “responsible“ the value “Meier”).

**Ergocompbase is an abstract class, i.e. this class does not directly create objects.**

One basic principle in the DELMIA Process Engineer is the organisation of data in a database (**Manufacturing Hub**).

Many parts belong to the product, process and resource areas. The base type of these areas is **ergocompbase**. Of course, products, processes and resources have different properties and are therefore organised in technical and organizational components.

### Technical Components

ergocompproductdefault (XDOErgoCompProductDefault)

ergocompprocessdefault (XDOErgoCompProcessDefault)

ergocompplantdefault (XDOErgoCompPlantDefault)

### Organizational Components

ergocomporgproduct (XDOErgoCompOrgProduct)

ergocomporgprocess (XDOErgoCompOrgProcess)

ergocomporgplant (XDOErgoCompOrgPlant)

## 2.1.1.3 ErgoPlanTypeSet

### 2.1.1.3.1 Plantype Pool

The plantype pool principle is characterized by the fact that the plantypes are no longer administered in the project-specific plantype sets as before, but as a plantype pool in the Project Library that is independent of projects.

Besides the plantypes, this plantype pool contains parent-child relations of the plantypes, too. The plantype pool is referenced by the projects and is **not copied** to the project when a new project is created.

Instead of the plantype sets plantypeproxy sets have been introduced on the project level. They do not contain plantypes, but project-specific plantypeproxies (plantype substitutes) to be able to define user rights with regard to plantypes within a project, for example, and to be able to configure plantype-dependent folders in the Project Library.

### 2.1.1.3.2 Plantype

Each project created within the DELMIA Process Engineer® must be assigned a plantype set. The assigned plantype set defines the project structure according to the included planning types i.e. the three hierarchical levels below the three planning areas, product, process, resource and the arrangement of the hierarchical levels.

Using the parent-child relations, which you can define between the plantypes, you can organise the planning process.

All objects in the Project Library shown at one level as children of a plantype (folders) belong to the **ergocompbase** type. Please refer to the [Figure 1](#).

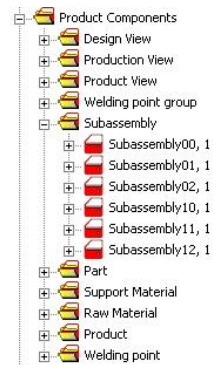


Figure 1: Project Library Section

Therefore Project Library objects are:

|              |                        |
|--------------|------------------------|
| Base Type    | Ergocompbase           |
| Derived Type | Ergocompproductdefault |
| Plantype     | Subassembly            |

**How can you obtain information using the script about whether it is a plantype or a type?**



```
Call
Information = Data.GetAttributebyId(object_id, attribute)
```

Table 1: Attributes and Information

| Attributes    | Information   |
|---------------|---|
| Ergoplantype  | The object ID of the plantype of this base object   |
| Plantype_name | The plantype name of the base object  |
| Plantype_area | The plantype area (product, process or resource) of the base object   |
| Pprtype       | A Whole Number With The Values 0,1, and 2<br>0 = Resource<br>1 = Product<br>2 = Process   |
| Typename      | An unambiguous character combination (GUID), for example "6fff3887-675d-4631-8724-cfc4a1b4792e" which serves for identifying the plantype of the object |

#### 2.1.1.4 Attribute

##### Generic and Non-Generic Attributes

Each attribute is characterised by a special set of attributes. Many attributes are generic. Generic means that the attribute value is stored directly in the database.

Attributes whose value is not directly stored in the database, are non-generic attributes. Non-generic attributes are mostly transient attributes (see below: Persistent and transient attributes).

The attributes **1,2,3** for example have the values **a, b** and **c**.

The sum of attribute **1(a)** and attribute **2(b)** is the same as attribute **3(c)**. Thus, there are two possibilities:

- If the value *c* of the 3<sup>rd</sup> attribute is directly stored in the database, this attribute is a generic attribute.
- If attribute 3, however, is defined by the sum function and by its relations to attributes 1 and 2, then attribute 3 is non-generic.

### What is a configured name or a physical name?

Nearly all attributes are determined in the configuration. Each attribute in the configuration has a **configured name** and a **physical (internal) name**. Exceptions are the transient (temporary) attributes. The configured attribute name is represented by the "m\_name" attribute.

### Persistent and Transient Attributes

Most of the attributes are **persistent**, i.e. their current value is stored in the database and is not subject to changes.

The value of a **temporary (transient)** attribute is stored during the program runtime; this information, however, is lost when exiting the program. The value of a transient attribute is only visible and usable for the current user. Transient attributes can but needn't have a physical name.

#### 2.1.1.5 Inheritance

One principle of modern programs is inheritance. Inheritance means inheriting particular properties from so-called base objects. This can be made clearer using the attribute "name" as an example:

The "name" attribute owns almost every component. If other components inherit this attribute from the base class, then all these components have a "name" attribute. Using **Overwrite** you can change the value of this attribute in the corresponding components, however, you cannot change the attribute itself. This process is called overwriting the basic class attribute.

## 2.1.2 Ergoitems

The objects in the Project Library that do not belong to the PPR area, belong to the **ergoitem** or **defaultimpl** type. They form their own class, additional information and extensions.

An example for an ergoitem is the **Script** or **SA-Codes** type.

#### 2.1.2.1 Referencing or Associating Base Objects

The most important links between plantypes are:

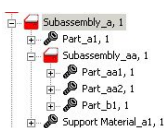
- Subcompitem
- Relationships

## 2.1.3 Subcompitem: relationship\_nodes (XDOSubCompItem)

The class **relationship\_nodes** (internal name XDOSubComp-Item, heading plural) is the basic class for all bill of materials entries (BOMs).

A typical example to explain this term is a subassembly that can consist of parts and of other subassemblies (product, plantype).

The children of a parent are called **subcompitem**. This term is independent of the area.





Do these children also belong to the **ergocompbase** type? They belong to both types.

**subcompitem** and **ergocompbase** are different objects between which a link exists. **subcompitem** and **ergocompbase** have different positions in the PPR Navigator.

As soon as an object from the Project Library is linked to the corresponding Product, Process or Resource View, a **subcompitem** type is generated. If you link a **subcompitem** with more than one parent, you have two **subcompitem**s representing the same **ergocompbase**. The Figure 2 serves to illustrate this process.

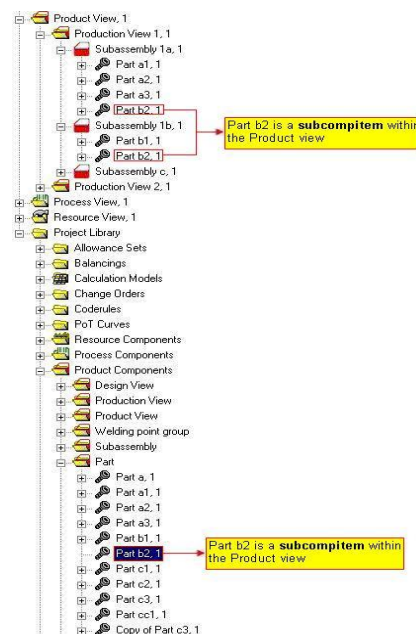
Bill of materials entries of processes and products have additional properties and are therefore classified in their own sub-classes of SubCompItem (SubCompItemPro and SubCompItemProcess).

The basic class of all Ergo Components (ergocompbase) has a list of bill of materials entries. Therefore a bill of materials entry consists of data relevant to bills of materials along with a reference to a ERGOcomponent. The Ergo Components that can be referred to is defined via the plantypes. The Ergo Components can be a part of the same project or a part of the system library. Please refer to the Figure 4.



### Note

*If an attribute of a subcombitem changes, then the attribute in the Project Library changes as well.*



**Figure 2: Relation between ergocompbase and subcompitem**

**How ergocompbase-subcompitem (relationship\_nodes) relations are found using a script?**

Use the call:

```
call base_id = Data.GetAttributebyId(sci_id, "ergocompbase")
```

PE 5.15



```
base_id = Data.GetAttributebyId(sci_id, "relationobject2")
```

On the subcompitem node you will receive the desired ID. When calling an ergocompbase node, you will only receive its ID.



### Note

Since an ergocompbase can have an unlimited number of subcompitem (relationship\_nodes), it is not possible to display them as well. You can have the ID of the parent (not the ergocompbase or *relationobject2*) assigned with the following call.



```
Parent_base_id =
```

```
Data.GetAttributebyId(sci_id,"ergocompbase-parent")
```

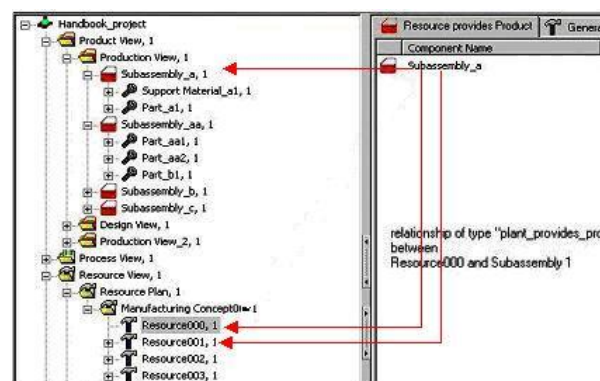
PE 5.15

```
base_id = Data.GetAttributebyId(sci_id, "relationobject1")
```

Using a call as shown in you will receive, for example, the “subassembly aa” ID or the “subassembly b” ID; this depends on which “Part b1” (subcompitem) this script command has been executed.

## 2.1.3.1 Relationship

Relationships (links) are associative objects. A link links exactly two objects from the ergocompbase type. All object links between the PPR Navigator views are relationships as well, for example, between the Product and the Resource Views; these relationships are of the **plant\_provides\_prod** type (resource provides product) or of the **proc\_uses\_plant** type (process uses resource).



**Figure 3: Relations in List View**

In the standard configuration the relations between the objects are not displayed directly. Instead the linked components and their properties are shown in the list view.

The relation between the objects is of the **relationship** type.

The objects linked to this relation are of the **relationobject** type.

Using the script you can obtain the relation name:

```
relationname = Data.GetAttributebyId(rel_id, "relationname")
```

To receive the linked object (relationobject), use the following call in a script:

```
relationobject_1_id = Data.GetLinkedObjectAttributebyId(relationship_id, relationobject_2_id, "oid")
```



In the following figure the object structure is displayed graphically. Connection lines ending with an arrow indicate an inheritance. Thus ErgoCompGraphic inherits all properties from ergocompbase. Please refer to the [Figure 4](#).

### 2.1.3.2 SubCompView

An instance of this class defines a view on the bill of material entry (BOM) of the Ergo Components.

An example is the **BOM view** and the **BOM view** of a graphic element. This class contains a list of instances of the class XDOSubCompViewItem which also belong to the class XDOSubCompViewItemItem or XDOSubCompViewItemItemList. This allows for the creation of any arbitrary Sub-BOM structure. An instance of the class XDOSubCompViewItemItem contains a reference to a BOM entry (XDOSubCompItem) of the Ergo Components.



How to hide the bill of materials entry in the resource view:

- 1) In the Configuration Manager select the ERGOcomponent entry: **ergocompbase** (XDOErgoCompBase)
- 2) In the parent-child relations select the entry: **subcompview:: subcompview** and in the *parent-child information* in the *Tree view* and in the *list* set the *display value* to No.

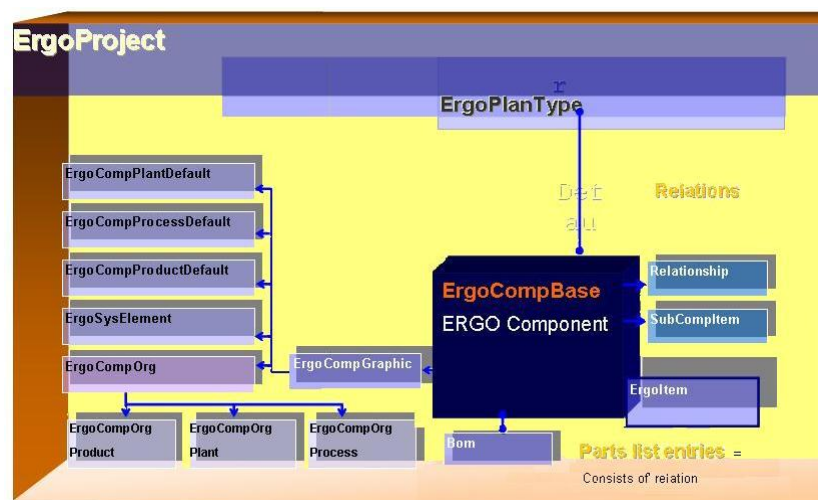


Figure 4: Simplified data structure in the DELMIA Process Engineer®

## 2.2 Language Settings

Language settings are exclusively changed by the administrator with access rights to the server.

For all languages set additionally, a translation file (RLG-file) must be created before.

### 2.2.1 Setting Languages

In Process Engineer German and English are set as standard; you may select these from the general settings. To set another language proceed as follows:

- 1) Edit the language directly in the registration editor, or

- 2) Create a registration file with the file extension **".reg"**. You may create the file in a standard text editor.
- The *Values (valuen)* correspond to the number of languages to be set. The registration file should contain the following.

### Example

#### REGEDIT4

```
[HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\ergoplan\option-configuration\common\languageid]
"default"="1033"
"functiongroup"="0"
"globaldata"="0"
"label"="Select Language:"
"ordernumber"="80"
"pagenumber"="15"
"type"="4"
"value1"="1033|English"
"value2"="1031|German"
"value3"="1036|French"
"value4"="1055|Turkish"
"value5"="1034|Spanisch"
"value6"="1049|Russian"
```

**Figure 5: Registration File with Six Values**

The composition of the value includes the **decimal value** and the **name** of the respective language. In the registration file shown, the character string values *valuen* are identified by the figures 1 to 6. A further language would thus be identified by the string value **7**, etc...

The corresponding values can be extracted from the table below. *Please refer to the [Table 2](#).*

### Example

#### Examples for String Values

*For Afrikaans*

"valuen"="1078|Afrikaans"

*For Greek*

"valuen"="1032|Greek"

### 2.2.1.1 Creating an RLG File

In the RLG file all terms are translated for the respective language. To create an RLG file it must be accessible and identifiable. For identification of the RLG file always use the respective *hex value* of the language listed in the table. This *hex value* is included once at the beginning of the RLG file.

In the example shown the term for Login is translated. The *hex value* for French is 040:

```
040C
french
User authorization
Connector
```

**Figure 6: Example for the Structure of an RLG File**



### Caution

*Contact DELMIA Support if you wish to create RLG files in other languages. You will be advised on which files need to be used.*

**Table 2: Table of Values for Language Codes**

| Local Descriptions            | Short Strings | Hex Value     | Decimal Value |
|-------------------------------|---------------|---------------|---------------|
| Afrikaans                     | af            | 0x0436        | 1078          |
| Albanian                      | sq            | 0x041C        | 1052          |
| Arabic – United Arab Emirates | ar-ae         | 0x3801        | 14337         |
| Arabic - Bahrain              | ar-bh         | 0x3C01        | 15361         |
| Arabic - Algeria              | ar-dz         | 0x1401        | 5121          |
| Arabic - Egypt                | ar-eg         | 0x0C01        | 3073          |
| Arabic - Iraq                 | ar-iq         | 0x0801        | 2049          |
| <b>Arabic - Jordan</b>        | <b>ar-jo</b>  | <b>0x2C01</b> | <b>11265</b>  |
| Arabic - Kuwait               | ar-kw         | 0x3401        | 13313         |
| Arabic - Lebanon              | ar-lb         | 0x3001        | 12289         |
| Arabic - Libya                | ar-ly         | 0x1001        | 4097          |
| Arabic - Morocco              | ar-ma         | 0x1801        | 6145          |
| Arabic - Oman                 | ar-om         | 0x2001        | 8193          |
| Arabic - Qatar                | ar-qa         | 0x4001        | 16385         |
| Arabic - Saudi Arabia         | ar-sa         | 0x0401        | 1025          |
| Arabic - Syria                | ar-sy         | 0x2801        | 10241         |
| Arabic - Tunisia              | ar-tn         | 0x1C01        | 7169          |
| Arabic - Yemen                | ar-ye         | 0x2401        | 9217          |
| Armenian                      | hy            | 0x042B        | 1067          |
| Azeri – Latin                 | az-az         | 0x042C        | 1068          |
| Azeri – Cyrillic              | az-az         | 0x082C        | 2092          |
| Basque                        | eu            | 0x042D        | 1069          |
| Belarusian                    | be            | 0x0423        | 1059          |
| Bulgarian                     | bg            | 0x0402        | 1026          |
| Catalan                       | ca            | 0x0403        | 1027          |
| Chinese - China               | zh-cn         | 0x0804        | 2052          |
| Chinese - Hong Kong S.A.R.    | zh-hk         | 0x0C04        | 3076          |
| Chinese – Macau S.A.R         | zh-mo         | 0x1404        | 5124          |
| Chinese - Singapore           | zh-sg         | 0x1004        | 4100          |
| Chinese - Taiwan              | zh-tw         | 0x0404        | 1028          |
| Croatian                      | hr            | 0x041A        | 1050          |
| Czech                         | cs            | 0x0405        | 1029          |
| Danish                        | da            | 0x0406        | 1030          |
| Dutch – The Netherlands       | nl-nl         | 0x0413        | 1043          |
| Dutch - Belgium               | nl-be         | 0x0813        | 2067          |

| Local Descriptions       | Short Strings | Hex Value | Decimal Value |
|--------------------------|---------------|-----------|---------------|
| English - Australia      | en-au         | 0x0C09    | 3081          |
| English - Belize         | en-bz         | 0x2809    | 10249         |
| English - Canada         | en-ca         | 0x1009    | 4105          |
| English – Carribbean     | en-cb         | 0x2409    | 9225          |
| English - Ireland        | en-ie         | 0x1809    | 6153          |
| English - Jamaica        | en-jm         | 0x2009    | 8201          |
| English - New Zealand    | en-nz         | 0x1409    | 5129          |
| English – Phillippines   | en-ph         | 0x3409    | 13321         |
| English - South Africa   | en-za         | 0x1C09    | 7177          |
| English - Trinidad       | en-tt         | 0x2C09    | 11273         |
| English - United Kingdom | en-gb         | 0x0809    | 2057          |
| English - United States  | en-us         | 0x0409    | 1033          |
| Estonian                 | et            | 0x0425    | 1061          |
| Farsi                    | fa            | 0x0429    | 1065          |
| Finnish                  | fi            | 0x040B    | 1035          |
| Faroese                  | fo            | 0x0438    | 1080          |
| French - France          | fr-fr         | 0x040C    | 1036          |
| French - Belgium         | fr-be         | 0x080C    | 2060          |
| French - Canada          | fr-ca         | 0x0C0C    | 3084          |
| French - Luxembourg      | fr-lu         | 0x140C    | 5132          |
| French - Switzerland     | fr-ch         | 0x100C    | 4108          |
| Gaelic – Ireland         | gd-ie         | 0x083C    | 2108          |
| Gaelic - Scotland        | gd            | 0x043C    | 1084          |
| German - Germany         | de-de         | 0x0407    | 1031          |
| German - Austria         | de-at         | 0x0C07    | 3079          |
| German - Liechtenstein   | de-li         | 0x1407    | 5127          |
| German - Luxembourg      | de-lu         | 0x1007    | 4103          |
| German - Switzerland     | de-ch         | 0x0807    | 2055          |
| Greek                    | el            | 0x0408    | 1032          |
| Hebrew                   | he            | 0x040D    | 1037          |
| Hindi                    | hi            | 0x0439    | 1081          |
| Hungarian                | hu            | 0x040E    | 1038          |
| Icelandic                | is            | 0x040F    | 1039          |
| Indonesian               | id            | 0x0421    | 1057          |
| Italian - Italy          | it-it         | 0x0410    | 1040          |
| Italian - Switzerland    | it-ch         | 0x0810    | 2064          |

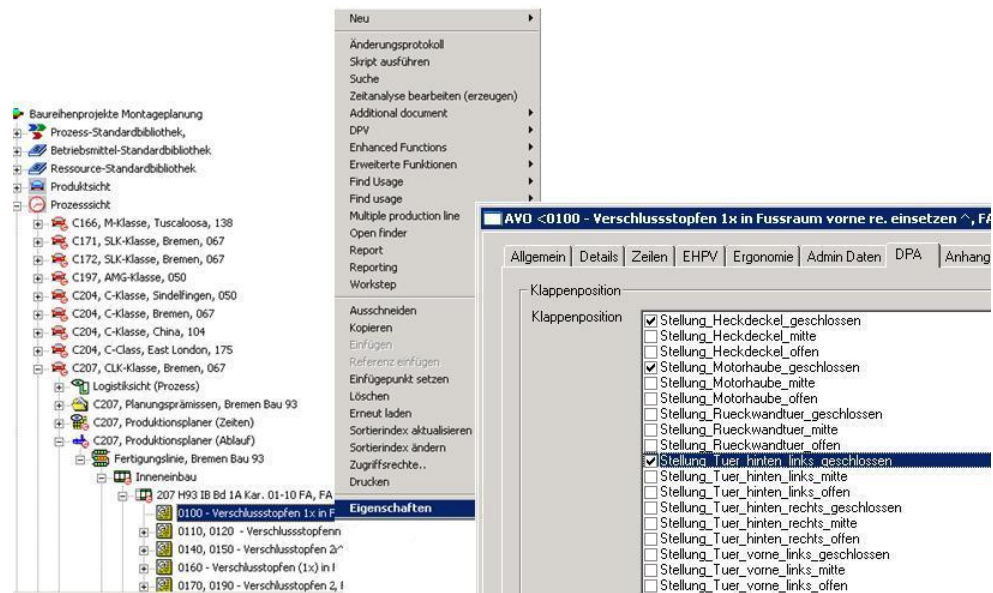
| Local Descriptions           | Short Strings | Hex Value | Decimal Value |
|------------------------------|---------------|-----------|---------------|
| Japanese                     | ja            | 0x0411    | 1041          |
| Korean                       | ko            | 0x0412    | 1042          |
| Latvian                      | lv            | 0x0426    | 1062          |
| Lithuanian                   | lt            | 0x0427    | 1063          |
| FYRO Macedonian              | mk            | 0x042F    | 1071          |
| Malay - Malaysia             | ms-my         | 0x043E    | 1086          |
| Malay – Brunei               | ms-bn         | 0x083E    | 2110          |
| Maltese                      | mt            | 0x043A    | 1082          |
| Marathi                      | mr            | 0x044E    | 1102          |
| Norwegian - Bokmål           | no-no         | 0x0414    | 1044          |
| Norwegian – Nynorsk          | no-no         | 0x0814    | 2068          |
| Polish                       | pl            | 0x0415    | 1045          |
| Portuguese - Portugal        | pt-pt         | 0x0816    | 2070          |
| Portuguese - Brazil          | pt-br         | 0x0416    | 1046          |
| Raeto-Romance                | rm            | 0x0417    | 1047          |
| Romanian - Romania           | ro            | 0x0418    | 1048          |
| Romanian - Moldova           | ro-mo         | 0x0818    | 2072          |
| Russian                      | ru            | 0x0419    | 1049          |
| Russian - Moldova            | ru-mo         | 0x0819    | 2073          |
| Sanskrit                     | sa            | 0x044F    | 1103          |
| Serbian - Cyrillic           | sr-sp         | 0x0C1A    | 3098          |
| Serbian – Latin              | sr-sp         | 0x081A    | 2074          |
| Setsuana                     | tn            | 0x0432    | 1074          |
| Slovenian                    | sl            | 0x0424    | 1060          |
| Slovak                       | sk            | 0x041B    | 1051          |
| Sorbian                      | sb            | 0x042E    | 1070          |
| Spanish - Spain              | es-es         | 0x0C0A    | 1034          |
| Spanish - Argentina          | es-ar         | 0x2C0A    | 11274         |
| Spanish - Bolivia            | es-bo         | 0x400A    | 16394         |
| Spanish - Chile              | es-cl         | 0x340A    | 13322         |
| Spanish - Colombia           | es-co         | 0x240A    | 9226          |
| Spanish - Costa Rica         | es-cr         | 0x140A    | 5130          |
| Spanish - Dominican Republic | es-do         | 0x1C0A    | 7178          |
| Spanish - Ecuador            | es-ec         | 0x300A    | 12298         |
| Spanish - Guatemala          | es-gt         | 0x100A    | 4106          |
| Spanish - Honduras           | es-hn         | 0x480A    | 18442         |

| Local Descriptions    | Short Strings | Hex Value | Decimal Value |
|-----------------------|---------------|-----------|---------------|
| Spanish - Mexico      | es-mx         | 0x080A    | 2058          |
| Spanish - Nicaragua   | es-ni         | 0x4C0A    | 19466         |
| Spanish - Panama      | es-pa         | 0x180A    | 6154          |
| Spanish - Peru        | es-pe         | 0x280A    | 10250         |
| Spanish - Puerto Rico | es-pr         | 0x500A    | 20490         |
| Spanish - Paraguay    | es-py         | 0x3C0A    | 15370         |
| Spanish - El Salvador | es-sv         | 0x440A    | 17418         |
| Spanish - Uruguay     | es-uy         | 0x380A    | 14346         |
| Spanish - Venezuela   | es-ve         | 0x200A    | 8202          |
| Sutu                  | sx            | 0x0430    | 1072          |
| Swahili               | sw            | 0x0441    | 1089          |
| Swedish - Sweden      | sv-se         | 0x041D    | 1053          |
| Swedish - Finland     | sv-fi         | 0x081D    | 2077          |
| Tamil                 | ta            | 0x0449    | 1097          |
| Tatar                 | tt            | 0X0444    | 1092          |
| Thai                  | th            | 0x041E    | 1054          |
| Turkish               | tr            | 0x041F    | 1055          |
| Tsonga                | ts            | 0x0431    | 1073          |
| Ukrainian             | uk            | 0x0422    | 1058          |
| Urdu                  | ur            | 0x0420    | 1056          |
| Uzbek – Cyrillic      | uz-uz         | 0x0843    | 2115          |
| Uzbek – Latin         | uz-uz         | 0x0443    | 1091          |
| Vietnamese            | vi            | 0x042A    | 1066          |
| Xhosa                 | xh            | 0x0434    | 1076          |
| Yiddish               | yi            | 0x043D    | 1085          |
| Zulu                  | zu            | 0x0435    | 1077          |

### Language Selection

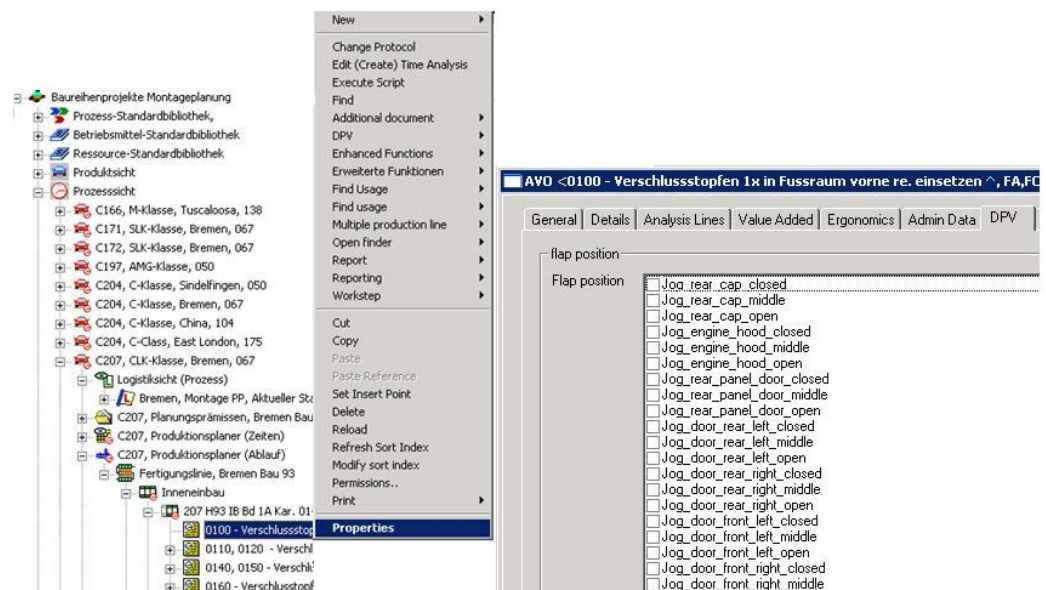
- 1) Start DPE in German Language.
- 2) Right-click 0100 – Verschlussstopfen, select properties and make selection of checkboxes in DPA tab.





**Figure 7: German Language Settings – Checkbox Selection**

- 3) Open **Tools > Settings > Change**.
- 4) Select English Language and press **OK**.
- 5) Close DPE in German language and start again in English language.
- 6) Right-click 0100 – Verschlussstopfen, select properties and open DPA tab.
- 7) The checkbox selection made in German language settings is lost.



**Figure 8: English Language Settings - Checkbox Selection**



### Caution

*Language restriction feature is available in DPE. Changes made to the settings in DPE opened with German language for first time will not be available in the DPE opened with English language for next time.*

*For Example, the checkbox selection in multiselection list box of any attribute in the properties dialog of any component in the PPR tree in German will not*

*be available in the DPE opened with English language, due to mismatch between English and German text in the database.*

## 2.3 Minimum Necessary Access Rights for Users

The following are the minimum access rights for users.

**Table 3: Minimum Access Rights for Users**

| Directory                                  | Access Rights  | Comment  | Directory of Client or Server (C/S) |
|--|----------------|--|-------------------------------------|
| ...\data                                   | read           |  | C,S                                 |
| Data\DatenkartenE5                         | read           |  | C,S                                 |
| data\help                                  | read           |  | S                                   |
| data\lsegrim                               | read           | everyone who uses layout planning or creates WSCs              | S                                   |
| data\Koerper                               | read and write | everyone who uses layout planning or creates WSCs              | S                                   |
|  | read           | everyone   | S                                   |
| data\poolingsrv                            | read           |  | S                                   |
|  | read and write | for administrators   | S                                   |
| data\pprclientdata                         | read           |  | S                                   |
| data\scripts                               | read           |  | S                                   |
| data\Temp                                  | read and write |  | S                                   |
| data\view_bitmaps                          | read and write | everyone who uses E5 graphic                                   | S                                   |
| data\Bitmaps                               | read           | everyone   | C                                   |
|  | read and write | everyone who is allowed to create bitmaps                      | C                                   |
| data\Bitmaps_Overlay                       | read           | everyone   | C                                   |
|  | read and write | everyone who is allowed to create overlay bitmaps              | C,S                                 |
| data\pprclientdata\cadpath<br>data\cadpath | read           | everyone who wants to view CAD data files in E5                | C,S                                 |
|  | read and write | everyone who wants to create and store CAD data files using E5 | S                                   |
| data\pprclientdata                         | read           |  | S                                   |
| data\Formulare                             | read           |  | S                                   |
| data\help                                  | read           |  | S                                   |

| Directory                   | Access Rights  | Comment  | Directory of Client or Server (C/S) |
|-----------------------------|----------------|--|-------------------------------------|
| data\pprclientdata\Import   | read           | everyone who wants to use the import scripts included in the demo projects | C                                   |
| data\Misc<br>data\PPRLoader | read           |  | C                                   |
|                             |                | Depending on the customization at the customer side                        | C                                   |
| data\product_cadpath        | read           | everyone who wants to view CAD data files in E5                            | C                                   |
|                             | read and write | everyone who wants to create and store CAD data files using E5             | C                                   |
| Data\rlg                    | read           |  | C                                   |
|                             | read and write | for administrators   |                                     |

## 2.4 Saving User Profiles in a CITRIX Environment

### 2.4.1 General Information

DELMIA Process Engineer® saves the setting of the individual user profiles of a client workstation in the key "HKEY\_CURRENT\_USER" using the Windows registration editor. If DPE is used in a CITRIX environment, the current user profile in the registration editor is reflected by CITRIX. To save the user-specific settings of the registration editor centrally, a dedicated profile server may be configured additionally in CITRIX. However, if more than one CITRIX login with the same DPE user name is registered and profile servers for DPE installations are running, management and saving of user profiles with the Windows registration editor turns out to be complicated and prone to produce errors.

To avert this, user specific data is served in the Oracle database as of the release DPE5.16SP4.

The user profile "*Delmia\ergoplan*" is saved in the E5 Oracle database instead of the Windows registration editor key "HKEY\_CURRENT\_USER". Since the user profile is saved by a database transaction, only "*saved*" profiles can be read by a different CITRIX login. The user profile is saved for the DPE user and not for the Windows user as is the case in the registration editor.

To be able to save user settings for individual user profiles in the DPE database, asynchronous saving for the PPR Client has been implemented. During asynchronous saving, the data is collected every 20 seconds and then saved to the DPE database.

You can switch over between the two saving options.

Since some settings need to be available prior to accessing the Oracle database, some settings continue to be saved in the Windows registration editor.

## 2.4.2 Activating and Deactivating the Save Mode

### 2.4.2.1 Settings

To enable the user profile in the database, a DPE administrator must enter the following settings for the entire DPE installation. Use the maintenance tool to do this, then, by way of the *Global* tab select the root directory 'Ergoplan'

- 1) Open the maintenance tool in **Tools < Settings < Maintenance**.
- 2) Select the **ErgoPlan** root directory from the Global tab. There cannot be an existing key ergoplan present.  
Notice the user profile saved in the registry of the Current User tab.

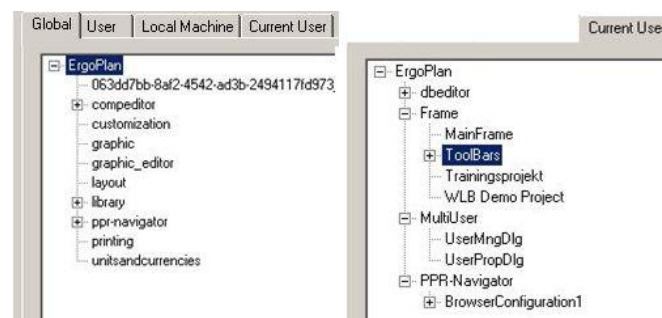


Figure 9: Global Tab

- 3) Select the **ErgoPlan** root directory from the **Global** tab.
- 4) To create a new key, click the **New Key** button.
- 5) Enter the following values in the open dialog:  
In the Key field: **ergoplan/EPSessionDataMng**  
In the Name field: **UserProfileSDMStorage**  
In the Value field: **1**

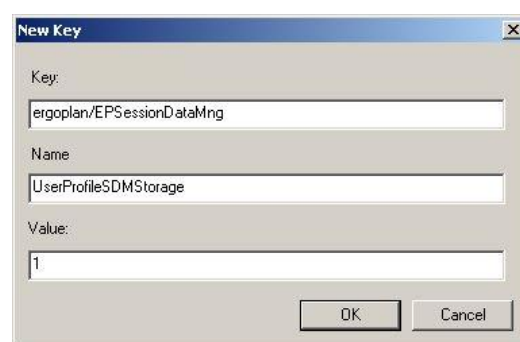


Figure 10: New Key

- 6) Click **OK** to close the dialog.
- 7) Select the **EPSessionDataMng** entry and create two new values, by activating the **New Value** button. Enter the following values:

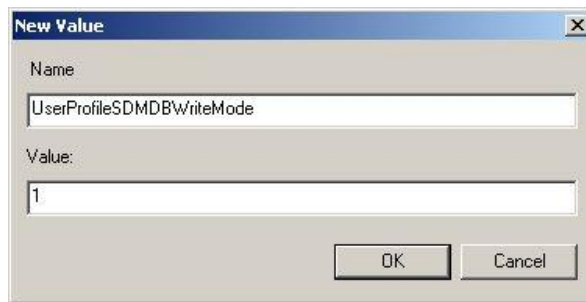


Figure 11: New Value

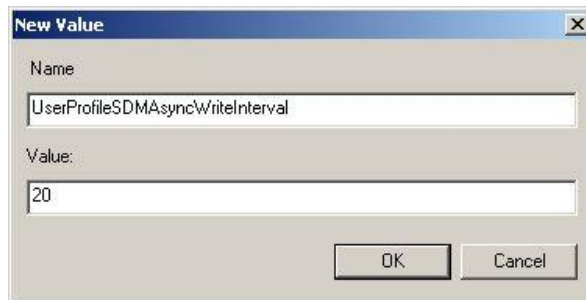
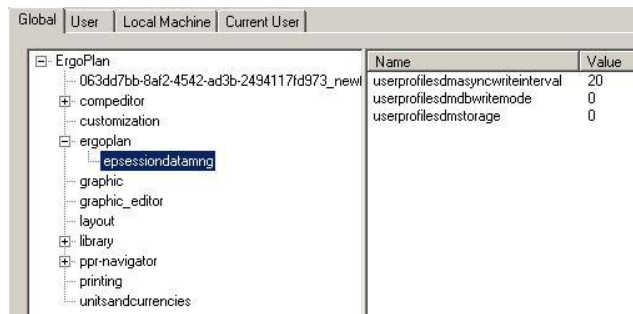


Figure 12: New Value

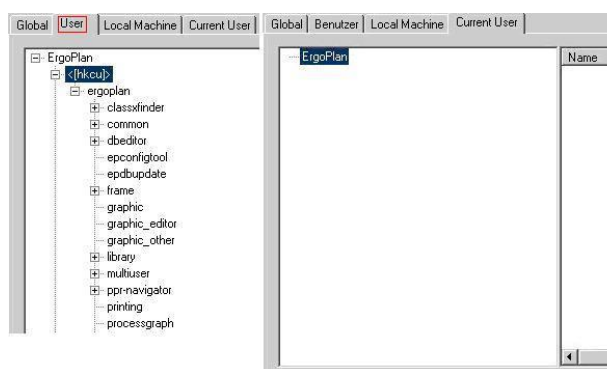
Once again a summary of the three values:

- **The value `UserProfileSDMStorage`:** Enter the following value in maintenance tool to activate the synchronous storage mode: `ergoplan\UserProfileSDMStorage = '1'`. By default, the entry is off, setting "0". The synchronous storage mode will now save all modifications. Activating this option, may cause a decrease in performance.
- **The value `UserProfileSDMDBWriteMode`:** To activate the asynchronous storage mode the following value must be entered in the maintenance tool: `ergoplan\UserProfileSDMDBWriteMode = '1'`. Default is asynchronous mode, setting "1".
- **The value `UserProfileSDMAsyncWriteInterval`:** This value indicates how often the user profile is saved under asynchronous storage: `rgoplan/EPSessionDataMng/ UserProfileSDMAsyncWriteInterval = '20'`. The default setting for the value is 20, which means the user profile is saved every 20 seconds. This value can vary in the range of 3 to 120. If a value smaller than three is entered, the user profile will be saved every three seconds. If a value larger than 120 is entered, the setting will be reset to save the user profile every 120 seconds. The default setting, 20 seconds, has proven to be the most effective setting.



**Figure 13: UserProfileSDMAsyncWriteInterval Setting**

After the setting has been made; notice that the entries saved in the database are displayed under the **<[hkcu]>** key in the *User* tab. The *Current User* tab is empty. Please refer to the [Figure 14](#).



**Figure 14: User Tab**

If save setting for the database is deactivated, the user profile will once again be displayed in the *Current User* tab.

Global settings can be overwritten by local settings when saving the user profile. The local settings, found at HKEY\_LOCAL\_MACHINE, can be used independently of the global settings. When using local settings the global settings are disregarded. If the user profile for a single client is saved in the database, the global settings are no longer active.



### Caution

*These settings are only to be used in case of repairs or during the setup phase. After those actions have been completed, the global settings should be reactivated. If the settings are incorrect, the DELMIA Process Engineer® may not function properly.*

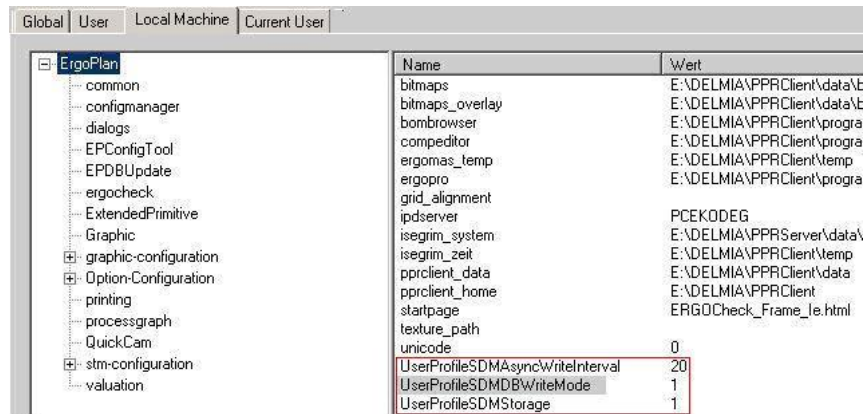
The maintenance tool is also used to make local settings:

- 1) Create three new values by selecting the **ErgoPlan** root directory by way of the *Local Machine* tab.
- The value *UserProfileSDMStorage*: Enter the following value in the maintenance tool to activate the synchronous storage mode `ergoplan\UserProfileSDMStorage = '1'`  
By default, the entry is off, setting "0".
- The value *UserProfileSDMDBWriteMode*: To activate the asynchronous storage mode the following value must be entered in the maintenance tool

`ergoplan\UserProfileSDMDBWriteMode = '1'`

Default is asynchronous mode, setting "1".

- The value *UserProfileSDMAsyncWriteInterval*  
This value indicates how often the user profile is saved under asynchronous storage `UserProfileSDMAsyncWriteInterval = '20'`  
The default setting for the value should be set at 20.



| Name                             | Wert                                     |
|----------------------------------|--|
| bitmaps                          | E:\DELMIA\PPRClient\data\bitmaps         |
| bitmaps_overlay                  | E:\DELMIA\PPRClient\data\bitmaps_overlay |
| bombrowser                       | E:\DELMIA\PPRClient\program              |
| compeditor                       | E:\DELMIA\PPRClient\program              |
| ergomas_temp                     | E:\DELMIA\PPRClient\temp                 |
| ergopro                          | E:\DELMIA\PPRClient\program              |
| grid_alignment                   |  |
| ipdserver                        | PCEKODEG                                 |
| isegrim_system                   | E:\DELMIA\PPRServer\data\isegrim_system  |
| isegrim_zeit                     | E:\DELMIA\PPRClient\temp                 |
| pprclient_data                   | E:\DELMIA\PPRClient\data                 |
| pprclient_home                   | E:\DELMIA\PPRClient                      |
| startpage                        | ERGOCHECK_Frame_1e.html                  |
| texture_path                     |  |
| unicode                          | 0  |
| UserProfileSDMAsyncWriteInterval | 20                                       |
| UserProfileSDMDBWriteMode        | 1  |
| UserProfileSDMStorage            | 1  |

**Figure 15: UserProfileSDMAsyncWriteInterval Settings**

While switching the user profile from the registry to database, only some of the entries will be copied into the database. But, you may add these values later, if required.

## 2.5 Basic Rules for the Configuration of Types and Plantypes

### 2.5.1 Types Basic Rules

In the DELMIA Process Engineer you have extensive configuration possibilities. Using the type and plantype configuration you can individually configure the following:

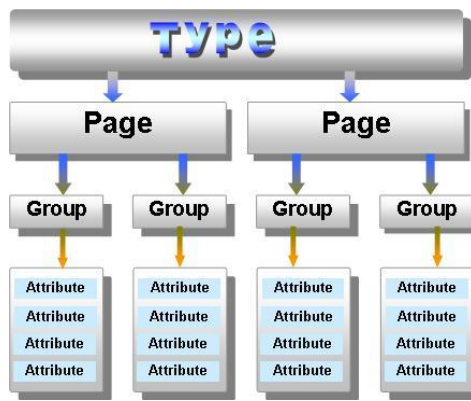
You can configure the design of the dialogs almost without any restrictions.

You can add additional attributes.

In the following chapters the basic rules for the configuration of types and plantypes are explained.

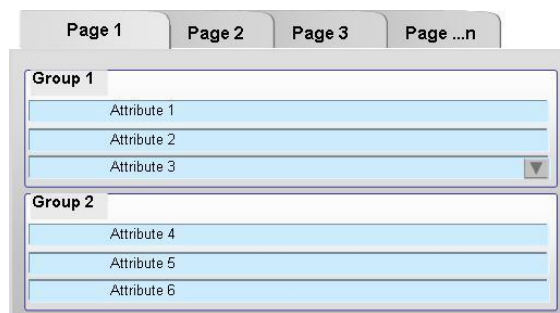
A type contains all data relevant for the configuration of a data object. A type contains a list of pages, groups and attributes defining the "look" and "feel" of a data object in the user interface.





**Figure 16: Types Schematic**

In [Figure 16](#) the schematic structure of a type is displayed.



**Figure 17: Dialog Layout using Pages and Groups and Attributes**

Using pages tabs are created in the user dialog for entering data object properties. In the tabs groups can be defined, in which multiple attributes can be combined according to subject.



### Note

*In the DELMIA Process Engineer types are valid cross-module or cross-project. Therefore changes on the type level have effects on the whole program and on all existing projects.*



### Caution

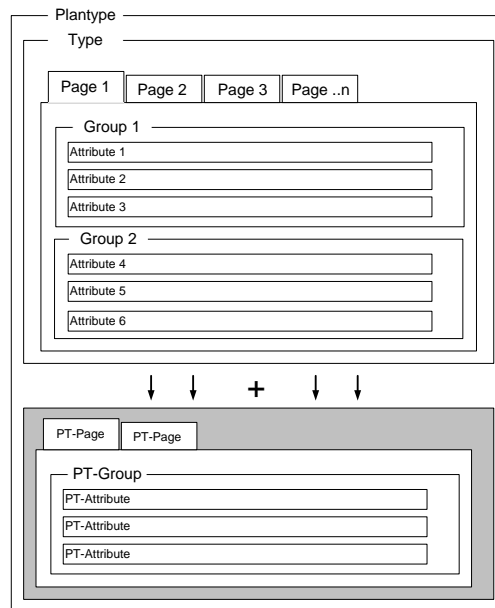
*Improper changes on the type level can lead to a total loss of data. Therefore, changes should only be made by trained administrators.*

## 2.5.2 Plantypes Basic Rules

Plantypes are classes derived from types (Configuration Manager) that can be created and configured by users. For this reason, each plantype has pages, groups and attributes of the parent type (Configuration Manager). In addition, the following can be configured for a plantype by the user:

Attributes, groups, and pages





**Figure 18: Types and Plantypes in the DELMIA Process Engineer®**

Each plantype refers to a specific planning area. At present plantypes can be defined for the following areas.

### Resource Plantype

Resources are all technical and organizational objects necessary for production. Typical resources are machines or tools, typical organizational resources are production lines or workstations.

### Process Plantype

Processes are all actions or tasks necessary for the production of a product or of its subassemblies or individual parts. Processes are defined using: process name – process number – process time.

### Product Plantype

The product view looks at the product or the manufactured item. In this view bills of materials of the products are referenced.

Products represent on the one hand complete structures which contain: Sub-assemblies, parts, and auxiliary materials.

Individual subassemblies, parts or auxiliary materials can also be products. Products are unambiguously defined by name, number, planning code, code rule etc. The Process Engineer enables the planning of variants controlled by code rules. Within the areas a distinction is made between **technical** and **organizational** objects.

A Technical Object describes a planning item (for example a part, work sequence, operating resource).

Technical objects can be defined in such a way that additional objects can be created under this object.

These objects are called **technical structure items**.

An Organizational Object describes a project structure item (for example a plant structure, department, planning structure). They serve the purpose of mapping the planning, process or manufacturing structure within a project, among other things.

These objects are also called **organizational structure items**.

### 2.5.3 Importing and Exporting Plantype Sets



For more information on importing and exporting plantype sets, *Please refer to the [System Library Manual](#).*

## 3. Creating Plantype Set and Plantype Structures

### 3.1 General

A plantype set defines the project structure according to the planning types included. To create an object in a project it is necessary to first create the corresponding plantypes in a plantype set and to link them. By linking the plantypes you can define hierarchies that preset the planning structure like a form of a template.

This is performed in three steps:

- Step 1: Creating a plantype set
- Step 2: Creating a plantype
- Step 3: Linking plantypes

A plantype set does not only provide information regarding the structure. If you create a new component in the PPR Navigator, a sequence of queries is started:

- Which view does this component belong to?
- What type should be created?
- What pages and groups are to be displayed how and where?
- What attributes does it have?
- Where are the attributes determined, in the configuration or in the plantype set?

The plantype set contains all the information and provides the PPR Navigator with the information, which then creates a component. Plantypes are important for describing component properties.

For information on how to generate and edit plantype sets please refer to the following sections.



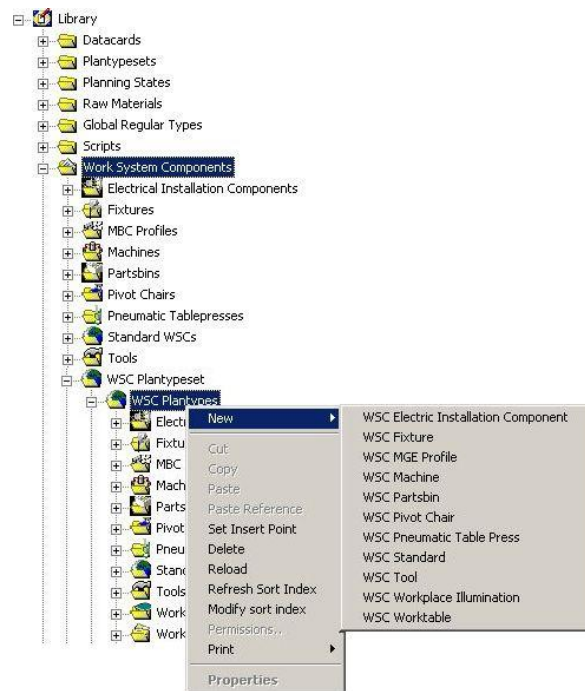
#### Caution

*Since version 5.10 the project plantype set has only one reference to the plantype set of the General Library. This means that the changes can only be made in the plantype set of the General Library.*

#### 3.1.1 The Plantype Set for System Elements

The **WSC plantype** set has an exceptional position within the plantype sets. This plantype set **cannot** be structured.

The plantype set for system elements is found in the system library under **Work System Components < Plantype Set for System Elements** and not in the plantype set file. Please refer to the [Figure 19](#).



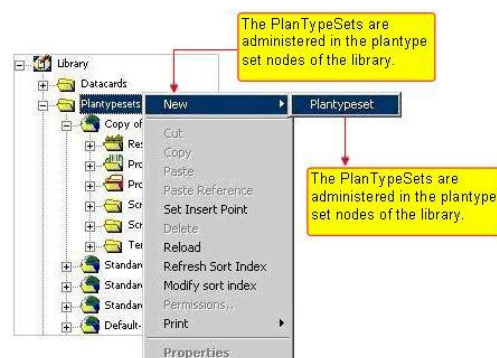
**Figure 19: Creating a New Plantype Set for System Elements**

New plantype sets are created in the general library **Work System Components < Plantype Set for System Elements < Plantype for System Elements**. Since this plantype set is referenced by all projects, any changes made on the WSC plantype set will affect all projects.

Creating and editing plantypes for system elements does not differ from “normal” plantypes. That is why the following paragraphs will not discuss plantypes for system elements explicitly.

## 3.2 Creating Plantype Set Structure

To create a plantype set open the General **Library** and right-click on the **Plantypesets** folder in the library. In the following context menu select the “New Plantypeset” option.



**Figure 20: Creating a New Plantype Set**

In the following Properties dialog you can enter the **Typeset Name** of the plantype set and its abbreviation.

Using a simple editor you can enter general remarks regarding the plantype set in the “Notes” tab.

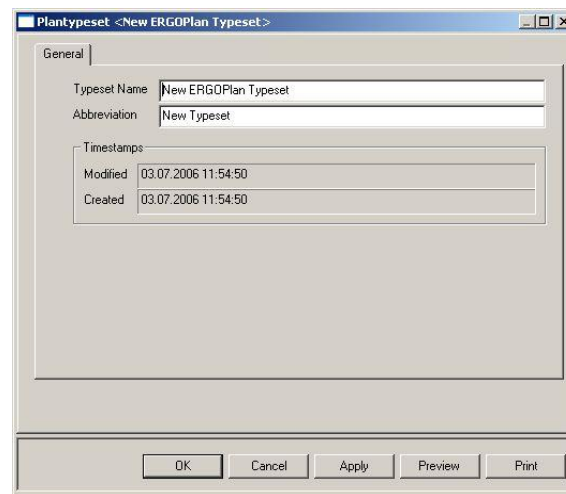


Figure 21: Plantype Set Properties

### 3.2.1 Creating a Plantype

After you have created a plantype it is now time to animate the plantype set, i.e. in this case to create plantypes. For this purpose 3 plantype areas (views) are available. *Please refer to the [Table 4](#).*



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

- The “**Resource plantypes**”, used, for example, to create a plantype for resources.
  - The “**Process plantypes**”, used, for example, to create a plantype for a work sequence.
  - The “**Product plantypes**”, used, for example, to create a plantype for a subassembly.
- 1) To create a plantype, right-click one of the plantype areas (**Resource**, **Process** or **Product** plantype).
  - 2) In the following context menu you can create a new plantype using the **New** option.

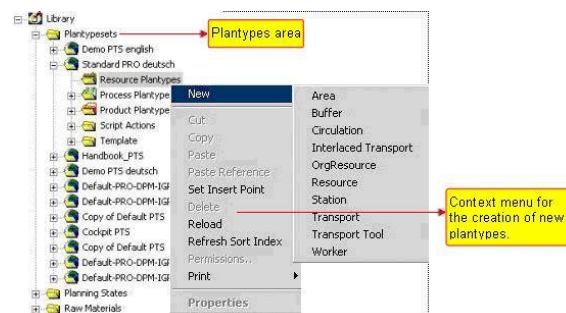


Figure 22: Creating a Resource Plantype

Depending on the plantype area the following plantypes can be created:

Table 4: Plantype Overview

| Plantype area   | Plantype                               | Comment   |
|---|--|---|
| <b>Plantypes</b><br><b>Resources</b><br><div> Area<br/> Buffer<br/> Circulation<br/> Interlaced Transport<br/> OrgResource<br/> Resource<br/> Station<br/> Transport<br/> Transport Tool<br/> Worker </div>   | Area                                   | Using this plantype you can structure areas (for example supply areas and transport areas, etc.).   |
|   | Worker                                 | Using this plantype you can define workers as Resource (this is applied, for example, in the Manufacturing Concept).  |
|   | OrgResource                            | Using OrgResource you can create organizational structures, to which resources, such as stations or operating resources, can be assigned during the planning process. |
|   | Buffer                                 | Buffers are used, for example, in the Manufacturing Concept of the DELMIA Process Engineer.   |
|   | Resource                               | Using this plantype you can create plantypes for your operating resources (for example, drill, screw)   |
|   | Station                                | Using this plantype you can create plantypes for workstations.  |
|   | Transport                              | You can set the type of transport with the help of this plantype (e.g. chained transports, or forward and back transports).   |
|   | Circulation                            | Using this plantype, for example, you can structure single conveyor sections.   |
|   | Transport Tool                         | Using this plantype you can create plantypes for transport trolleys etc.  |
|   | Interlaced Transport                   | Interlaced (chained) transports (e.g. lines) are defined with the help of this plantype.  |
| <b>Plantypes</b><br><b>Process</b><br><div> Behavior<br/> OrgProcess<br/> Process<br/> Simulation Feature<br/> Timeanalysis<br/> Timeanalysis AZB<br/> Timeanalysis FOR<br/> Timeanalysis MEK<br/> Timeanalysis MOST<br/> Timeanalysis MTM1<br/> Timeanalysis MTM2<br/> Timeanalysis RWF<br/> Timeanalysis SAM<br/> Timeanalysis STD<br/> Timeanalysis UAS<br/> Work Instruction </div> | Behavior                               | Behavior processes are used in V5 and can contain further processes.  |
|   | OrgProcess                             | Using OrgProcesses you can create organizational structures to which processes such as plans can be assigned during the planning process.                             |
|   | Process                                | Using this plantype you can create plantypes for processes (e.g., workplans, main and sub-processes).   |
|   | Timeanalysis und Timeanalysis AZB .... | Plantypes for process analysis and all analytical methods can be created using this.  |
|   | Work Instruction                       | Work instructions are used in ENOVIA.   |
| <b>Plantypes</b><br><b>Product</b><br><div> Fastener<br/> OrgProduct<br/> Product </div>  | OrgProduct                             | Using OrgProducts you can create organizational structures to which subassemblies and parts can be assigned during the planning process, for example.                 |
|   | Product                                | Using this plantype you can create plantypes for products (e.g. modules, subassemblies, parts).   |
|   | Fastener                               | Using this plantype you can create plantypes for fasteners.   |



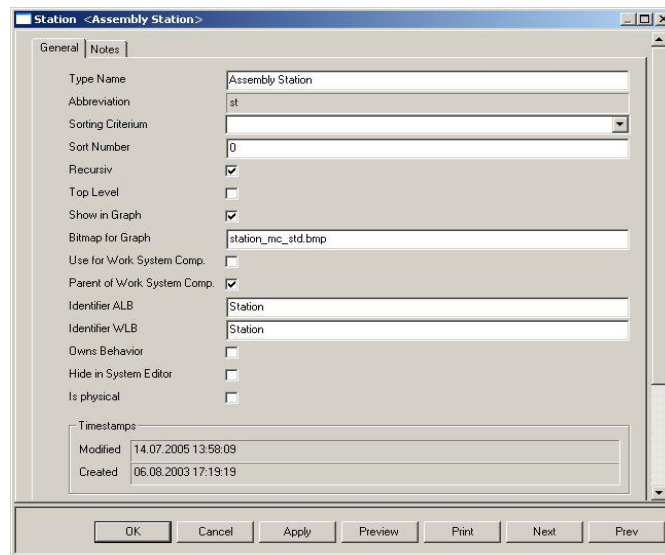
## Note

Organizational plantypes can contain non-organizational (technical) plantypes. Organizational plantypes can be generated only via organizational nodes.

### 3.2.1.1 Plantype Properties

After you have created a new **plantype** the Properties window is opened automatically; here you can enter the type name, the abbreviation, etc.

The Properties window offers you tabs where you can enter “General” Properties, “Notes” and “Additional” information.






**Figure 23: Plantype Properties (Technical Resource)**

The tabs of the Properties dialog are described in detail below.




#### The “General Properties” of a plantype

**Table 5: Plantype Properties**

| Properties   | Description  |
|--------------|--|
| Type Name    | The name of the plantype.  |
| Abbreviation | <p>The abbreviation of the plantype. The short name of a plantype must always be unambiguous. If you execute the following tip, you should display the short name only when it is write-protected and you should not change the existing short names.</p> <p> <i>If the abbreviation is not shown by default, you could set it to default. Proceed as follows:</i></p> <p>Open the configuration manager ( <i>Tools / Database Utilities / Configuration tool</i> ).</p> <p>search for the „ergoplantype“ entry in the “<i>List of Types</i>”.</p> <p>search for the “nameshort” entry in the “<i>List of Attributes</i>”.</p> <p>Enable here at least the “Show in editor” and the “Show in browser” options</p> <p>Before leaving, save the changes in the Configuration Manager.</p> |

| Properties   | Description  |
|--|--|
| Sort number  | With this number the place is determined where the Plantype is shown in the tree structure of the PPR Navigator. The higher the sort number chosen, the further down the plantype is displayed in the tree structure.  |
| Sorting criterium<br><div> <div>A-B</div> <div>AA-AI</div> <div>AA-AM</div> </div> | <p>Specify here the sorting criterium of the plantype in the Project Library. There are four possibilities:</p> <p>A-B: The components of this type are saved in folders from A-Z depending on the first letter.</p> <p>AA-AI: The components of this type are saved in folders from AA-ZZ (AA-AI, AJ-AR, AS-AZ etc.) depending on the first letter. Using this option you can achieve the highest sorting density.</p> <p>AA-AM: The components of this type are saved in folders from AA-ZZ (AA-AM, AN-AZ etc.) depending on the first letter.</p> <p>none: no folders are created</p>   |
| Top Level  | <p>If a plantype is defined as Top Level, the objects created with this plantype appear on the highest hierarchical level of a project.</p> <p>The top level property of a plantype has to be defined in the plantype set of the project. To modify it, open the project and open therein the plantype set. Right-click on the plantype and select 'Properties' in the context menu."</p>  |
| Recursiv   | Here it is determined whether a plantype is defined recursive, i.e. a plantype can contain children of the same plantype. A subassembly can, for example, contain further subassemblies.   |
| Show in Graph  | This option determines whether a plantype is shown in the DELMIA Industrial Engineer Process Engineer.   |
| Bitmap for Graph   | Here you can define a bitmap to be shown in the Process Graph. If no bitmap is defined, a standard bitmap is shown. The bitmap is scaled to 84 x 36 pixels.  |
| Use for Work System Comp.  | <p>This option determines whether a plantype for automatically created system items can be used. If you link a system item with a product or with process objects, a BOM object is created automatically under the plantype in the Project Library with the system item name that activated the checkbox.</p> <p> <b>Caution</b><br/> <i>This is only valid for resource objects. In the case of product or process objects this enabled option has no importance. This checkbox can only be activated with a plantype. If it is enabled with another resource object, an entry set previously is disabled.</i></p> |
| Parent for Work System Component   | <p> <b>Caution</b><br/> <i>This applies only to resource objects.</i></p> <p>With this option you can set whether a work system component can be assigned to this plantype. This is necessary only whenever you drag a work system component to a resource by using drag &amp; drop.</p> <p>If, for example, you want to use work system component in a station graphic you have to activate this option on the plantype station. The same applies to all resource plantypes to which you want to assign work system component.</p>   |
| Identification ALB   | The description is found in the chapter Configuring Automatic Line Balancing: <a href="#">Attribute identifieralb (Identification ALB)</a> .   |



| Properties            | Description  |
|-----------------------|--|
| Identification WLB    | The description is found in the chapter Configuring Work Load Balancing: <a href="#">Attribute Identifierwlb (Identification WLB)</a>  |
| Owens Behavior        | <p>This applies to technical resource objects only.</p> <p>This option defines whether it is possible to create a behavior process for this plantype. The checkbox can be set for one plantype only. If activated on a different resource object, the previous entry is deactivated.</p> <p>V5 uses behavior processes, which can contain further processes. In the process view, a plantype of the behavior type must exist. If this is not the case, an error message will appear. To stop error messages, create a plantype of the behavior type.</p>  <p>Always, only one behavior process can be created on a resource. Behavior processes are bound to the resource on which they have been created. When the resource is copied or deleted, the behavior process is copied or deleted as well.</p> <p>Access rights, planning status or filters cannot be allocated to behavior processes.</p> <p>If this option is activated, the context menu of this resource will contain an entry, which can be used to create a behavior process.</p>  |
| Hide in System Editor |  <p><b>Caution</b><br/>Only for resource objects.</p> <p>With this option configure whether to show (deactivated) or hide (activated) this plantype in the system editor of V5.</p>   |
| Is physical           |  |



### Note

*The “Type Name” and the “Abbreviation” of a plantype must always be chosen unambiguously within a plantype set.*

### “Notes” tab

In the “Notes” tab you can create commentaries using a simple editor.

## 3.2.2 Editing a Plantype

By right-clicking a context menu appears with the following options to edit a plantype:

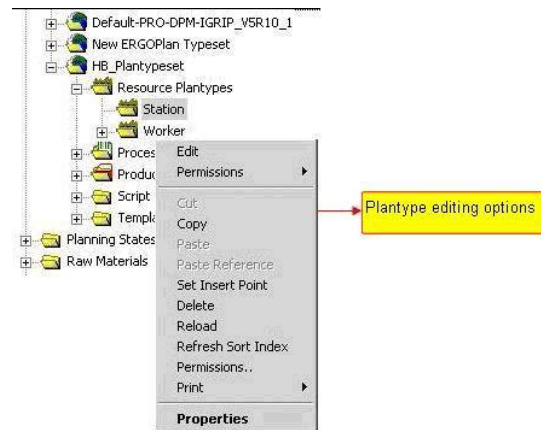
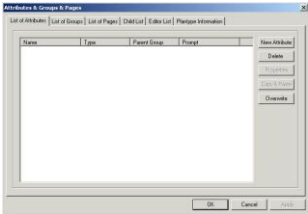
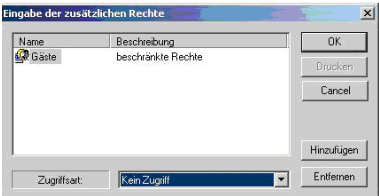
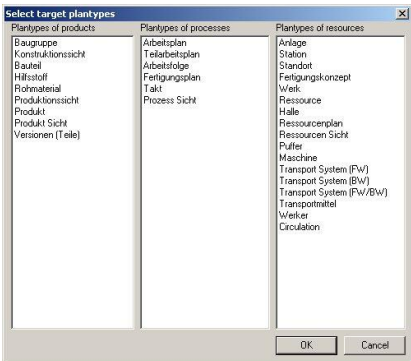
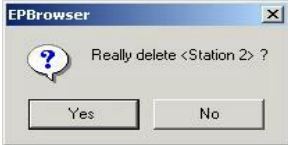
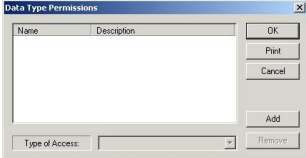
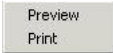
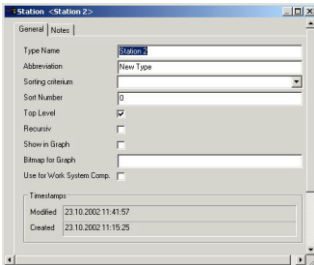


Figure 24: Editing a Plantype

### 3.2.2.1 The Plantype Context Menu

Table 6: Plantype Properties

| Properties   | Description   |
|--|---|
| <b>Edit</b><br>  | <p>By clicking the “Edit” option the plantype editor is started. In the plantype editor you can define additional attributes, groups and pages (see the following chapter).</p>   |
| <b>Permissions</b><br><br> | <p>Transferring rights to other types.</p> <p>Using this menu item you can transfer rights of a selected plantype to other plantypes. <i>Please refer to the <a href="#">User Management Manual</a>.</i></p> <p>You have three possibilities to determine the rights of children:</p> <p>Remove rights: Using this menu item you can remove user rights from multiple types. In the first step you should determine which rights are to be removed. In the next step you can specify the plantype where the previously selected right should be removed.</p> <p>Add rights: Using this menu item you can add user rights to other types. This change affects all selected types.</p> <p>Overwrite rights: Using this menu item all rights of the selected types are transferred to the selected types in the dialog that is opened.</p> |
| <b>Copy</b>  | <p>Using this menu item you can copy a plantype to add it again as reference.</p>   |

| Properties   | Description   |
|--|---|
| <b>Paste Reference</b>   | You can create a link using this menu item if you have copied a plantype before. <i>Copy</i> and <i>Paste reference</i> are only possible within <b>one</b> view.   |
| <b>Set Insert Point</b>  | Using the “ <b>Set Insert Point</b> ” option you can determine the order of the sub-objects displayed in the object list.   |
| <b>Delete</b><br>       | Using the “ <b>Delete</b> ” option, the plantype is deleted after confirmation.   |
| <b>Reload</b>  | Using the “ <b>Reload</b> ” option the display area of the object properties is refreshed.  |
| <b>Refresh Sort Index</b>  | <i>Please refer to the <a href="#">PPR Navigator Manual</a> for more details.</i>   |
| <b>Permissions</b><br> | Using the “ <b>Permissions...</b> ” option you can determine the access rights to plantypes. <i>Please refer to the <a href="#">User Management Manual</a>.</i>   |
| <b>Print</b><br>      | Printing plantype information using the “ <b>Print</b> ” option. You have the possibility to show the plantype in a preview or to directly send it to a printer.<br><br>A requirement is the existence of a corresponding print form. |
| <b>Properties</b><br> | Using the “ <b>Properties</b> ” option you can open the dialog described before to define or to change object attributes.   |

### 3.2.3 Creating Plantype Hierarchies

#### Example

In the following example it is explained how you can create a hierarchical structure in your plantype set. The goal of this chapter is to replicate the following hierarchical structure in a plantype set.

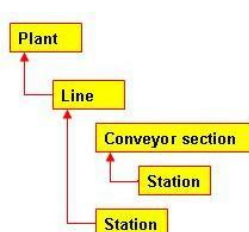


Figure 25: Example for a Hierarchical Structure

Since DELMIA Process Engineer views a plant as a resource, you have to select a resource plantype for plant. For this purpose by right-clicking on the **“Plantypes Resource”** folder you can create the following resources using the **New < Station** option.

- Plant (define as Top Level)
- Line
- Conveyor section
- Station

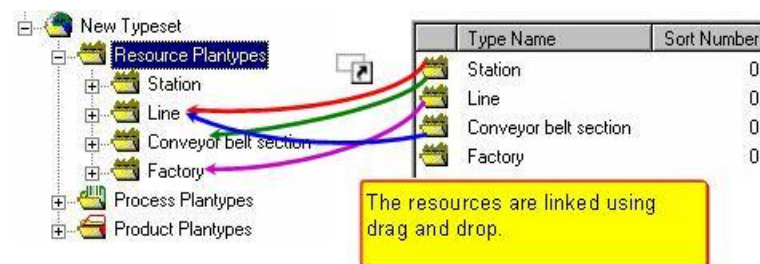
Afterwards link the plantypes using drag and drop as follows:



For more information on drag and drop, Please refer to the [PPR Navigator Manual](#).

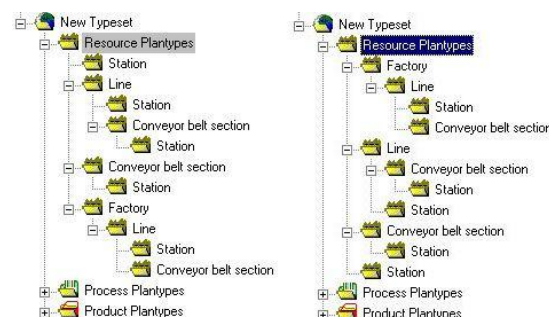
**Table 7: Plantype Links**

| Plantype         | Link With | Plantype         |
|------------------|-----------|------------------|
| Line             |           | Plant            |
| Conveyor section |           | Line             |
| Station          |           | Conveyor section |
| Station          |           | Line             |



**Figure 26: Linking Plantypes using “Drag and Drop”**

Depending on the sort index specified, the completed plantype set in the library looks as follows:



**Figure 27: The Completed Plantype Set Example**

If you now create a new project the plantype set that you have just created appears in the **plantype set** selection. Select your plantype set and confirm by clicking **“OK”**. In the project you can now map a plant using the plantype set you have just created.



For more information, *Please refer to the PPR Navigator Manual.*

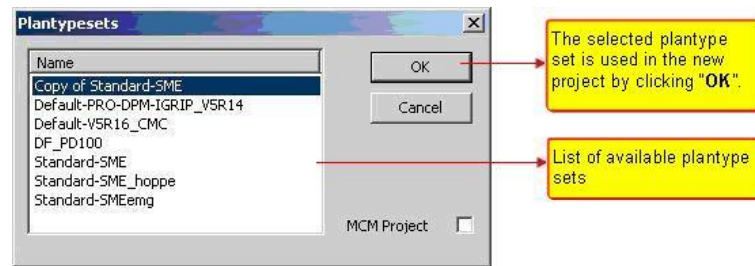


Figure 28: Plantype Set Selection

### 3.3 Creating Attributes, Groups, and Pages

You have the possibility to define additional attributes, groups and pages for each plantype. This enables you to adapt the program to your individual needs. Using new attributes you can add as many data fields to a plantype as you want, which can then be filled with information by the DELMIA Process Engineer user during the planning process.

In the DELMIA Process Engineer you have the possibility to add new attributes, groups and pages to a plantype, as well.

- 1) **In the plantype set area of the library:** Right-click a plantype and select the **Edit** option from the context menu. *Please refer to the Figure 29.*

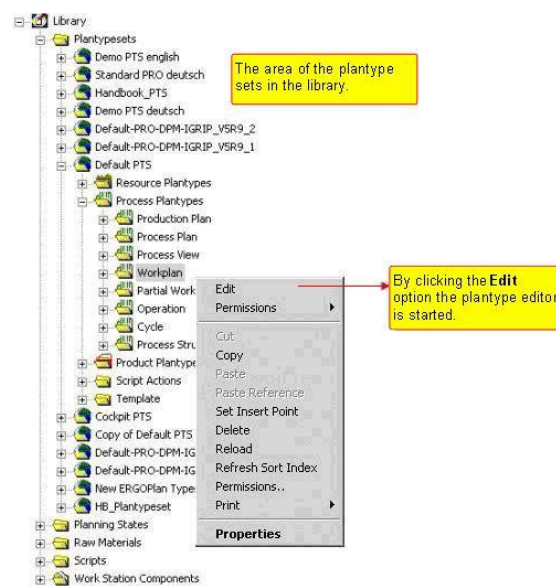
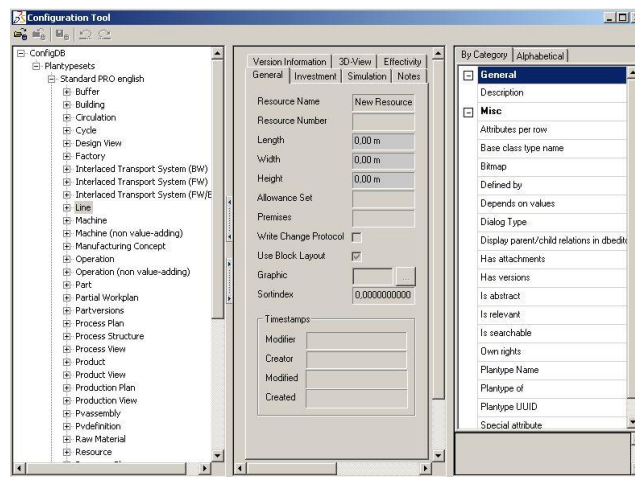


Figure 29: Calling the Plantype Editor in the General Library

Once you have selected **Edit** in the context menu of the plantype, the *Configuration tool* dialog opens. In this dialog, the plantype set and the plantype, which have been selected in the system library, are always shown. *Please refer to the Figure 30.*

In the *Configuration tool* dialog you can edit attributes, groups and pages as well as the plantype information of a plantype.



**Figure 30: Configuration Tool Dialog with Selected Plantype Set**

The functions listed below can be edited using the configuration tool:

Read chapter concerning this [The Configuration Manager](#).

- **Attributes:** New attributes can only be created from the plantype set. To edit the plantype-specific attributes of a plantype. *Please refer to the [Editing Attributes of a Type](#) and [Attributes](#).*
- **Groups:** To edit plantype-specific groups. *Please refer to the [Editing Groups of a Type](#).*
- **Pages:** To edit plantype-specific pages. *Please refer to the [Editing Pages of a Type](#).*
- **Child list:** Settings of the PPR relations, in which views they are displayed or what they are called. *Please refer to the [Configuration of Parent-Child Relations](#).*
- **Menu items:** The context menu entries of this plantype are controlled in this list. *Overwrite* displays all available menu entries. You can change, display, or hide these context menu entries for the plantypes as well as set them as standard entries. New context menu entries can not be created here. *Please refer to the [Editors for the Context Menu](#).*
- **Plantype Information:** Here you can assign an icon to a plantype. You can also determine whether a change log should be created for objects of this plantype. *Please refer to the [Editing the Properties of Types](#).*

### 3.3.1 Attributes



#### Caution

*Attributes added after starting a new project do not only have an effect on newly created objects, but also have an effect on existing objects.*

Self-defined attributes, defined on the plantype level from Version 5.10 onwards can be shown as search criterion in the “**Finder**”.



#### “External ID” attribute

The *External ID* attribute with the attribute designation **externalid** and the internal name **m\_strExternalId** plays a central role while importing data using



the PPR Loader and has been implemented exclusively for the import process. Since only the unambiguous identification (ID) of the objects is stored in this attribute, this attribute should only be used for the data import.

### Overwrite

You have the possibility to overwrite existing attributes, defined with types (configuration manger). Thus, for example, you can display an attribute on another page or you can limit the entry length, etc.



### Note

*There are unused attributes for all components (special or free attributes). These attributes are for overwriting. For more information, Please refer to the [Editing Attributes of a Type](#) in the Configuration chapter*

### Copy

Attributes can be copied. It is thereby important to assign a **new attribute name**. This is a special case of creating new attributes, so to speak. The setting of the existing attribute are transferred. In the case of newly created plantype attributes this attribute is completely independent of the original afterwards. In the case of overwritten attributes you have to distinguish between two cases:

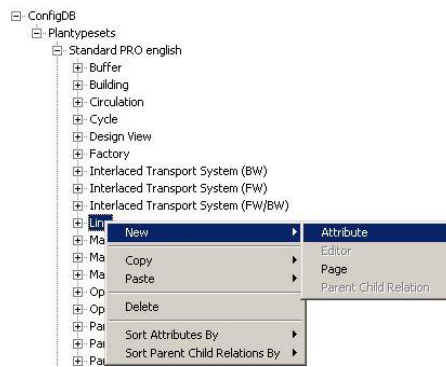
- If the overwritten field is identified by the server through internal attribute names (that is the case with the minority of the attributes, for example, with *modifier* or *creator*), then these two attributes are dependent on each other afterwards. Therefore, only one of the attributes must be overwriteable.
- The majority of the attributes, however, is identified in the DELMIA Process Engineer by its attribute name. As the name has to be changed when copying, these attributes are completely independent of the original afterwards.

### 3.3.1.1 Creating a New Attribute

If you cannot find suitable attributes with the type (*Overwrite*), you can create your own attributes (= plantype attributes).

- 1) Select the plantype in the plantype set and open the context menu by clicking with the right mouse button.
- 2) Select **New < Attribute**. New attributes are also stored *permanently* in the database and you can search for them if you have assigned values to the attributes.
- 3) The properties of an attribute can be displayed according to category or alphabetically and can then be edited. Click on one of the tab in the properties window.

By Category | Alphabetical

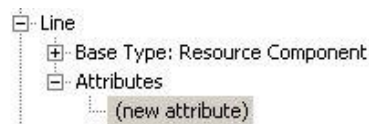


**Figure 31: Creating a New Attribute via the Context Menu**

4) Set the properties for the new attribute.

For a description of the fields, *Please refer to the* [Table 14](#).

If no attribute exists as yet for the plantype, a new folder attribute is created for the plantype.



**Figure 32: New Folder for Attributes**

## 3.4 Configuration Examples

The following chapter contains examples on how you can configure the interface and the database in the DELMIA Process Engineer using plantypes.

### 3.4.1 Configuration of Plantypes

The following chapter contains examples showing what you can achieve with the configuration of plantypes.

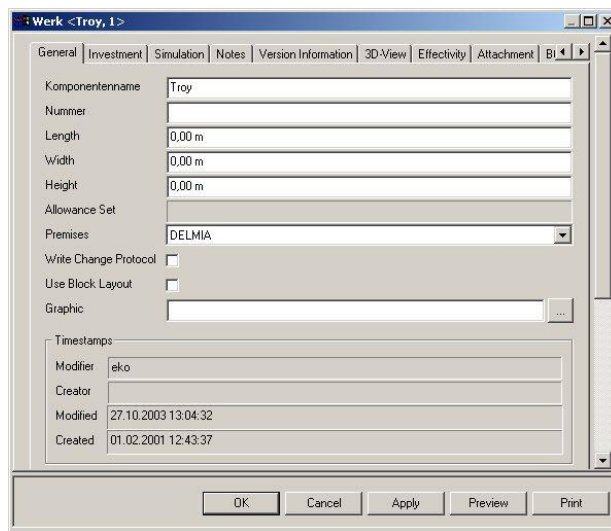
#### 3.4.1.1 Creating an Attribute

##### Example

This example shows how you can create an attribute. Create a plantype set as described in chapter [Creating Plantype Set Structure](#).

Create a project using the plantype set you created and create a plant. By right-clicking on the plant and by selecting Properties in the context menu, the following window is opened:

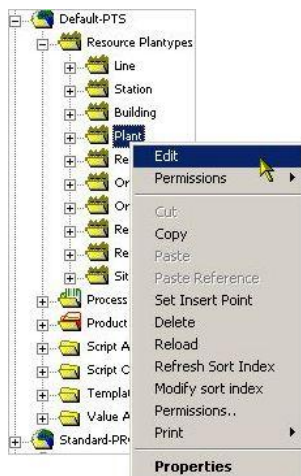




**Figure 33: Plant Properties Example**

As you can see, the entry of the plant site (for example, Untertürkheim, Ingolstadt, Wolfsburg, etc.) has not yet been included. For this purpose we are now going to define a new attribute on the plantype set level. To do this, exit the Plant Properties dialog and open the node **“New ProjectPTS”** in the project.

- 1) Open the node “Plantype Resources”.
- 2) Right-click on the plantype **“Plant”**.
- 3) Select the **“Edit”** menu item to open the Configuration Tool.



**Figure 34: Editing the Plant plantype**

The *Configuration tool* dialog opens with the plantype set you have selected in the system library.

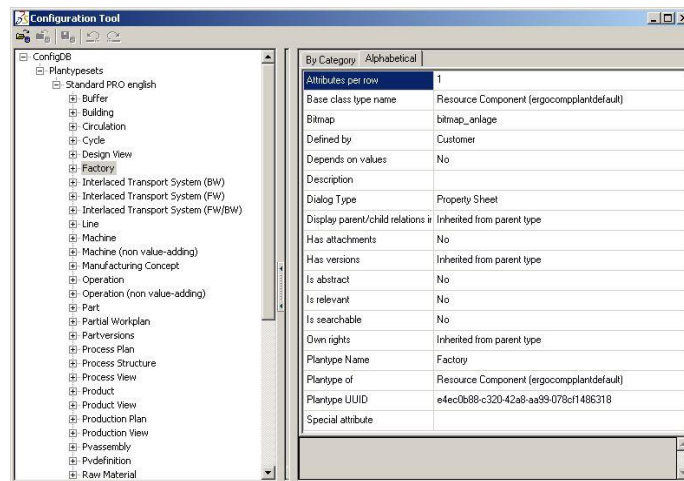


Figure 35: Configuration Tool Dialog

- 4) For the plantype Factory create a new attribute with the following properties:

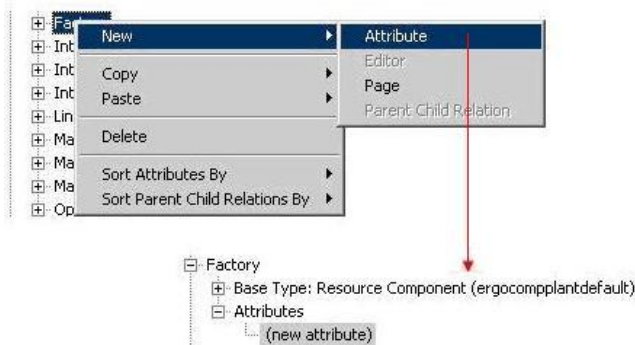


Figure 36: Creating a New Attribute

- 5) Enter the following for the properties of the new attribute. [Figure 53](#) shows the input for the properties.



- 6) Save your entries by clicking on the Save icon.
- 7) Open the Plant Properties again. You have now added a simple editor for entering a site in the Properties dialog. *Please refer to the [Figure 37](#).*

**Figure 37: Properties Dialog with “Example\_Site” Input Field**

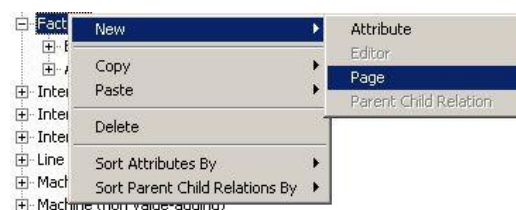
### Example

The aim of the following example is to optically delimit the newly created input field “Example site” from the other input fields with the help of a frame. In the DELMIA Process Engineer such a frame is called group. This group should furthermore be displayed on a new page (under a new tab). Proceed as follows:

- 1) First create a new page.
- 2) Create a new group for the page.

### 3.4.1.2 Creating a New Page

- 1) In the configuration tool, select the plantype Factory.
- 2) Open the context menu using the right mouse button. Select **New < Page**.



**Figure 38: Creating a New Page via the Context Menu**

- 3) The new page is shown in the Layout directory. Select the new page and set the following properties:

| By Category        | Alphabetical |
|--------------------|--------------|
| Attributes per row | 0            |
| Caption            | Site         |
| Defined by         | Customer     |
| Description        |              |
| Display in browser | Yes          |
| Display in editor  | Yes          |
| Display order      | 0            |
| Number             | 11000        |
| Visible usage data | Yes          |

Figure 39: Properties of the “Site” page

**Example**

- 4) To display the attribute site in the properties dialog of the plantype Factory in a separate group, create a new group for the new page (*Please refer to the Figure 39*).
- 5) Open the Layout directory and select the new page site.
- 6) With the right mouse button open the context menu. Select **New < Group**.

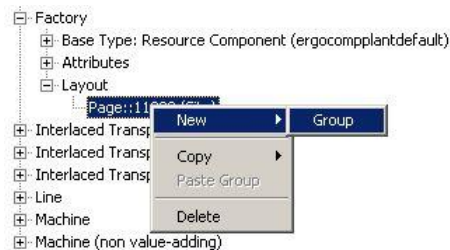


Figure 40: Creating a New Group via the Context Menu



- 7) The new group is displayed in the Layout directory under the page *Site*.
- 8) Select the group and set the following properties:

| By Category   | Alphabetical       |
|---------------|--------------------|
| Caption       | Site               |
| Defined by    | Customer           |
| Description   |                    |
| Display order | 0                  |
| Number        | 10100              |
| Page          | Page::11000 (Site) |

Figure 41: Properties of the “Site” Group

- Assign the *Group/page site* to the **attribute Site** previously created (*Please refer to the Figure 37*) in the field group of the properties dialog.

|       |  |
|-------|--|
| Group | Group::10100 (Site)-Page::11000 (Site) |
|-------|--|

Figure 42: Assigning the Group/Page to the Attribute Site

- 9) Right-click to open the “**Plant**” properties. The “Site” input field is now displayed in a separate tab.

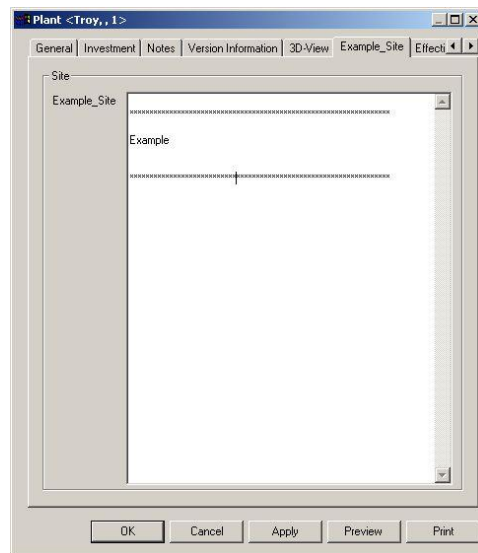


Figure 43: The “Plant” Properties Dialog with the “Site” Register

### 3.4.2 Configuration Example for Advanced Users

#### Example

In the following example it is shown what configuration possibilities you have using the inheritance mechanism in the DELMIA Process Engineer.

As already described in the chapter “Basic rules for the configuration” plantypes are derived from types. This means that they inherit the properties and the attributes of the type.

If you want an attribute of a plantype to have different properties than defined in the type you have to create an attribute with the following properties:

- “**Attribute name**” on the plantype level must match with “**Attribute name**” on the type level.
- “**Physical name**” on the plantype level must match with “**Physical name**” on the type level.
- „**Data type**“ must be identical.

All other properties can be changed without restriction. This procedure is also designated as overwriting attribute properties on the plantype level.

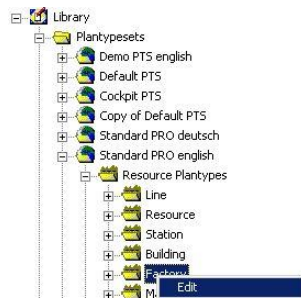
#### 3.4.2.1 Inheritance in the DELMIA Process Engineer®

#### Example

The following example shows how you can overwrite attributes in a simple manner.

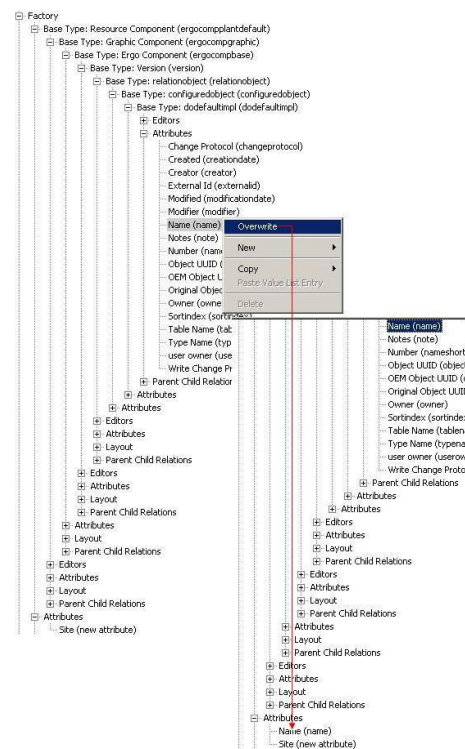
The target is to hide the “**name**” for a certain plantype – in the example, the name for the plantype *Factory* is hidden.

- 1) To open the *Configuration tool* dialog, click the plantype *Factory* in the system library using the right mouse button, then select the menu item *Edit* in the context menu.



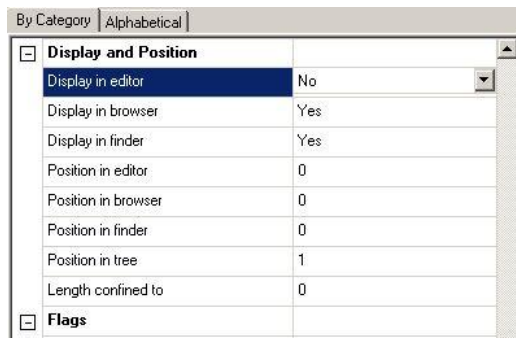
**Figure 44: Open the Configuration Tool on the Plantype Factory**

- 2) Open the hierarchy of the plantype until you find the attribute to be overwritten in the Attribute directory. In the example, the attribute *name* is defined in the *base type: dodefaultimpl*.



**Figure 45: Plantype Hierarchy**

- 3) Select the attribute *name*.
- 4) Open the context menu using the right mouse button. Select *Overwrite*.  
Upon overwriting, the attribute *name* is immediately assigned to the Attribute directory of the plantype Factory.  
Thus the attribute *name* has been passed on to Factory and can be changed here. These changes do **not** influence the original attribute.
- 5) Now select the overwritten attribute.
- 6) In the properties, set the value for *Display in editor* to *No*.



| By Category   Alphabetical                           |     |
|--|-----|
| <input type="checkbox"/> <b>Display and Position</b> |     |
| Display in editor                                    | No  |
| Display in browser                                   | Yes |
| Display in finder                                    | Yes |
| Position in editor                                   | 0   |
| Position in browser                                  | 0   |
| Position in finder                                   | 0   |
| Position in tree                                     | 1   |
| Length confined to                                   | 0   |
| <input type="checkbox"/> <b>Flags</b>                |     |

**Figure 46: Properties of Attribute Name**

7) Save the new data in the configuration tool.

From now on the name is no longer shown in the Object Properties dialog of this plantype. To stay with the example, the field name is no longer shown in the properties dialog of the plantype Factory.

## 4. The Configuration Manager

### 4.1 General Information

#### 4.1.1 Browser-based Configuration Manager

The most important to begin with: All settings made in the Configuration Manager are adopted online upon saving. It is not necessary to close the Configuration Manager to activate the changes, as was required in the previous version.

Thus you may leave the Configuration Manager open as long as you are making changes and immediately continue working in Process Engineer with the changed configuration. Changed configuration data will only become visible in Process Engineer when the view is refreshed.

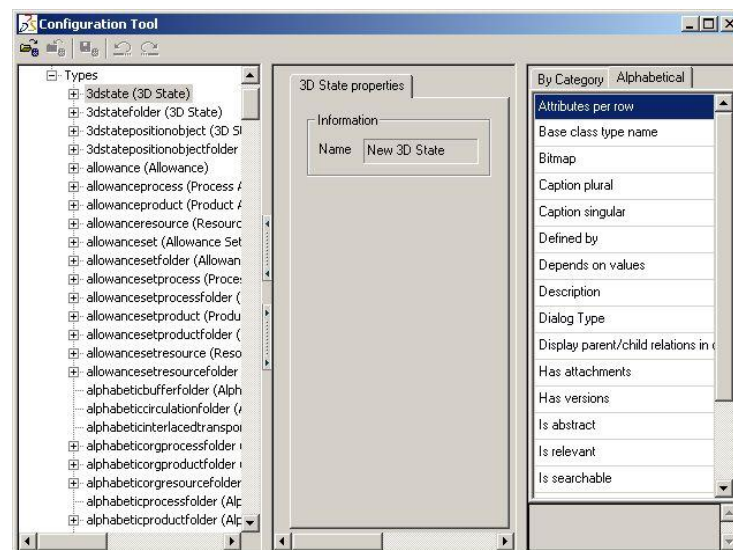


Figure 47: Configuration Tool

##### 4.1.1.1 Operating Mode

Operation and function have been improved substantially. The user guidance has been changed to the generally familiar browser technique, completely replacing the guidance by dialogs used in the previous version. Also, the functions are now easier to operate. All changes are displayed in one interface. It is no longer necessary to jump from dialog to dialog. The browser-based Configuration Manager still offers the same scope of functions.

The Configuration Manager is subdivided into three areas. Each of the three areas serves a particular function.

##### The three areas

- In the left area all types that may be used for configuration are listed. Depending on the type selected, additional information will be displayed in the other two areas. The types are sorted alphabetically.



- In the center area an abbreviated preview of the attributes, groups and pages (layout or page) is displayed, comprising the most important properties of the respective type selected.
- In the right area all properties are listed explicitly.

The layout of the three areas can be pre-defined with **Tools / Settings / Change.../Configuration Tool / Window Layout** (key combinations **Ctrl** + **F7**).

#### 4.1.1.2 Configuration of the Types

Besides the plantypes, it is possible to configure types directly in the DELMIA Process Engineer as well. This is useful if a change should have an effect on all projects and on all plantypes derived from one type.

The Configuration Manager is provided for this purpose; you can use it to edit types in multiple ways.

##### Configuration Manager

The Configuration manager is a tool *independent* of project types and plantypes used for editing of types, and especially of **all** ErgoComponents. Changes made here have an effect on all projects and all plantype sets.

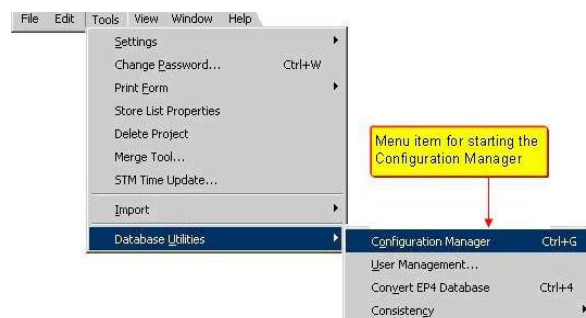


##### Caution

*Improper changes on the type level can lead to a complete loss of data. Therefore, changes should only be made by trained administrators.*

## 4.2 Opening the Configuration Manager

The Configuration Manager is started in the PPR - Navigator via the menu **Tools < Database Utilities** by selecting the menu item "Configuration Manager" or by pressing the keys **Ctrl** + **G**.








**Figure 48: Opening the Configuration Manager**

The **Configuration manager** offers a large set of utilities for editing of configurations. The individual tools for editing configurations and types are described in detail below. Please refer to the [Figure 47](#) and [Table 8](#).

### 4.2.1 Tool Bar

Table 8: Tool Bar Description

| Icon  | Description  |
|---|--|
|    | <b>Insert configuration</b><br>This Icon allows to open an INI or XML file for viewing and editing. These two file types are used to check a configuration before importing it; e. g. when changing from Process Engineer version PE 5.xxx to 5.yyy<br>Prior to the import, such configurations have no effect on the current configuration. |
|    | <b>Close configuration</b><br>Close an INI or XML file using this icon.  |
|    | <b>Save</b><br>This icon is used to save the changes to the current configuration to the database and in the INI or XML file.<br>This icon is active only when changes have been made in the Configuration Manager or in the INI or XML file.  |
|  | <b>Undo</b><br>Undoes the last change or deletes the entry last made.  |
|  | <b>Redo, restore</b><br>Is active only when the button <i>Undo</i> has been pressed once before.<br>Undoes the action that has been triggered by pressing the button <i>Undo</i> .   |

## 4.3 Working with the Browser-based Configuration Manager

The new Configuration Manager offers the same scope of functions as the previous Configuration Manager. However, the operation has changed considerably. If you have worked with the previous Configuration Manager, you will know the operating steps. Nonetheless, the following brief summary repeats the operating procedures used in the previous Configuration Manager.

- 1) Open the Configuration Manager.
- 2) Choose the type to be edited from the list of types.
- 3) Depending on the operation to be performed on the type, select the corresponding tab (list of groups, list of pages, list of attributes, editors for the context menu or list of parent-child relations).
- 4) Call up the page to be edited, select the attribute etc. and open it for editing.

- 5) To activate the changes, all dialogs had to be closed and the Configuration Manager had to be terminated.

### Summary

All functions that could be edited in the previous Configuration Manager are also available in the new Configuration Manager. The next section explains the new user interface of the Configuration Manager.

## 4.3.1 Using Configuration Manager

The user interface of the Configuration Manager puts a powerful tool for the administration and editing of types, and especially **all** ERGOComponents, at your disposal.


- 1) For all types in the directory you can select the corresponding functions of the selected type via context menus.
- 2) Open a context menu by selecting the type in the directory and then pressing the right mouse button.
- 3) You can sort types in the directory. To select sorting, open the context menu on the top node.
- 4) From the context menu select *ConfigDB <Sort Types By*. For instance, if you select sorting by *Type Name*, the listing in the directory will correspond to the order in the previous version of the Configuration Manager. *Please refer to the [List of Types](#).*



### Caution

*“Browser” is used for internal purposes only and is deactivated in the standard version of DELMIA Process Engineer.*



- 5) In the center area of the browser a preview of the configured types is shown. You can select the preview options about the context menu:
  - Sides of this type
  - Sides of this type and additional the inherited sides
- 6) Sides of this type and additional also this one inherited
- 7) In the left area the properties of the selected type are shown. *Please refer to the [Editing the Properties of Types](#).*
- 8) The plus sign  next to the type indicates that sub-objects for this type exist in the directory.
  - New in the editing of types is that the pass-on hierarchy is displayed.
  - For instance, if you open a type in the directory, it will also be indicated from which base type the opened type has been derived. The base type always is the type from which the selected type is derived. This always is the type of the next higher hierarchical level.
  - You may continue this indefinitely until you have found the type from which all objects of this type have been derived.

- 9) The attributes, context menu entries, parent-child relations and the layout of the selected type can be edited as in the previous Configuration Manager, however, without having to open a new area - this is new.
  - 10) Compared to the previous version (Configuration Manager), the **Overwrite** function has been simplified considerably; only the attributes existing in the base type can be overwritten: For example, you can use this function to overwrite attributes, parent-child relations, pages or context menu entries (**Overwrite** context menu). *Please refer to the [Using a Value List](#).*
  - 11) Changes of the configuration take effect immediately after saving.
- You do **not** need to close the Configuration Manager to continue working in PE.




### Note

*In the new Configuration Manager it is no longer possible to copy the properties of an attribute.*

## 4.4 List of Types

After the initial start of the Configuration Manager, you will see the entry **ConfigDB** in the left browser area.

Under this entry the current configuration is displayed.

Under all types (objects) of the Configuration Manager that are identified by a  further types (sub-objects) are listed.

- 1) Click the plus sign  to display these types.

### Opening the configuration

- 2) Click the plus sign  to open a configuration for editing.

➤ Under the entry **ConfigDB < Types** you will find a list of all types. (This list of the browser is used for internal purposes only and is deactivated in the standard version of DELMIA Process Engineer).

### 4.4.1 The Context Menu

Most functions of the Configuration Manager are selected from the context menus.

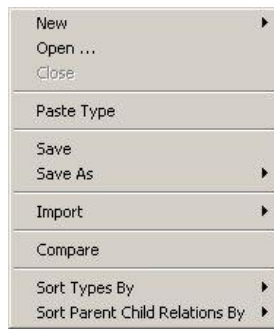


Figure 49: Context Menu

### New

The menu item *New* allows to create a new type or global attributes. Please refer to the [Global Layout Design](#). This function should be used only by an **administrator**. As a rule, you should **not** create any new types on your own.

### Open ...

The menu item *Open* allows to open an INI or XML file for viewing and editing. These two file types are used to check a configuration before importing it; e.g. when changing from Process Engineer version PE 5.xxx to 5.yyy.

Prior to the import, such configurations have no effect on the current configuration. This function is also available via an icon in the tool bar.

### Close

Close an INI or XML file using this menu item.

### Save

The menu item *Save* is used to save the changes made in the current configuration to the database and in the INI or XML file.

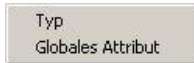
The menu item is active only when changes have been made in the Configuration Manager or in the INI or XML file.

### Save as

This menu item is used to export the configuration.

### Exporting a configuration

With the help of import and export mechanisms configurations can be exchanged between different computers.



### Note

*It is recommended to perform an export before changing a configuration. If the configuration is damaged, the original status can be restored.*

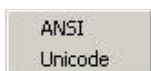


### Caution

*The ini files should not be changed manually (for example, using a text editor). If an unadmissible change has been made, this can lead to an incomplete import.*

In the DELMIA Process Engineer configurations can be saved in text files. The Configuration Manager offers a series of functions enabling the DELMIA Process Engineer database to be administered.

Two export formats are available: *ANSI* and *Unicode*:



**ANSI:** The ANSI format is used for standard installations.

**Unicode:** The Unicode format is used when additional characters are needed; e. g. if you need Japanese or Chinese characters for your work.

- 1) Once you have selected the export format, determine the export path and filename for the export file.
- 2) Click the “Open” button to save the configuration of the database in a configuration file. The file set in the input field of the Configuration Manager in the first step is used for this.

### Context menu Import Importing a configuration



#### Note

*The text files must correspond to the format of the DELMIA Process Engineer for configuration files and end in “**ini**” or “**xml**”.*

*A standard ini file named “**ergoplan.ini**” is located in the directory ...< IPDClient < Program < Bin. A standard configuration can be created with the aid of this file (e. g. in case of emergency, when the database configuration has been damaged).*

In the first step the path and name of the import file are set.

- 1) Open the context menu on the top node of the Configuration Manager and select **Import**.
- 2) Select the update mode.
  - More information concerning the update mode is found in chapter [Update Mode](#).
- 3) Select the file to be imported in the file selector.
- 4) Once you have specified the update mode and the configuration file, the selected file can be imported.
- 5) With the left mouse button click the *Open* button; the update procedure is started.
- 6) After completion of the update, quit the Configuration Manager by clicking *Close*.

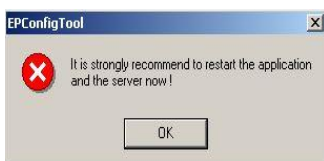
Thereafter terminate the IPD client. After restarting the IPD client the new configuration is available.

#### Update Mode

The *Configuration Manager* can be used to update an existing configuration.

- 1) For this purpose, in the input field of the Configuration Manager enter a configuration file containing update information.

In the Configuration Manager different update modes are offered:



Customer Update  
Customer Update - Make Entries Invisible  
Update All

## Customer Update

In this mode the existing configuration is preserved. The new functions contained in the update, such as new attributes, pages, groups and parent-child relations are added to the configuration.

### Customer Update - Make Entries Invisible

As in the first case, the existing configuration is preserved. However, the functions contained in the update are imported but not shown.

## Update All

In this mode the existing configuration is overwritten completely.

### Context Menu *Compare*

Different configurations can be compared to one another in version PE 5.12. The changes are displayed when two configurations are compared, but they can not be edited.

The compare function only allows to compare configurations files, but not plantype sets.

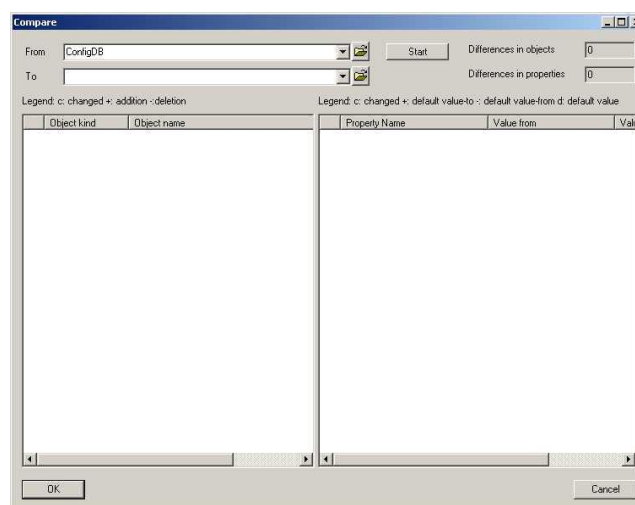
It make sense to compare the configurations, before importing a new or another configuration.

- You have several configuration files for import available and want to test them in advance.
- You want to compare the changes in the current configuration with an older version.

## Starting the Comparison

- 1) Click the menu item *Compare*.

- A dialog opens in which you can select the two configuration files to be compared with the two input fields *From* and *To*. Please refer to the [Figure 50](#).



**Figure 50: Compare Dialog when Opened for the First Time**

- 2) After you have selected the configuration files, start the comparison with the **Start** button.







### Context Menu *Sort Types By*

Via the context menu *Sort Types by* you can select the criterion for sorting of types in the browser of the Configuration Manager, selectable are:

Type Name, Prompt, Physical Name, or Base Type

The selected sorting criterion always will be displayed in first place in the respective line. The components are displayed in the browser according to the selected sorting. Sorting is alphabetic.



**Figure 52: Context Menu “Sort Types By”**

#### Sorting by Type Name

If you select **Type Name**, these will be sorted alphabetically. The Type Name identifies the name of the respective type. The name of the respective type is shown in the properties, that is, in the right display area in the Configuration Manager.

#### Sorting by Prompt

If you select **Prompt** (Heading), sorting is by the prompt (external value) of the attribute. External prompts include, for example, process components such as process or operation, product components such as subassembly or part, and resource components such as machine, line, or employee.

#### Sorting by Physical Name

If you select **Physical Name**, sorting is by the internal physical name of the attribute. The physical names of attributes include, for example, ergocompprocessdefault, ergocomppgraphic, premisesfolder or types beginning with XDO.

#### Sorting by Base Type

Types are normally derived from base types (base class). The base type is in first place in the respective line, the corresponding type is always put in parentheses. Types that appear merely in parentheses in this sorted listing, do not have a base type.

## 4.5 Editing Types

### 4.5.1 The Context Menu

For a type you can define new attributes, sides, editors and parents child relations.

This function should be used only by an **administrator**.



**Figure 53: Context Menu for Types**




## Note

*Deletion of types is deactivated as standard in the customer version of DELMIA Process Engineer.*

It is not possible to delete new pages and parent-child relations. Attributes and editors can not be created via the context menu.

## 4.5.2 Editing Types

- 1) To edit a type, open the list by double-clicking the entry *Types*, or click the  sign.

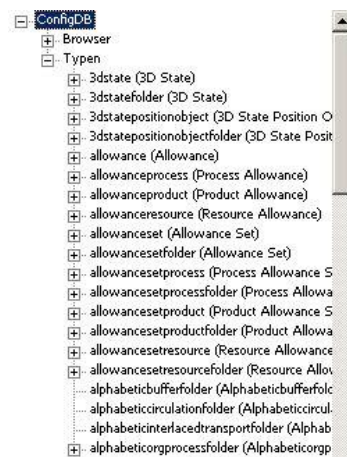


Figure 54: List of Types

- 2) In the list select the type to be edited.



Figure 55: Select Type

Each type may have up to five sub-entries. These are:

- **The base type:** The base class of the type is specified in this field. The *base class* is the type from which the selected type was derived. When you open the base type, its base class is displayed. You may continue this indefinitely until you have found the type from which all objects of this type have been derived.
- **The editors, list of menu items:** This list is used to control the context menu entries of this type. From the base types, you can display all available menu items for this type with *Overwrite*. You can change, display, or hide these context menu entries for the plantypes as well as set them as standard entries. New context menu entries can not be created here.
- **List of attributes:** Here, you can edit the attributes of a type.

- **Layout, pages and groups:** In the list of pages you can edit or create type-specific pages and groups.
- **Parent-child relations:** All pre-defined parent-child relations (PPR relations) are displayed here. A parent-child relation is a relation between two types. You can edit existing relations (in which view and under what name they are to be displayed) or define new parent-child relations. These relations determine which types can be assigned to one another.

To configure DELMIA Process Engineer®, not all types must or should be edited. The editable components most frequently used begin with the type designation “ergocomp”. Ergoitems are an exception from this rule.

All editable components start with the type name “ergocomp”.

In the following table the most frequently used components are listed and explained again.

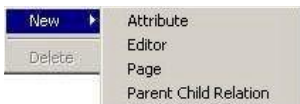
**Table 9: Editable Process Engineer Items**

| Type Name              | Physical Name              | Meaning   |
|------------------------|----------------------------|---|
| ergoitem               | XDOErgoitem                | Ergoitem, item for short, is a class, which stores additional information belonging to a component (XDOErgoCompBase), project, and a BOM entry (XDOSubCompltem).  |
| ergocompbase           | XDOErgoCompBase            | The class XDOErgoCompBase is the base class for all so-called ERGO-Components, components for short.<br><br>Each component references a plan type (XDOErgoPlanType), the project it belongs to and has a list of ERGOItems (see XDOErgoitem) such as cost analyses, work positions etc. |
| ergocompgraphic        | XDOErgoCompGraphik         | All attributes of the graphic are defined here (for example, system items).   |
| Ergocomporgplant       | XDOErgoCompOrgRessource    | Attributes of the organizational resources are defined here.  |
| Ergocomporgprocess     | XDOErgoCompOrgProcess      | The organizational processes are defined here.  |
| Ergocomporgproduct     | XDOErgoCompOrgProduct      | The organizational product nodes can be edited here.  |
| Ergocompplantdefault   | XDOErgoCompPlantDefault    | Attributes of a resource are defined here.  |
| Ergocompprocessdefault | XDOErgoCompProcess Default | Attributes of a process are defined here.   |
| Ergocompproductdefault | XDOErgoCompProductDefault  | Attributes of a product are defined here.   |

#### 4.5.2.1 Working with Types

##### New

For a type you can define new attributes, sides, editors and parents child relations.



### Overwrite

If you have used the previous version of the Configuration Manager, you will be familiar with the Overwrite function.

- In it the attributes and relations of groups, pages and editors could be overwritten directly.
- The Overwrite button was always active.

➤ Clicking displayed a list of all objects available for overwriting.

This function has been changed:

In the new Configuration Manager you need to:

- Select the object to be overwritten from the base type or the base types
- Open the context menu and select *Overwrite*
- The result is displayed under groups, pages, or editors

### Change

Changes can be undone

- However, this is possible only if you have not saved between individual changes.
- Undoing works like stack management. Please note here as well: As long as one action does not exclude another, the stack can be processed.

### Processing

The context menu functions are available in the browser on different hierarchical levels.



### Note

*You can open the context menu only on an individual object. The context menu is not available for individual directories of the respective types in the browser, such as for the sub-entries of the types for attributes, editors, parent-child relations.*

- Edit the properties of an object in the right display area of the Configuration Manager.
- Objects of both types and base types can be edited.
- Newly created pages or parent-child relations are displayed in the respective directories of the types in the browser and can be edited there.





#### 4.5.2.2 Editing the Properties of Types






To display the properties of an object, select the respective object in the browser. The respective properties of the selected type are shown in the right display area of the Configuration Manager.







|    |  |                               |
|----|--|-------------------------------|
| 1  | Attributes per row                         | 1                             |
| 2  | Base class type name                       | dodefaultimpl (dodefaultimpl) |
| 3  | Bitmap                                     |                               |
| 4  | Caption plural                             | Plantypes                     |
| 5  | Caption singular                           | Plantype                      |
| 6  | CCZ Membership                             | Inherited from parent type    |
| 7  | Defined by                                 | Delmia                        |
| 8  | Depends on values                          | No                            |
| 9  | Description                                |                               |
| 10 | Dialog Type                                | Property Sheet                |
| 11 | Display parent/child relations in dbeditor | Inherited from parent type    |
| 12 | Has attachments                            | No                            |
| 13 | Has versions                               | Inherited from parent type    |
| 14 | Is abstract                                | Yes                           |
| 15 | Is configured                              | No                            |
| 16 | Is relevant                                | No                            |
| 17 | Is searchable                              | Yes                           |
| 18 | Name                                       | ergoplanotype                 |
| 19 | Own rights                                 | Yes                           |
| 20 | Physical name                              | XDOErgoPlanType               |
| 21 | Special attribute                          |                               |

**Figure 56: The Properties of a Type**  
**Description of the Properties**


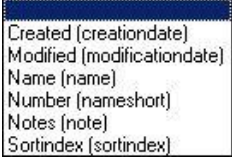
**Table 10: Description of Type Properties**

| No. | Property  | Meaning  |
|-----|---|--|
| 1   | Attributes per row  | Number of rows in the Properties dialog of an object of this type. You can enter values in this field if you have selected "Dialog" as the value for "Dialog type". Also see No. 9 under dialog type in the table.   |
| 2   | Base class type name<br> | The base class of the type is specified in this field. Clicking once in the field opens a drop-down menu, from which the base class may be selected by clicking with the mouse.<br><br><i>Note: This function is disabled in the standard version of DELMIA Process Engineer.</i>   |
| 3   | Bitmap<br>               | Standard icon that is used for objects of this type in the tree structure. Change the bitmap by clicking the button <i>Select</i> with the left mouse button. (Details see chapter "Assigning bit-maps").  |
| 4   | Caption plural  | Caption in plural (e. g., "resource components"). This caption is used for screen output when several of the objects of this type exist.   |
| 5   | Caption   | Caption (e. g. "resource component"). This caption is used for screen output.  |
| 6   | CCZ Membership<br>       | Inherited from parent type = undefined, the state of basetype is deciding.<br><br>No = Default setting is No, new versions can be created.<br><br>CCZ Owner = You can only create versions for PPR-components, which are indicated with option <i>CCZ Owner</i> or <i>No CCZ Owner</i> . With creating a new version of a CCZ Owner, new versions of its <i>CCZ Members</i> are created automatically.****<br><br>CCZ Member = When a component is customized as |

| No. | Property  | Meaning  |
|-----|---|--|
|     |   | <p>member of a CCZ, attempts to check out this component, delete, or modify its planning state will be rejected with an appropriate error message.</p> <p>The standard versioning behaviour with/without Manufacturing Change Management will retain unmodified. <i>Please refer to the <a href="#">Has versions</a>.</i></p>  |
| 7   | Defined by  | <p>Specifies whether this type or plantype has been created by a user. Needed to distinguish from internal program types or plantypes.</p> <p>This is especially important for an upgrade since user-defined plantypes are not changed for a software upgrade, plantype set update, or import, depending on the setting.</p>  <p><b>Note:</b> This function is disabled in the standard version of DELMIA Process Engineer and it is executed by the program. The value for new plantypes is always set to 'customer'.</p>  |
| 8   | Depends on values   | If a special attribute (also see No. 75) has been selected for display properties, define here, whether it is to be used (Yes), or if the standard sections are to be used (No).   |
| 9   | Description   | <p>This field can be used to enter a description of the configuration of this type or plantype. It can be used as a documentation field for these types or plantypes.</p> <p>For example, who created this plantype and why it was created can be set here.</p> <p>Or you can use the description field as a log of changes.</p>   |
| 10  | Dialog Type<br>                                | <p>Defines the type of <i>Properties</i> dialog.</p> <p>Selecting "Dialog" results in a "simple" dialog.</p> <p>Selecting "Property Sheet" results in a dialog with multiple "Properties" (with multiple tabs).</p>  |
| 11  | Display parent/child relations in dbeditor<br> | <p>Using this function you can display relations visible in the object list in the Properties dialog of an object, too.</p> <p>No = do not display</p> <p>Yes = display</p> <p><i>Inherited from parent type</i> = display relations as defined in the base type</p>   |
| 12  | Has attachments<br>                            | <p>If "Yes" is selected, an attachment page is added to all objects of this type.</p> <p>If "No" is selected, no attachment page is added to all objects of this type.</p>   |
| 13  | Has versions<br>                               | <p>Specifies whether this type is allowed to have versions or not.</p> <p>Whether the property versions is created or not can be inherited by the respective parent type or parent plantype. The property <i>Has versions</i> must be set to "Inherited from parent type".</p> <p>Only plantypes that can have versions are taken into consideration in the change management. If a plantype is to be taken into consideration by the change management, the property <i>Has versions</i> must be set to "Yes".</p> <p>When using the Engineering Hub to Manufacturing Hub Connection (kurz EH to MH), the product plantypes must <i>not</i> be versionable. The property <i>Has versions</i> must be set to "No" for these plantypes. The consequence of this is that the change management no longer takes these product plantypes into consideration.</p> |

| No. | Property   | Meaning   |
|-----|--|---|
|     |  | <p>If „Only Planning State Support“ is selected, you can set planning state outside from versions mechanisms.</p> <p>The settings you make under <i>Has versions</i> take immediate effect on the function <i>Is configured</i> (no. 15).</p>   |
| 14  | Is abstract<br>     | <p>Marks this type as abstract class. This mark is used for internal purposes and should not be changed.</p> <p><b>Note:</b> This function is disabled in the standard version of DELMIA Process Engineer.</p>  |
| 15  | Is configured<br>  | <p>This function refers to the change management and specifies if and how the type or plantype for the change management is used.</p> <p>Following possibilities are available:</p> <p>No = no change management</p> <p>Yes = for change management; standard configuration mode</p> <p>Max. Configured = for change management; maximal configuration mode</p> <p>Besides the function <i>Is configured</i>, the function no.12 <i>Has versions</i> Owns Version is also important.</p> <p>If plantypes are not configurable, so an extended effectivity can not assign to them.</p> <p>In case of standard configuration we differentiate between plantypes, which are versionable or not.</p> <p>For versionable plantypes new objects or checked out version receive the validity from the instruction for modification.</p> <p>During check in of a version, the extended effectivity is subjected to each possible version with a planning state, which agrees with effectiveness from the selected instruction for modification. There is no special operation for not versionable plantypes, but which are configured. This requires a modification assignment, which is selected when these objects are created.</p> <p>Maximal configuration means, that plantypes have either “always” or “never valid” extended effectivities.</p> <p>Maximal configured plantypes don't support parallel versions.</p> |
| 16  | Is relevant<br>   | <p>When this option is marked, a protocol of changes for objects of this type is created (Yes), or not created (No).</p>  |
| 17  | Is searchable<br> | <p>You can specify here whether you can search for this type using the Finder.</p>  |
| 18  | Name<br>          | <p>The designation of the type is indicated here.</p> <p><b>Note:</b> This function is deactivated in the standard version of DELMIA Process Engineers. Only for new types or plantypes a new name can be registered.</p>   |
| 19  | Own rights<br>    | <p>Using this function you can see whether rights have been assigned for a type.</p> <p>There are three different options for the checkbox:</p> <p>No = no rights can be assigned or linked to this type.</p>   |



| No. | Property   | Meaning   |
|-----|--|---|
|     |  | Yes = rights are assigned to this type<br><i>Inherited from</i><br><i>parent type</i> = undefined; the status of the base type determines the setting.  |
| 20  | Physical name  | Here, the physical name of the type is specified.<br><br><i>Note: This function is disabled in the standard version of DELMIA Process Engineer.</i>  |
| 21  | Special attribute<br> | Using this function you can select an attribute with special properties and an own bitmap. If you have created an attribute with a value list and have assigned a bitmap to every entry in the value list, you can specify the attribute here. The corresponding bitmap is then shown in the object structure of the PPR Navigator depending on the entry in the attribute.<br><br>In this way, for example, it is possible to display the status of a data object in the object structure. |

#### 4.5.2.3 Assigning Bitmaps

If you do not like the bitmaps used to display objects of a certain plantype in the tree structure, you can assign different bitmaps to this plantype in the **"Plantype Information"** tab. To do this, click the **Select** button (*Please refer to the Table 10 by bitmap*). The **"Select icon"** window opens, where you can select or import a bitmap. The same applies to the Manufacturing Concept.

When creating or importing a bitmap the following must be observed:

- Small bitmaps must have the size 18 x 17 pixels (width x height), large bitmaps must have the size 30-90 x 30-90 pixels.
- The bitmaps must be saved with up to 256 colors. If the number of colors exceeds this value, then it is not possible to perform the import; if the number of colors is smaller than this value, only the file name is displayed and not the icon.

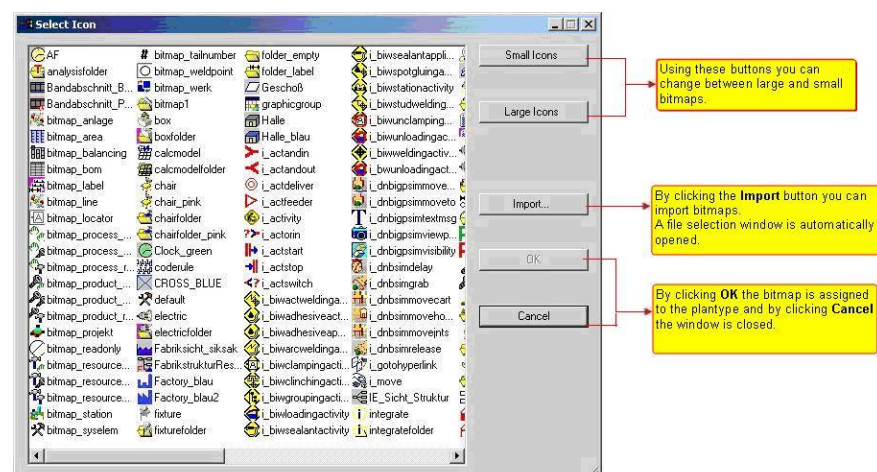


Figure 57: Select Bitmap Dialog



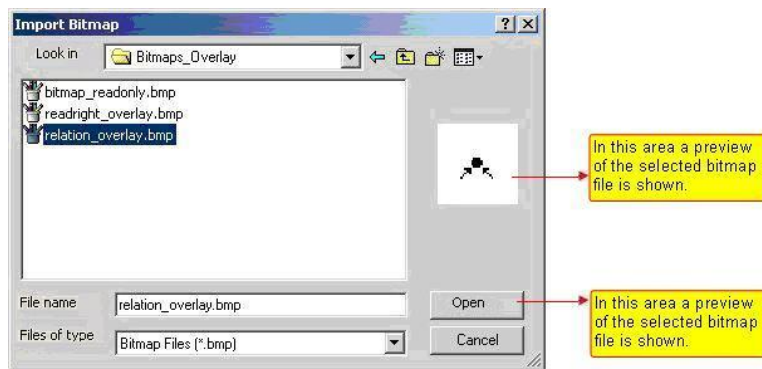


Figure 58: Importing a Bitmap

#### 4.5.2.4 Transparent Colors



##### Caution

The first pixel of a bitmap is the pixel at the **LOWER LEFT**. The color of the first pixel of a bitmap is interpreted as a **transparent** color. This means that all pixels with this color will be displayed in DPE with the respective background color of the program. In the standard configuration "magenta" is defined as the transparent color in DPE. The following example as well as the section [EPBitmapChanger](#) provide more detailed information on this topic.

Table 11: Example for Transparent Colors

| Bitmap | Display in Browser |
|--------|--------------------|
|        |                    |
|        |                    |
|        |                    |

#### 4.5.2.5 EPBitmapChanger

The program *EPBitmapChanger* replaces the first pixel (**lower left**) with the color "magenta" in all bitmaps that are saved in a pre-defined directory.

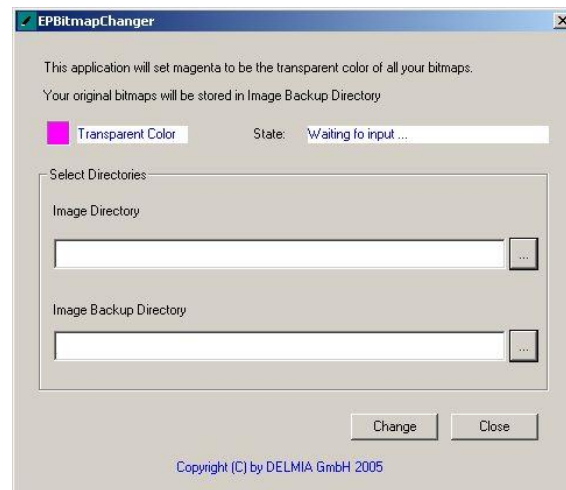
An image is scanned and saved line by line. This is started in the lower left corner. The image files contain the number of the palette entry (except for 24 bits) -- this means that it is impossible to know anything about the colors without knowing the palette. In 24 bit bitmaps, the information consists of one pixel with 3 individual bytes each -- they immediately specify the corresponding color intensities (red, green, and blue values). Therefore 24 bit bitmaps do not contain any transparency information.

If bitmaps are saved as 24 bit bitmaps, the transparency of the bitmaps will not be displayed correctly.



##### Executing the program *EPBitmapChanger*

- 1) Open the directory your DELMIA Process Engineer® Client installation.  
**\\DELMIA\\PPRClient\\program\\bin**
- 2) Start the application **EPBitmapChanger32.exe**.
- 3) The *EPBitmapChanger* opens.



**Figure 59: EPBitmap Changer**

In the standard configuration "magenta" is defined as the transparent color in DPE.

The current program step is displayed under **State**. The following program steps are possible:

- Waiting for input...: Waiting for the selection of the bitmap and backup directory.
- Backup images: A backup of the bitmap directory is created.
- Changing images: The bitmaps from the bitmap directory are cycled through and the first pixel of every bitmap is set to "magenta".
- Operation successfully done: The process has been successfully completed.
- Errors occurred: Errors have occurred.



- 4) Select the directory of which the bitmaps are to be changed. The standard directory for bitmaps can be found under **\\DELMIA\PPRClient\data\Bitmaps\**.



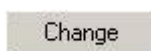
- 5) Select the backup directory.



### Note

*Before starting the EPBitmapChanger, check to ensure that the following prerequisites for execution have been fulfilled:*

- 1) *The backup directory must exist and must not be write-protected (complete access is necessary).*
- 2) *Complete access to the bitmap directory is also necessary (all bitmaps in the bitmap directory must not be write-protected).*
- 3) *There must be enough storage space on the drive with the backup directory in order to accommodate the entire bitmap directory.*



- 6) Use **Change** in order to start changing the bitmaps.

- All bitmaps that are located in the bitmap directory are changed.


Close

- All bitmaps from the bitmap directory are copied to the backup directory before they are changed. If the path is not specified, a backup of the original bitmap is not created.
  - The transparent color is set to "Magenta".
  - Subdirectories are not taken into consideration.
- 7) Leave the *EPBitmapChangers* via **Close**.

### 4.5.3 Editors for the Context Menu

The menu items for a context menu are defined for the types, under the editors, in the Configuration Manager. Note, that context functions are not provided for all types, and that for such types there will not be an Editors directory. In Process Engineer, context menus are listed in the tree structure and in the lists for editing of objects.

- 1) Open a context menu by right-clicking the selected object.
- 2) To edit the context menu items, first unfold the type and than the editors in the browser.
- 3) All objects displayed in the Editors directory can be edited.
- 4) Select an object. In the right display area of the Configuration Manager the properties are shown.




|                                |                               |
|--------------------------------|-------------------------------|
| Browser ID                     | ergoexplorer-projectlibrary   |
| Browser on multi select        | No                            |
| Browser on single select       | No                            |
| Command                        | Assign OrgIDs to children     |
| Default for this type          | No                            |
| Default new editor             | No                            |
| Defined by                     | Delmia                        |
| Description                    |                               |
| Menutext                       | Assign OrgIDs to children     |
| Open in new frame              | No                            |
| Program ID                     | ERGOEXPLORER.ERGOEXPLORERView |
| Properties dialog remains open | No                            |




**Figure 60: Properties of the Editors for Context Menu Items**


#### Description of Properties - Context Menu

**Table 12: Context Menu Properties**

| Property                | Meaning  |
|-------------------------|--|
| Browser ID              | <p>Determines in which program area (browser) the menu item is available.</p>  <p><i>This field is disabled in the standard version of DELMIA Process Engineer.</i></p> |
| Browser on multi select | <p>Multiple select: When the function is activated, the function appears in the context menu and in the list view.</p>   |

**Example**

| Property  | Meaning  |
|---|--|
|   | Whereas the single selection can be enabled independently, the multiple selection is always enabled together with the single selection.  |
| Browser on single select  | <p>Single select: If activated, the function is displayed in the context menu of the object structure and in the list view (only one item selected).</p> <p>You can specify, whether the menu item is to be displayed in the context menu.</p>   |
| <p><b>Example:</b></p> <ol style="list-style-type: none"> <li>1) Select ergocompprocessdefault from the List of Types.</li> <li>2) Select the <b>Open\ Open in IGRIP</b> item from the "Editors for Popup Menus".</li> <li>3) Depending on what is enabled you will see in the context menu.</li> <li>4) For a work sequence that either is: <ul style="list-style-type: none"> <li>➤ no entry (no checkmark is set)</li> <li>➤ disabled entry, (You have only enabled the "single selection", but you have selected several entries) or</li> <li>➤ the "Open in IGRIP" entry. (You have activated "Multiple selection" and you see the entry for each type of selection)</li> </ul> </li> </ol> <div> <div>1<br/>Export to IGRIP<br/>Open (new frame)<br/>Open (this frame)<br/>Open in VS<br/>Open in VNC</div> <div>2<br/>Export to IGRIP<br/>Open (new frame)<br/>Open (this frame)<br/>Open in IGRIP<br/>Open in VS<br/>Open in VNC</div> <div>3<br/>Export to IGRIP<br/>Open (new frame)<br/>Open (this frame)<br/><b>Open in IGRIP</b><br/>Open in VS<br/>Open in VNC</div> </div> |  |
| Command   | <p>A fixed menu text is used within the software to clearly identify the functionality to be performed.</p>  <p><i>Note: This field is disabled in the standard version of DELMIA Process Engineer.</i></p>           |
| Default for this type   | <p>Default for this type</p> <p>Using this option you can automatically determine, whether the editor should be started automatically by double-clicking the object, and whether the editor is to be opened in a new frame.</p>  |
| Default new editor  | <p>With this option you can specify whether the editor entry should be started automatically when <i>New</i> is selected while the object is selected.</p>   |
| Defined by  | <p>Specifies whether this type has been created by a user. Needed to distinguish from internal program types.</p>  <p><i>Note: This function is disabled in the standard version of DELMIA Process Engineers.</i></p> |
| Description   | <p>This field can be used to enter a description of this context menu's configuration. You can use this field for documentation purposes regarding this context menu as well as any possible changes.</p>  |
| Menutext  | <p>The text displayed in the context menu.</p>   |
| Open in new frame   |  <p><i>Note: The function "Open in new frame" is disabled in the standard version of DELMIA Process Engineer.</i></p>   |

| Property                       | Meaning  |
|--------------------------------|--|
| Program ID                     | <p>The editor to be executed when selecting the menu item is entered here.</p>  <p><i>Note: This field is disabled in the standard version of DELMIA Process Engineers.</i></p>   |
| Properties dialog remains open | <p>This function is normally disabled. It is only enabled if another dialog is called from a properties dialog opened via the context menu item.</p> <p>What is the effect of this function?</p> <p>For example, if you enable the “Car Body Position” button in the properties dialog of a process, another dialog in which you can set the car body position opens.</p> <p>If the function “Properties dialog remains open” is activated, the properties dialog of the process remains open. After leaving the car body position dialog, you will be returned to the properties dialog of the process.</p> <p>If the checkbox “Properties dialog remains open” is not enabled, only the dialog for setting the car body position will be opened and the properties dialog of the process will be closed. Both dialogs are closed upon exiting this dialog.</p> |

## 4.5.4 Editing Attributes of a Type

The list of attributes of a type is identical with the list already described on the **plant type** level. Please refer to the [Creating an Attribute](#).

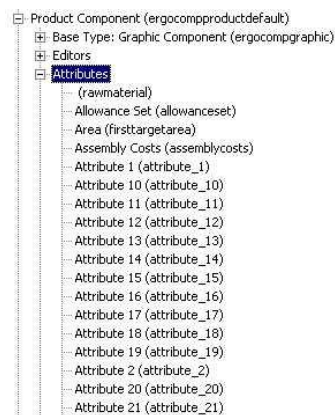


Figure 61: Configuration List of Attributes

### 4.5.4.1 New Attributes

The following restrictions apply to editing new attributes of a type:

- Editing of a “**Physical name**” is disabled
- Editing of “**Defined by**” is disabled
- Editing of “**In AVSet**” is disabled

### 4.5.4.2 Free Attributes

The following restrictions apply to editing attributes of a type:

- Editing of an “**attribute name**” is disabled
- Editing of a “**Physical name**” is disabled
- Editing of a “**Data type**” is disabled
- Editing of “**Is read only for scripts**” is disabled
- Editing of “**Defined by**” is disabled
- Editing of “**In AVSet**” is disabled

In [Figure 61](#) you can see the attributes with the names *attribute\_1* to *attribute\_55* and designations attribute 1 - attribute 55. These are the so-called “free”, “unused” attributes.

### Free Attributes

These attributes are designed to be overwritten in a plantype. Apart from the string attributes *attribute\_1* to *attribute\_55* there are also the double attributes *dbl\_attribute\_1* to *dbl\_attribute\_10*, i.e. all attributes which, for example, belong to the type *ergocompprocessdefault*.

As standard, these attributes belong to a technical process object. In the database all these attributes are listed in **one** table. These attributes are automatically read when process data is to be accessed, for instance, to get the name and number. The number of attributes varies among the different types. These attributes can and should be configured arbitrarily.

Since process objects can vary (e.g. test process, process non-value adding, process value adding, etc.), you have the opportunity to overwrite the attribute previously configured on the type *ergocompprocessdefault* when editing the plantype set. New attributes on the plantype must be defined only if more attributes are required than are available as free attributes in the type.

### Advantages of Overwriting of Free Attributes

You need to edit the attributes only once and then can re-use them in the various plantypes at any time.

Database access to the overwritten attributes is faster than access to newly defined attributes, since newly defined attributes are saved in a separate table in the database. Therefore, two tables must always be accessed whenever a process object with new attributes is accessed.

#### 4.5.4.3 Unique Keys

One or several attributes where the property ‘Is Key’ has the value ‘Yes’ and the key scope is defined. The key for an object in DELMIA IPD consists of all key attributes of the Plantype and is unique in its scope. User-Defined Attributes can NEVER be key attributes.

### Conditions

You can define the effectivity of a key attribute via the property *Key Scope*.

- **Global:** The value of the attribute of this type must exist only once in the database
- **Project:** The value of the attribute must exist only once in the project
- **Parent:** The value of this attribute must exist only once for all objects of this plantype.

There can only be defined one key scope per object type.

### Key Attribute

| Key              | Constraints |
|------------------|-------------|
| Is generated key | No          |
| Is key           | Yes         |
| Key Scope        |             |

The key concept is customization based. This means you can customize which objects and which attribute(s) must be unique. On the other hand, the key concept is type based. This means, only the objects of the same type can be checked for uniqueness. The maximum scope 'global' are all objects of a specific type.

In order to define a key attribute, select an attribute and set the property "Is Key" to 'Yes'. You thus have an attribute that is unambiguous for either all objects of this type in the database, all object of this type in the project, or all objects of this plantype.

This key attribute is always checked when an object of this type is created or the object is changed.

Whenever you create an object without modifying the key attribute, the key attribute has a standard value. This works only when an object is created for the first time; you can not again create an object of this type or plantype since the value (default value) already exists in the database.

In this case you can either modify the key attribute or you can set the property "generated key" to 'Yes'.

A key attribute is generated when

- creating or
  - copying
- an object of this type .

You can define several attributes as key attributes (by enabling the isKey-Flag).

The key then consists of all the attributes taken as a whole. To guarantee the uniqueness of the key, only one attribute has to differ between objects.

Whenever you copy or create a supply project, the objects keep their existing key with the defined keys and meaning "Project".

If the defined key scope is "global" and there is a key attribute defined as generated key attribute, new keys will be generated.

### **What is the scope "Parent" exactly and what is the difference to "Project"?**

The meaning of scope 'parent' depends on the type at which the key attribute is defined.

For components, the scope parent means 'all components of this **plantype** in this project' beginning with DPE 5.15.

For components, the scope project means 'all components of this **type** (e.g. all products) in this project'.

In previous releases the scope 'parent' was undefined and behaved like scope 'project'.

### **Example**

#### **Example:**

For example: overwrite the attribute 'name' at plantype 'Part' and at plantype 'Assembly', both productdefault.

At part set it to it 'is Key' = true and scope to 'project'. At assembly it 'is key' = false.

If you modify the name at an assembly, there is no check at all.

If you modify the name at a part, there is a check done on uniqueness over all parts, assemblies and other products in the project.



Now at plantype 'Part' set the scope of key attribute 'name' to 'parent'. If you modify the name at a part now, there is a check done on uniqueness over all parts.

### Restrictions of the Key Mechanism

When the project name (and the same applies to other entities and attributes) is defined having properties "Is Key" = Yes but

"Is Generated Key" = False,

then it is actually possible to create entities with the same default attribute value.

### Reason

On creation of a new entity, the properties of the new entity are initialized by the application server **without** checking key constraints.

Then **initially** the properties sheet in the client opens and if you do not change any property and just press OK, actually **nothing** is stored, because nothing has changed on the client side.

The properties sheet compares the old and new values and stores only the differences!

How to overcome the problem?

a) Set "Is Generated Key" = true: Then the server computes a unique value for the attribute, **before** the properties sheet is opened.

b) Set "Mandatory Value" = True and Default Value = "" (Empty).

This forces the user to provide a non-empty value in the properties sheet, which in turn means a difference between old and new value.

This difference has to be stored by pressing OK - and hence the constraint check (again server-side) will be applied.

**Table 13: Properties of an Attribute**


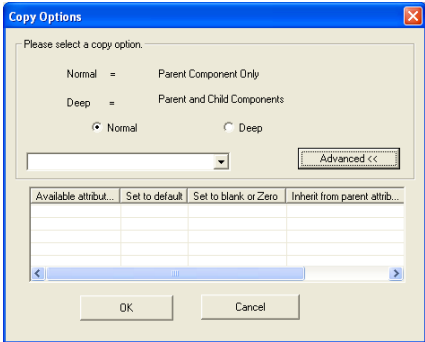
|      | Attributes           | Properties                         |
|------|----------------------|------------------------------------|
| [-]  | <b>Basics</b>        |                                    |
| (1)  | Name                 | attribute_1                        |
| (2)  | Physical name        | m_strAttribute_1                   |
| (3)  | Prompt               | Attribute 1                        |
| (4)  | Data Type            | String                             |
| (5)  | Control type         | Edit                               |
| (6)  | Group                | Group: 2400()-Page::2100 (Details) |
| (7)  | Description          |                                    |
| (8)  | Use master           | No                                 |
| [-]  | Data Type Definition |                                    |
| (9)  | Length               | 32                                 |
| (10) | Precision            | 2                                  |
| (11) | Unit category        |                                    |

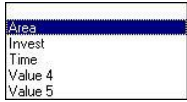
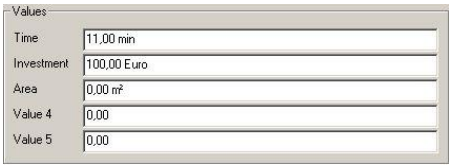
|      | Attributes                           | Properties                      |
|------|--------------------------------------|---------------------------------|
| (12) | Unit                                 |                                 |
| (13) | Integer min                          | 0                               |
| (14) | Integer max                          | 2147483647                      |
| (15) | Double min                           | 0.0000000000000000              |
| (16) | Extended copy                        | No                              |
| (17) | Double max                           | 999999999999.999900800000000000 |
| (18) | Default value                        |                                 |
| (19) | Copy prefix                          |                                 |
| [-]  | <b>Data Type Reference (Pointer)</b> |                                 |
| (20) | Type Attribute Name                  |                                 |
| (21) | Value scope                          |                                 |
| (22) | Target                               |                                 |
| [-]  | Display and Position                 |                                 |
| (23) | Display in editor                    | Yes                             |
| (24) | Display in browser                   | No                              |
| (25) | Display In finder                    | No                              |
| (26) | Position in editor                   | 2401                            |
| (27) | Position in browser                  | 0                               |
| (28) | Position in finder                   | 0                               |
| (29) | Position in tree                     | 0                               |
| (30) | Length confined to                   | 0                               |
| [-]  | <b>Flags</b>                         |                                 |
| (31) | Is read only                         | No                              |
| (32) | Is read only for scripts             | No                              |
| (33) | Is printable                         | Yes                             |
| (34) | Change in integrate state            | No                              |
| (35) | Change in release state              | No                              |
| (36) | Is transient                         | No                              |
| (37) | In AVSet                             | No                              |
| (38) | Mandatory value                      | No                              |
| (39) | Is relevant                          | No                              |
| (40) | Own Rights                           | No                              |

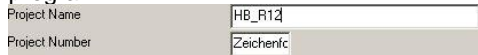


|      | Attributes              | Properties |
|------|-------------------------|------------|
| (41) | Constraint[09AZ]        | Yes        |
| (42) | Defined by              | DELMIA     |
| [-]  | <b>Permission</b>       |            |
| (43) | Used for permission     | No         |
| (44) | Mask value              |            |
| [-]  | <b>Key Constraints</b>  |            |
| `    | Is generated key        | No         |
| (46) | Is key                  | No         |
| (47) | Key Scope               |            |
| [-]  | <b>Assigned Buttons</b> |            |
| (48) | Can have Macro assigned | No         |
| (49) | Position                | Bottom     |





Table 14: Properties of an Attribute

|   | Property      | Meaning  |
|---|---------------|--|
|   | <b>Basics</b> |  |
| 1 | Name          | <p>This value always must be unique. If the "Attributname" is already used in the database, you cannot save the properties of this attribute.</p> <p>Special characters as well as blanks shouldn't be used for <i>new</i> attributes. Should a name or an attribute have two or more words, these words should be linked with an underscore „ _ „. For example: Floor_4711</p>  |
| 2 | Physical name | This field is always disabled and is used for internal purposes by DELMIA Process Engineer®.   |
| 3 | Prompt        | Is the prompt of an input field in the properties dialog of an object.   |
| 4 | Data Type     | <p>With this property you can define the data type (string, integer, float value, etc.) of the attribute.</p> <p>The single data type are explained in the Appendix <a href="#">Data Types</a>.</p>  |
| 5 | Control Type  | Display type of the input field: Selection list, option field, etc. The single types are explained in the Appendix <a href="#">Control Type</a>  |
| 6 | Group - Page  | Group in which the field will be shown in the properties dialog (only already existing groups can be selected).  |
| 7 | Description   | <p>In this field you can enter a description for this attribute. The description is also displayed in the properties dialog as a tool tip.</p> <p><i>Please refer to the <a href="#">Extending the Text Information for Description</a></i></p>  |
| 8 | Use master    | <p>You can define a manufacturing concept as the master for multiple usage.</p> <p>Using the Use master function you can determine whether this attribute is to be used for multiple usage (slave) or not.</p> <p>If you set this field to Yes, the value of the master ERGOcomponent is used. This attribute is disabled in the slave manufacturing concept and cannot be changed. If the value of this attribute is changed in the master manufacturing concept, it is also changed in the slave</p> |

|                             | Property           | Meaning  |
|-----------------------------|--------------------|--|
|                             |                    | <p>manufacturing concept.</p> <p>If you set this field to <i>No</i>, you can change the attribute in the multiple usage.</p>  <p><i>Note:</i> You cannot define a key attribute and an attribute with the use master property at the same time.</p>   |
| <b>Data Type Definition</b> |                    |  |
| 9                           | Length             | Maximum length (character string:) Number of characters.<br>Numbers: Number of figures.  |
| 10                          | Precision          | In the case of float values: number of decimal places.   |
| 11                          | Unit Category      | Here it is determined which units belong to the data type (e.g. currency, angle, density, power, etc.).  |
| 12                          | Unit               | Is depending on the selected unit category. Only if a unit category was selected, the unit can be indicated. The standard unit can be defined in a list <(e. g. unit category = time ->unit is minutes (min)).   |
| 13                          | Integer - Min      | The minimum value of an integer attribute is determined, here.   |
| 14                          | Integer - Max      | The maximum value of an integer attribute is determined in this field.   |
| 15                          | Double value - Min | The minimum value of a float attribute is determined, here.  |
| 16                          | Extended Copy      | <p>Possible values :</p> <p>0 =&gt; No<br/>1 =&gt; Set to default<br/>2 =&gt; Set to empty or zero<br/>3 =&gt; Inherit from parent type<br/>4 =&gt; Set to dynamic value.</p>  <p>When set to 0, attribute will not be considered for enhanced copy</p> <p>Clicking the Advanced button offers users an expanded dialog box that includes a plantype field and a grid selection table. In the plantype box, a list appears containing the plantype names of the copied objects on which enhanced copy can be performed.</p> <p>The user can select a plantype. Depending on the plantype, the grid selection table is populated with those attributes whose property "Extended copy" is set to true.</p> <p><b><u>Inherit from parent attribute:</u></b> The attribute value of the resultant object/s will be inherited from its parent object attribute. If the parent attribute does not exist then the existing default copy mechanism will work for that attribute.</p> <p><b><u>Set to default:</u></b> The attribute value of the resultant object/s will be set to default value.</p> <p><b><u>Set to blank or zero:</u></b> The attribute value of the resultant object/s will be set to blank/ zero.</p> <p>For data types that are neither "String" nor "numeric" the effect will</p> |

|                             | Property   | Meaning  |
|-----------------------------|--|--|
|                             |  | be blank (empty). Except for Date\Time type the effect is it displays Current System Date\Time.<br><b>Set to dynamic value:</b> The attribute value of resultant object/s will be set to dynamic value give by the user.   |
| 17                          | Extended Copy for Supplier Integration   | Used for Supplier integration solution.<br>0=>No<br>1=>Set to default<br>2=>Set to empty or zero   |
| 18                          | Default value  | Specifies a default value; this value is entered into the input field when creating a new object.  |
| 19                          | Copy prefix  | Prefix that is automatically added to the name of the attribute when copying an object of this plantype (for example, "Copy of").  |
| <b>Data Type Reference</b>  |  |  |
| 20                          | Type Attribute Name  | Determines that the available values for this attribute are calculated dynamically at runtime. If, for example, "ergocompprocessdefault.name" has been selected, the names of all processes are offered in a list box.   |
| 21                          | Value Scope  | Here it is specified whether the names of all processes in the data base are offered, the names of all processes in the current project or the names of all processes under the current parent node.   |
| 22                          | Target<br> | Here you specify which attribute will be evaluated as the target.<br>   |
| <b>Display and Position</b> |  |  |
| 23                          | Display in editor  | If you set this field to Yes, the attribute is shown in the properties dialog of the plantype.   |
| 24                          | Display in browser   | If you set this field to Yes, the attribute is shown in the browser (Listview).  |
| 25                          | Display in finder  | If you set this field to Yes, you can search for this attribute.   |
| 26                          | Position in editor   | Specifies the position / order for display in the group of the <i>Properties</i> dialog of an object (ascending order).  |
| 27                          | Position in browser  | Specifies the position / order for display in the group in the object properties area (ascending order).   |
| 28                          | Position in finder   | Specifies the position / order for display in the finder (ascending order).  |
| 29                          | Position in tree   | Specifies the position / order for display in the group in the object properties area (ascending order).   |
| 30                          | Length confined to   | With this function available in version PE 5.12 you can set the length of the input field. In this way you can control the appearance of your input fields.<br><br>What needs consideration?<br><br>The Datatype must be <i>String</i> , <i>Int</i> , <i>Float</i> or <i>Pointer</i> .<br><br>The type of control must be <i>Edit</i> or <i>RTF Edit</i> .<br><br>The character sequence length (point 31) can be longer than the length of the input field. The text which is not displayed can be displayed by moving the mouse or keyboard (see example: The character sequence length is 32; length is 5). |
| 31                          | Fixed height   | Stores the height of the variable height control associated with the   |

|    | Property                   | Meaning  |
|----|----------------------------|--|
|    |                            | attribute.<br>Default value is 0. Can be any valid positive integer value.<br><br>If this value is 0 then the height of control is considered to be variable.  |
|    | <b>Flags</b>               |  |
| 32 | Is read only               | The attribute is disabled in the properties dialog and values cannot be entered.   |
| 33 | Is read only for scripting | Attributes in which this function is enabled can only be read by scripts. The attribute value cannot be changed by scripts.  |
| 34 | Is printable               | It is specified here whether the attribute is printable or not. Only for new attributes is it important that the name of the new attributes hasn't includes a sign with a special character. <i>Please refer to the <a href="#">Name</a>.</i>  |
| 35 | Change in integrate state  | If you set this field to Yes and the <i>Status Complete</i> has been assigned, you can still change this attribute.  |
| 36 | Change in release state    | If you set this field to Yes and the Status Released has been assigned, you can still change this attribute.   |
| 37 | Is transient               | The value of a temporary (transient) attribute is stored during the program runtime; this information, however, is lost when exiting the program.<br>  |
| 38 | In AVSet                   | This field is processed internally. This field records the assignment of the newly created attribute to the plantype.<br><br><i>Note: This field is disabled in the standard version of DELMIA Process Engineers.</i>   |
| 39 | Mandatory Value            | If you select this field, the attribute is marked as relevant for changes. Therefore, as soon as the user makes a change to this attribute (by entering a new value), a change protocol is created for the object.   |
| 40 | Is relevant                | If you select this field with yes, the attribute is marked as relevant for changes. This means that as soon as the user makes a change to this attribute (by entering a new value) a change log is created for the object.   |
| 41 | Own rights                 | Determines whether this attribute can be controlled by user access rights or not. If you set this field to <i>yes</i> , the entry <i>Permissions...</i> is activated in the context menu for this attribute.<br><br>Via <i>Permissions</i> you can assign users and groups to this attribute and thus design user-specific layouts. <i>Please refer to the <a href="#">User Access Rights for Pages, Groups and Attributes</a>.</i>  |
| 42 | Constraint (09AZ)          | In the Properties window you can enter all letters, numbers, and special characters.<br>If an attribute property Constraint (09AZ) is "YES, the lower case letters are converted to upper case letters.  |
| 43 | Defined by                 | Specifies whether this attribute has been created by a user. Is used to differentiate internal program attributes and should therefore always be set to 'customer' for new attributes. This is important, especially for an upgrade, since user-defined attributes are not changed when upgrading.<br><br><i>Note: This function is disabled in the standard version of DELMIA Process Engineer and it is executed by the program. The value is always set to 'customer' for new attributes and overwritten</i> |

|    | Property   | Meaning   |
|----|--|---|
|    |  | <i>attributes.</i>  |
|    | <b>Permission</b>  |   |
| 44 | Use for permission   | Is not used at this time  |
| 45 | Mask value   | Is not used at this time  |
|    | <b>Key Constraints</b>   |   |
| 46 | Is generated key   | <p>If you set this field to Yes, this key attribute is used when creating or copying.</p> <p>If you have assigned several key attributes to a type, the key attribute of which the function <i>Is generated key</i> is activated is used whenever this type is created or copied.</p> <p>If there are several key attributes, only one attribute may be selected as the <i>Is generated key</i>. The attribute must be of data type 'Integer' or 'String' and of the control type 'edit', and it must have a length of at least 6 characters as well as an internal name</p>  <p><i>Note:</i> One attribute must have put "Is Generated key", the error message „No key found which can be generated. New object is not unique.“ otherwise appears. Please refer to the <a href="#">Unique Keys</a>.</p> |
| 47 | is key   | <p>A key attribute determines the unambiguousness of the uses or contents. The key attribute must be considered in the context of the entries of the field "Meaning".</p> <p>Limitation: The unambiguousness of the key can not be guaranteed if several objects with the same keys are created at the same time (Multi-user environment).</p>  <p><i>Note:</i> The creation of an object is completed only after the saving finishes. Please refer to the <a href="#">Unique Keys</a>.</p>  |
| 48 | Key Scope<br> | <p>Where is the attribute used:</p> <p>Global: The attribute value should only be available once in the whole system.</p> <p>Project: The attribute value should only be available once in the project.</p> <p>Parent: The value of this attribute must exist only once for all objects of this plantype. Please refer to the <a href="#">Unique Keys</a>.</p>  |
|    | <b>Assigned Buttons</b>  |   |
| 49 | Can have Macro assigned  | Determines whether a script can be assigned to this attribute by means of script assignment. If a script action is defined for this attribute, a new button appears next to the attribute. The position of the button is determined by the <i>Position</i> property.  |
| 50 | Position<br>  | If a script is to be executed on this page, define the position of the button here.   |

### 4.5.5 Using a Value List

For all attributes a value list can be created. A value list may have an indefinite number of entries. Attributes for which a value list has been created are identified by a plus sign (+) in the browser.



A value list is generally created for a radio button, a list box, or a combobox. Always proceed in the same way: The selection is available in the properties of an attribute in the field **Control type**.

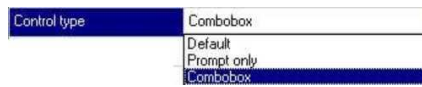


Figure 62: Properties of Attributes – Select Control Type

#### 4.5.5.1 Creating a Value List

The value list determines the individual entries to be displayed in a list of selectable values. You need to create each entry of a value list individually. The meaning of the individual entries can be selected freely.

A value list is created via the context menu of an attribute.

- 1) In the *Attribute* directory of a type or base type select the attribute that you wish to create a value list for.

➤ A new entry is created below the attribute.



Figure 63: Create Value List Entry via the Context Menu

#### Editing the properties of a value list entry

- 3) Select the entry to edit the properties.



|   |          |
|---|----------|
| <input type="checkbox"/> <b>Basics</b>  |          |
| Item name                               | name     |
| Item value                              |          |
| Description                             |          |
| <input type="checkbox"/> <b>Display</b> |          |
| Display order                           | 0        |
| Bitmap                                  |          |
| <input type="checkbox"/> <b>Flags</b>   |          |
| Is hidden                               | No       |
| Defined by                              | Customer |

Figure 64: Properties for a New Value List Entry

#### Description of the Properties

Table 15: Description of the Properties of a Value List Entry

| Property    | Meaning   |
|-------------|---|
| Item name   | In this field write the name for the value list entry to be displayed in the value list.<br>The item name is case sensitive.  |
| Item value  | The item value for the field depends on the data type of the attribute for which the value list entry is created.<br>The item value is case insensitive.<br><br>If the attribute is of the data type <b>String</b> , for instance, any value (figures and letters) can be used; for the data type <b>Integer</b> only figures can be entered as the item value. |
| Description | In this field write the description for the value list entry.   |

| Property      | Meaning   |
|---------------|---|
| Display order | In this field enter the order of the entry for display in the value list; e. g., if you enter zero, the value will always appear as the first entry, for instance when displayed in a combobox. The numbering is ascending.   |
| Bitmap        | <p>This field allows to assign a bitmap to the entry.</p> <p>To do so, click into the field and then click the  button to select the bitmap.</p> <p>Concerning use of bitmaps see <a href="#">Table 10</a> field 15.</p>   |
| Is hidden     | <p>This field controls the display of the entry in the value list, e. g., for display of entries in a combobox. As default, the field is set to <b>No</b>, the value is thus displayed.</p>  <p>The selection list allows you to hide the entry. To do so select <b>Yes</b>.</p> |

#### 4.5.5.2 Defining a Value List Entry to Display an Empty Field

For all attributes an empty value list entry can be defined which is to be displayed as empty in the value list. An empty value list entry allows to assign an empty, or undefined, value (even later on), although there are defined value list entries for the attribute as well.

The empty entry is independent of the display order; it is always displayed as the first entry for the attribute.

##### Defining an Empty Value List Entry

To define an empty value list entry, enter EMPTY in the field *Item name* in the properties of the entry. For an empty entry it is not required to enter an item value.

Proceed exactly as outlined in the previous section, when creating an empty value list entry. *Please refer to the [Creating a Value List](#).*

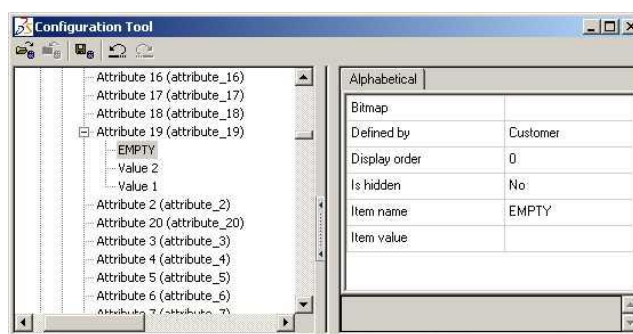


Figure 65: Properties of Empty Value List Entry – Display in Browser

##### Display of an Empty Value List Entry

In the example shown, three values are assigned to the attribute *Sample value list*. If you select none of the defined values (Value1, Value2), the attribute will always be displayed with the empty field.

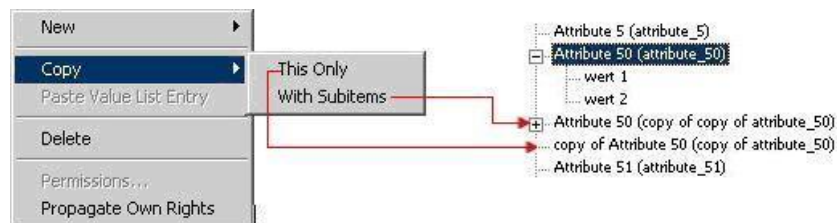


**Figure 66: Empty Field for Display of an Attribute in a Combobox**

### 4.5.5.3 Copying Attributes

In order to copy an attribute, select it and then open the context menu.

- 1) If you select **copy** in the context menu, you can choose between copying the attribute itself and copying the attribute with an added value list in a submenu.

**Figure 67: Select Copy in the Context Menu**

- 2) In order to add the copied attribute, select a type or a plantype and select **add** in the context menu.
- The copied attribute is added under the selected type.

## 4.5.6 Extending the Text Information for Description

If you enter text in the description field of an attribute, page, or group, this text appears as the tool tip in the properties dialog when the mouse is moved over the designation of the attribute, page, or group.

### Example for an Attribute

**Figure 68: Attribute**

If the tool tip should not or may not become excessively large (the tool tip cannot become larger than the storage space of the attribute; that is max 1000 to 4000 characters), or if a detailed description shall be added to the attribute, an additional text can be added.

### How to add a text information to the description

#### Define the storage location of the text information:

Enter the path to the additional text in the maintenance tool.

- 1) Open the maintenance tool via the menu **Tools < Settings < Maintenance**.
- 2) Select the entry *ErgoPlan* on the *Global* page and create a new key with the following entries:

|       |                                   |
|-------|-----------------------------------|
| Key:  | customization                     |
| Name: | contextHelpOnCustomizationObjects |

Value: The path, for the file with the additional information.

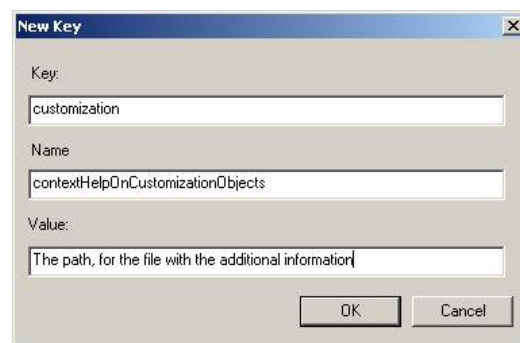


Figure 69: New Key

This key may be overwritten by an entry on the *Local machine* tab. It is important for it to exist.

### Publishing the information about the additional text in the description

- Next to the normal tool tip text the following entry must be made: `<document>Infotext.htm</document>`. Enter the filename instead of *Infotext.htm*.
- Next to the tool tip text the following will appear now: *“Press F1 for more info...”*. If you press F1 now, the additional text will open. For the file to be opened, a suitable program must be installed on the computer. HTML pages, for instance, are opened with Internet Explorer.

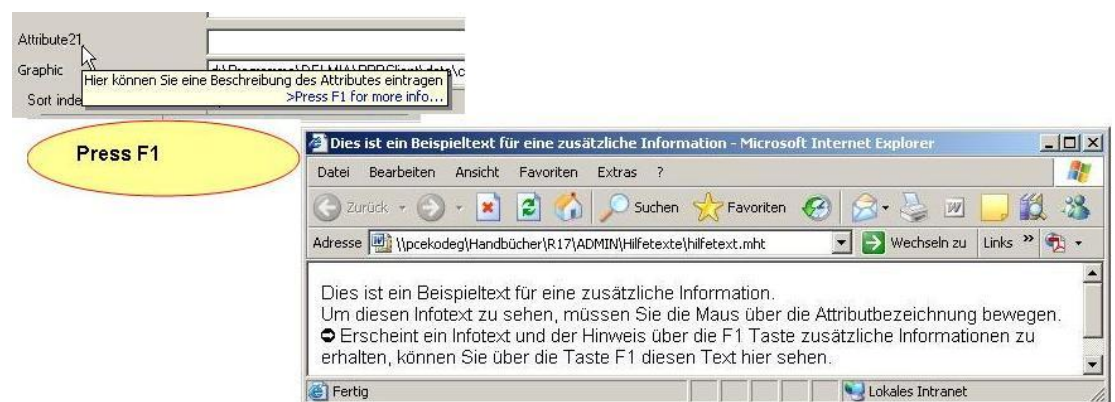


Figure 70: Press F1 for More Information



### Note

The settings allow to specify the size of a tool tip and set the delay until the tool tip appears.

## 4.5.7 Editing Pages of a Type

New pages can be created for types and base types, which can then be displayed in the browser, editor, and list view. The data to be displayed on a page is exclusively defined by means of groups, which in turn must have attributes assigned to them. Groups are always created in conjunction with a page. You can use attributes that are available for the respective type or base type.



## Note

To display pages in a PPR Navigator (browser, editor, list view), at least one attribute per page must be configured. Pages for which no attribute has been configured are **not** shown in the PPR Navigator.

### 4.5.7.1 Creating a New Page

New pages are created via the context menu. The context menu is available for types and base types.

Proceed as follows

- 1) Open the context menu, select **New** and then **Page**.

The new page is displayed at the center area of the Configuration Manager and is listed under the Layout directory in the browser of the Configuration Manager. Please refer to the [Table 16](#).

The new page is shown in the Layout directory as **Page:: xxxx (Page)**. In the example it is designated:

**Page:: 12000 (Page).**

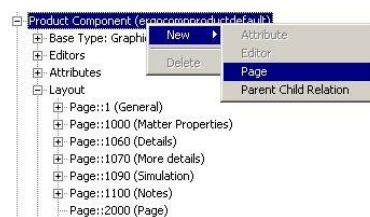


Figure 71: Creating a New Page via the Context Menu

### 4.5.7.2 Copying Pages

In order to copy a page, select it and then open the context menu.

- 1) If you select **Copy** in the context menu, you can choose from a submenu between copying the page itself and copying the page with assigned groups.
- 2) In order to insert the copied page, select a type or a plantype and select **Add** in the context menu.
- 3) The copied page is inserted under the selected type.

### 4.5.7.3 Overwriting Pages

Only existing pages for a base type can be overwritten. These pages are saved in the Layout dictionary of the respective base type. Overwriting copies the page. The copied page is exclusively assigned to the type selected in the browser. The pass-on hierarchy is disregarded when pages are overwritten (several hierarchical levels of base types).

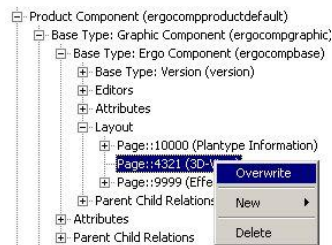
Thus, it does not matter from which hierarchical level of the base type you overwrite a page, the page will always be assigned only to the type that is selected.

- The page overwritten is then shown in the Layout directory of the selected type and can be edited individually. Changes only apply to the page that has been overwritten.
- The page thus created can be displayed for this type either in the editor, browser, or list view.

Proceed as follows

- 1) In the browser first select the type in which you want to overwrite a page derived from the base type.

- 2) Select the base type.
- 3) Open the Layout directory, then open the context menu on the selected page.
- 4) Select the menu item *Overwrite*. The page is assigned to the selected type and is located in its Layout directory.



**Figure 72: Overwriting Pages – Context Menu**

Defining the properties of a (new) page. Please refer to the [Table 16](#).

|                         |                      |
|-------------------------|----------------------|
| Alignment               | Horizontal           |
| Attributes per row      | 1                    |
| Can have Macro assigned | No                   |
| Caption                 | Plantype Information |
| Defined by              | Delmia               |
| Description             |                      |
| Display in browser      | No                   |
| Display in editor       | No                   |
| Display order           | 0                    |
| Number                  | 10000                |
| Own Rights              | No                   |
| Position                | Bottom               |

**Figure 73: The Properties of a Page**

**Table 16: Description of the Properties of a New Page**

| Property                | Meaning   |
|-------------------------|---|
| Alignment:              | In this field you can enter a description of the group. You can document additional notes and changes in the description.   |
| Attributes per row      | Determines the number of columns of the group displayed on the pages.   |
| Can have Macro assigned | Determines whether a script can be assigned to this page by means of script assignment. If a script action is defined for this attribute, a new button appears next to the page. The position of the button is determined by the Position property. |
| Caption                 | Title of the page (register title).   |
| Defined by              | Specifies whether this page has been created by a user. Needed to distinguish from internal program pages.  |
| Description             | In this field you can enter a description for this page. The description is also displayed in the properties dialog as a tool tip.<br><i>Please refer to the <a href="#">Extending the Text Information for Description</a></i>                     |
| Display in browser      | Specifies whether this page (except in the properties window) is to be shown in the object properties display area as well.   |
| Display in editor       | In this field you can enter a description of the group. You can document additional notes and changes in the description.   |



| Property      | Meaning   |
|---------------|---|
| Display order | Determines the order of the pages. The default value must be identical to the number (page number) for already created objects.<br><br>In older versions, page positioning followed the page number. In order not to change the existing display order, the same value as in the page number should also be found in the display order field. |
| Number        | Number of the page. In older versions this number controlled the display order. The higher the number, the further to the right the page was displayed in the "Properties".<br><br>As of version PE 5.13 the number is assigned by the system exclusively. When a new page is created, its number is generated automatically.                 |
| Own rights:   | Determines whether this page can be controlled by user access rights or not. If you set this field to yes, the entry Permissions... is activated in the context menu for this page.<br><br>Via Permissions you can assign users and groups to this page and thus design user-specific layouts.  |
| Position      | If a script is to be executed on this page, define the position of the button here.   |

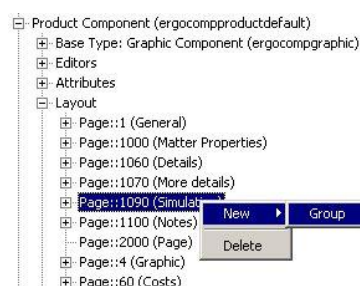
## 4.5.8 Editing Groups of a Type

To be able to display the contents of a new page, groups must be created, and then attributes must be assigned to these groups. The attributes define and name the respective fields of a group. At least one attribute must be assigned to a group. Groups may be shown in a frame, for instance, when displayed in the editor. To allow for this, you need to define a caption. If you do not define a caption, the group is displayed as an individual field in the editor.

### 4.5.8.1 Creating a New Group

Groups for a new page are created directly on the new page via the context menu. The new pages are always saved in the Layout directory. Proceed as follows

- 1) Select the type or base type and open the Layout directory.
- 2) In the Layout directory select the (new) page and open the context menu.
- 3) Select *Group* in the context menu. The new group is created and displayed below the page in the browser.



**Figure 74: Creating a New Group for a Page**

- The new group is displayed below the page.





- If a group is hidden, that is, if it is not assigned to a page, it appears in the browser as an *un-assigned group*.

- 4) To edit the properties of the group, select the group. *Please refer to the Table 17.*

|                         |                                    |
|-------------------------|------------------------------------|
| Alignment               | Vertical                           |
| Attributes per row      | 1                                  |
| Can have Macro assigned | No                                 |
| Caption                 |                                    |
| Defined by              | Delmia                             |
| Description             |                                    |
| Display in browser      |                                    |
| Display in editor       |                                    |
| Display order           | 0                                  |
| Group                   |                                    |
| Number                  | 9001                               |
| Own Rights              | No                                 |
| Page                    | Page::10000 (Plantype Information) |
| Position                | Bottom                             |

**Figure 75: The Properties of a Group**

**Table 17: Description of Group Properties**

| Property                | Meaning  |
|-------------------------|--|
| Alignment               | In this field you can enter a description of the group. You can document additional notes and changes in the description.  |
| Attributes per row      | Determines the number of columns of the group displayed on the pages.  |
| Can have Macro assigned | Determines whether a script can be assigned to this group by means of script assignment. If a script action is defined for this attribute, a new button appears next to the attribute. The position of the button is determined by the <i>Position</i> property.                                     |
| Caption                 | Title of the group.  |
| Defined by              | Specifies whether this field has been created by a user. It is necessary to distinguish user-defined from program-internal groups and should therefore always be selected as a default.  |
| Description             | In this field you can enter a description of the group. You can document additional notes and changes in the description. The description is also displayed in the properties dialog as a tool tip. <i>Please refer to the <a href="#">Extending the Text Information for Description</a></i>        |
| Display in browser      | Specifies whether this page (except in the properties window) is to be shown in the object properties display area as well.  |
| Display in editor       | In this field you can enter a description of the group. You can document additional notes and changes in the description.  |
| Display order           | Group order on the page. The higher the number, the further down the group is displayed on this page.  |
| Group                   | Group in which the group is displayed - the group must already have been created or overwritten. If this property is set, it overwrites the Page property. <i>Please refer to the <a href="#">Creating a Group under a Group</a>.</i>  |
| Number                  | The number is assigned by the system and cannot be changed.  |
| Own rights              | Determines whether this group can be controlled by user access rights or not. If you set this field to yes, the entry Permissions... is activated in the context menu for this group.<br><br>Via Permissions you can assign users and groups to this attribute and thus design user-specific layouts |

| Property | Meaning   |
|----------|---|
| Page     | Page on which the group is displayed - must already have been created or overwritten. If a script is to be executed on this page, define the position of the button here. If this property is set, it overwrites the <i>Group</i> property. |
| Position | If a script is to be executed on this page, define the position of the button here.   |

#### 4.5.8.2 Creating a Group under a Group

Groups may contain further groups. There are two ways of defining a group in a group.

- 1) Groups for a group are created directly in the group via the context menu. The new group is always saved in the Layout directory.

Proceed as follows

- Select the type or plantype or base type and then open the Layout folder.
- In the Layout folder select the (new) group and open the context menu.
- Select **New < Group** in the context menu. The new group is created and displayed below the group in the browser.

If several groups exist, select a group and change to the properties of that group.

Select the *Group* property and select a different group.

- Any settings for the *Page* property are thereby overwritten and the group is moved below the group currently selected.

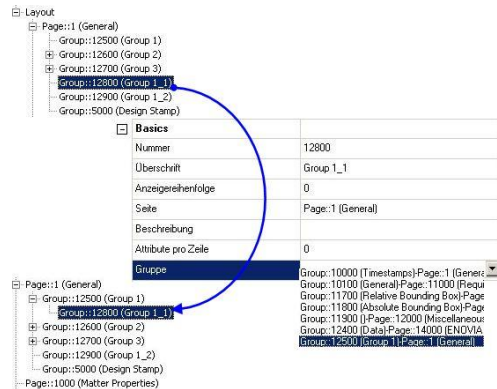


Figure 76: Group in a Group

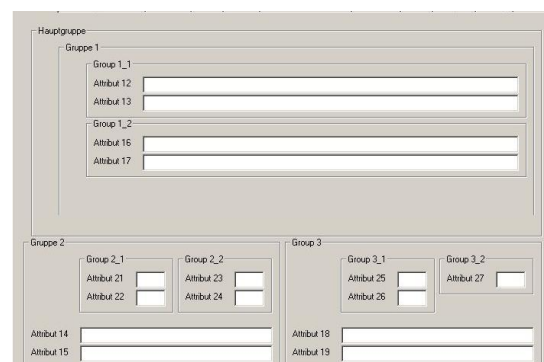


Figure 77: Example for Grouping of Groups

### 4.5.8.3 Assigning Attributes to a New Group

An attribute can belong to and be assigned to only one group.

Proceed as follows

- 1) To assign an attribute to a (new) group, select the attribute in the *Attributes* directory of the type or base type. The attribute *Cost Center* is shown in the example.
- 2) Select *Group* in the list of properties for the attribute. Assign the attribute to a group via the combobox. In the example it is the (new) group 2000.
- 3) This way you can assign any number of attributes to a group. These attributes are used to design the display of the (new) page.

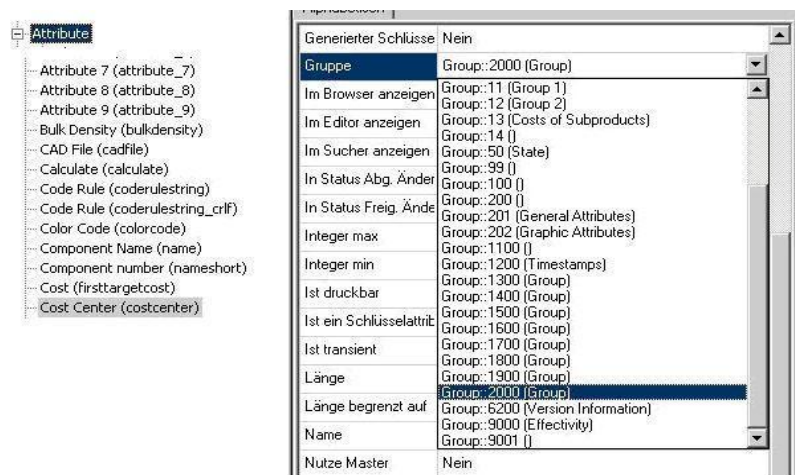


Figure 78: Assigning an Attribute to a New Group

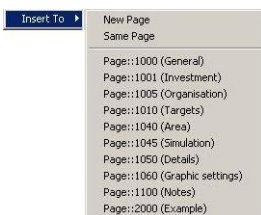
### 4.5.8.4 Overwriting a Group



#### Note

When you overwrite a page, the groups listed under it are **not** overwritten with it.

When overwriting a group you always need to indicate the page to which the group is to be written.



### 4.5.8.5 Copying a Group

For copying of a group proceed as described for copying of pages and attributes.

The only difference is **adding** of the copied group. The *Add group* entry is found in the context menu opened on a page or group.

## 4.6 Layout Editing for Pages, Groups and Attributes

### 4.6.1 User Access Rights for Pages, Groups and Attributes

In user manual administration you have already learnt in description of properties for pages, groups and attributes that starting from Version DPE 5.17 you can define own user rights for these display elements. Description of pages, groups and attributes can be pre-defined for any individual user.



For more information on the user access rights for pages, groups and attributes, *Please refer to the [User Management Manual](#).*

### 4.6.2 Layout Editing in General

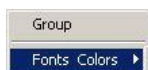
Starting from version PE 5.17SP2 you can edit attribute notation, attribute entry and group headline with a background and arrange them in color. The modifications of configuration are saved in database and are available for all users.

Therefore you have two possibilities:

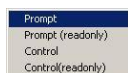
- You configure directly the type or plantype and adjust it to your needs.
- You define global rules which are dependent on attribute properties and then apply to all types and plantypes. Global rules have always priority before the local settings.

### 4.6.3 Local Layout Design

The aim of local layout editing is to highlight certain groups or attributes and to extend the overview of dialogs or to direct the attention of user on important elements.



- 1) Select the group or attribute which you want to edit and choose in context menu **New < Font Colors**.
  - If you have selected a group you can only edit the notation that means the group headline. *Please refer to the [Figure 79](#)*
  - If you have selected an **Attribute** then you have the choices between notation of an attribute (Prompt) and input field (Control). With both properties readonly and not readonly attributes are not distinguished.



All four options can be edited. If a property is already configured you cannot select it any longer during renewed opening of context menu. Already created layout assignments can be edited or deleted. In order to delete an entry select the entry and choose in context menu *Delete*.

In the following example the groups *Organizational Data* and *Timestamps* in project node were indicated in each case with different color notations. The modifications are visible directly in the middle part of configuration manager after saving.

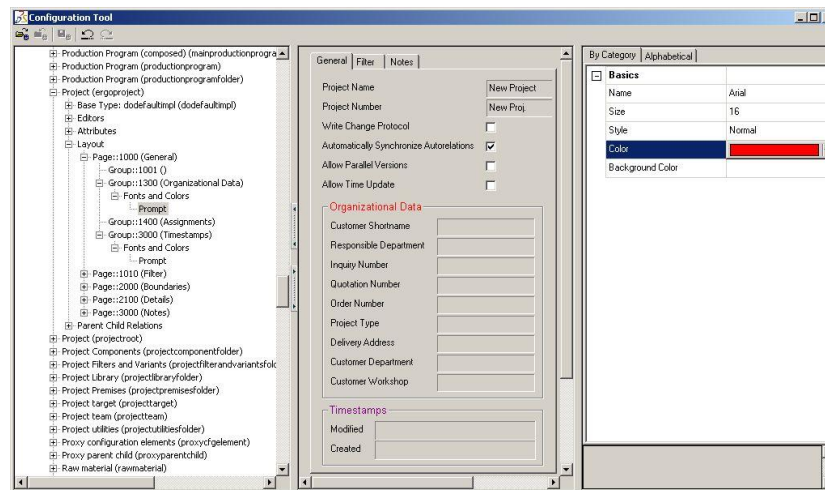
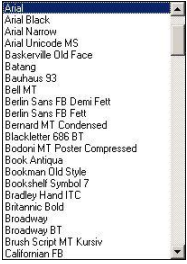





Figure 79: Example

### 4.6.3.1 Properties of Fonts and Colors

Table 18: Properties of Fonts and Color

| Property   | Meaning  |
|--|--|
| <b>Name</b><br>               | <p>Specifies the font.</p> <p>Select in field <i>Name</i> a font.</p> <p>The number of fonts is dependent on the number of fonts installed on the client.</p>  |
| <b>Size</b>  | <p>Specifies size of font.</p> <p>Edit the size of font of text and figures. Give a size of font in field <i>Size</i>. The indicated size of fonts indicated in field <i>Size</i> is dependent on selected fonts in field <i>Name</i> and on the active printer.</p>   |
| <b>Style</b><br>              | <p>Specifies style sheet.</p> <p>Select the style sheet which is to be used.</p>   |
| <b>Color</b><br>            | <p>Specifies color of the text.</p> <p>Formats the entry in color which you have selected. Select in field <i>Color</i> a color. With the button <i>Other...</i> you can add new colors:</p> <ol style="list-style-type: none"> <li>1) Click on the filed <i>Color</i> and then on the scroll bar arrow. <ul style="list-style-type: none"> <li>➤ It opens a color menu.</li> </ul> </li> <li>2) Click "Other..." <ul style="list-style-type: none"> <li>➤ It opens dialog "Color" in which you can select more colors and create them by yourself.</li> </ul> </li> </ol> |
| <b>Background Color</b><br> | <p>Specifies background color.</p> <p>Formats the background of entry in the color which you have selected. Select in field <i>Color</i> a color. With the button "Other..." you can add new colors:</p> <ol style="list-style-type: none"> <li>➤ Click on the filed <i>Color</i> and then on the scroll bar arrow.</li> <li>➤ It opens a color menu.</li> <li>2) Click "Other..." <ul style="list-style-type: none"> <li>➤ It opens dialog "Color" in which you can select more colors and create them by yourself.</li> </ul> </li> </ol>                                |

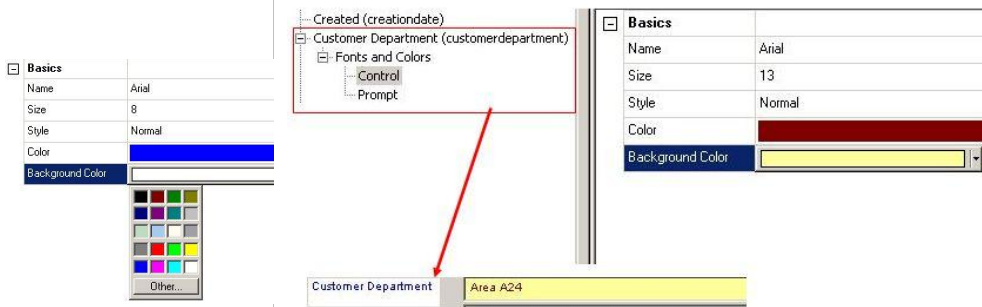


Figure 80: Example of a Configured Attribute

#### 4.6.4 Global Layout Design

The aim of global layout editing is to display certain properties of an attribute. The global layout design is specified directly in configuration manager. In order to extend DPE for these functions starting from version 5.17SP2 new object classes were introduced:

You can create global attributes on the highest node (ConfigDB) of your configuration.

- 1) Select the node and choose in context menu **New < Global Attribute**.



Figure 81: Global Attributes

- It opens a list with the possible properties of an attribute which are applicable for the layout design.

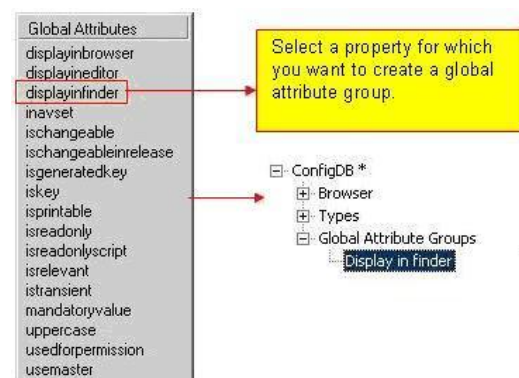


Figure 82: List of Attributes

Select a property for which you want to create a global attribute group.

In order to edit the new created group entry you select it and choose in context menu **New**.





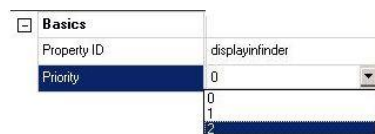
**Figure 83: New Group Entry**

- As in local layout design you have here also the choice between notation of an attribute (Prompt) and input field (Control). With both properties you can also distinguished between readonly and not readonly attributes. All four options can be edited. If a property is already configured you cannot select it any longer during renewed opening of context menu. The editing is the same as in local layout editing.
- All attributes which own the selected property are displayed in property dialog according to the made settings.

Since there are certain properties of an attributes which can occur multiple times you can specify with a prioritization which property is to be displayed first. The properties of an attribute are described in chapter Editing Attributes of a Type. So it is possible that there are attributes which can be displayed e.g. in finder and also in browser and are a key attribute in addition. If global attribute groups are created for these three properties you have to decide which property is more important.

In order to check the priority of an entry, select the entry.

- In the properties you can see the priority assigned by system. The highest priority is 0.



**Figure 84: Specify the Priority**

You can only change the priority if there are several properties in the group.

### Note

*Settings which are made for global attribute groups have always priority before local settings.*

Example for Editing of priority:

*isMandatory* has priority 0 and *isPrintable* has priority 1

Priority of *isMandatory* is changed manually to 1.

- The program recognizes the new priority of *isMandatory* and change the priority of *isPrintable* to 0.

Already created group entries can be edited or deleted. In order to delete an entry select the entry and choose in context menu **Delete**.



#### 4.6.4.1 Properties of Fonts and Colors for Global Attributes Groups

The properties are the same as described in local layout design. *Please refer to the [Properties of Fonts and Colors](#)*

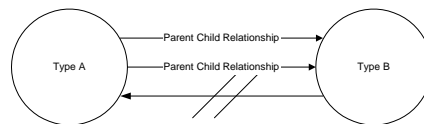
#### 4.6.4.2 Overwrite of Attributes and Groups with Configured Layout Properties

With overwriting of attributes and groups layout properties are also overwritten.

## 4.7 Configuration of Parent-Child Relations

### 4.7.1 Specifying Relations

A parent-child relation is a relation between two types. Using this relation you can determine which types can be assigned to each other.



**Figure 85: Parent-Child Relations**

The partners in a parent/child relation are often not equal. Thus, as shown in [Figure 85](#), for example, type A can have multiple relations to type B. This means that there are multiple possibilities to assign objects of type B to type A. On the other side it must not necessarily be possible to assign objects of type B to type A. This is only possible, if a parent/child relation is defined.

The parent-child relations are shown and edited in the browser of the Configuration Manager in the directory **Parent Child Relations** of a type.

#### 4.7.1.1 New Parent-Child Relation

To create a new parent-child relation, proceed as follows:

- 1) On a type open the context menu and select **New < Parent Child Relation**.



**Figure 86: New Parent-Child Relations**

- From the list select the type to which the parent-child relation is to be established.



**Figure 87: Selecting a Parent-Child Relation**







The new parent-child relation is shown in the directory Parent Child Relation. In the next step, edit the properties of the new parent-child relation. *Please refer to the [Table 19](#) and [Table 20](#).*


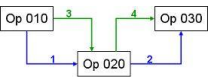
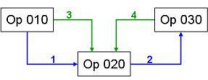



## 4.7.2 Properties of a Parent-Child Relation

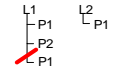

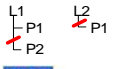

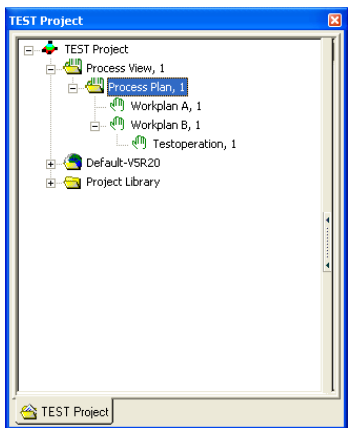
| Basics |  |                        |
|--------|--|------------------------|
| 1      | Parent type  | balancing              |
| 2      | Child type   | allowancesetprocess    |
| 3      | Child list   | allowancesetprocess    |
| 4      | Prompt   | process allowance sets |
| 5      | Child set  | m_pAllowanceSet        |
| 6      | FPR - relation                                       | No                     |
| 7      | Inheritance  | Inherit to all         |
| 8      | Relation class                                       |                        |
| 9      | Child parent relation                                |                        |
| 10     | Color  |                        |
| 11     | Description  |                        |
| 12     | Loopcheck group                                      | No loopcheck           |
| Flags  |  |                        |
| 13     | Is relevant  | No                     |
| 14     | Is enabled   | Yes                    |
| 15     | Is autorelation                                      | No                     |
| 16     | Change in integrate state                            | No                     |
| 17     | Change in release state                              | No                     |
| 18     | Inherit effectivity                                  | Yes                    |
| 19     | Show in graphic                                      | No                     |
| 20     | Is unique for parent and child                       | No                     |
| 21     | Is unique for parent                                 | No                     |
| 22     | Is autorelationpath                                  | No                     |
| 23     | Copy link to child                                   | No                     |
| 24     | Is system internal                                   | No                     |
| 25     | Copy link to child (Change Management Controlled)    | No                     |
| 26     | Copy link to child (versioning)                      | No                     |
| 27     | Is configured  | No                     |
| 28     | Is a simple relationship                             | No                     |
| 29     | Must have unexposed object for source                | No                     |
| 30     | Consider for Activation by Link                      | No                     |
| 31     | Exclude from Manufacturing Change Management and CCZ | No                     |
| 32     | Must have unexposed object for target                | No                     |
| 33     | Defined by   | Delmia                 |
| Enum   |  |                        |
| 34     | Owner type   | No Owner               |

**Figure 88: Properties of a Parent-Child Relation**

Table 19: Properties of a Parent-Child Relation





|    | Property              | Comment   |
|----|-----------------------|---|
|    | <b>Basics</b>         |   |
| 1  | Parent type           | <p>Here, the parent type is displayed.</p>  <p><i>Note: This function is disabled in the standard version of DELMIA Process Engineer.</i></p>  |
| 2  | Child type            | <p>Display field:</p> <p>Here, the child type is displayed. Via this function you can later change or replace the type.</p>   |
| 3  | Child List            | <p>Write a name for the child list here. Between identical parent and child types the name must be unique. The name may be used only once in this relation for this (child) type or plantype.</p> <p>If a parent-child relation exists with several different plantypes and types, the same name for the child list can be used for these relations.</p>  |
| 4  | Prompt                | In the PPR Navigator in the object properties display area the relations between the objects of different types are displayed. You can enter the description (Prompt) of the tab in the object properties display area of the PPR Navigator.  |
| 5  | Child set             | <p>This function is intended for internal use only, to add, remove generic implementation (parent knows its children).</p>  <p><i>Note: This function is disabled in the standard version of DELMIA Process Engineer.</i></p>   |
| 6  | PPR - relation        | You can determine here, whether it is a product, process or resource relation.  |
| 7  | Inheritance           | <p>The inheritance property determines whether a parent-child relation is also passed on to the types and plantypes derived from the present type.</p>   |
| 8  | Relation class        | You can determine a relation class here. This function is intended for internal use only; it describes the relation class that is to be created, if it is a relation.   |
| 9  | Child parent relation | <p>Specification of the parent-child relation. See also Child set, however, the child knows its parents, but not vice versa.</p>  <p><i>Note: This function is disabled in the standard version of DELMIA Process Engineer.</i></p>  |
| 10 | Color                 | <p>Define the color for relations in the process graph and the manufacturing concept.</p> <p>Existing relations in the process graph and manufacturing concept can be highlighted by the color selection. The color selection is based on the respective relations selected: for example, for the relations process runs before process or process must precede process, etc.</p>                    |
| 11 | Description           | In this field you can enter a description of the relation. You can document additional notes and changes in the description.  |
| 12 | Loopcheck group       | <p>Loopcheck group is an extension of the function "Loopcheck". With the Loopcheck group multiple relation types can be checked for loop developments, whereas the "simple" check could examine only one relation at a time. In order to combine relations to a group, all relations that belong to the same group must have the same group name entered. Currently three groups are available.</p>  |

|    | Property                         | Comment   |
|----|----------------------------------|---|
|    |                                  | <p>Further groups can be created. For more information on creating further groups, please refer to DELMIA Support.</p>  <p><b>Note:</b> The loopcheck group can affect the performance.</p> <p>The following must be considered when configuring the loopcheck group:</p> <p>Only relations that combine the elements of an area are allowed in a group.</p> <p>Both directions of a parent-child relation must be covered with the check consistently, meaning that the “_reverse” relation must be included in the same group.</p> <p>Examples:</p> <p><code>ergocompprocessdefault:: process_mustprecede_process</code></p> <p><code>ergocompprocessdefault:: process_mustprecede _process_reverse</code></p> <p><code>ergocompprocessdefault:: process_runsbefore_process</code></p> <p><code>ergocompprocessdefault:: process_runsbefore _process_reverse</code></p> <p>A correctly created relation</p>  <p>An error message appears while creating the relation 4.</p>  <p>An error message appears while creating the relation 4.</p>  <p>► No Loopcheck: Using this relation development of loops will not be checked.</p> <p>► Simple: This Loopcheck occurs between components of the same type only (process_runs_before_process). Example: Preventing a Rel1 - Rel2 - Rel3 - Rel1.</p> <p>► Group 1 etc: Defines the relation as part of a group, which is checked for development of loops together with other relations that were assigned to this group as well.</p> |
|    | <b>Flags</b>                     |   |
| 13 | <b>Is relevant</b>               | You can specify here whether a change log is to be created if a type B object is assigned to a type A object.   |
| 14 | <b>Is enabled</b>                | <p>You can hide relations on plantype level here.</p>  <p><b>Note:</b> This function is disabled in the standard version of DELMIA Process Engineer.</p>   |
| 15 | <b>Is autorelation</b>           | <p>PPR relations are automatically created.</p>  <p><b>Note:</b> This function is disabled in the standard version of DELMIA Process Engineer.</p>   |
| 16 | <b>Change in integrate state</b> | Defines whether the relation can still be changed after the status “integrated” has been assigned.  |

|    | Property                       | Comment  |
|----|--------------------------------|--|
| 17 | Change in release state        | Defines whether the relation can still be changed after the status "released" has been assigned.   |
| 18 | Inherit effectivity            | <p>If this function is enabled, effectivities for code rules / line numbers are inherited. This function is relevant only for the following relations:</p> <p>PROC_FIRSTPROCESSES_PROD<br/> PROC_CREATES_PROD<br/> PROC_CREATES_PROD<br/> PROC_PROCESSES_PROD<br/> PROC_REMOVES_PROD<br/> PROC_FIRSTPROCESSES_PROD<br/> PLANT_PROVIDES_PROD</p> <p>Enabling this function has no effect on other relations.</p> <p>For details <i>Please refer to the <a href="#">Example of "Inherit Effectivity"</a>.</i></p>  |
| 19 | Show in graphic                | If the checkbox "Show in Graphic" is enabled, the graphic of the children is displayed depending on the relation. For more detailed information <i>Please refer to the <a href="#">Autorelations</a>.</i>  |
| 20 | Is unique for parent and child | <p>This relation can be generated only once with this parent and child (with this parentname, childname, childlistname and configname).</p> <p>Functions as in PE 5.11.</p>   <p><b>Note:</b> This function is disabled in the standard version of DELMIA Process Engineer.</p>  |
| 21 | Is unique for parents          | <p>This relation can be generated only once with this type</p>   <p>Using standard PTS create Process structure with two Workplans (workplan A and workplan B) under "process plan" and a "Testoperation" under workplan B.</p>  <p>1) In configuration Overwrite relationship Workplan - Operation: nodes</p> |

|    | Property   | Comment  |
|----|--|--|
|    |  | <p>2. Overwrite relationship Workplan - Operation: nodes_reverse</p> <p>3. Set "Is unique for parent" to <b>Yes</b> on both relations</p> <p>4. Create a new Operation as a child to Workplan A.</p> <p>5. Create a second operation as a child to Workplan A - not allowed (throws up warning message <b>"The parent already has a relation with the given type"</b>).</p> <p><br/> <b>Note:</b></p> <p><i>When you try to create a second operation under "Workplan A", system checks for all the nodes relation on Workplan type, since an operation is already created (in step 4) under workplan A system does not allow to have more than one operation as child for workplan type.</i></p> <p>6. Create TestOperation as a child to Workplan A.</p> <p><b>Note:</b></p> <p><br/> <i>Except "operation" system allows to create multiple TestOperations. Reason "Is unique for parent" is only set on Operation relation.</i></p> <p>7. Create an operation as a child to Workplan B.</p> <p><b>Note:</b></p> <p><br/> <i>There already exist a plantype "Test Operation" under Workplan B. now when you try to create a "Operation" under "Workplan", system checks for all the nodes relation on Workplan type and throws up warning message (The parent already has a relation with the given type). Since "TestOperation" and "Operation" both are of type "ergocompprocessdefault" and "Is unique for parent" check is at type level. Hence it does not allow to have more than one child for Workplan type.</i></p> <p>8. Create a Testoperation as a child to Workplan B.</p> <p><b>Note:</b></p> <p><br/> <i>Except "operation" system allows to create multiple TestOperations. Reason "Is unique for parent" is only set on Operation relation.</i></p> |
| 22 | Is autorelationpath  | If the function "Is autorelationpath" is enabled, an autorelation is created. For more detailed information <i>Please refer to the <a href="#">Autorelations</a>.</i>  |
| 23 | Copy Link to Child   | If a component is copied, a link to its children is created.   |
| 24 | Is system internal   | If a relation is only valid for system-internal areas like versioning or autorelations, then this checkbox must be enabled.  |
| 25 | Copy link to child object (Change Mangement)   | If a component is copied, then a link is created to its children. This applies only to change management.  |
| 26 | Copy link to child (Versioning)  | If a component is copied, then a link is created to its children. Applies only to versions without change management.  |
| 27 | Is configured<br> | <p>This function refers to the change management and specifies if and how the relation for the change management is used.</p> <p>Following possibilities are available:</p>  |



|    | Property                              | Comment  |
|----|---------------------------------------|--|
|    |                                       | <p><b>No</b> = no change management</p> <p><b>Yes</b> = for change management; standard configuration mode</p> <p><b>Max. Configured</b> = for change management; maximal configuration mode</p> <p>Besides the function Is configured, the function no.12 <a href="#">Has versions</a> Owns Version is also important.</p> <p>If plantypes are not configurable, so an extended effectivity can not assign to them.</p> <p>In case of standard configuration we differentiate between plantypes, which are versionable or not.</p> <p>For versionable plantypes new objects or checked out version receive the validity from the instruction for modification.</p> <p>During check in of a version, the extended effectivity is subjected to each possible version with a planning state, which agrees with effectiveness from the selected instruction for modification. There is no special operation for not versionable plantypes, but which are configured. This requires a modification assignment, which is selected when these objects are created.</p> <p>Maximal configuration means, that plantypes have either "always" or "never valid" extended effectivities.</p> <p>Maximal configured plantypes don't support parallel versions.</p> |
| 28 | Is a simple relationship              |  <p><b>Note:</b> This function is disabled in the standard version of DELMIA Process Engineer.</p>   |
| 29 | Must have unexposed object for source |  <p><b>Note:</b> This function is disabled in the standard version of DELMIA Process Engineer.</p>  |
| 30 | Consider for Activation by Link       | Consider for Activation by Link  |
| 31 | Exclude from Manufacturing            | <p>If this property is set to Yes, relations that have an owner with a version count are not considered in the MCM and Configuration Control Zone.</p> <p>The property must be adjusted consistently for both directions of the parent-child relation.</p>   |
| 32 | Must have unexposed object for target |  <p><b>Note:</b> This function is disabled in the standard version of DELMIA Process Engineer.</p>  |
| 33 | Defined by                            | <p>Specifies whether this relation has been created by a user. Needed to distinguish from internal program types.</p>  <p><b>Note:</b> This function is disabled in the standard version of the DELMIA Process Engineers.</p>   |
|    | Enum                                  |  |
| 34 | Owner Type                            | <p>A relation is taken into consideration in a Change Management Controlled (MCM) project only if the relation has an owner and if this owner is a 'versionable' component. Set the owning relationships of the relation under the function <i>Owner type</i>.</p> <p>The following settings are available:</p> <p><b>No Owner</b></p>   |

| Property | Comment  |
|----------|--|
|          | <p>If a relation does not have an owner, it is not within the range of applicability of the change management.</p> <p><b>Source Owner</b><br/>The source PPR component is the owner of the relation.</p> <p><b>Target Owner</b><br/>The target PPR component is the owner of the relation. The 'Source owner' and 'Target owner' specify the source and target plantypes with regard to the owner of the relation. If the relation of the API (<i>application programming interface</i>) is created, the information can be derived during the runtime and the owner does not need to be explicitly specified.</p> <p><b>Explicit Owner</b><br/>The owner of the PPR component can be freely defined. An explicit owner can be an object from any valid plantype. In DPE it is not possible to derive this information if the relation is created, and therefore it is necessary to make a reference to the explicit user for the API.</p> <p><b>Common Parent Owner</b><br/>The owner is always the direct parent of the linked source and target components (common parent). The target according to the relation <i>relationship_nodes</i> (BOM), DPE is not able to derive this information, so the reference to the common parent owner to the API must be made when the relation is created. source – and target component (common father). The aim, according to the relation <i>relationship_nodes</i> (BOM), DPE is not in the position to derive this information, therefore the note to the common parent must be led to API, when the relation is created.</p> |

### 4.7.3 The Parent-Child Information

#### Eltern-Kind Information

The information concerning the parent-child relation first determines, whether it is to be displayed in the tree view and in the list view of the PPR Navigator; and second it determines how the parent-child relation is to be displayed.

Create the information concerning the parent-child relation via the context menu of the type or plantype of the parent-child relation.

- 1) Open the context menu and select **New < Parent Child Information**.



**Figure 89: Create Information for Parent-Child Relation via the Context Menu**

- 2) To edit the properties of the parent-child information, select the parent-child relation in the browser.
- The parent-child information is found in the browser under the type or plantype of the parent-child relation. Please refer to the [Table 20](#).









**Figure 90: Selecting the Parent-Child Information in the Browser**

| By Category |                  | Alphabetical                         |
|-------------|------------------|--------------------------------------|
| 1           | Browser ID       | ergoexplorer-projectlibrary          |
| 2           | Default relation | No                                   |
| 3           | Defined by       | Delmia                               |
| 4           | Description      |                                      |
| 5           | Folder name      |                                      |
| 6           | Is in listview   | Yes                                  |
| 7           | Is in treeview   | No                                   |
| 8           | Is read only     | No                                   |
| 9           | Program ID       | ErgoExplorer.EPAdvancedDataComponent |
| 10          | With a 'new' ent | No                                   |

Figure 91: Parent-Child Information

Table 20: Description of Properties

|    | Property                   | Comment   |
|----|----------------------------|---|
| 1  | Browser ID                 |  <p><i>Note:</i> For pre-configured relations this function is disabled in the standard version of DELMIA Process Engineer.</p> <p>It is active only for new parent-child relations. For a new relation, via this function define the location in which the relation is to be displayed.</p> |
| 2  | Default relation           | Determine via this function, whether this parent-child relation is to be activated as the default in the relations selection dialog of the PPR Navigator.   |
| 3  | Defined by                 | <p>Specifies whether this relation has been created by a user. Needed to distinguish it from internal program types.</p>  <p><i>Note:</i> This function is disabled in the standard version of DELMIA Process Engineer.</p>  |
| 4  | Description                | In this field you can enter a description. You can document additional notes and changes in the description.  |
| 5  | Folder name                |  <p><i>Note:</i> This function is disabled in the standard version of DELMIA Process Engineer.</p>   |
| 6  | Is in listview             | Determine via this function, whether the parent-child relation is to be shown in the <i>Listview</i> of the PPR Navigator.  |
| 7  | Is in treeview             | Determine via this function, whether the parent-child relation is to be shown in the <i>Treeview</i> of the PPR Navigator.  |
| 8  | Is read only               |  <p><i>Note:</i> This function is disabled in the standard version of DELMIA Process Engineer.</p>   |
| 9  | Program ID                 |  <p><i>Note:</i> This function is disabled in the standard version of DELMIA Process Engineer.</p>   |
| 10 | With a 'new' entry in menu |  <p><i>Note:</i> This function is disabled in the standard version of DELMIA Process Engineer.</p>   |

## 4.7.4 Deleting Parent-Child Relations



### Caution

*You should not delete existing parent-child relations. When such relations are deleted, information may be lost, such as existing autorelations.*

*You may, however, delete new parent-child relations that you have created yourself.*

### Deleting Parent-Child Relations

Parent-child relations and corresponding information are deleted via the context menu.



**Figure 92: Delete Parent-Child Relations**

### Deleting Parent-Child Information

Additionally, you can delete parent-child information individually, without having to delete the type or plantype.

- 1) To do so, in the browser select the parent-child information and open the context menu.
- 2) Click *Delete*.

### Configuring Relation

For every relation 2 PCs (parent/child) and 1 type are needed in the configuration.

### Example

Parent/Child Relation: *Childlistname pprelationtype relationclassname autorelationpath* (Yes = 1, No = 0) *autorelation* (Yes = 1, No = 0))

### PCs in the case of relations with different types

| Type | Name      | Relation          | RealTypeName    | set |
|------|-----------|-------------------|-----------------|-----|
| A B  | childlist | pprelation        | XDORelationship | 1 0 |
| B A  | childlist | ppreverserelation | XDORelationship | 0 0 |

Is only allowed to be in one direction=1 maximum! Must be 0 in the case of non-autorelations!

XDORelationship or derived class (exception: not in the case of "nodes" or "refnodes"!!)

### PCs in the case of relations with different types

|     |                   |                   |                 |     |
|-----|-------------------|-------------------|-----------------|-----|
| A A | childlist         | pprelation        | XDORelationship | 1 0 |
| A A | childlist_reverse | ppreverserelation | XDORelationship | 0 0 |

### Type:

*Name:* relationship\_childlist (in both cases)

*RealTypeName:* XDORelationship or derived classes

*Parent type* relationship or derived type

## 4.7.5 Autorelations

### 4.7.5.1 Introduction to Autorelations

Autorelations are special relations that are automatically created and deleted.

Autorelations are created if a chain of relations with an enabled autorelation path checkbox is generated between 2 components for which an autorelation is defined and if the autorelation has not been created before. The autorelation object remembers ALL paths that are “valid” for it.

If a chain is interrupted, which belongs to one (or multiple) autorelation paths or to one (or multiple) autorelations, the sub-path is removed from the corresponding autorelation object. If an autorelation object does not have any more sub-paths, the object is *marked* as deleted. It is not actually deleted because it is necessary for *moving* or *using* during **Versioning**.

**A “chain” can be created if**

- a) a child component is created.
- b) a relation is created.
- c) a component is moved.
- b) a relation is moved.
- e) a new version of a component is used.
- f) a new version of a relation is used.

**A “chain” can be interrupted if**

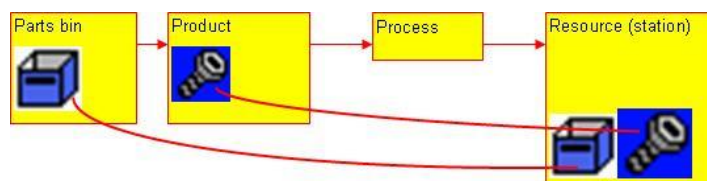
- a) a child component is deleted.
- b) a relation is deleted.
- c) a component is moved.
- b) a relation is moved.
- e) a new version of a component is used.
- f) a new version of a relation is used.

**Figure 93: Autorelation****Example****Example**

You assign a parts container to a product (which has its own graphics).

This product is made by means of a process running on a resource.

Thus, a chain is established. The purpose of chaining is to ensure that the product and the parts container are displayed in the resource.

**Figure 94: Chain Structure**

You can directly link the product to a resource. The product is visible in the resource view, the container is not. No autorelation has been created.

However, if you create the following links:

- Parts container to product
- Product to process
- Process to resource, an autorelation results (product and container become visible in the resource view).

Autorelations always depend on the type and the parent-child relation defined. Concerning parent-child relations, the following functions are pertinent

- The function **is autorelation**
- The function **is autorelationpath**
- The function **show in graphics**

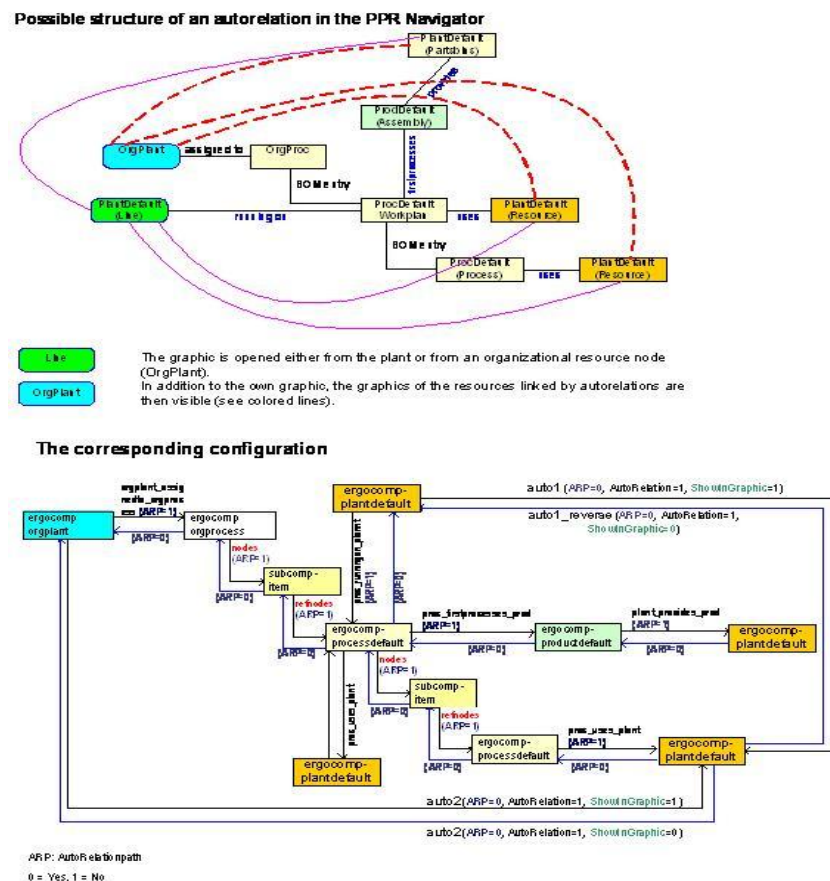


## Caution

*Autorelations are bidirectional: If the function “Show in graphic” is enabled in both directions, an endless loop will result. If components of an autorelation (parents) are deleted, the underlying components are deleted as well.*

The following example shows how autorelations need to be configured for given requirements.

- Some autorelations cannot be re-configured in the standard installation. You can only influence the display of graphics.
- The setup of the autorelations can also be seen in the example "Inherit effectivity".



### Figure 95: Autocorrelations Configuration

## 4.8 Configuration Samples

The following chapter contains examples on how you can configure the interface and the database in the DELMIA Process Engineer using types.



## 4.8.1 Example for the Configuration of Types

### Example

Even though you cannot create attributes on type level, various possibilities for configuration are offered, however. The configuration on type level is suitable above all if a change in the configuration should have effects not only on the plantype, but also project-wide on entire plantype areas. In the following chapters some of the configuration possibilities are shown as an example.



### Caution

*Changes on type level have effects beyond the limits of projects and plantype sets. Therefore the configuration should be made with great care and only by trained administrators.*

#### 4.8.1.1 Creating a Group on the Type Level

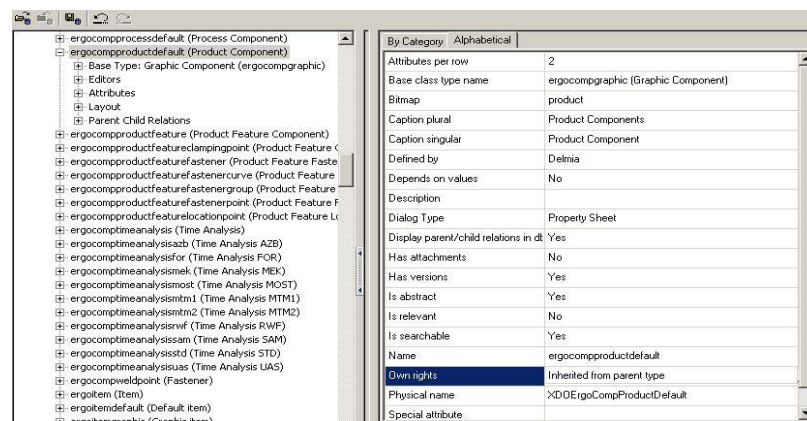
The following example illustrates how a group is created on the type level.

The aim of this exercise is to create a group for the designation and the number of a product component. You will notice that creating a group on the type level hardly differs from the procedure followed on the plantype level.

- 1) Open the **Configuration tool** in the “**Tools < Database Utilities**” menu.
- 2) From the list of types select the type “ergocompproductdefault” and then open the Layout directory.

**Ergocompproductdefault** (Product Component (ergocompproductdefault))

= the type from which the product plantypes are derived.



**Figure 96: The Type “ergocompproductdefault”**

**Figure 97: Properties of an Assembly before the Changes**

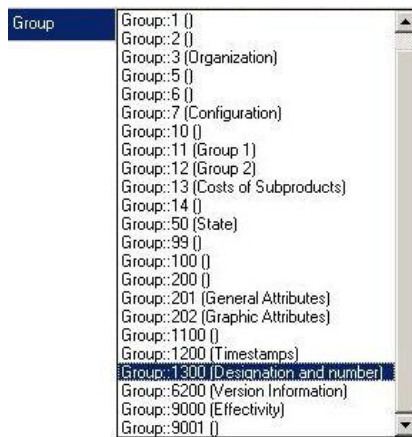
As shown in [Figure 97](#), the first eleven attributes of a group are put together without a caption. The component name and the component number are now to be put together in a separate new group. To achieve that, you need to assign a caption in the properties of the new group. In the example it would read “Designation and number”.

- 1) In the Layout directory select the page **General** and select *New / Group* from the context menu.
- You have created a new group.
- 2) Select the new group (it has the highest number and is designated *Group*) and enter the properties shown below (you cannot change the number):

|               |                        |
|---------------|------------------------|
| Caption       | Designation and number |
| Defined by    | Customer               |
| Display order | 0                      |
| Number        | 1300                   |
| Page          | Page::1 (General)      |

**Figure 98: Properties of the Group “Designation and Number”**

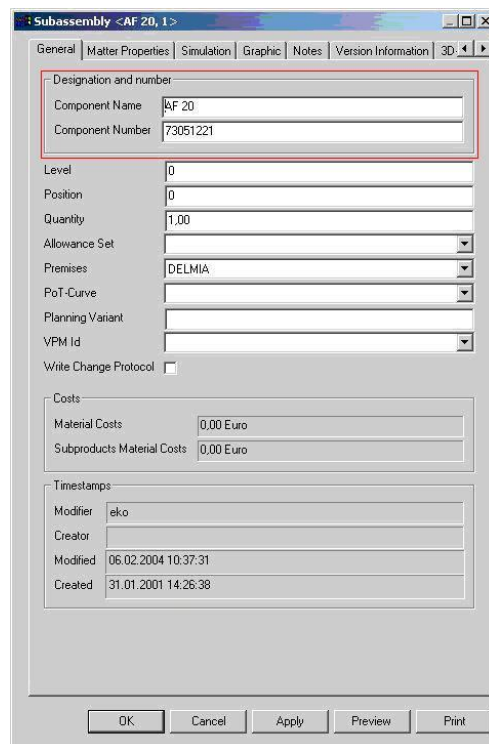
- 3) To assign the two attributes **Component name (name)** and **Component number (nameshort)** to the new group, open the Attribute directory.
- 4) Open the two attributes and change the values in the input field “group” as follows:



**Figure 99: The Input Field “Group”**

5) Now open any product in a project.

- “Number” and “Designation” of the product are now put together in the group “Designation and number”.



**Figure 100: The Group “Designation and Number”**

## 4.8.2 Examples for Attributes per Row and Display Order



### Note

*If you find the designations of groups, pages or attributes displayed in English in the Configuration Manager, but not in the properties dialog, this is caused by an internal translation mechanism. Familiar terms, that is, terms that are used frequently in DELMIA Process Engineer®, are translated internally.*

### 4.8.2.1 Defining the Attributes per Row

Attributes per row

If you set the number of *Attributes per row* higher than 1 for a page (this defines the number of columns for the layout of the properties dialog), the layout of the page will change substantially, as illustrated in [Figure 101](#). Please refer to the [Defining the Display Order](#).

Figure 101: The Group “Designation and Number”

### 4.8.2.2 Defining the Display Order

To achieve the page design shown in [Figure 101](#), the *display order* of the groups also had to be changed. The display order determines the sequence in which the groups are arranged in the columns.

The display order was changed as follows:

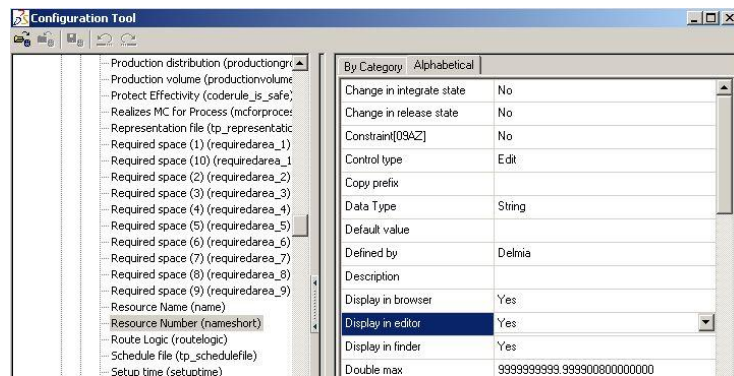
Table 21: Schematic for the Definition of the Display Order

| Group Designation      | Display Order, Old | Display Order, New |
|------------------------|--------------------|--------------------|
| Designation and number | 1                  | 1                  |
| Group without caption  | 2                  | 3                  |
| Costs                  | 3                  | 2                  |
| Timestamps             | 4                  | 4                  |

### 4.8.3 Example for Hiding of an Attribute

This example shows how you can hide an attribute in a properties dialog. To do so, start the *Configuration Manager* and select the type “**ergocomplant-default**”, for instance.

- 1) In the **list of attributes** search the attribute with the type designation (name of attribute) **nameshort ( Number)**. Change to the properties of the attribute.



**Figure 102: The Attribute “nameshort”**

- 2) Set the entry **Display in editor** to **No**. Save the change.

The number is now hidden in the *Properties* dialog of the resource. The number or short name can be shown again at any time, by setting the entry **Display in editor** to **Yes**. As you see, it is not necessary to delete an attribute to remove it from a dialog.

#### 4.8.4 Example for the Configuration of “Free” Attributes

##### Example

In DELMIA Process Engineer so called “free” attributes have been defined. Free attributes can be recognized by their designation, which is structured in the following way:

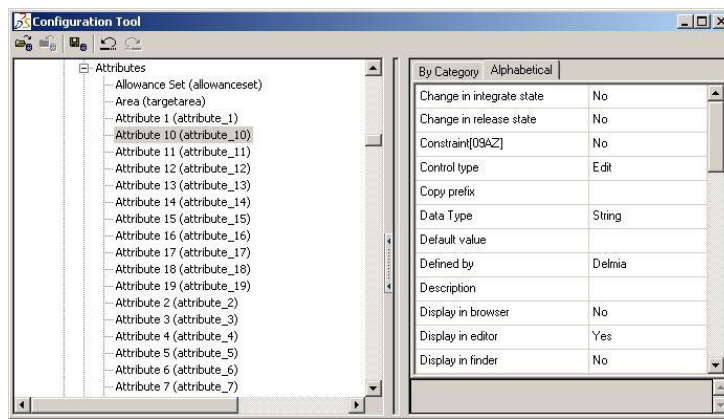
**attribute\_xx**: String attributes, freely configurable by the user.

**dbl\_attribute\_xx**: “Float” type attributes, freely configurable by the user.

The following example shows how you can use the free attributes for your purposes.

Suppose, for example, that your superior wants a responsible person to be defined for every resource object. The name of the responsible person is to be printed on corresponding print forms. It is also to be possible to use the name of the responsible person as a search criterion in the Finder.

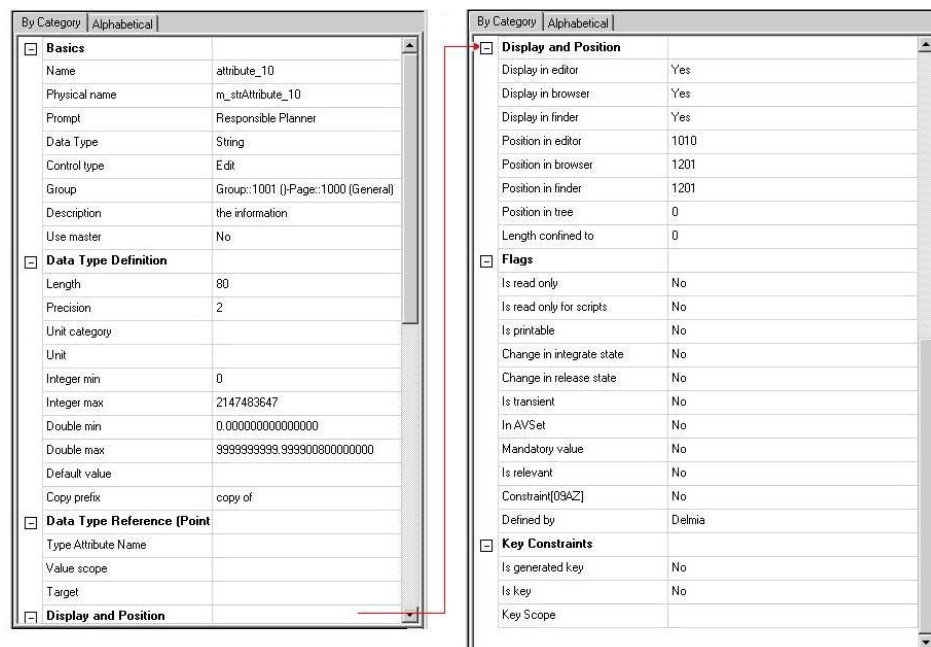
- 1) Start the *Configuration Manager* and select the type “**ergocomplantdefault**”.
- 2) In the **list of attributes** search the attribute with the type designation (name of attribute) **attribute\_10 (Attribute 10)**. Change to the properties of the attribute.



**Figure 103: The Free Attribute “attribute\_10”**

Change the properties according to [Figure 104](#).

For instance, identify the name of *Attribute 10* as *Responsible Planner*.



**Figure 104: Properties of the Attribute “Responsible Planner”**

1) Save the configuration.

➤ The new attribute is now available in the resource properties.

Figure 105: Resource Properties

#### 4.8.4.1 Value Lists and Special Attributes

Suppose your superior is satisfied with this configuration, but while using the Navigator, he now wants to see at a glance which person is responsible for what resource. To allow you achieve that, he hands you a list of all persons that might be involved. Also, he wants to be able to display resources to which no responsible person has been assigned.

- 1) Again select the attribute *Responsible Planner*.
- 2) Change the Control type from *Edit* to **Combobox**, picking the corresponding entry from the list.
- 3) Now create a value list with the following entries. Please refer to the [Using a Value List](#).

Table 22: Value List Entries

| Item Name    | Item Value | Order | Bitmap |
|--------------|------------|-------|--------|
| Mr. Meyer    | ME         | 10    |        |
| Mrs. Muller  | MM         | 20    |        |
| Mr. Perry    | PG         | 30    |        |
| Mrs. Schmitt | SM         | 40    |        |
| Mr. Jackson  | JA         | 50    |        |
| EMPTY        |            | 60    |        |

The bitmaps can be created using any graphics program (e. g. Paint). The bitmaps must have a size of 18 x 17 pixels (width x height) and must be saved in 256 color mode.

- 4) Now select the type **ergocomplantdefault** and change the properties of the type.

Control type Combobox

Special attribute Responsible Planner (attribute\_10)

Internal shelf planning data (alb\_shelfpl)

Allowance Set (allowanceset)

Attribute 1 (attribute\_1)

Responsible Planner (attribute\_10)

Attribute 11 (attribute\_11)

Attribute 12 (attribute\_12)



- 5) Search the entry **Special attribute**. From the list now opening select the attribute *Responsible Planner*.
  - 6) Search the entry **Depends on values**. Enter Yes for it.
  - 7) Save the changes.
- In the PPR Navigator the resources are now shown with different icons. The assigned icons identify the individual employees. Thus, your superior can with the help of the colored icons quickly get an overview of who is responsible for a plan, without even having to read the individual names.

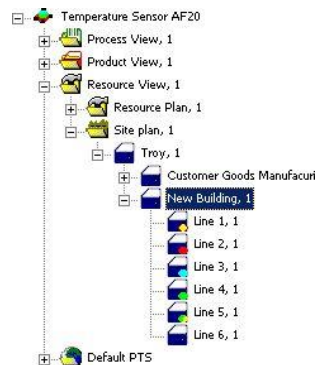


Figure 106: The Result of the Configuration

#### 4.8.5 Example of Displaying a Folder in the Project Library

The arrangement of the folders in the project library have changed as of version PE 5.14. This may lead to some folders not being displayed.

The folder *Tools* will be displayed in the following example. The folder *Tools* contains two subfolders: **Scripts** and **VBA projects**. This folder is deactivated as a standard since in a productive environment scripts are not constantly created in the project library.

- 1) Start the *Configuration Manager* and select the type with the heading **Project Library** and the name **projectlibraryfolder**.
- 2) Look for the entry **projectutilitiesfolder** in the **List of the parent-child relations**.
- 3) Set the entry **Tree view** to Yes in the parent-child information. Please refer to the [Figure 107](#).

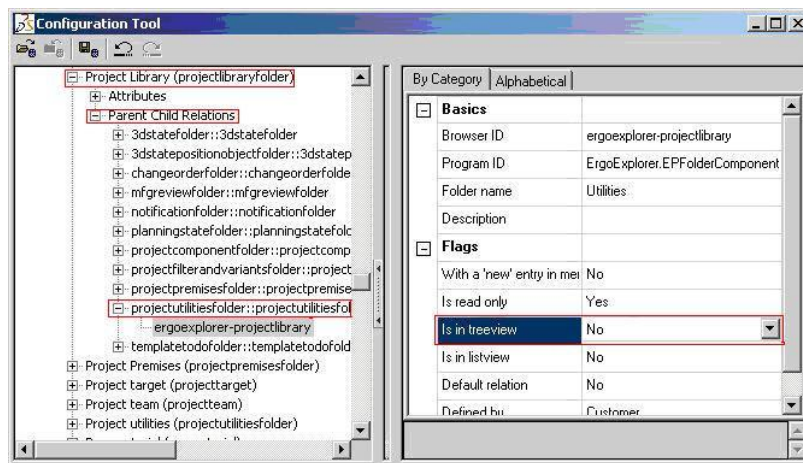


Figure 107: Display Folder in Project Library

- 4) Save the changes in the Configuration Manager.
  - 5) Update the project library (e.g. key **F5**).
- The folder Tools is displayed in the project library. You can also hide the folder in the same way.

#### 4.8.6 Example of Displaying a Context Menu Entry

As of version PE 5.14 the context menu entry **Execute Script** was deactivated as a standard. This is to prevent the execution of scripts on components which are not designed for this purpose. Scripts can be executed on the corresponding type or plantype by means of script assignment. The context menu entry can be re-activated in a pilot project or in a test phase.



#### Note

*The context menu entry 'Execute Script' should always be set to the standard value - not activated in the productive environment.*

Since the context menu entry 'Execute Script' applies to all types it is defined on the lowest level and is inherited by all objects derived from it.

- 1) Start the *Configuration Manager* and select the type with the heading and the name **dodefaultimpl**.
- 2) Look for the entry *ScriptEngineLoader.ScriptEngineLoaderManager::ergoexplorer-projects::Execute script* in the **List of editors**.
- 3) Set both entries to *Single selection in the browser* and *Multiple selection in the browser* to **Yes**. Please refer to the [Figure 108](#).

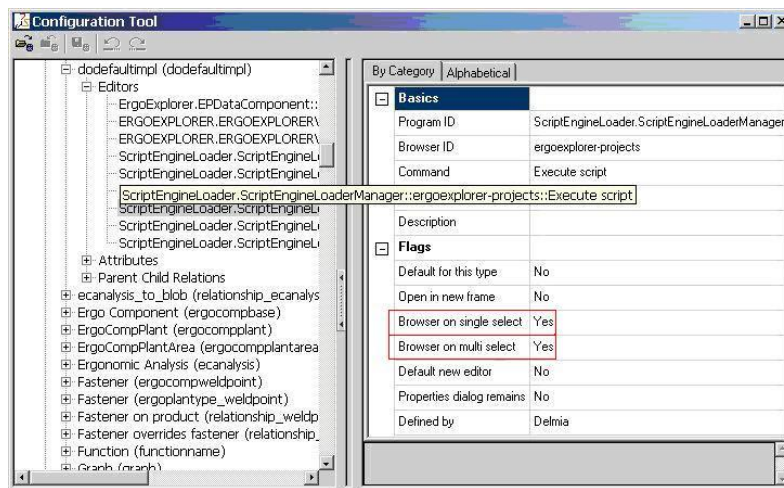
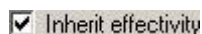


Figure 108: Display a Context Menu Entry

- You can call up the context menu entry on all objects of the PPR navigator.

### 4.8.7 Example of "Inherit Effectivity"



In version PE 5.11, the process inherited the effectivity from the product every time a relation between a process and product was generated. In order to prevent this the "protect effectivity" checkbox on the target object had to be enabled. Furthermore it was not possible to permit inheritances of certain types.

As a consequence, whenever a relation between a process and a product or between a resource and a product is generated and the "inherit effectivity" function of the process or resource has not been enabled, the code rule of the process or resource is replaced by the code rule of the product. The inheritance of line numbers behaves identically and is also affected by the "Inherit effectivity" checkbox.

In version PE 5.12, an additional functionality for the inheritance of effectivities (code rule / line numbers) was introduced with the **Inherit effectivity** function.

Since the inheritance of effectivities for code rules/line numbers is sensible only for relations between product/process and resources/product, it should be used only for these relations.

The function is enabled in the default configuration. The DELMIA Process Engineer behaves identically to version PE 5.11 whenever the checkbox is enabled.



#### Caution

*The settings take effect only for relations between product/process and resource/product. The settings are ignored by all other relations.*

The function must always be identical for both directions.

If, for example, the function for:

```
Ergocompprocessdefault, Ergocompproductdefault, Proc_creates_prod
```

is enabled, then it must also be enabled for the opposite direction

Ergocompproductdefault, Ergocompprocessdefault, Proc\_creates\_prod.

The code rule/line number effectivity is tested only for product/process and resource/process relations. Therefore only the functions of the following relations take effect.

PROC\_FIRSTPROCESSES\_PROD  
 PROC\_CREATES\_PROD  
 PROC\_CREATES\_PROD  
 PROC\_PROCESSES\_PROD  
 PROC\_REMOVES\_PROD  
 PROC\_FIRSTPROCESSES\_PROD  
 PLANT\_PROVIDES\_PROD

### Initial Situation

A line with three stations (= resource)

Three processes (Proc 1-3)

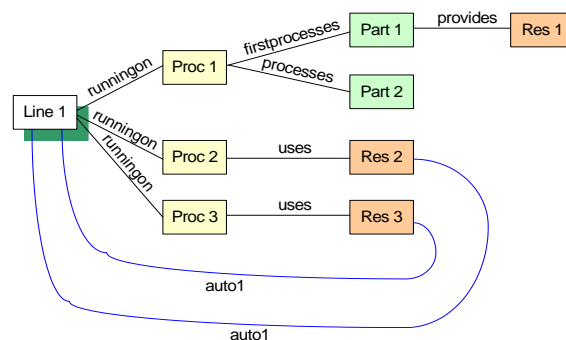
Two products (Part 1 and Part 2)

The following relations are generated:

- Proc 1 firstprocesses Part 1
- Proc 2 processes Part 2
- Part 1 provides Res1
- Proc 2 uses Res 2
- Proc 3 uses Res 3
- Line 1 runningon Proc 1
- Line 1 runningon Proc 2

The following code rules are assigned:

- Proc 1: +a/b.
- Proc 2: +d+c.
- Part 1 +a+b.
- Part 2 +c.
- Res 1 : +d.



The "Inherit effectivity" function is **not** enabled for all of the following relations.

PROC\_FIRSTPROCESSES\_PROD  
 PROC\_CREATES\_PROD  
 PROC\_CREATES\_PROD  
 PROC\_PROCESSES\_PROD  
 PROC\_REMOVES\_PROD  
 PROC\_FIRSTPROCESSES\_PROD  
 PLANT\_PROVIDES\_PROD

Garbage collection  
Rebuild Coderules  
Reparse Coderules  
Replace Codes

Thus all effectivities which could take effect are switched off

You have to restart the Process Engineer in order for the changes to take effect.

### Case 1

- 1) On the project node call up the context menu **Coderules < Rebuild Coderules**

The result will look like the example

| Comp...   | V | Coderul... | Object ... | Old Coderule | New Coderule | Status             |
|-----------|---|------------|------------|--------------|--------------|--------------------|
| Station 3 | 1 | CR         | C          | .            | .            | coderule unchanged |
| Station 2 | 1 | CR         | C          | .            | .            | coderule unchanged |
| Station 1 | 1 | CR         | C          | +d.          | .            | ok                 |
| proc 3    | 1 | CR         | C          | .            | .            | ok                 |
| proc 2    | 1 | CR         | C          | +d+c.        | .            | ok                 |
| proc 1    | 1 | CR         | C          | +a/b.        | .            | ok                 |
| Line 1    | 1 | CR         | C          | .            | .            | coderule unchanged |
| Factory   | 1 | CR         | C          | .            | .            | ok                 |
| Cycle     | 1 | CR         | C          | .            | .            | ok                 |
| Building  | 1 | CR         | C          | .            | .            | ok                 |

Figure 109: Example of "Inherit Effectivity"; Case 1

- The code rules for process 1 and 2 as well as for station 1 have been deleted.

### Case 2

- 1) If you now enable the "Inherit effectivity" function for ROC\_FIRSTPROCESSES\_PROD (for both directions), the following statement is made when the code rule is formed anew:

| Component | V | Coderul... | Object ... | Old Coderule | New Coderule | Status             |
|-----------|---|------------|------------|--------------|--------------|--------------------|
| proc 2    | 1 | CR         | C          | +d+c.        | .            | ok                 |
| proc 3    | 1 | CR         | C          | .            | .            | ok                 |
| Cycle     | 1 | CR         | C          | .            | .            | ok                 |
| proc 1    | 1 | CR         | C          | +a/b.        | +a+b.        | ok                 |
| Factory   | 1 | CR         | C          | .            | .            | ok                 |
| Building  | 1 | CR         | C          | .            | .            | ok                 |
| Line 1    | 1 | CR         | C          | .            | +a+b.        | ok                 |
| Station 1 | 1 | CR         | C          | +d.          | .            | ok                 |
| Station 2 | 1 | CR         | C          | .            | .            | coderule unchanged |
| Station 3 | 1 | CR         | C          | .            | .            | coderule unchanged |

Figure 110: Example of "Inherit Effectivity"; Case 2

- The code rules for process 1 and the line have +a+b as a new code rule.

### Case 3

- 1) If you now enable the "Inherit effectivity" function for PROC\_PROCESSES\_PROD (for both directions), the following statement is made when the code rule is formed anew:

A new product part 3 was previously generated. Its code rule is +d+e.

Part 3 is linked to Proc 1 with the relation "processes"

Effectivity

Begin: 04.09.2003

End: 04.09.2003

Labels:

Planning Code:

Coderule: **+a+b&d+e.**

Protect effectivity: ☐

Frequency: 100.00 %

Line numbers:

| Component | V | Coderul... | Object ... | Old Coderule | New Coderule | Status             |
|-----------|---|------------|------------|--------------|--------------|--------------------|
| proc 2    | 1 | CR         | C          | +            | +c,          | ok:                |
| proc 3    | 1 | CR         | C          | +            | +            | ok:                |
| Cycle     | 1 | CR         | C          | +            | +            | ok:                |
| proc 1    | 1 | CR         | C          | +a+b&a+b,    | +a+b&d+e,    | coderule unchanged |
| Factory   | 1 | CR         | C          | +            | +            | ok:                |
| Building  | 1 | CR         | C          | +            | +            | ok:                |
| Line 1    | 1 | CR         | C          | +a+b,        | +a+b&d+e     | ok:                |
| Station 1 | 1 | CR         | C          | +            | +            | ok:                |
| Station 2 | 1 | CR         | C          | +            | +c,          | ok:                |
| Station 3 | 1 | CR         | C          | +            | +            | coderule unchanged |

Figure 111: Example of "Inherit Effectivity"; Case 3

- The code rules for the line have been created.

#### Case 4

- 1) If you now enable the "Inherit effectivity" function for PLANT\_PROVIDES\_PROD (for both directions) the following statement is made when the code rule is formed anew:

A new product part 4 was previously generated. Its code rule is +e.

Part 4 is dragged to the line by drag and drop.

| Component | V | Coderul... | Object ... | Old Coderule | New Coderule  | Status             |
|-----------|---|------------|------------|--------------|---------------|--------------------|
| proc 2    | 1 | CR         | C          | +c,          | +c,           | coderule unchanged |
| proc 3    | 1 | CR         | C          | +            | +             | ok                 |
| Cycle     | 1 | CR         | C          | +            | +             | ok                 |
| proc 1    | 1 | CR         | C          | +a+b&d+e     | +a+b&d+e,     | ok                 |
| Factory   | 1 | CR         | C          | +            | +             | ok                 |
| Building  | 1 | CR         | C          | +            | +             | ok                 |
| Line 1    | 1 | CR         | C          | +a+b&d+e     | +e&a+b&d+e&c, | ok                 |
| Station 1 | 1 | CR         | C          | +a+b,        | +a+b,         | coderule unchanged |
| Station 2 | 1 | CR         | C          | +c,          | +c,           | coderule unchanged |
| Station 3 | 1 | CR         | C          | +            | +             | coderule unchanged |

Figure 112: Example of "Inherit Effectivity"; Case 4

- The code rules for the line and the station were generated.

## 5. Configuration of Work Load Balancing

### 5.1 General information

You can start the configuration of Work Load Balancing via the context menu of the project node; **Extra < Balancing WLB Configuration**.



**Figure 113: Open Balancing Configuration**

The configuration of Work Load Balancing should always be started using the context menu of the project node.

It is, however, still possible to start Work Load Balancing from another node. Therefore you will be shown two possibilities for configuring with which you can start Work Load Balancing on planttypes in the context menu of which the starting of the Work Load Balancing was not planned.

#### 5.1.1 Configuration of the Context Menu Entries

##### 5.1.1.1 Configuration Manager

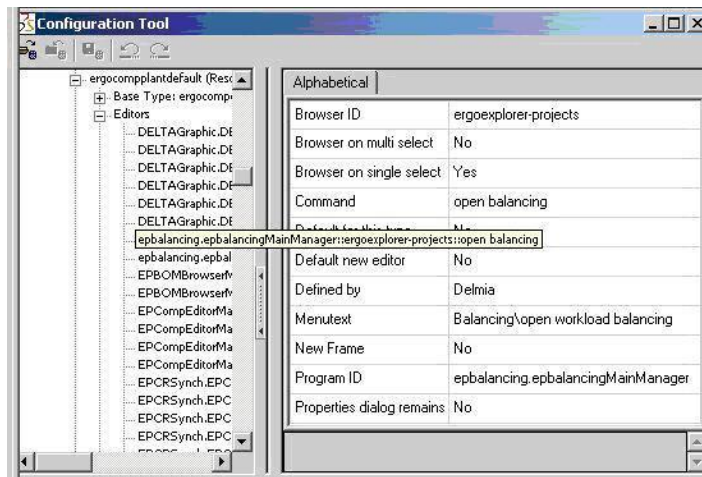
In a new installation, the process described in the following is not necessary since the entries have already been configured. You must create two new context menu entries if they are not available.

If the two entries are to appear in the context menu of every resource, choose the Configuration Manager for configuring.

- 1) Open the Configuration Manager and select the entry ***ergocomplantdefault***.
- 2) Open the folder Editors and search for the entries for the menu text ***Balancing\Open Workload Balancing (WLB)***, the command ***open balancing*** ***Balancing\Utilization utilization***, the command ***utilization***.

The context menu entry is listed in the Balancing folder which is likewise found in the Editors folder of the type *ergocomplantdefault*.





**Figure 114: Configuration Tool: Opening the Balancing Process**

- 3) Enter a call text in the menu text input field, for example *open workload balancing* and activate **Browser on single select**.
- 4) Save the changes.

### 5.1.1.2 Plantype Set

If the context menu entry is to be opened only with a certain **plantype**, you must overwrite the entry in the plantype set. To do this open the plantype set in the system library for which the entries will be made:

- 1) Select a **plantype**, e.g. the line and open it via the context menu *Edit*.
  - The Configuration Manager opens.
- 2) In the list of editors of the base type you can now search again for the aforementioned entry. Overwrite the entry. The procedure is the same as that previously described.
  - After closing and saving, the context menu entries will appear only on the line level.

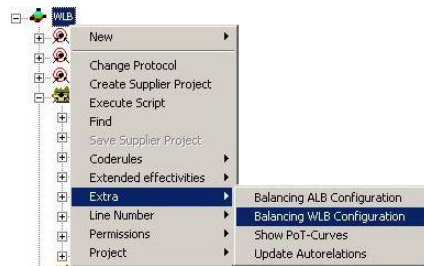
## 5.2 Opening the Configuration for Work Load Balancing



### Caution

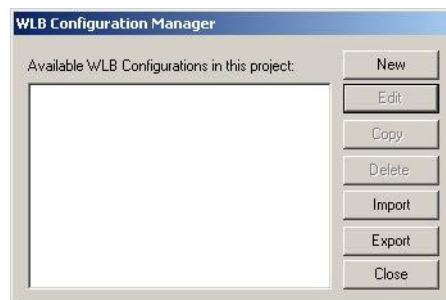
*The Work Load Balancing configuration is linked to certain boundary conditions which should be absolutely observed. If these conditions are observed, no error messages will occur.*

- 1) Open the balancing configuration using the context menu of a project.



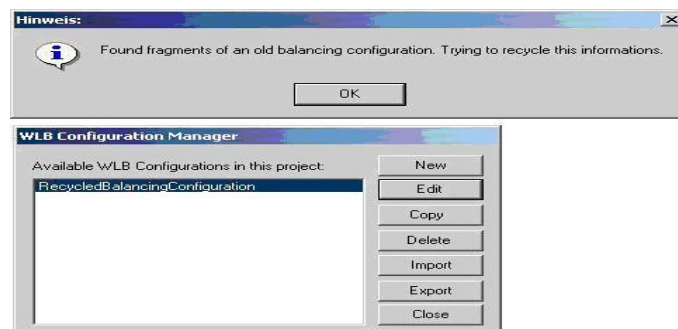
**Figure 115: Open Balancing Configuration**

The dialog for configuring the Work Load Balancing is opened.



**Figure 116: WLB Balancing – Configuration Manager**

If you have already worked with older versions of DPE and have created a configuration for Work Load Balancing, the system will first inform you that the work load balancing configuration is no longer current and the administrator should open the work load balancing configuration so that it is automatically updated.



**Figure 117: Work Load Balancing Configuration**

The previously available configuration is adapted to the new version, i.e. if a valid and previously functioning configuration is available, it is updated.

Numerous new settings options have been included in the work load balancing configuration since version PE 5.15. When updating, the previous settings are transferred and standard values are set for the new settings. You should check the set standard values.

**Note**

*Before starting a work load balancing with an updated configuration, you should re-check the configuration and, if necessary, adjust it to your requirements.*

You can create several configurations. The description of the dialog is as follows:

**New**

You can create a new configuration using the **New** button.

**Edit**

A previously existing configuration can be edited and changed using the Edit button.

**Copy & Paste**

You can duplicate the selected configuration with the Copy button.

**Delete**

You can delete the selected configuration with the Delete button.

**Import**

You can import configurations using the Import button. Only files of the type *Balancing Configurations Files* with **.bal** extensions can be imported.

**Export**

Either all or only the selected configurations are exported using the Export button.

A dialog for selecting the saving location is opened. If the place and name of the export file were set, you can select between all configurations and those selected.



**Figure 118: Export WLB Configuration**

The exported file is saved with the extension **.bal**. All previously existing configurations are exported.

**Close**

Closes the dialog.

**Note**

*If you use the material provision view, the automatic synchronization of the autorelations on project level must be switched off.*

Project <WLB\_HB>

General | Filter | Notes

Project Name: WLB\_HB

Project Number: WLB

Write Change Protocol: ☒

Automatically Synchronize Autorelations: ☐

Allow Parallel Versions: ☒

Allow Time Update: ☒

Figure 119: Automatic Synchronization of the Autorelations

## 5.3 Adjusting the Plantype Set

In the *Work Load Balancing* manual the various possible types of work load balancings are described, such as 3 level balancing, tentative work load balancing, area work load balancing, and fine work load balancing. As you already know from the Work Load Balancing manual, there are several possibilities for the work load balancing. In order to be able to use all the types of work load balancing, changes must be made to the plantype set. Therefore these adjustments to the plantype set will be discussed before the actual configuration is described.



For more information, Please refer to the [Work Load Balancing Manual](#).

### 5.3.1 Creating New Plantypes

#### 5.3.1.1 General Information

A plantype must be created specifically for the non-balanced processes of the work load balancing. This plantype is of the type *ergocomplantdefault* and it must be a child of the plantype on which the work load balancing is started.

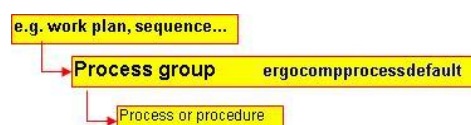
**Start object** (line, system, or every plantype on which a work load balancing is started)



Figure 120: Plantype ergocomplantdefault

#### 5.3.1.2 Processes

If you want to combine processes into groups in order to use these groups for a first tentative work load balancing, you must also create a new plantype or use a previously existing one. But this is not absolutely necessary. Nothing changes in this case with regard to the prerequisites.



Scheme of the setup of the tentative work load balancing



### 5.3.1.3 Three Level Balancing

If you want to use 3 level balancing, you must extend your planttype set.

All 3 levels are unambiguously dependent on each other. For the configuration of level 3, this means: for every level a separate configuration has to be created, and it must build on the respective previous configuration.

- **Level 1 work load balancing:** tentative work load balancing e.g. of stations of a line with processes or process groups.  
The level 1 work load balancing is the prerequisite for the level 2 work load balancing.
- **Level 2 work load balancing:** area work load balancing – is the management of processes that are distributed in groups in an organizational structure. A prerequisite is a level 1 work load balancing. In this way you could, for example, align the capacities, which could then lead to a restructuring of the processes.  
Level 2 work load balancing is the prerequisite for the level 3 work load balancing.
- **Level 3 work load balancing:** Group work load balancing – e.g. the work load balancing of the individual workplaces of a group. A prerequisite is a level 2 work load balancing.

An organizational structure must be available or must be created for the 3 level work load balancing. The structure must be organized as follows:

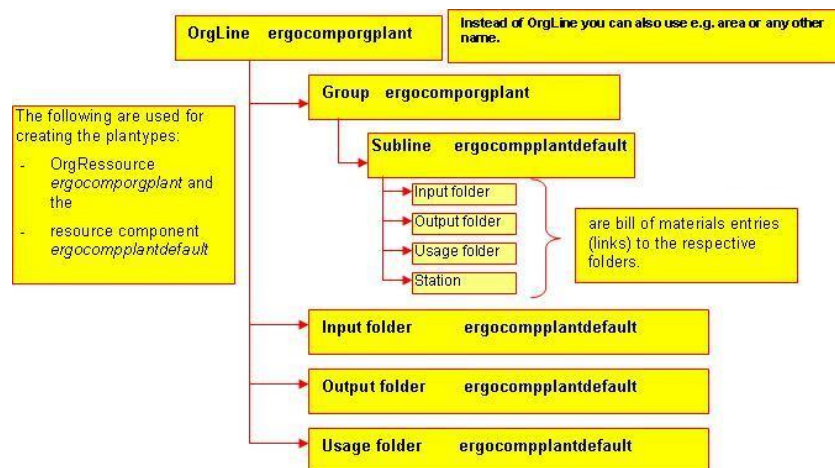


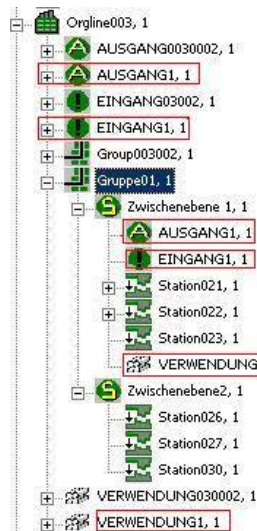
Figure 121: Organizational Structure for 3 Level Work Load Balancing

- **Workplaces** are not visible in the tree structure and they are created per work load balancing (they are a part of the work load balancing). In the plantype set there must be an organizational plantype for the workplaces.



**Figure 122: Organizational Plantype for the Workplaces**

- For every group under the organizational node OrgLine,
  - one *output folder*,
  - one *input folder*input, and output folders of the group must be linked to the intermediate level (subline) of this group as a bill of materials entry. Precisely the same usage, input, and output folders must also be linked in the bill of materials exactly below the OrgLine (for which the line work load balancing was created, and one *usage folder* must be set.
- The **usage**, input, and output folders of the group must be linked to the intermediate level (subline) of this group as a bill of materials entry. Precisely the same usage, input, and output folders must also be linked in the bill of materials exactly below the OrgLine (for which the line work load balancing was created). Only one usage, input, and output folder may exist for the group fine work load balancing. This means that the line work load balancing has one usage, input, and output folder per group.
- The **stations** for which the workplaces are balanced must always be assigned to the intermediate level.
- The **stations** beneath the group must be a portion of the stations that are used for the tentative work load balancing.
- A group can have several intermediate levels (sublines), however it may have only one usage folder, one input folder, and one output folder per group. *Please refer to the [Subline](#).*



**Figure 123: Organizational Node OrgLine**

### Subline

Between the group and the station there must be one intermediate level, in the example it is the *Subline*, which contains the cycle time.

### Attributes



Table 23: Subline

| Attribute | Meaning    | Type  | Range | Comment    |
|-----------|------------|-------|-------|------------|
| cycletime | Cycle time | Float |       | cycle time |

**Input Folder**

The processes of the group assigned to the stations in the tentative work load balancing are assigned via the input folder in the level 2 work load balancing. The processes are assigned via the input folder of the group.

The processes are assigned to the workplaces via the input folder of the group in the level 3 work load balancing.

**Output Folder**

The processes that were not planned or could not be planned in the fine work load balancing are marked via the output folder. The level 2 work load balancing is thus instructed to move the processes to another input folder of a group.

In the level 3 work load balancing, the processes that could not be assigned to any workplace in the group are assigned to the output folder and then re-distributed via the level 2 work load balancing.

**Usage Folder**

The usage folder has only an informative function in the level 2 balancing.

In the level 2 work load balancing, the processes which are assigned to the workplaces via the input folder of a group are also displayed in the usage folder for informative purposes.

The objective is that **all** processes are displayed in the usage folder.

Table 24: Usage Folder

| Attribute        | Meaning                  | Type   | Range | Comment |
|------------------|--------------------------|--------|-------|---------|
| Name             | Name of the folder       | String |       |         |
| Nameshort        | Short name of the folder | String |       |         |
| Materialareawith |                          | double |       |         |
| Length           |                          | double |       |         |
| Width            |                          | double |       |         |

**Additional Comments**

- 1) The groups must be in a bill of materials beneath the line of the line work load balancing. An unlimited number of intermediate levels can exist between the line and the groups. It is, however, important that bill of materials are always used ("relationship\_nodes").
- 2) The standard configuration name is read from the "*balancing\_customizationname*" of the starting object (line, group, ...). If the "*balancing\_customizationname*" is empty, the standard algorithm for determining the work load balancing configuration starts. *Please refer to the [Attribute "balancing\\_customizationname"](#).*
- 3) Valid work load balancings can be saved only if the objects used in them (lines, stations, processes, starting object) are temporally valid via the entire time range of the work load balancing.



- 4) If valid work load balancings are saved, the planning status of the objects used in it (lines, stations, processes, starting object) are set to valid (integrated).
- 5) In all objects set to 'valid', relations can be changed only if their child lists are configured correspondingly (change in integrate state). This affects both of the relations *proc\_runningon\_plant* and *proc\_runningon\_plant\_reverse* between the standard stations and processes used in the work load balancing.
- 6) Starting a level 2 work load balancing from the PPR-Navigator is not possible and it leads to an error message.
- 7) Starting a level 3 work load balancing from the PPR-Navigator is not possible and it leads to an error message.

Additional information on attributes can be found under [Additional Attributes](#).

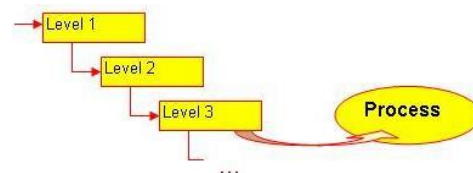
## 5.4 Configuring Work Load Balancing

Before starting with the configuration, you should check the prerequisites described in the following.

### 5.4.1 The Basic Prerequisites for a Raw Balancing

A structure consisting of three hierarchical levels must be available.

- The third level (not the last) must be able to be linked via a relation to a process or a process-like type.



**Figure 124: Three Hierarchical Level Structure**

- The selected relations must be set to **no** in the Configuration Manager under the tab of the *parent-child relations* (the entry "unambiguous for parents") if variants of the work load balancings are created and their multiple use is to be allowed. Please refer to the [Specifying Relations](#).
- A raw balancing is started on level 2.

#### 5.4.1.1 The Balancing Configuration Dialog

If you observe the above- mentioned conditions, the configuration should present no problems.

The default structure is as follows:

```

conveyor section,
    ↳ line,
        ↳ station,
  
```

↺ corresponding processes.

There are four tabs available to you for configuring Work Load Balancing.

## 5.4.2 General Settings



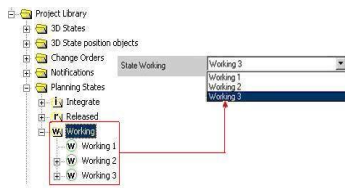
### Caution

*A configuration that is saved and used for the work load balancing should not be changed afterward. Changes to an existing configuration that is being used can lead to errors in the execution of the work load balancing. If you want to change an existing configuration that is being used, copy the configuration and then make the changes.*

Figure 125: Balancing Configuration

### 5.4.2.1 General Data

- **Configuration Name:** Enter the name of the configuration here. This name is shown to you when you open the Work Load Balancing configuration.
- **Input Folder:** - For a level 1 work load balancing and a raw balancing, select the plantype you will use for the non-balanced processes here. The selection field is available only if you have configured a plantype for it (tab views: Display *Additional Grid View* and tab: Plantype forms).  
- For a level 2 work load balancing select the plantype which you will use as a input folder or output folder here. The correct entry is available to you in the selection field only if you have already configured a plantype for it (tab Views: Display *Additional Grid View* and tab: Plantype forms).  
- For a level 3 work load balancing select the plantype which you will use as a input folder or output folder here. The correct entry is available to you in the selection field only if you have already configured a plantype for it (tab Views: Display *Additional Grid View* and tab: Plantype forms).
- **Master Configuration:** Select the master configuration here. You must not select a master configuration for a level 1 work load balancing and a raw balancing. The entry must be NONE.  
For a level 2, select the corresponding level 1 work load balancing. If you have selected a corresponding name for the configuration name, the selection will be easier.  
For a level 3, select the corresponding level 2 work load balancing.



- **State Working:** Here select the working state that applies to this workload balancing configuration. In case there are multiple working states in the system, the right state for this workload balancing can be selected here. Only existing working states will be shown in the combobox. There are two further fields available for a **level 2 work load balancing**
- **Use Cycle time from First Level associated Balancing Configuration:** This field must be activated if the cycle time of the starting object of the corresponding level 1 work load balancing is to be applied.
- **Save Working Balancings, which begin in the past is allowed:** This field must be activated, if changes to a previously existing work load balancing of which the starting date is either in the past or today are made. If the field is not activated, the work load balancing can not be saved, if the starting date is in the past or is today.

There are also two further fields available for a **level 3 work load balancing**

- **Use Cycle time from first level associated balancing configuration:** This field must be activated if the cycle time of the starting object of the corresponding level 1 work load balancing is to be applied.
- **Allowed number of planning balancings in the same group and period:** Here you can set the maximum number of level 3 work load balancings for the same group and same effectivities range. A number of 1 to 10 is possible.
- **MC Collaboration:** This field must be activated if you want to start a work load balancing from the manufacturing concept. If the context menu entry for opening the work load balancing is not available in your manufacturing concept, it must be configured. Proceed as follows:

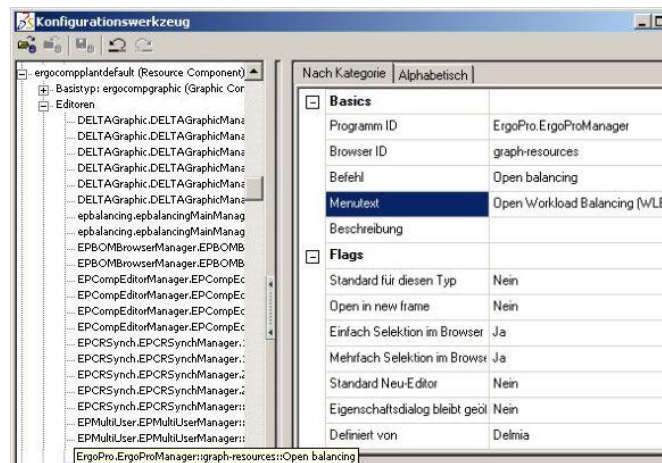
1) Open the Configuration Manager and select the entry **ergocomplantdefault**.

2) Create a new editor via the context menu of the type *ergocomplantdefault*.

→ The following must be specified in the properties of the editor:

**Program ID**            **ErgoPro.ErgoProManager**  
**Browser ID**           **graph resources** (selection from Combobox)  
**Command**            **Open Balancing**  
**Menu text**            **Open Workload Balancing (WLB).**

- 3) Activate the single and multiple selection in the browser.  
 4) Save the changes.



**Figure 126: Configuration Tool: Editor for Opening the Work Load Balancing in the Manufacturing Concept**

- **Write Graphic Station Offsets to Database for Layout:** Activate this field if you want to save a graphic of the station for layout editing.
- **Multiple Process Usage is Allowed:** Activate this field if you want to use the same process several times in your work load balancing.
- **Enable Show Tools:** If you wish to activate the context menu entry *Show Tools* in the table view, activate this field. Once this field is active, the process or the process group can be selected in the table view (the same applies to the Enable Show Parts field) and the context menu entry *Show Tools* can be called up (in case tools are linked). *Please refer to the Figure 127.* Links to parts or tools are displayed in a dialog. This dialog cannot be edited.
- **Enable Show Parts:** If you wish to activate the context menu entry *Show Parts* in the table view, activate this field. Select in the table view, the process, or the process group and then open the context menu. Links of the processes or between the process assembly and parts will be shown in a dialog. The dialog cannot be edited.

| No. | Name          | Short name    | Time [min] | Time with | Freq. [%] |
|-----|---------------|---------------|------------|-----------|-----------|
|     | Station001    | tech.resource |            |           |           |
| 1   | ProcessGroup0 | Process Group | 0.58       | 0.58      | 100       |
| 2   | ProcessGroup1 | Process Group | 0.77       | 0.77      | 100       |
| 3   | Pro           | Group         | 0.81       | 0.81      | 100       |
| 4   | Pro           | Group         | 0.41       | 0.41      | 100       |
| 5   | Pro           | Group         | 0.37       | 0.37      | 100       |
|     | calc          |               |            |           |           |
|     | Sta           | source        |            |           |           |
| 6   | Pro           | Group         | 0.95       | 0.95      | 100       |
| 7   | Pro           | Group         | 0.77       | 0.77      | 100       |
| 8   | Pro           | Group         | 0.47       | 0.47      | 100       |
| 9   | Pro           | Group         | 0.65       | 0.65      | 100       |

**Figure 127: Enable Show Tools and Parts**

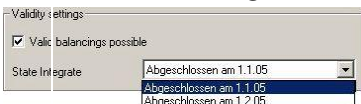
☒ Enable Partsbinmanagement

- **Enable Partsbinmanagement:** This field is active only, if no material provision view is activated. WLB takes over the relation management (Station >> parts bin, parts bin >> part and process >> part) if the material provision view is activated. Activate this field, if you want WLB to take over the relation management

even if the material provision view is switched off. The autorelation function must be switched off so that part bins are not shown repeatedly.

- **Enable Toolmanagement:** Activate this field, if the relation *balancing\_resource\_uses\_resource* is to be created between tool and station. Usually, a tool is linked to a process and the process is linked to a station. The tool will be linked to the station, if the tool management is activated and the tool and the relation are configured correctly between tool and process.
- **Enable Locking:** This option is available for Level 1 workload balancing only. This option must not be deactivated in the workload balancing levels 2 and 3. This field is active by default and must not be deactivated. If a workload balancing is opened for editing, then all processes that are used in the workload balancing are locked and cannot be changed by other users. If this option is deactivated, another user could edit or even delete processes that you use in the workload balancing.

#### 5.4.2.2 Validity Settings



Here you can select the planning status that you want to use for the work load balancing. In the selection list you can select an entry only if you have created one or more complete planning states in the project library. Deactivate the entry *Valid balancings possible* if there is no valid planning status.

#### 5.4.2.3 Number of Workplaces

Enter the number of workplaces here. If you open a level 3 work load balancing, you will be asked for the number of workplaces. Only the range that you set here is available to you as a selection in the work load balancing.

#### 5.4.2.4 Balance

Here you can set whether the work load balancing is executed for technical nodes or for workplaces. A work load balancing for workplaces is sensible only for a level 3 work load balancing or a raw balancing.

#### 5.4.2.5 Effectivity Representation

The period of time for the effectivities of a work load balancing can be set as days or calendar weeks. If you select calendar weeks you can additionally specify the starting date of the week.

**Weeks Starting on:** Set the start of a calendar week here. You can select between Sunday and Monday.

#### 5.4.2.6 Process Settings

- **Balance Process Groups:** Activate this entry if you only want to balance process groups.
- **Balance Single Processes:** Activate this entry if you only want to balance processes.

### 5.4.3 Views

The *Work Load Balancing* generally provides four views for the processing of the work load balancing:

Balancinglist with balanced processes: **Standard Grid View**

Non-balanced processes: **Additional Grid View**

Column diagram: **Diagram View**

Material provision view: **Area View**

The views you want to see in your work load balancing are set in the first column. In the other columns you can then set additional options for the respective view.

A view must always be displayed.

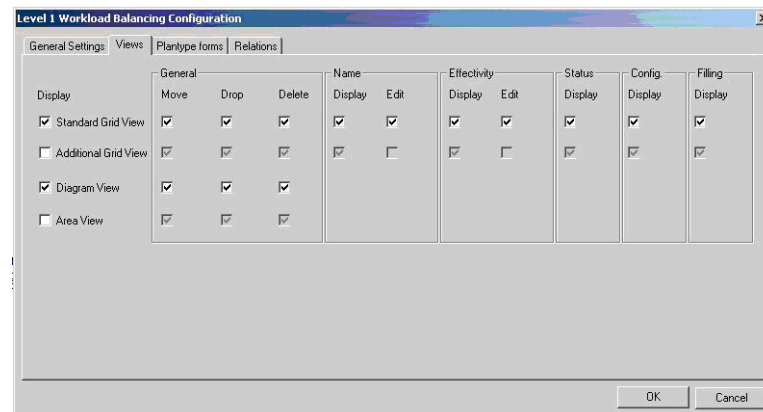


Figure 128: Balancing Configuration

#### 5.4.3.1 General Information

If you want to edit a view, you can set the editing level here. Selections include

- Moving within the view
- Inserting into the view by using drag and drop
- Deleting components from the work load balancing (not the object itself)

If you have not selected an option, you can use this view for displaying purposes only.

##### Name

Set whether the name of the work load balancing is to be displayed, and if it is displayed, whether it can be edited.

##### Effectivity

Here you can set whether the effectivities time period of the work load balancing is to be displayed, and is it displayed, whether it can be edited.

##### Status

Shows the planning status.

##### Config

Shows the name of the currently used Work Load Balancing configuration.

##### Filling

Is an additional attribute ("*balancing\_additional\_data\_01*") on the type **XDO-Balancing**, which you can use for displaying additional information.



## 5.4.4 Plantype Forms

| Component type | Attribute             | Value         | Attribute Lists |
|----------------|-----------------------|---------------|-----------------|
| Line sequence  | ergocomplantdefault   | identifierwlb | Building        |
| Line           | ergocomplantdefault   | identifierwlb | Line            |
| Station std    | ergocomplantstation   | identifierwlb | Station         |
| Process        | ergocomprocessdefault | identifierwlb | Operation       |
| Station add    | ergocomplantdefault   | identifierwlb | E/A-Order       |
| Part           | ergocomproductdefault | name          | Part            |
| Parts Bin      | ergocomplantdefault   | name          | Partsin         |
| Workplace      | ergocomprocess        | name          | Workplace       |

Figure 129: Balancing Configuration, Tab Plantype Forms

### 5.4.4.1 Component Type

Enter the types in here. All types of the Configuration Manager will be offered in a pop-up menu.

#### Attribute

In this field you can define the attribute that is to be used for the identification of the plantypes. All attributes of this type will be offered in a pop-up menu.

#### Value

In this field you can define the value of the attribute that you have selected in the field *Attribute*. Please refer to the [Value Input Field](#).

#### Attribute Lists

By using the button *Attribute Lists* you set the attributes of the previously selected type which are to be displayed in the work load balancing form. This attribute list refers only to the plantypes. Please refer to the [Attribute Lists](#).

#### Attribute Input Field

A plantype and its properties are described with multiple attributes. You can enter the attribute for the identification of the plantype, here.

All attributes of the type **Plantype** (XDOErgoPlanType) will be offered in a pop-up menu.

- Here you can enter the attribute you wish to use to identify the plantype. To identify this plantype, you still need the value of the selected attribute. You can get this value from the plantype set or its plantype. In the example of a station below, the value of the attribute *nameshort* is determined.

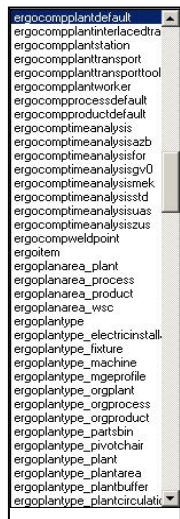
#### Value Input Field

This is the value for identifying the attribute.

### Example

#### Example

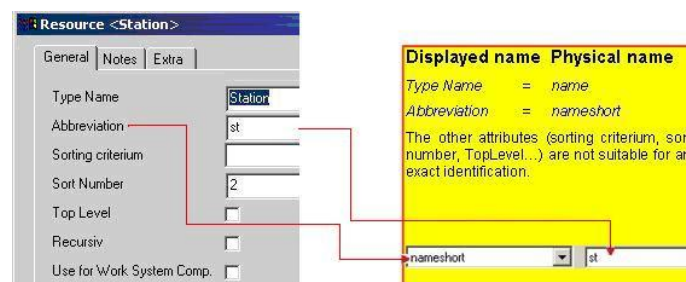
The Station type has an attribute with the attribute designation *nameshort* and the designation (displayed name) *Abbreviation*. The value that can be assigned to this attribute depends on the attribute definition. In this example you can enter any character string (string data type). Thus, the plantype Station assigns the attribute *nameshort* with the value **st**.





In order to use several plantypes at the same time in the balancing list, the identifying characteristic (value of the identification attribute) must be the same. Attributes such as *nameshort*, which are unique for each plantype, are therefore not suitable as identifying characteristics.

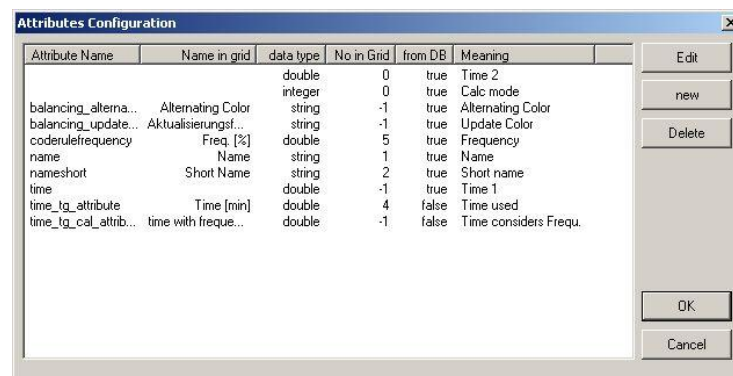
- 1) This value is necessary when configuring the balancing process.
  - 2) Open the resource view in the plantype set of your project.
  - 3) Open the plantype "Station".
- The abbreviation **st** corresponds to the value needed in the Work Load Balancing process. If the abbreviation is not displayed, proceed as described below.



**Figure 130: Attributes of the Station Resource Type**

### Attribute Lists

In the List of Attributes you will find all attributes which are displayed as a default in the balancing form. Some of these attributes are necessary to enable the balancing to function smoothly and therefore cannot be deleted. You can add additional attributes to these Lists of Attributes or you can change and hide existing attributes.



**Figure 131: Process Attributes**

To add a new attribute proceed as follows:

- 1) By using the "New" button you start the **Insert Attribute** editor where you can define the properties of an attribute.

**Figure 132: New Attribute****OK button**

Using this button the attribute is permanently saved in the database.

**Cancel button**

Using this button you can exit the dialog without saving the changes.

**Table 25: New Attribute Dialog Description**

| Properties             | Type            | Comment  |
|------------------------|-----------------|--|
| Attribute name         | Input field     | Enter the name of the new attribute here. You can find the correct name in the configuration or in the plan-type set   |
| Data type              | Selection field | Here you can see the data type of the selected attribute. The data types are described under Data Types in this manual.  |
| Database attribute     | Checkbox        | Database attributes are saved permanently in the database. If this checkbox is enabled, the attribute is a database attribute. All new attributes that you are creating should be database attributes. In the raw balancing configuration there are two attributes that are not database attributes. The value of these attributes is calculated using other attributes. These attributes should not be changed. |
| Show Attribute in Grid | Checkbox        | By checkmarking this checkbox you can determine whether this attribute should be displayed in the table (balancing list) or not. The subsequent properties are only visible after enabling this checkbox   |
| Name in grid           | Input field     | Enter the name here that should be displayed in the balancing list   |
| Col no. in grid        | Rotating field  | Select the column number of the balancing list where   |

| Properties     | Type            | Comment   |
|----------------|-----------------|---|
|                |                 | the attribute should be displayed. If the same numbers are used with multiple attributes, the attribute values are separated in the balancing list by an underscore, '_'.   |
| Grid col width | Rotating field  | You can set the column width for the balancing list here.   |
| Align text     | Selection field | You can specify the text position here: left, right or center.  |
| Format string  | Input field     | This field is dependent on the data type. With it you can influence the output format of the attribute. Entries you can make here are described in more detail under Data Types. If you make wrong entries here, severe errors can occur. |
| Output factor  | Input field     | This field is dependent on the data type and is only displayed when selecting a double data type. An entry in this field is multiplied by the attribute value as in the case of time conversions from minutes to seconds, for example     |

#### Deleting Attributes

To delete an attribute, select it and click the “Delete” button.

#### 5.4.4.2 Attribute Identifierwlb (Identification WLB)

In order to balance plantypes of the same kind simultaneously, these plantypes must be identifiable for balancing.

For this reason a new attribute for Work Load Balancing and a new attribute for Automatic Line Balancing were introduced in version PE 5.12.

The attribute **identifierwlb** is used to balance several plantypes of the same kind (for example in processes: test process, process non-value added, process value added). The allocation of this attribute calls the plantype for the balancing process. This is illustrated using an example:

#### Example PE 5.12:

#### Example

What should you do if you want to balance several process types?

In the first step you have to show the attribute **identifierwlb** in one or in all plantypes.

- 1) Select the plantype in the Configuration Manager and show the attribute *identifierwlb*.

|                  |                                    |
|------------------|------------------------------------|
| Name             | identifiervlb                      |
| Internet Name    | m_stridentifiervlb                 |
| Bezeichnung      | Identifizierung WLB                |
| Datentyp         | String                             |
| Typ des Controls | Standard                           |
| Gruppe           | Group::1001 (Page::1000 (General)) |
| Beschreibung     |                                    |
| Nutze Master     | Nein                               |

**Data Type Definition**

**Display and Position**

|                     |      |
|---------------------|------|
| Im Editor anzeigen  | Ja   |
| Im Browser anzeigen | Nein |
| Im Sucher anzeigen  | Nein |
| Position im Editor  | 105  |
| Position im Browser | 105  |
| Position im Sucher  | 0    |
| Position im Baum    | 0    |
| Länge begrenzt auf  | n    |

**Attribute Properties | Advanced Properties**

|                |                         |   |
|----------------|-------------------------|---|
| Type Name      | ergoplanttype           | <input type="checkbox"/> Is relevant          |
| Attribute Name | identifiervlb           | <input type="checkbox"/> Is Printable         |
| Physical Name  | m_stridentifiervlb      | <input type="checkbox"/> Use master           |
| Prompt         | WLB                     | <input type="checkbox"/> Read Only            |
| Copy Prefix    |                         | <input type="checkbox"/> Read-only for script |
| Defined by     | Default                 | <input type="checkbox"/> Constraint(034Z)     |
| Group - Page   | [1001] - [1000] General |   |

**Display**

|                     |     |  |
|---------------------|-----|--|
| Position in tree    | 0   |  |
| Position in Editor  | 100 | <input checked="" type="checkbox"/> Display in Editor  |
| Position in Finder  | 0   | <input type="checkbox"/> Display in Finder             |
| Position in Browser | 100 | <input checked="" type="checkbox"/> Display in Browser |

Figure 133: Attribute *identifiervlb*

- 2) Enter the same value (e.g. "wlb") in the attribute *identifiervlb* in all plantypes (system library / plantype sets) to be used for balancing processes.
  - 3) In the configuration of Work Load Balancing you must now select the attribute *identifiervlb* and enter the previously entered value for the process configuration.
- All plantypes with this value (wlb) in the attribute *identifiervlb* are used for balancing.

## 5.4.5 Additional Attributes

### 5.4.5.1 Attribute "balancing\_update\_identifier"

The attribute (data type: String) "*balancing\_update\_identifier*" can be used for highlighting or marking individual processes. The process is then shown in the work load balancing in color (red). You can find the attribute under the type XDOErgoCompProcessDefault.

How do you use this attribute?

- 1) In your plantype set select the process or process group plantype and select Edit in the context menu.
- 2) In the open Configuration Manager overwrite the attribute "*balancing\_update\_identifier*" of the base type *ergocompprocessdefault* for the plantype.

- 3) The attribute must be displayed on a page and must be able to be edited.
  - 4) Save your changes in the Configuration Manager.
- You can find the new attribute in the properties of a process.

As long as the attribute is empty, the process is not marked.

- If you enter a value into the attribute, the process is marked for the work load balancing. The value is shown in the work load balancing and the process is highlighted in color. Please refer to the [Figure 134](#).

| Nr. | Name       | Process number  | Time [min] | time with | Freq. [%] | Aktualisierungsfarbe | Alternating Color |
|-----|------------|-----------------|------------|-----------|-----------|----------------------|-------------------|
| 37  | Process084 | TechRes.Process | 0.02       | 0.022410  | 100       |                      |                   |
| 38  | Process085 | TechRes.Process | 0.10       | 0.100461  | 100       |                      |                   |
| 39  | Process086 | TechRes.Process | 0.12       | 0.115469  | 100       |                      |                   |
| 40  | Process087 | TechRes.Process | 0.05       | 0.051907  | 100       |                      |                   |
| 41  | Process088 | TechRes.Process | 0.08       | 0.082434  | 100       |                      |                   |
| 42  | Process089 | TechRes.Process | 0.15       | 0.147448  | 100       |                      |                   |
| 43  | Process090 | TechRes.Process | 0.04       | 0.042468  | 100       |                      |                   |
| 44  | Process091 | TechRes.Process | 0.11       | 0.110966  | 100       | a                    |                   |
| 45  | Process092 | TechRes.Process | 0.07       | 0.068237  | 100       |                      |                   |

Figure 134: Attribute Updating Color

#### 5.4.5.2 Attribute "balancing\_alternating\_identifier"

The attribute (data type: String) "*balancing\_alternating\_identifier*" can be used for grouping processes. The processes marked as such are then shown in the work load balancing with alternating colors whenever the value changes (green). In [Figure 135](#) the attribute was used to show the group affiliation of the processes. You can find the attribute under the type XDOErgoCompProcessDefault.

How do you use this attribute?

- 1) In order to show the attribute, proceed exactly as described for the attribute "*balancing\_update\_identifier*".

- In the properties of a process you can now find the new attribute

|  |                 |
|--|-----------------|
| Balancing Alternating Color Identifier | ProcessGroup048 |
|--|-----------------|

- If you enter a value into the attribute, the process is marked for the work load balancing. The value is shown in the work load balancing and the process is highlighted in color. Please refer to the [Figure 135](#).

| Nr. | Name         | Process number     | Time [min] | time with | Freq. [%] | Aktuali | Alternating Color |
|-----|--------------|--------------------|------------|-----------|-----------|---------|-------------------|
| 37  | Process084   | TechRes.Process    | 0,02       | 0,022410  | 100       |         |                   |
| 38  | Process085   | TechRes.Process    | 0,10       | 0,100461  | 100       |         |                   |
| 39  | Process086   | TechRes.Process    | 0,12       | 0,115469  | 100       |         |                   |
| 40  | Process087   | TechRes.Process    | 0,05       | 0,051907  | 100       |         | ProcessGroup044   |
| 41  | Process088   | TechRes.Process    | 0,08       | 0,082434  | 100       |         | ProcessGroup044   |
| 42  | Process089   | TechRes.Process    | 0,15       | 0,147448  | 100       |         | ProcessGroup045   |
| 43  | Process090   | TechRes.Process    | 0,04       | 0,042468  | 100       |         | ProcessGroup045   |
| 44  | Process091   | TechRes.Process    | 0,11       | 0,110366  | 100       | a       | ProcessGroup045   |
| 45  | Process092   | TechRes.Process    | 0,07       | 0,068237  | 100       |         | ProcessGroup046   |
| 46  | Process093   | TechRes.Process    | 0,19       | 0,187672  | 100       |         | ProcessGroup047   |
| 47  | Process094   | TechRes.Process    | 0,07       | 0,068236  | 100       |         | ProcessGroup047   |
| 48  | Process095   | TechRes.Process    | 0,06       | 0,064157  | 100       |         | ProcessGroup048   |
| 49  | Process096   | TechRes.Process    | 0,02       | 0,022474  | 100       |         | ProcessGroup048   |
| 50  | Process097   | TechRes.Process    | 0,08       | 0,080680  | 100       |         | ProcessGroup049   |
| 51  | Process098   | TechRes.Process    | 0,20       | 0,197533  | 100       |         | ProcessGroup049   |
| 52  | Process099   | TechRes.Process    | 0,06       | 0,058821  | 100       |         | ProcessGroup050   |
| 53  | Process100   | TechRes.Process    | 0,12       | 0,120119  | 100       |         | ProcessGroup050   |
|     | Berechnete Z |                    |            |           |           |         |                   |
|     | AUSGANG00    | TechRes.Output-Fol |            |           |           |         |                   |
|     | Berechnete Z |                    |            |           |           |         |                   |

**Figure 135: Attribute 'balancing\_alternating\_identifier'**

In contrast to the attribute "balancing\_update\_identifier" with which individual processes are highlighted, the attribute "*balancing\_alternating\_identifier*" is filled with different values of several processes, so that the color of the highlighting of the processes changes whenever the value is changed.

#### 5.4.5.3 Attribute "balancing\_number\_of\_workplaces"

The standard number of workplaces per station is taken from the attribute „*balancing\_number\_of\_workplaces*“ (data type: Integer) of the starting object of the level 3 work load balancing (here group) (only the part with whole numbers).

If the attribute is not configured or the value of the attribute is not within the interval defined in the configuration for the number of workplaces, a dialog is opened with the help of which the number of workplaces is set in the work load balancing.

You can find the attribute under the type XD OErgoCompBase.

How do you use this attribute?

- 1) The attribute must be configured on the starting object of the level 3 work load balancing (e.g. group). In order to show the attribute, proceed exactly as described for the attribute "*balancing\_update\_identifier*".
  - 2) The attribute should be write-protected and given a standard value. It does not have to be visible on the screen.
- As soon as this attribute is set, the prompt for the number of workplaces no longer appears when the level 3 work load balancing starts (Figure 136) and the work load balancing is started with the previous value.



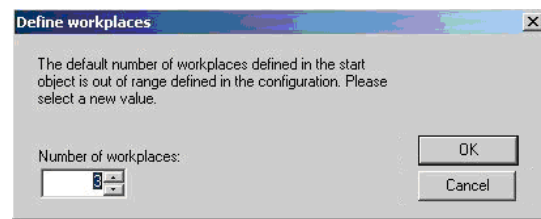


Figure 136: Dialog for Specifying the Number of Workplaces

#### 5.4.5.4 Attribute "balancing\_customizationname"

If you have started a work load balancing on an object in the resource view, the Work Load Balancing takes note of the names of configuration and upon the next opening it starts the work load balancing with the same configuration.

In order not to display the selection dialog when there are several dialogs, use the attribute "*balancing\_customizationname*".

Proceed as follows:

- 1) If you always want to start with the same configuration in PPR-Navigator or for a new work load balancing in the resource view, you must configure the attribute "*balancing\_customizationname*" (data type String) on your starting object and enter the name of the work load balancing configuration.
- 2) You can overwrite the attribute in the Configuration Manager of the Process Engineers of *XDOErgoCompBase*.

For a level 3 work load balancing you have e.g. three starting objects (line, organizational line, and group), and three different configuration names for the respective starting objects.

#### 5.4.5.5 Attribute "graphicname"

**Graphic macro for displaying the station in the layout**

This is configured on the plantype Station. This attribute is not shown on the screen and it can not be edited.

*M balancing\_stationarea\_default* must be entered as a standard value. This is the name of the macro.

#### 5.4.5.6 Attributes of the Station

The macro accesses the data of the station. Therefore the following attributes must be available at the station:

Table 26: Station Attributes

| Attribute        | Meaning | Type   | Range | Comment |
|------------------|---------|--------|-------|---------|
| Materialareawith |         | double |       |         |
| Length           |         | double |       |         |
| Width            |         | double |       |         |

If the attributes are incorrect or contain incorrect values, the interplay between material provision view and the layout is incorrect (stations are then shown incorrectly and parts bins are not shown at the right places).



### 5.4.5.7 Cycle Time

If no cycle time is specified in the resource to which you open the balancing process, a message will indicate this. Even if a cycle time is not defined, you can still open the balancing process.

- 1) Confirm the message with **OK**. The balancing process will then be opened. As a default, the balancing process then uses a cycle time of one minute.



**Figure 137: Message whenever no cycle time has been specified**

- 2) First check the properties dialog to see if there is a “Cycle time” input field.
  - If not found, you must create a “Cycle time” input field using an attribute.

You can show the attribute in the plant type set or in the configuration manager.

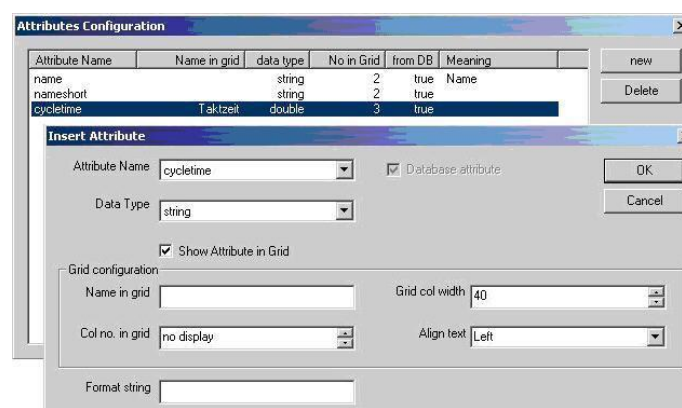
The following attributes can be used as the cycle time:

All double attributes

The attribute « **cycletime** »

The attribute “cycletime” is intended as the default for the cycle time. Now you just need to show it on a page.

- 3) Then open the configuration of the balancing process.
- 4) Open the attribute list of the line (Level 2 = line).
  - For example in your configuration, the attribute **dbl\_attribute\_10** is pre-defined for the cycle time. This attribute is the attribute shown on Level 2 (the line in the example).



**Figure 138: Attribute Configuration**

- 5) Open this attribute.
- 6) In the selection field for attribute name select the attribute *cycle time*.

7) Click **OK** to leave the dialog.

You can use any double attribute for the cycle time, but you have to assign this attribute to the meaning 'Cycle Time' in the configuration of the balancing process.

## 5.4.6 Relations

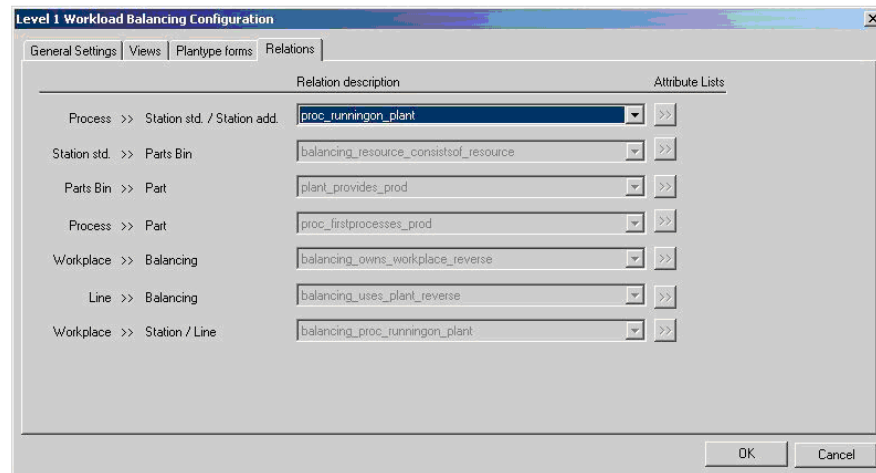


Figure 139: Balancing - Configuration; Tab Relations

### 5.4.6.1 Relation Description

Select the relation to be used in the selection list. Only the relations that are possible between the two plantypes and which have been created in the system are offered.

The following relations are used specially by Work Load Balancing:

**balancing\_owns\_workplace** and **balancing\_owns\_workplace\_reverse**:

Relation from work load balancing object to the workplace.

➤ In Status complete change: Yes

**balancing\_proc\_runnigon\_plant**: Relation from the workplace to the resource (e.g. station).

➤ In Status complete change: Yes

**balancing\_resource\_consistsof\_resource**: Relation of station, usage folder to the parts bin.

➤ In Status complete change: Yes

**balancing (ergocomplantdefault - balancing)**: Relation from the balancing object to the resource.

➤ In Status complete change: Yes

**balancing (ergocomporplant - balancing)**: Relation from work load balancing object to the organizational resource.

➤ In Status complete change: Yes

**balancing\_uses\_plant** and **balancing\_uses\_plant\_reverse**: Relation from the balancing object to the resource.

- In Status complete change: Yes

**balancing\_resource\_consistsof\_resource:** Relation from Tool to Station.

- Change in the status complete: Yes

Using the script action below (AFTER), you can have the internal name of a relation displayed.



```
function sa_link(parent_id, rel_ids, listname)
    relname = Data.GetAttributeById(rel_ids(0), "relationname")
    MsgBox(relname)
end function
```

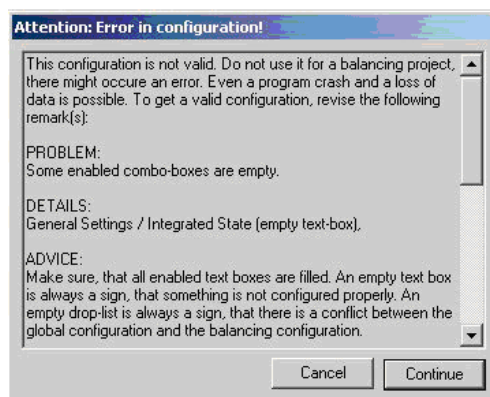
### 5.4.7 Using Change Management with Work Load Balancing



For more information, Please refer to the Appendix in the [Work Load Balancing User's Manual](#).

### 5.4.8 Error Messages in the Configuration of Work Load Balancing

If a configuration is left by clicking OK, the Work Load Balancing checks its integrity. If the Work Load Balancing recognizes an error in the configuration, you will be informed of this.



**Figure 140: Message if an Error Appears in the Configuration**

The message from Figure 140 indicates to you that 'Valid balancing possible' is activated, but no planning status has been selected.

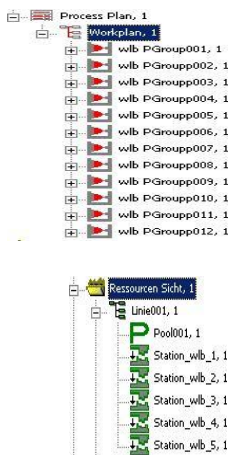
- 1) By clicking *abort* you return to the configuration and you can rectify the error.

If you leave the configuration despite the error warning, you will not be able to execute the work load balancing without errors. You will receive an alert when starting the work load balancing.

### 5.4.9 Example of a 3 level Work Load Balancing

There is a process structure with 12 process groups.

These process groups are distributed amongst five stations.



- 1) You can link the process groups to the stations in the PPR-Navigator or start the work load balancing on the line as shown in the example.

If you create the links in the PPR-Navigator, the WLB configuration is not called up and error messages may appear when opening the work load balancing. The multiple usage of processes could be used as an example. If you have deactivated them in the WLB configuration, but have linked the same process several times in a station via the PPR-Navigator, you will receive an error message when opening the work load balancing and only one process is shown in the balancinglist.

The corresponding configuration - for level 1 or raw balancing - looks like this:

Figure 141: Work Load Balancing Configuration

|  | General                             |                                     |                                     | Name                                |                                     | Effectivity                         |                                     | Status                              | Config                              | Filling                  |
|--|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|
| Display  | Move                                | Drop                                | Delete                              | Display                             | Edit                                | Display                             | Edit                                | Display                             | Display                             | Display                  |
| <input checked="" type="checkbox"/> Standard Grid View   | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> Additional Grid View | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> Diagram View         | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |                                     |                                     |                                     |                                     | <input checked="" type="checkbox"/> |                                     |                          |
| <input checked="" type="checkbox"/> Area View            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |                                     |                                     |                                     |                                     |                                     |                                     |                          |

Figure 142: Level 1 Work Load Balancing Configuration A

| Component type | Attribute | Value        | Attribute Lists |
|----------------|-----------|--------------|-----------------|
| Line sequence  | name      | Building     | >>              |
| Line           | name      | Line         | >>              |
| Station std    | name      | Station      | >>              |
| Process        | name      | ProcessGroup | >>              |
| Station add    | name      | Pool         | >>              |
| Tool           | name      | Tool         | >>              |
| Part           | name      | Part         | >>              |
| Parts Bin      | name      | Partbin      | >>              |
| Workplace      | name      | Workplace    | >>              |

Figure 143: Level 1 Work Load Balancing Configuration B

| Relation description                  | Attribute Lists                       |
|---------------------------------------|---------------------------------------|
| Process >> Station std. / Station add | proc_runningon_plant                  |
| Station std. >> Parts Bin             | balancing_resource_consistof_resource |
| Station std. >> Tool                  | balancing_resource_uses_resource      |
| Parts Bin >> Part                     | plant_provides_prod                   |
| Process >> Part                       | proc_firstprocesses_prod              |
| Process >> Tool                       | proc_uses_plant                       |
| Workplace >> Balancing                | balancing_owns_workplace_reverse      |
| Line >> Balancing                     | balancing_uses_plant_reverse          |
| Workplace >> Station / Line           | balancing_proc_runningon_plant        |

Figure 144: Level 1 Work Load Balancing Configuration C

The work load balancing is started on the line.

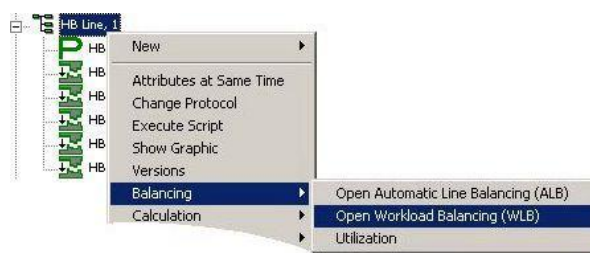
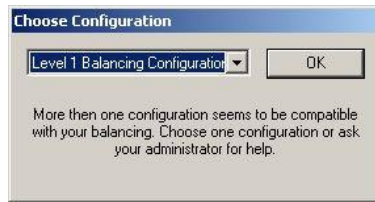


Figure 145: Work Load Balancing Started

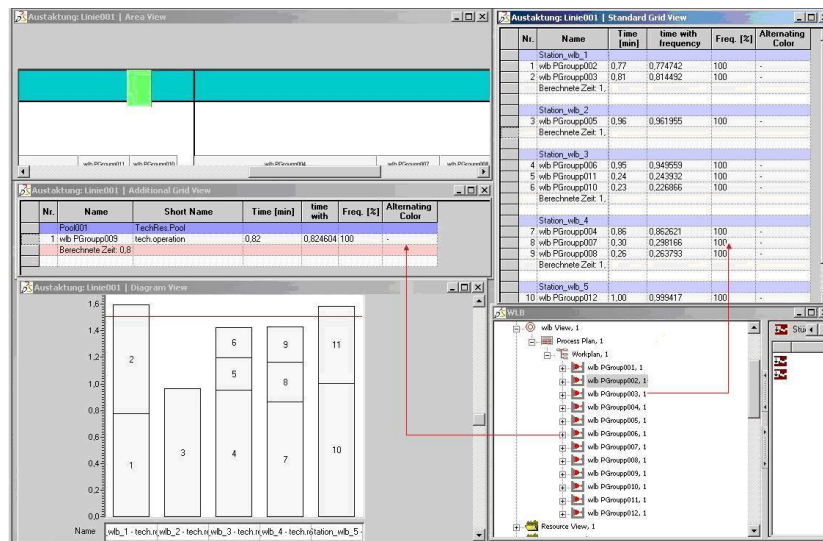
If there are several configurations, the corresponding one should be selected



**Figure 146: Choose Configuration**

The four views of the work load balancing open.

The processes can now be assigned to the stations from the PPR-Navigator.



**Figure 147: Four Views of Work Load Balancing**

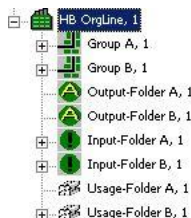
Save the work load balancing.

In the next step, a level 2 work load balancing is to be created. To do this, several objects must be created and linked to one another.

An additional organizational node is created in the resource view.

This organizational node should contain two groups.

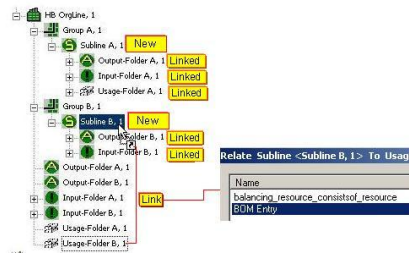
If it has two groups, an output folder, input folder, and a usage folder must also be created for both groups. The structure then looks like the following figure. *Please refer to the Figure 148.*



**Figure 148: Work Load Balancing Structure**

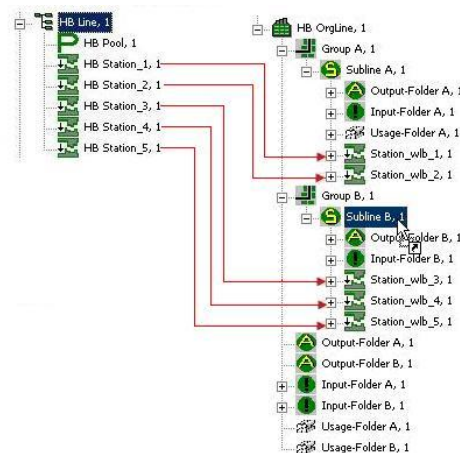
Every group must receive at least one intermediate level (Subline).

Every one of the intermediate levels (Subline) must be linked to an output folder, input folder and a usage folder as a bill of materials entry.



**Figure 149: BOM Entry**

The stations that should be under the group must be a part of the stations of the level 1 work load balancing. Therefore the stations of the level 1 work load balancing must also be linked as a bill of materials entry in the groups. Please refer to the [Figure 150](#).



**Figure 150: Link stations with the Subline**

The resources view is now opened on the organizational node. (Context menu on **HB OrgLine > Open in > Process Engineer**).

Create a new work load balancing on the organizational node; (Context menu on **HB OrgLine < New < Work load balancing**).

The corresponding configuration of a level 2 work load balancing looks like the following:

**Figure 151: Level 2 Work Load Balancing**



The level 2 work load balancing can be executed with process groups or with processes. If process groups were used in the level 1 work load balancing, the processes of this process group are used for the work load balancing in the level 2 work load balancing, assuming **Balance single processes** was activated.

If **Balance single processes** has not been activated, only process groups are balanced.

The input directory is dependent on the planttype and must be selected therefore.

| Display  | General                             |                                     |                                     | Name                                |                                     | Effectivity                         |                                     | Status                              | Config.                             | Filling                  |
|--|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|
|  | Move                                | Drop                                | Delete                              | Display                             | Edit                                | Display                             | Edit                                | Display                             | Display                             | Display                  |
| <input checked="" type="checkbox"/> Standard Grid View   | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> Additional Grid View | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> Diagram View                    | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            |                                     |                                     |                                     |                                     |                                     |                                     |                          |
| <input type="checkbox"/> Area View                       | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            |                                     |                                     |                                     |                                     |                                     |                                     |                          |

Figure 152: Level 2 Work Load Balancing Customization A

| Component type | Attribute             | Value         | Attribute Lists |
|----------------|-----------------------|---------------|-----------------|
| Line sequence  | ergocomplantdefault   | name          | Building        |
| Line           | ergocomplant          | name          | OrgLine         |
| Station std.   | ergocomplantdefault   | name          | Usage-Folder    |
| Process        | ergocomprocessdefault | name          | Process         |
| Station add.   | ergocomplantdefault   | identifiervlb | I/O Folder      |
| Tool           | ergocomplantdefault   | name          | Tool            |
| Part           | ergocomproductdefault | name          | Part            |
| Parts Bin      | ergocomplantdefault   | name          | Partsbin        |
| Workplace      | ergocomorgprocess     | name          | Workplace       |

Figure 153: Level 2 Work Load Balancing Customization B

Figure 154: Level 2 Work Load Balancing Customization C

The two views of the level 2 work load balancing are opened.

| Nr.                    | Name       | Process number  | Time [min] | time with | Freq. [2] | Aktualisierungsstatus | Alternating Color |
|------------------------|------------|-----------------|------------|-----------|-----------|-----------------------|-------------------|
| <b>Output-Folder A</b> |            |                 |            |           |           |                       |                   |
| Berechnete Zeit        |            |                 |            |           |           |                       |                   |
| <b>Output-Folder B</b> |            |                 |            |           |           |                       |                   |
| Berechnete Zeit        |            |                 |            |           |           |                       |                   |
| <b>Input-Folder A</b>  |            |                 |            |           |           |                       |                   |
| 1                      | Process003 | TechRes Process | 0.00       | 0.003504  | 100       |                       |                   |
| 2                      | Process004 | TechRes Process | 0.19       | 0.190161  | 100       |                       |                   |
| 3                      | Process005 | TechRes Process | 0.18       | 0.177260  | 100       |                       |                   |
| 4                      | Process006 | TechRes Process | 0.01       | 0.011338  | 100       |                       |                   |
| 5                      | Process009 | TechRes Process | 0.22       | 0.217862  | 100       |                       |                   |
| 6                      | Process010 | TechRes Process | 0.01       | 0.014059  | 100       |                       |                   |
| Berechnete Zeit        |            |                 |            |           |           |                       |                   |
| <b>Input-Folder B</b>  |            |                 |            |           |           |                       |                   |
| 7                      | Process011 | TechRes Process | 0.09       | 0.091005  | 100       |                       |                   |
| 8                      | Process012 | TechRes Process | 0.13       | 0.131217  | 100       |                       |                   |
| 9                      | Process021 | TechRes Process | 0.13       | 0.133469  | 100       |                       |                   |
| 10                     | Process022 | TechRes Process | 0.03       | 0.026592  | 100       |                       |                   |
| 11                     | Process019 | TechRes Process | 0.17       | 0.172778  | 100       |                       |                   |
| 12                     | Process020 | TechRes Process | 0.25       | 0.245001  | 100       |                       |                   |
| 13                     | Process023 | TechRes Process | 0.17       | 0.169044  | 100       |                       |                   |
| 14                     | Process024 | TechRes Process | 0.00       | 0.003525  | 100       |                       |                   |
| 15                     | Process001 | TechRes Process | 0.07       | 0.072391  | 100       |                       |                   |
| 16                     | Process002 | TechRes Process | 0.08       | 0.075487  | 100       |                       |                   |
| 17                     | Process007 | TechRes Process | 0.20       | 0.197620  | 100       |                       |                   |
| 18                     | Process008 | TechRes Process | 0.09       | 0.093384  | 100       |                       |                   |
| 19                     | Process013 | TechRes Process | 0.16       | 0.155674  | 100       |                       |                   |
| 20                     | Process014 | TechRes Process | 0.16       | 0.161956  | 100       |                       |                   |
| 21                     | Process015 | TechRes Process | 0.07       | 0.069636  | 100       |                       |                   |
| 22                     | Process016 | TechRes Process | 0.21       | 0.207451  | 100       |                       |                   |
| Berechnete Zeit        |            |                 |            |           |           |                       |                   |

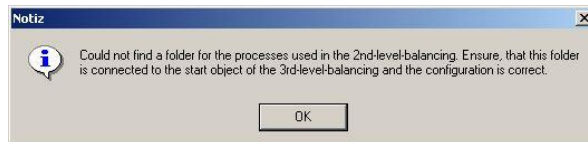
Figure 155: Level 2 Work Load Balancing for Processes



For information on how to work with the level 2 work load balancing, *Please refer to the [Work Load Balancing Manual](#).*

Save the work load balancing.

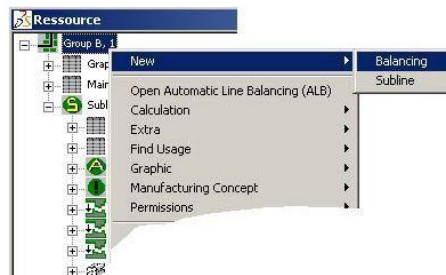
If when saving the work load balancing you receive the following error message, the link either to an input, output, or usage folder is missing. The work load balancing can therefore not be saved. Only after all folders have been linked correctly can the work load balancing be saved.



**Figure 156: Error Message**

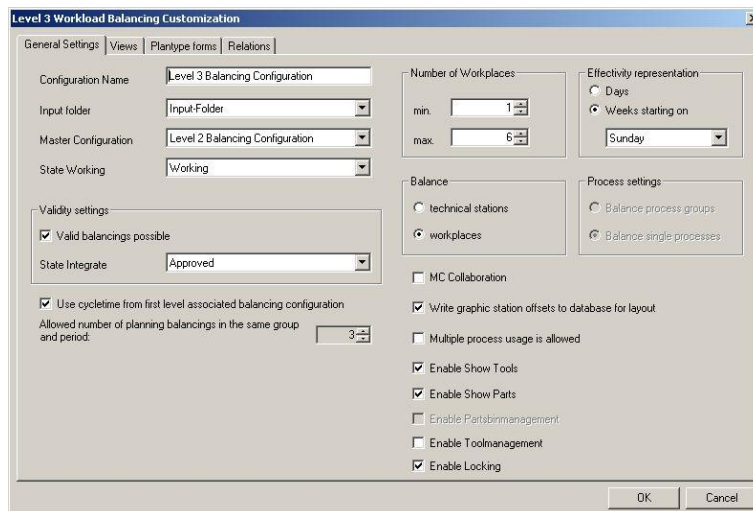
In the next step, a level 3 work load balancing is to be created. No further objects must be created for this.

- 1) The resource view is opened on a group, e.g. Group A. (**Context menu on group A < Open in < Process Engineer**).
- 2) Create a new work load balancing on group A; (**Context menu on group A < New < Work Load Balancing**).

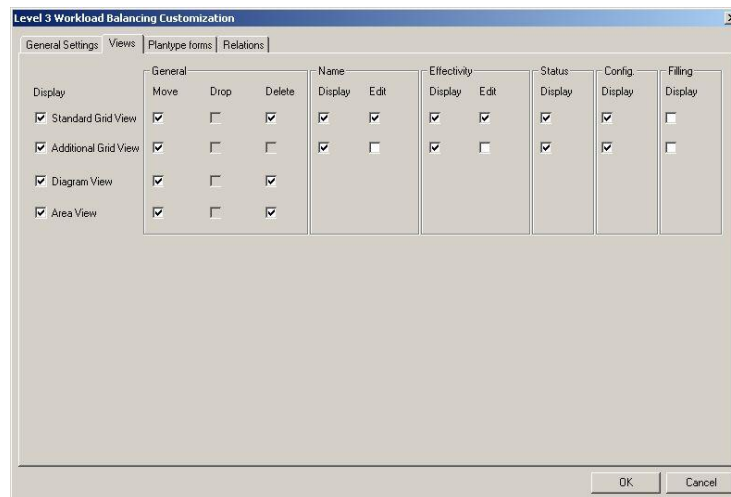


**Figure 157: Create a New Work Load Balancing**

The corresponding configuration looks like the following (a level 3 balancing or fine work load balancing):



**Figure 158: Level 3 Work Load Balancing Customization A**

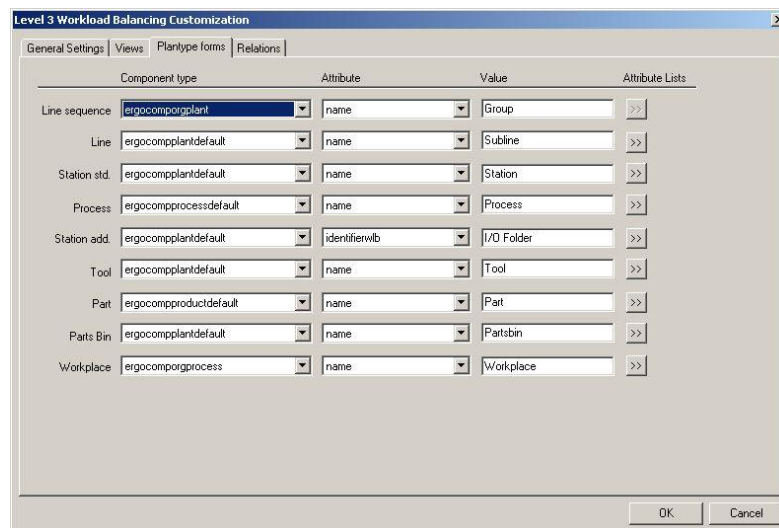


The dialog box 'Level 3 Workload Balancing Customization' has tabs for General Settings, Views, Planttype forms, and Relations. The 'Views' tab is active, showing a table of view settings.

| Display  | General                             |                          |                                     | Name                                |                                     | Effectivity                         |                                     | Status                              | Config                              | Filling                  |
|--|-------------------------------------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|
|  | Move                                | Drop                     | Delete                              | Display                             | Edit                                | Display                             | Edit                                | Display                             | Display                             | Display                  |
| <input checked="" type="checkbox"/> Standard Grid View   | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> Additional Grid View | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> Diagram View         | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |                                     |                                     |                                     |                                     | <input checked="" type="checkbox"/> |                                     |                          |
| <input checked="" type="checkbox"/> Area View            | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |                                     |                                     |                                     |                                     |                                     |                                     |                          |

Buttons: OK, Cancel

Figure 159: Level 3 Work Load Balancing Customization B



The dialog box 'Level 3 Workload Balancing Customization' has tabs for General Settings, Views, Planttype forms, and Relations. The 'Relations' tab is active, showing a table of component types and their attributes.

| Component type | Attribute             | Value        | Attribute Lists |
|----------------|-----------------------|--------------|-----------------|
| Line sequence  | ergocompplant         | name         | Group           |
| Line           | ergocomplantdefault   | name         | Subline         |
| Station std.   | ergocomplantdefault   | name         | Station         |
| Process        | ergocomprocessdefault | name         | Process         |
| Station add.   | ergocomplantdefault   | identifierwb | I/O Folder      |
| Tool           | ergocomplantdefault   | name         | Tool            |
| Part           | ergocomproductdefault | name         | Part            |
| Parts Bin      | ergocomplantdefault   | name         | Partsin         |
| Workplace      | ergocomprocess        | name         | Workplace       |

Buttons: OK, Cancel

Figure 160: Level 3 Work Load Balancing Customization C

- 3) Save the work load balancing configuration. For more information on how to work with the level 3 work load balancing, *please refer to the [Configuring Work Load Balancing](#)*.

## 6. Configuration of Automatic Line Balancing

Automatic Line Balancing (ALB) and WLB are configured in the same dialog in version PE 5.12. If you have already worked with WLB in previous versions, you are already familiar with the basic setup of this dialog; the structure and mode of functioning of the new dialog are identical. The dialog is opened using the context menu on the project node.

You can customize the configuration of ALB according to your needs. In earlier versions certain ALB settings had to be made using the Configuration Manager; they can be made more quickly in the new dialog of version PE 5.12.

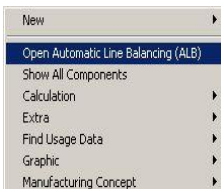
You can also make these settings using the Configuration Manager, as in previous versions. The procedure for doing this is explained briefly below.

The setting for the linking process generated plantypes is no longer required; it has been removed without a replacement.

### 6.1 Individual Configuration (only Version PE 5.10, 5.11)

Automatic Line Balancing is completely integrated in the DELMIA Process Engineer®. To be able to use Automatic Line Balancing, the context menu entry must be enabled before starting.

- 1) Open the Configuration Manager and select the entry **ergocomplantdefault**.
- 2) Open the *Editors* directory and search for the menu text entry **Balancing** and the browser ID **epcompeditor-component**.
- 3) Enter a call text in the menu text input field, for example *Start Automatic Line Balancing* and activate **Browser on single select**. Save the configuration.
- 4) The entry is now available in the context menu of a resource.



### 6.2 New Installation

To work with the Automatic Line Balancing module the following adjustments need to be made in the configuration and in the plantype set after a reinstallation:

Changing of values of car body positions in the configuration.

The module Automatic Line Balancing supports twelve car body positions, which are listed in the table. *Please refer to the Table 27:*

**Table 27: Car Body Positions**

| Name              | Abbreviation |
|-------------------|--------------|
| Rear left         | HL           |
| Rear center left  | HML          |
| Front center left | VML          |

| Name               | Abbreviation |
|--------------------|--------------|
| Front left         | VL           |
| Rear center        | HM           |
| Inside rear        | IH           |
| Inside front       | IV           |
| Front center       | VM           |
| Rear right         | HR           |
| Rear center right  | HMR          |
| Front center right | VMR          |
| Front right        | VR           |

- 1) To change the values, open the Configuration Manager and search the type ***ergocompprocessdefault***.
- 2) Select the attribute ***carbodyposition***.
- 3) Under the attribute *carbodyposition* hide the values already entered in the value list.
- 4) Then enter the new values as described in the [Table 28](#).

**Table 28: New Values**

| Item Name | Item Value | Order |
|-----------|------------|-------|
| HL        | 1          | 15    |
| HML       | 2          | 16    |
| VML       | 4          | 17    |
| VL        | 8          | 18    |
| HM        | 16         | 19    |
| IH        | 32         | 20    |
| IV        | 64         | 21    |
| VM        | 128        | 22    |
| HR        | 256        | 23    |
| HMR       | 512        | 24    |
| VMR       | 1024       | 25    |
| VR        | 2048       | 26    |

As a result you will get the following value list. Please refer to the [Figure 161](#).

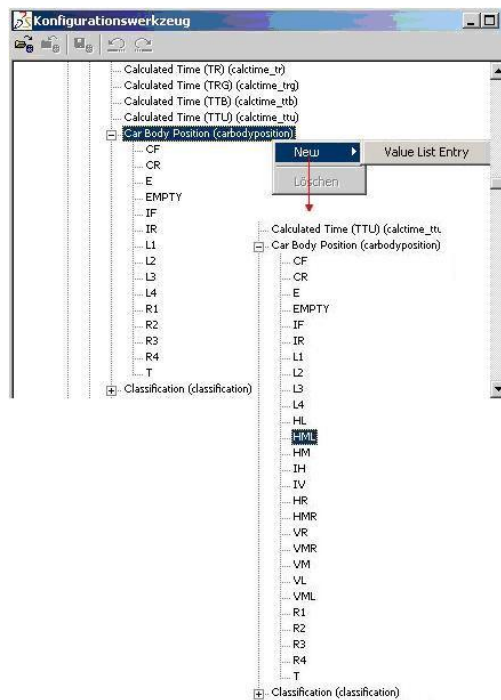


Figure 161: Value list of the attribute carbodyposition

### 6.2.1 Creating New Plantypes

The Automatic Line Balancing uses shelf libraries for planning shelves. The shelf libraries contain the respective shelves which can be used for planning. The purpose behind shelf libraries is to be able to use alternative shelves in the ALB. Shelf libraries must be created in templates of the project library.

Two new type resource plantypes must be created for planning shelves, depending on the plantype set used:

- Shelf
- Shelf Library

The plantype **shelf** is a child of the shelf library and the plantype used for the stations, e.g. in the "machine" or "work location".

The shelf library must be a child of the resource view.

Whenever the **Container** plantype is mentioned in Automatic Line Balancing, this refers to the plantype used for system elements. You can also create your own type resource plantype for the container.

#### To Create Shelves and Shelf Library Plantypes:

- 1) Open the plantype set you want to edit in the system library.
- 2) You can create two new resource plantypes: select the plantype "Resource" in the context menu: New / Resource and enter the name **Shelf Library** and **Shelves**. You also must enter a name for the identification ALB attribute.



Figure 162: New Plantype Shelf

### Shelf

- 1) Link the shelf to the shelf library and the plantype which is being used as stations.
- 2) Link the shelf library to the resource view.

### Material Area

The Automatic Line Balancing uses the plantype material area for the layout:  
The data model von Automatic Line Balancing can be found in the appendix.

**Material area** = area ergocomplantarea

The material area must be a child of the station.

## 6.2.2 Attributes

### 6.2.2.1 Material Area, Ergocomplantarea

A material area must have the following attributes:

**Name** of the material area.

*name*

**Number**

*nameshort*

Overwrite the following attributes from *ergocomplantdefault*:

**Page of the material area:**

*Containerareaside* 0 = undefined  
1 = left  
2 = right

**Length [mm]:**

*Length*

**Width [mm]:**

*Width*

**restricted area:**

*materialarea\_locklength\_left*

**Graphic macro for displaying the material area**

*graphicname*

This attribute is not shown on the screen and it can not be edited.

The graphic macro ***alb\_materialarea\_default*** must be available in the directory `\\DELMIA\\PPRServer\\data\\segrim\\Graphik`.

**6.2.2.2 Creating Containers**

A container must have the following attributes:

**Possible shelves:**

Internal shelf planning data (*alb\_shelfplanningattributes*)

**Number of levels of the standard shelf:**

Number of levels (*numberoflevels*)

There should be exactly one standard shelf in every shelf library for every one to six levels. ALB uses these shelves for the automatic balancing process. Non-standard shelves must be added manually.

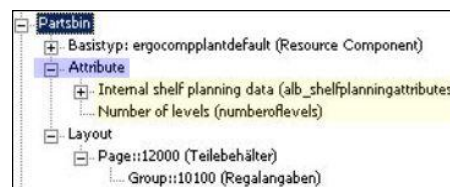
A template can contain several shelf libraries. If there are several shelf libraries available (even from various templates), ALB, while loading, asks you which shelf library should be used.

The standard shelf refers to the shelf which is automatically added in the automatic balancing process, assuming the container does not fit in any existing shelf.

You can enter a number between 1 and a freely definable upper level limit for the shelf levels (in the following example, the upper limit is 6 levels). In doing this you set the number of shelf levels for the standard shelf that can be used in the shelf library. Make sure that the container fits on its shelf.

The container can be placed on shelves of which the number of levels is specified in the attribute *possible shelves*.

The standard shelf should be located in the *possible shelves*; otherwise ALB issues an error message when loading and ignores the shelf attributes of the container.



**Figure 163: Error Message**

The attribute **Internal shelf planning data (*alb\_shelfplanningattributes*)** in *ergocomplantdefault* specifies in which shelves containers can be placed and should be used for the number of *possible shelves*.

- 1) Overwrite this attribute for container plantypes. Then create a value list. The values must be entered as described in the following table.

Low-value bits (entered as an internal value) specify whether the container can be stored in the respective shelf. 12 (= 1100 binary) therefore means that the container fits into standard shelves 3 and 4.

In the following table you will find the values described for shelves from one to six levels:

**Table 29: Value Calculation for Shelf Levels**

| First Shelf | Last Shelf | Value |
|-------------|------------|-------|
| 1           | 1          | 1     |
| 1           | 2          | 3     |
| 1           | 3          | 7     |
| 1           | 4          | 15    |
| 1           | 5          | 31    |
| 1           | 6          | 63    |
| 2           | 2          | 2     |
| 2           | 3          | 6     |
| 2           | 4          | 14    |
| 2           | 5          | 30    |
| 2           | 6          | 62    |
| 3           | 3          | 4     |
| 3           | 4          | 12    |
| 3           | 5          | 28    |
| 3           | 6          | 60    |
| 4           | 4          | 8     |
| 4           | 5          | 24    |
| 4           | 6          | 56    |
| 5           | 5          | 16    |
| 5           | 6          | 48    |
| 6           | 6          | 32    |

#### Formula for the evaluation

The value is calculated with the following formula:

$X_1$  = Levels of the first shelf

$X_2$  = Levels of the last shelf

$X_3$  = Levels of the last shelf - Levels of the first shelf = possible levels +1

$$2^{(\text{levels of the first shelf} - 1)} \cdot (2^{(\text{possible levels})} - 1)$$

$$\text{Value} = 2^{x1-1} \times 2^{x3} - 1$$

or

$$\text{Value} = 2^{x2} - 2^{(x1-1)}$$

You can use the values from the table *Value calculation for shelf levels* as an aid in creating the following value list. The value list is created in the same way as with the attribute *carbodposition*.

**Table 30: Value List**

| Display Value (Example) | Item Value | Display Order |
|-------------------------|------------|---------------|
| 1 level                 | 1          | 1             |
| 1 - 2 levels            | 3          | 2             |
| 1 - 3 levels            | 7          | 3             |
| 1 - 4 levels            | 15         | 4             |
| 1 - 5 levels            | 31         | 5             |
| 1 - 6 levels            | 63         | 6             |
| 2 levels                | 2          | 7             |
| 2 - 3 levels            | 6          | 8             |
| 2 - 4 levels            | 14         | 9             |
| 2 - 5 levels            | 30         | 10            |
| 2 - 6 levels            | 62         | 11            |
| 3 levels                | 4          | 12            |
| 3 - 4 levels            | 12         | 13            |
| 3 - 5 levels            | 28         | 14            |
| 3 - 6 levels            | 60         | 15            |
| 4 levels                | 8          | 16            |
| 4 - 5 levels            | 24         | 17            |
| 4 - 6 levels            | 56         | 18            |
| 5 levels                | 16         | 19            |
| 5 - 6 levels            | 48         | 20            |
| 6 levels                | 32         | 21            |

### 6.2.2.3 Creating Shelves

Shelves must be created in templates of the project library. In order to do this, select the planttype set in which you created the shelves and the shelf library.

The template must be opened as a new project in order for it to be edited.



For more information on “how templates are created in the project library”, *Please refer to the [Project Library Manual](#).*

A shelf must have the following attributes:

- **Name:** The name of the shelf
- **Number:** The number of the shelf
- **Standard shelf:** specifies whether the shelf is a standard shelf
- **Number of levels:** The number of levels of the shelf

- **Length, width, height:** The outer dimensions of the shelf
- **Inner length** (inner width, inner height): The inner dimensions of a compartment of the shelf. All compartments have the same dimensions.
- **Default shelf (*isdefaultshelf*):** specifies whether a shelf is a standard shelf
- **Inner length (*innerlength*):** specifies the inner length of a compartment of a shelf.
- **Number of levels (*numberoflevels*):** specifies the number of levels for shelves as well as the standard shelf for containers.

The parent-child relations between containers and shelves must be specified in the configuration tool for ALB:

*albspec\_Partsbin\_positioned\_in\_shelf c*

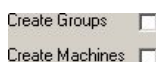
*albspec\_Partsbin\_positioned\_in\_shelf\_reverse* between *ergocomplantdefault* and *ergocomplplantdefault* in the relation class *XDORelationship*.

As of version PE 5.14SP2, new attributes have been introduced for shelves with the same heights, angles and overhangs. To display these attributes in the properties dialog, they must first be configured. A summary of all attributes and plantypes can be found in the appendix. *Please refer to the [Data model for Automatic Line Balancing](#).*

### 6.2.3 Additional Notes

- If a message appears, the workload balancing can still be continued. This message is only a note for used car body positions, which are not used by automatic line balancing (attribute *carbodyposition*).

Whenever you start a balancing process and select a manufacturing concept, you must deactivate *Create Groups* and *Create Machines*.



## 6.3 Preparing ALB plantype Sets

With this release, Automatic Line Balancing (ALB) produces results that can be displayed and read in V5 for stations and workplaces that have been created with Automatic Line Balancing. To obtain these types of results, it was necessary to change relationships and provide new plantypes for Automatic Line Balancing.

To get the most out of the ALB with the assistance of new plantypes, such as the *Logical Part Bin* (which works as the storage medium that creates the link between the material area and the container), or the plantype *DNBBehavior* (which is used to display *TSA* – resource dependant processes), the appropriate settings must first be made in the plantype set and in the ALB-configuration.

### 6.3.1 New Plantypes

With the assistance of the new plantypes, results of the Automatic line balancing can continue to be utilized and displayed in V5.



The table provides an overview of which new plantypes need configuring.

For more information, *Please refer to the sections “Configuring Automatic Line Balancing” for ALB Configuration “Planning resource-related processes” and “Providing data for part bin usage” in the [Automatic Line Balancing Manual](#).*

Attached is the table of the attributes, plantypes and relations that describe the Automatic Line Balancing data model.

**Table 31: Table of New Plantypes**

| Type                   | Plantype             |
|------------------------|----------------------|
| Ergocomplantworker     | Worker               |
| Ergocomprocessbehavior | DNBBehavior          |
| ergocomprocessdefault  | TSA                  |
| ergocomplantdefault    | ALB Logical Part Bin |

### 6.3.1.1 Important Terms – Glossary

The following table defines common terms:

**Table 32: Reference and Instance Objects**

| ALB PlanType     | Meaning  |
|------------------|--|
| Reference object | <p>Reference objects, such as a container create the base from which objects (instance objects) can be directed for planning. Only a single reference object is available in the system library.</p> <p><b>Example:</b> <a href="#">Container blue</a> is the reference object in the system library. All objects used for planning by container blue are allocated to this reference object. These are all objects (instances) that have been referenced to the reference object.</p> <p>Any changes to the reference object affect all objects that are allocated to it. For instance, by changing the length of container blue all referenced objects belonging to container blue will reflect this change.</p> |
| Instance object  | <p>Instance objects are similar to processes, container or material areas etc in that they are objects that are used for planning. Instance objects are derived from reference objects.</p> <p>Instance objects are based on the reference objects in the system library. The properties of the reference objects are determined in the plantype set.</p> <p>Copies of instance objects are independent objects that are used for planning. Changes made to the properties of a reference object do not affect copied instance objects.</p>  |

For explanations of the relationships relating to terms in this table, *Please refer to the [Plantype Relationships](#).*

**Table 33: Plantypes for processes**

| ALB PlanType | Meaning   |
|--------------|---|
| Process      | Processes are created in the DPE and are the basis for finding stations and workplaces. Processes are part of a larger processplan/workplan. Additionally, processes provide the basis for the TSA process. |

| ALB PlanType            | Meaning   |
|-------------------------|---|
| Processplan/Workplan    | Processplans and workplans are logical nodes in the process structure that contain all of the processes used for work load balancing.   |
| Resources with behavior | Resources that are marked with the <i>Owns Behavior</i> option are used to display TSA-processes in V5. TSA-processes can only be assigned to resources that are marked in this way.<br><br>Resources with behavior are marked with the <i>Owns Behavior</i> option in the plantype set.  |
| TSA                     | TSA processes (Task Specification Activities) are copies of the processes for a processplan/workplan that are used for work load balancing. TSA are resource-dependent processes that are only used for a particular resource, such as a station or workplace. These TSA processes enable the displaying of processes for resources in V5.<br><br>The parent node of a TSA process is always a DNBBehavior process.<br><br>TSA processes are created automatically in ALB by work load balancing. |
| Behavior process        | DNBBehavior processes are used in V5 to display TSA processes.<br><br>The parent node for Behavior processes created by ALB is always the workplace.<br><br>DNBBehavior processes are created automatically in ALB by work load balancing.  |

Table 34: Worker – Logical Partsbin

| ALB PlanType     | Meaning   |
|------------------|---|
| Logical Partsbin | A logical parts bin is a storage medium and the link between the material area or shelf and the parts bin.<br><br>The parent node of Logical Partsbin is a shelf or material area.<br><br>Logical partbins are created automatically in ALB by work load balancing. |
| Worker           | The number of workers is determined during work load balancing.<br><br>The parent node for the Worker plantype is always the Station plantype.  |

## 6.3.2 Adjusting ALB for R18 (Update)

The so-called reference-instance-model is the basis for displaying the ALB results in V5. In order to display the ALB results for this model, the following ALB-configuration settings need to be made.

### 6.3.2.1 ALB-Configuration Update – Overview

Along with the description of the settings described below, read the chapter entitled Plantype Relationships in the chapter entitled [Data model for Automatic Line Balancing](#) in the appendix of this user manual.

#### Dummy Part Bin Plantype

The *Dummy part bin* plantypes in the ALB-configuration must be replaced by the *Logical part bin* plantypes.



**Functionality**

*Dummy part bins* are created as *Logical part bin* type objects. A separate plantype is not required for *Dummy part bins*. The dimensions of *Dummy part bins* (such as length and width) are fixed in the *length* and *width* attributes. *Dummy part bins* in a plantype set must be assigned to the structure of a *Material area* and a *shelf* with the assistance of the parent/child relation.

**Logical Part Bin Attribute**

The following *Logical part bin* attributes must be configured:

**Table 35: Table of New Attributes**

| Attribute    | Meaning   |
|--------------|---|
| Partname     | An attribute used to show the name of the part.   |
| part number  | An attribute used to show the part number.  |
| process list | The process list attribute enables display of all processes that are linked to this part and parts bin. |
| TSAlist      | The TSAlist attribute enables display of all TSA-processes that are linked to this part and parts bin.  |
| part Id      | An attribute for the identification number (Id) of a part.  |
| part bin Id  | An attribute for the identification number (Id) of a part bin.  |



To insure the consistency of data from the ALB results, it is suggested that these attributes be configured as *read only*.

Why? These attributes are automatically setup with the appropriate values in the process of creating the Automatic line balancing. These results should not be manually changeable.

### 6.3.2.2 Data Model Scheme – Relationships between Plantypes Data Structure

A Logical parts bin is created by ALB for each Part bin that is planned in for Material area or Shelf: Material area, Shelf, Logical parts bin and Part bin are all linked together by way of parent/child-relations. The Logical part bin is either the child of the Material area or the Shelf. The Part bin is always the child of a Logical part bin.

The relation “balancing\_resourceconsists of\_resource” is no longer created by ALB to link a Material area with a Part bin. This relation is replaced by the parent/child-relations between the Material area and Shelf and the Logical parts bin.

The data model shows the new relationships for the R18 release of ALB:

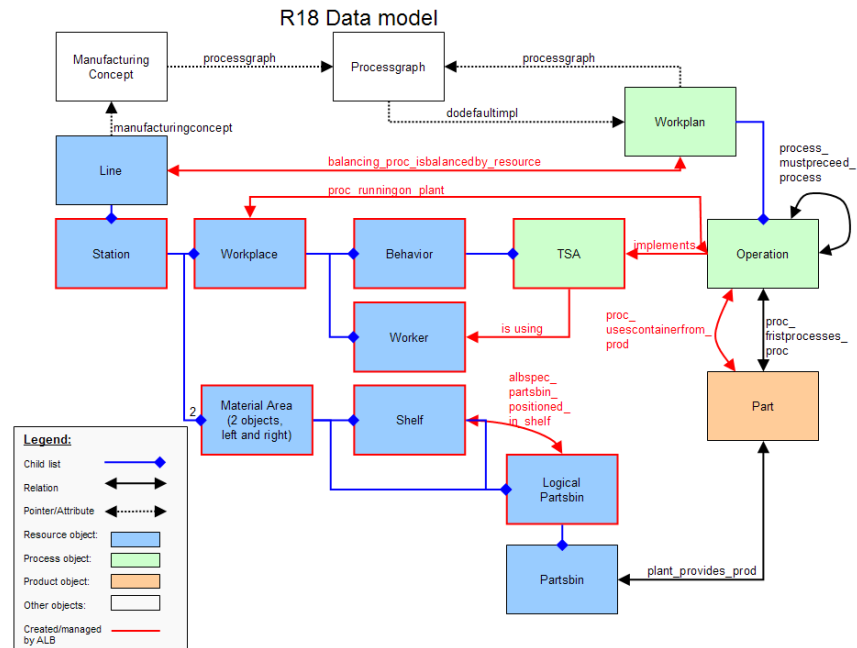


Figure 164: R18 Data Model – Relations

### 6.3.2.3 Updating an Older Version to R18

The following steps must be followed when updating an older version:

Carry out a data merge in the same manner as with previous updates

Configure the new plantypes in the plantype set

Check the ALB-configuration and configure the new plantypes and attributes

Once these steps have been taken, start ALB in the same manner as with preceding versions. Once the ALB-results have been saved, the new data structure is created, and in addition to this the ALB-results can now be displayed in the Reference-instance-model of V5.

## 6.3.3 Plantype Relationships

The relationships between plantypes should be determined in the plantype set.

A pre-configured plantype set is available with the update for the ALB-Demo-Project. The rest of the steps for the plantype set configuration must be carried out manually.

The names of plantypes used in the chapter are not set or in any way mandatory. In other words, feel free to choose other names for plantypes. It is important that the structures in the plantype set match those shown below.

### 6.3.3.1 Setting the Plantype Set Station

The *Workplace*, *Material area* and *Worker* are always linked with the station (parent node) by way of a parent/child-relation. These three plantypes are always part of the *Station* structure – which means they are so-called children of the *Station* parent node.

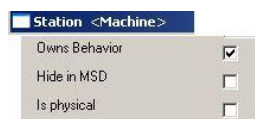


**Figure 165: Parent/child-relation for the Workplace**

### Workplace

In this example, (Machine) Station is the plantype for the workplace.

The *Owns Behavior* option must be activated in the properties dialog of the plantype for the workplace – in this example it is for the (Machine) station. Workplaces cannot be assigned to *TSA* processes unless this option has been activated.



**Figure 166: Activate the Owns Behavior Option**

### DNBBehavior – TSA

The *DNBBehavior* plantype enables the displaying of resource-dependant *TSA* processes in V5. For this reason, the *TSA* process plantype must always be assigned as the child of the parent node in the structure for the *DNBBehavior* plantype.



**Figure 167: DNBBehavior – TSA**

### Material Area – Logical Part Bin

A logical parts bin is a storage medium and the link between the *Material area* or *Shelf* and the *Parts bin*. This means that a *Logical parts bin* must have the *Material area* and *Shelf* plantypes as parent nodes.



**Figure 168: Logical Part Bin**

### Shelf – Logical Part Bin

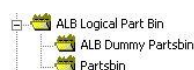
The *logical parts bin* must be part of the *Shelf* plantype structure.



**Figure 169: Logical Part Bin – Shelf**

### Parts bin – Logical Part Bin

The *Part bin* plantype must be part of the *Logical Part bin* plantype structure.



**Figure 170: Part Bin – Logical Part Bin**

## 6.4 Update

Pertains only to an update of versions prior to PE 5.10.

In case of an update from PE 5.10 to PE 5.11 or higher, you can skip over the following section.

### 6.4.1 Before Update

Before an update, a script must be executed that copies previously removed attribute values to another place. For this purpose you need to generate the following script and execute it directly in the Process View. The script brings about the following effect:



#### Caution

*The script only needs to be executed if you have already worked with Automatic Line Balancing in the previous version. If you have not yet used Automatic Line Balancing (most frequent case), for example, in PE 5.9, you should **not** execute the script before the update.*

- In the **proc\_firstprocesses\_prod** relation the values of the dbl\_defattribute\_1 attribute are copied to the nameshort attribute.
- In the **proc\_runningon\_plant** relation the values of the m\_strExternalId attribute are copied to the sortindex attribute.

```
Dim iNumberOfProcs
sub update_process(proc_id)
    REM = 1. if the object is a process => update the attributes
    tablename = data.GetAttributebyId(proc_id, "tablename")
    if tablename = "ergocompprocessdefault" then
        REM = 1.1 update relation "proc_firstprocesses_prod"
        REM > copy values from attr dbl_defattribute_1 into nameshort
        pfpp_id = data.GetFirstChild(proc_id, "proc_firstprocesses_prod")
        do while pfpp_id <> ""
            dblNumber = data.GetAttributebyId(pfpp_id, "dbl_defattribute_1")
            sNumber = CStr(dblNumber)
            data.SetAttributebyId pfpp_id, "nameshort", sNumber
            Rem SOLL: numberofitems STATT nameshort
            pfpp_id = data.GetNextChild(proc_id, "proc_firstprocesses_prod")
        loop
        REM = 1.2 update relation "proc_runningon_plant"
        REM > copy values from attribute m_strExternalId to sortindex
        prop_id = data.GetFirstChild(proc_id, "proc_runningon_plant")
        do while prop_id <> ""
            strSortindex = data.GetAttributebyId(prop_id, "m_strExternalId")
            if IsNumeric(strSortindex) then
                dSortindex = CInt(strSortindex)
                data.SetAttributebyId prop_id, "sortindex", dSortindex
            else
                data.SetAttributebyId prop_id, "sortindex", 0
            end if
            prop_id = data.GetNextChild(proc_id, "proc_runningon_plant")
        loop
        REM = 1.3 increase the number of updated processes
        iNumberOfProcs = iNumberOfProcs + 1
    end if
    REM = 2. if the process has childs, update them too
    proc_id1 = data.GetFirstChild(proc_id, "nodes")
    do while proc_id1 <> ""
```

```

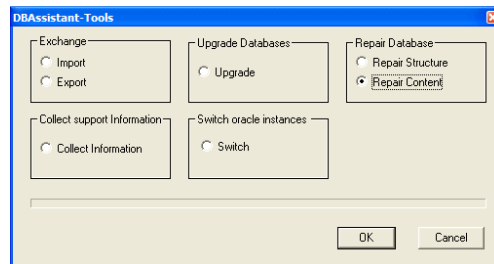
        update_process proc_id1
        proc_id1 = data.GetNextChild(proc_id, "nodes")
    loop
end sub
sub main(ap_id)
    iNumberOfProcs = 0
    proc_id = data.GetFirstChild(ap_id, "nodes")
    REM == 1. step through all processes
    do while proc_id <> ""
        update_process proc_id
        proc_id = data.GetNextChild(ap_id, "nodes")
    loop
    MsgBox(CStr(iNumberOfProcs) + " Prozesse upgedatet!")
End Sub

```

### 6.4.2 Update from R18 SP0/R18 SP1 to R18 SP2

To update the R18 SP0/R18 SP1 to R18 SP2, perform the following steps:

- 1) Start DBAssistant.



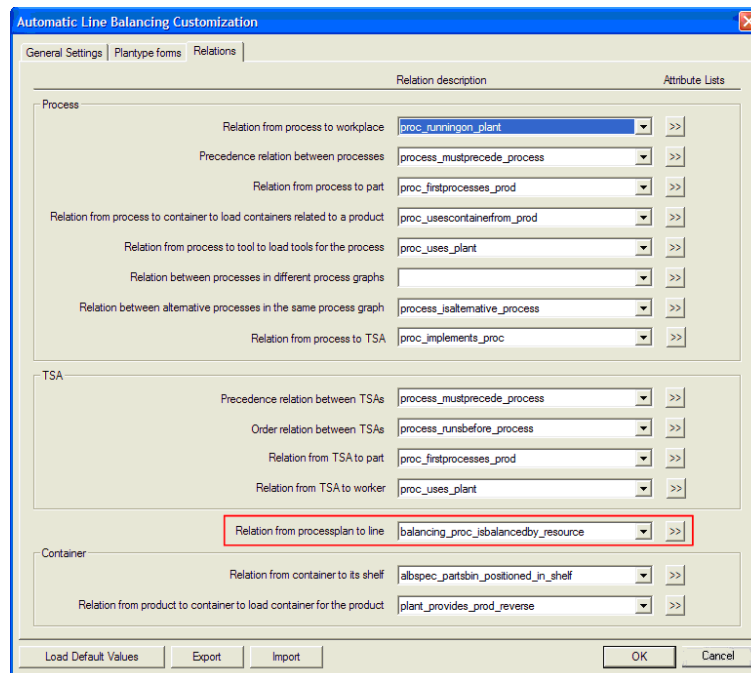
**Figure 171: DBAssistant Tools**

- 2) Select Option Repair Content under Repair Database.
- 3) Click **OK**.

### 6.4.3 Update from any Version to R18 SP2

To update the R18 SP2 from any version, perform the following steps:

- 1) Adopt ALB configuration.



**Figure 172: Automatic Line Balancing Customization**

- 2) Set Relation from processplan to line to “balancing\_proc\_isbalancedby\_resource”.
- 3) Click **OK**.

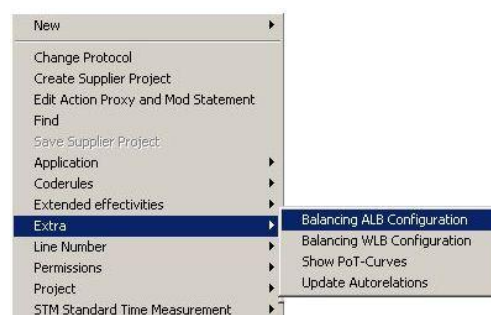
#### 6.4.4 After Update

You need to make the same adjustments in the planttype set as you would after a reinstallation in order to be able to work with the Automatic Line Balancing module after the update. For more detailed information *Please refer to the [New Installation](#)*.

In the appendix you will find the data model for the Automatic Line Balancing.

## 6.5 Opening the Configuration of Automatic Line Balancing

- 1) Open the balancing configuration using the context menu of a project.



**Figure 173: Open Balancing Configuration**

The dialog has three tabs necessary for the configuration of Automatic Line Balancing.

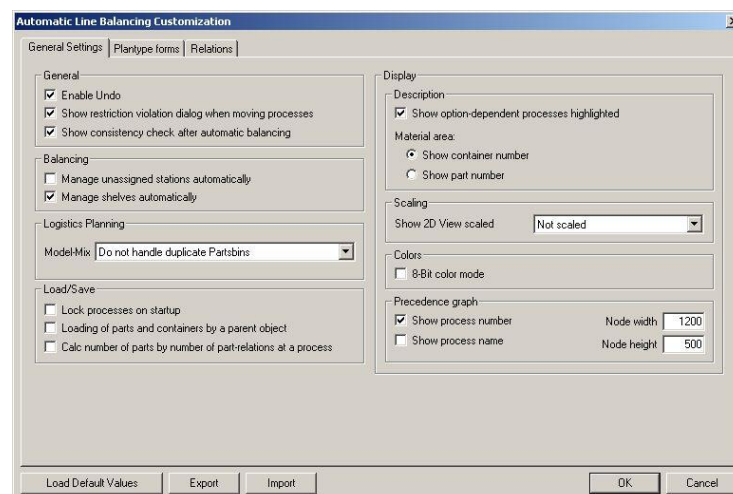
The configuration in Automatic Line Balancing is comparable to the configuration of Work Load Balancing.

The tab Plantype forms includes the configuration of the types and their attributes. The structure of balancing is set here, just as in Work Load Balancing.

- 2) The **types** must be set. All types of the Configuration Manager will be offered in a pop-up menu.
- 3) In the second step define the **attributes** required for identifying the plantypes. All attributes of the selected type will be offered in a pop-up menu.
- 4) The **value** of the previously selected attribute is entered.
- 5) In the list of the **attributes** are all attributes which are shown in the balancing form as a default. These attributes are required to enable the balancing process to function smoothly and therefore cannot be deleted. You can change these attributes.

## 6.6 The General Settings Tab

Set the general valid data for the display of balancing under the tab General Settings of the Automatic Line Balancing configuration. The user can change the basic settings in the balancing under *Options / Settings* and adapt them to the prerequisites of balancing. These changes apply only to this user. The Automatic Line Balancing configurations set in the basic settings apply to any other users who open the same balancing.



**Figure 174: Configuration of Automatic Line Balancing; tab 3 General**

- 1) Check *Undo*, if you want to enable the *Undo* function.
- 2) **Show restriction violation dialog when moving processes:**  
This function is activated in the standard configuration. If you [deactivate](#)



Show restriction violations on moving, the violation of restrictions will no longer be displayed in the performance view of the balancing when the processes are moved.

### 3) Show consistency check after automatic balancing:

This function is active in the standard configuration. If you [deactivate](#) Show consistency check after doing an automatic balancing, consistency checks will no longer be displayed after starting or re-starting the balancing.

### Balancing

- **Manage unassigned stations automatically:** Checkmark *Manage unassigned stations automatically* to enable the functions described below that option.
- **Manage shelves automatically:** If you enable Automatically manage shelves, empty shelves are removed in the balancing processes,
- When moving a process to another station to which part bins placed in a shelf are assigned, a new shelf is generated in this station as long as no appropriate shelf is available in this station.

### Logistics planning

- **Model Mix:** Determine the default value for the **Model Mix**. This setting is used to start every new work load balancing. The setting can be changed in the work load balancing.

### Load/Save

- **Lock processes on startup:** Enabling this field, the processes used for the balancing process can not be edited in the PPR-Navigator. Therefore enabling this field ensures, for example, that process times can be changed while working with the balancing process.
- **Loading of parts and containers by a parent object:** Whether you enable the *Loading of parts and containers by a parent object* function or not depend on your configuration. If the containers are linked to an organizational node of the product structure in the configuration (*Please refer to the Figure 175*), you must check the box. In the default configuration the load containers and containers with technical nodes are linked and you do not have to check the box.

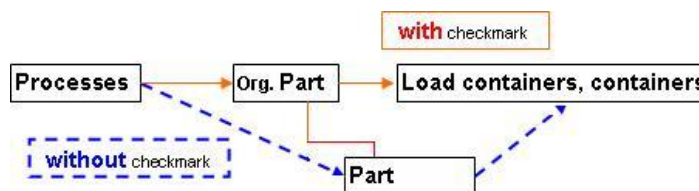


Figure 175: Schematic Diagram

### Calc. number of parts by numbers of parts-relations at a process:

- If this option is activated, all identical parts must be linked individually to a process - Instances model.
- If this option is deactivated, you need to link a part that is used several times only once to a process – Reference model.

### Description

☒ Manage shelves automatically

- Checkmark *Show option-dependent processes highlighted* to visually display processes to which special equipment parts are assigned. Processes are assigned special equipment parts according to a code rule.

### Material Surface

- If you activate *Show parts bin number*, the corresponding **parts bin number** for a loading unit is shown.
- If you activate *Show part number*, the corresponding **part number** of the part for the loading unit that is assigned to this loading unit is shown.

### Scaling

2D View scaled display: By default the 2D View is not displayed scaled. With the help of the two options (*Only Workplace Width, All*) you can display the 2D View in other scale modes. This might be helpful if the width of the workplace is too small so that the process cannot be displayed within the workplace.

### Color

Activate 8-bit color mode: Activate this mode only if you can display the graphics on the computer in 8-bit color mode.

### Precedence Graph

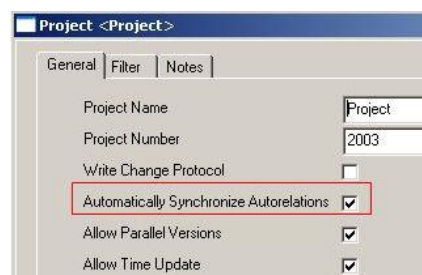
This area specifies data for the display of processes in the Priority Graph view.

- Checkmark *Show process number* and *Show process name* to display the processes with the corresponding names and numbers in the Priority Graph view.
- The Node width and Node height fields are both used to specify the size for displaying the process in the Priority Graph view. Node width and node height are specified in pixels.



### Caution

*If you use the ALB work load balancing, the automatic synchronization of the autorelations must be switched off.*



**Figure 176: Automatic Synchronization of the Autorelations**

## 6.7 Plantype Forms Tab

| Component type   | Attribute              | Value | Attribute Lists      |
|------------------|------------------------|-------|----------------------|
| Line             | ergocomplantdefault    | name  | Line                 |
| Station          | ergocomplantstation    | name  | Machine              |
| Workplace        | ergocomplantstation    | name  | Machine              |
| Material Area    | ergocomplantarea       | name  | Material Area        |
| Process          | ergocomprocessdefault  | name  | Operation            |
| TSA              | ergocomprocessdefault  | name  | TSA                  |
| Behavior         | ergocomprocessbehavior | name  | DNBBehavior          |
| Worker           | ergocomplantworker     | name  | Worker               |
| Part             | ergocomproductdefault  | name  | Part                 |
| Container        | ergocomplantdefault    | name  | Partbin              |
| Logical Part Bin | ergocomplantdefault    | name  | ALB Logical Part Bin |
| Tool             | ergocomplantdefault    | name  | Resource             |
| Shell            | ergocomplantdefault    | name  | Shell                |
| Shell Library    | ergocomplantdefault    | name  | Shell Library        |
| Processplan      | ergocomprocessdefault  | name  | Workplan             |

**Figure 177: Configuration of Automatic Line Balancing; tab Plantype Forms**

The data model for Automatic Line Balancing can be found in the appendix. The data model provides you with an overview and can be seen as a template for the standard configuration.

You can find all of the types of the data model in the dialogs of the configuration (tab 2 and 3).

### Button Attribute

The attribute list is displayed in its entirety and without error messages if the type, the attribute and its corresponding value have been completely and correctly filled out.

Only attributes for the set type are displayed in the attribute list.

- 1) The attribute list is be opened by clicking on the *Attributes* button.

| Attribute Name    | data type | Meaning                      |
|-------------------|-----------|------------------------------|
| name              | string    | Arbeitsplatzname             |
| nameshort         | string    | Arbeitsplatznummer           |
| carbodyposition   | string    | Arbeitsplatzbereich          |
| lockstatus        | integer   | Sperstatus                   |
| multicyclecount   | integer   | Mehrtaktanzahl               |
| multicyclelimit   | integer   | Mehrtaktlimit                |
| containerareaside | string    | Materialflächenseite         |
| workarea_color    | integer   | Farbe des Arbeitsplatzber... |
| workarea_name     | string    | Arbeitsplatzbereichsname     |
| objectuuid        | string    |                              |
| attribute_17      | string    |                              |

**Figure 178: Open Balancing Configuration**

- In the list you can see all attributes of the selected plantypes which are used by ALB as a default. These attributes are required to enable the balancing process to function smoothly, but they can be changed.

**Displayed**

- The name of the attribute (indicates the usage, important if changes are made to the attribute; for example, if the displayed usage of the attribute was or is to be configured under another attribute.)

The used data type

- And the meaning (example: "Setuptime" for name, and setup time for meaning).
- The displayed attributes can be edited.

2) After selecting the attribute click on the *Edit button*.

- The *Configure ALB Attribute* dialog will open. Editing is necessary when, for example, the setup time is not assigned to the attribute *setuptime*, but rather to a free attribute such as *dbl\_attribute\_1*. All available attributes of this plantype can be found in the combobox of the display field *attribute*.

If, for example, the name of the station (Figure 179) is not read from the attribute *name*, look for the attribute in the combo box which contains the name or the meaning of the station.



**Figure 179: Dialog Configure ALB Attribute**

- 3) After you have made all the settings, you must close the dialog again in order for the settings to take effect.
- 4) Confirm the entries with **OK**.

**Deleting Attributes**

To delete an attribute, select it and click the "**Delete**" button.

### 6.7.1 Inserting a New Attribute into the ALB Configuration

All attributes that by default are shown on the balancing form, are found on this list of attributes. Most of these attributes are required to enable the balancing process to function smoothly and therefore cannot be deleted. The following error message will be displayed:



**Figure 180: Deleting a Predefined Attribute**

Further attributes can be added to this list, or existing attributes can be changed and removed from the table.

To show fixed space in the layout, the *ALB Dummy Parts bin* attribute must be configured along with the configuration for the Dummy Container plantype. To insert a new attribute follow these directions:

### New Attribute

- 1) Use the "New" button to start the **insert attribute** editor. This editor can be used to define the properties of an attribute.

Figure 181: New Attribute

### OK button

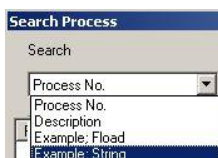
Using this button the attribute is permanently saved in the database.

### Cancel button

Using this button you can exit the dialog without saving the changes.

Table 36: New Attributes Dialog Description

| Properties     | Type            | Comment  |
|----------------|-----------------|--|
| Name           | Selection field | Select a name for the new attribute  |
| Meaning        | Display field   | This field cannot be edited. The default value for new attributes is set on user defined attribute   |
| Data type      | Selection field | Here you can see the data type of the selected attribute. The data types are described under Data Types in this manual..   |
| Searchable     | Checkbox        | To determine whether these attributes are (activate) or, are not (deactivate) to be used as search criterion, check this checkbox  |
| Show in dialog | Checkbox        | To determine whether these attributes are (activate) or, are not (deactivate) to be displayed, check this checkbox. The attribute properties will first be visible after this checkbox has been activated. |
| Name           | Input field     | Enter one of the names to be displayed in the balancing list   |



| Properties     | Type            | Comment   |
|----------------|-----------------|---|
| Show on page   | Rotary field    | Select the page on which the attribute is to be displayed   |
| Column number  | Rotary field    | Select the column number for the balancing list, in which the attribute is to be displayed. If multiple attributes use the same number, the values of the attributes should be separated by a hyphen "-" in the balancing list        |
| Alignment      | Selection field | You can specify the text position here: left, right or center   |
| Output factor  | Input field     | This field is dependent on the data type and is only displayed when selecting a double data type. An entry in this field is multiplied by the attribute value as in the case of time conversions from minutes to seconds, for example |
| Decimal places | Input field     | This field is also dependent on the data type. This is used to determine the decimal places for the attribute   |

### 6.7.2 Attribute identifieralb (Identification ALB)



#### Note

Just as in the configuration of WLB, there is a new attribute for ALB (*identifieralb*) which is used to balance several plantypes of the same kind (for example, with processes: test process, process non-value added, process value added)

The items station and workplace are exceptions. The attributes and values must be set using unique defined attributes (they can not be used twice) for these plantypes. These plantypes are generated by Automatic Line Balancing.

Plantypes of the same kind which are to be balanced simultaneously must be identifiable for balancing.

The new attribute (**identifieralb**) can be assigned to plantypes of the same kind several times. The allocation of this attribute calls the plantype for the balancing process.

#### Example

##### Example:

What should you do if you want to balance several process types?

In the first step you have to show the attribute **identifieralb** in one or in all plantypes.

- 1) Select the plantype in the Configuration Manager or the base class of all plantypes (internal name = XDOErgoplantyp; typename = ergoplantyp) and show the attribute *identifieralb*.
- 2) Enter the same value (e.g. "alb") in the attribute *identifieralb* in all plantypes to be used for balancing processes.

- 3) In the configuration of Automatic Line Balancing you must now select the attribute *identiferialb* and enter the previously entered value for the process configuration.
  - 4) Close the dialog or switch tabs. The items are then stored and enabled.
- All plantypes with this value (alb) in the attribute identiferialb are used for balancing

**Figure 182: Attribute Identiferialb**

### 6.7.3 Relations Tab

The procedure for configuring the relations is analogous to the plantype configuration. The proper relations must be selected in accordance with the selected plantypes.

If, for example, the relation *proc\_alike\_proc* is not displayed in the selection list, the relation must first be enabled.

**Figure 183: Configuration of Automatic Line Balancing; Tab Relations**

#### Load Default Values

Use this button to load the standard settings of the Automatic Line Balancing configuration (all three tabs are affected by this). The configuration shown in [Figure 184](#) serves as the basis, and the corresponding values are inserted in the configuration.

The data model shows the relationships between the different plan types exemplarily.



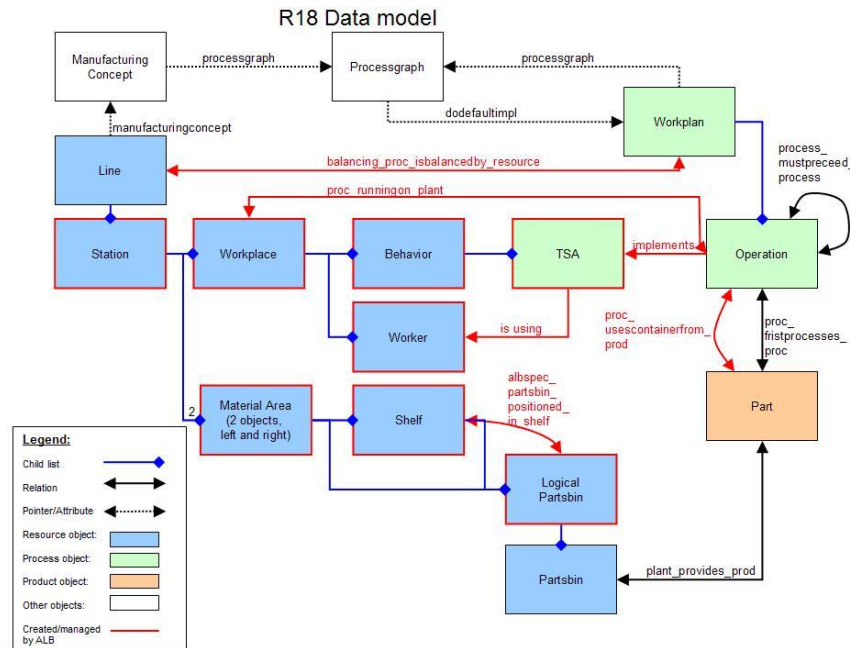


Figure 184: Default values for Automatic Line Balancing

### Import

You can import an Automatic Line Balancing configuration by using the Import button. Only files of type *ALB Configurations Files* with the **.alb** extension can be imported. The existing configuration is overwritten.

### Export

The configurations are exported using the Export button. A dialog for selecting the saving location is opened. The exported file is saved with the **.alb** extension.

### Showing Graphic

The *Show Graphic* function can only be activated at the start of a project with the assistance of a script action. Proceed as follows:

- 1) Create a new script in either the project library or system library: **Scripts < New < Script**.
- 2) Name the script and insert the following script into the source text field.  
Function `SA_OpenProject(id)`

'Necessary for the Feature 'Show Graphic' in ALB.

'Shows the Graphic of the object, stored in the Registry.

'Then deleted the entry, so the object is shown only once.

'DELMIA GmbH 17.01.2006 - MFM

'=====

Dim sOID

```
sOID = Data.ReadSessionData("E5Balancing", "OID_for_ShowGraphic",
"HKCU" )
```

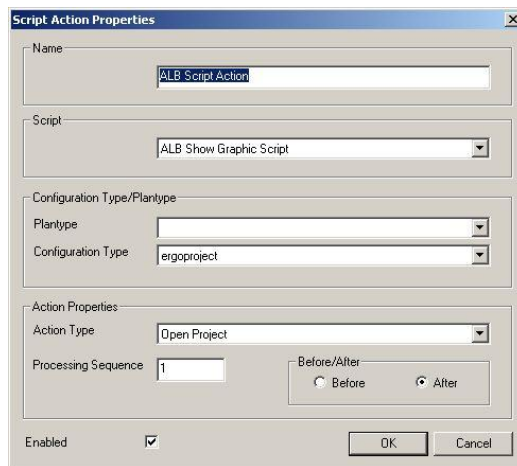
```
Call Data.WriteSessionData("E5Balancing", "OID_for_ShowGraphic",
"HKCU", "")
```

If Len(sOID) > 0 Then

```
Call Graphic.ShowGraphic( sOID , "" )  
End if
```

End Function

- 3) Save the script.
- 4) Create a new script action in either the project library or system library:  
**script action < new < script action** and enter the following values:



The image shows a 'Script Action Properties' dialog box with the following fields and settings:

- Name:** ALB Script Action
- Script:** ALB Show Graphic Script
- Configuration Type/Planttype:**
  - Planttype:** (empty dropdown)
  - Configuration Type:** ergoproject
- Action Properties:**
  - Action Type:** Open Project
  - Processing Sequence:** 1
  - Before/After:** After (selected)
- Enabled:** ☒
- Buttons:** OK, Cancel

**Figure 185: Script Action Properties**

- The next time the project is started, the Show Graphic in ALB function will be available.

## 7. Settings

### 7.1 Recommended Settings for the PPR Server

The following settings allow influencing the performance of the PPR server:

HKEY\_LOCAL\_MACHINE\SOFTWARE\DELMIA\IPDSERVER\Ergoplan\configuration  
ReadAhead = 100

HKEY\_LOCAL\_MACHINE\SOFTWARE\DELMIA\IPDSERVER\Ergoplan\Database\  
ReadAhead = 100

HKEY\_LOCAL\_MACHINE\SOFTWARE\DELMIA\EnableSBHeap = 1

HKEY\_LOCAL\_MACHINE\SOFTWARE\DELMIA\SBHeapThreshold = 1016

### 7.2 Recommended Settings for the PPR Client

The performance can be influenced negatively by a flood of settings which the PPR Client writes into the registry in course of time. You may gain performance by deleting the key HKEY\_CURRENT\_USER\Software\DELMIA. Note, however, that all user-specific settings will be lost when you do so.

The following settings allow influencing the performance of the PPR client:

HKEY\_LOCAL\_MACHINE\SOFTWARE\DELMIA\ERGOPlan\useclientcache = 1

### 7.3 Logging

Process Engineer provides logging functionalities to record system errors and warnings, as well as server- and user-specific activities.

#### 7.3.1 Standard Error Logging

Standard error logging in Process Engineer differentiates between critical and non-critical errors by categorizing them into the following error types:

- **Information and warnings:** Non-critical information that is helpful when debugging client connection problems or network-related issues.
- **Access Errors:** Errors that occur when a client requests faulty information, such as attempting to access an object that has been deleted or requesting unknown attributes for a given object type.
- **Server Errors:** Errors that occur in context of server services but allow the continued use of the server process, such as database errors or data consistency errors.
- **Critical Errors:** Errors that lead can lead from restricted usage of the server process to no usage at all, depending on the severity of the error. These might include critical database errors, memory issues, connection issues, etc.

Each entry in the server error log file is in the following format:

Severity: <category> <classification>

where <category> represents one of the following error types:

Information/Warning

Access Error

Server Error

Critical Error

and <classification> represents more detailed information about the specific error.



### Note

*This functionality is effective for logs generated from IPDServer, Lock Manager and Update Manager processes.*

## 7.3.1.1 Customization

### Enable/Disable Standard Error Logging

The standard error log file can be turned on or off via the “SwitchOffStandardErrorLog” registry key:

[HKEY\_LOCAL\_MACHINE\SOFTWARE\DELMIA\IPDSERVER\SwitchOffStandardErrorLog]

A value of “0” enables standard error logging, while a value of “1” disables it.

### Error Filtering

Errors can be filtered by type to control the contents of the standard error log file. This is determined by the value of the “ErrorLogSeverity” registry key:

[HKEY\_LOCAL\_MACHINE\SOFTWARE\DELMIA\IPDSERVER>ErrorLogSeverity]

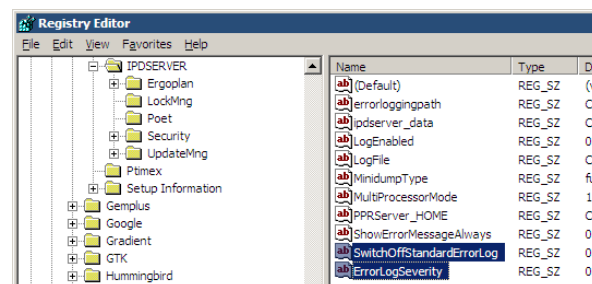
This key can be set to the following values:

0: Everything is logged to the standard error file

10: Everything is logged except Information and Warnings

20: Only Server Errors and Critical Errors are logged

30: Only Critical Errors that force the server into the emergency mode are logged. (In this case, the standard error log file is identical to the critical error log file.)



**Figure 186: “ErrorLogSeverity” Registry Key**



### Note

*These keys do not exist in the default installation and must be created in the proper location in the registry.*

Disabling standard error logging as described above does not affect critical error logging. Critical errors are always logged to a separate file that contains only critical errors.

## 7.3.2 Server Activity Logging

Server activity logging retains information regarding the start and termination of IPDServer processes, as well as changes in the various states of the process (emergency state, lock state, and availability state).

The resulting XML log file contains this information in the following format:

- **Start of an IPDServer process:** Activity ("IPDServer started") , <date>, <time>, <server machine>, <server process id>, <OS user name>
- **Termination of an IPDServer process:** Activity ("IPDServer terminated") , <date>, <time>, <server machine>, <server process id>, <kind of termination> (regular, irregular)
- **Change of the emergency state of an IPDServer process:** Activity ("IPDServer emergency"), <date>, <time>, <server machine>, <server process id>, <kind of emergency> , <List Affected clients <client machine> and <client process ids> >
- **Change of the lock state of an IPDServer process:** Activity ("IPDServer lock state"), <date>, <time>, <server machine>, <server process id>, <lock cause>, <List Affected clients <client machine> and <client process ids> >
- **Pool assignment of an IPDServer process:** Activity ("IPDServer pool assignment"), <date>, <time>, <server machine>, <server process id>, <Kind of pool assigned to>, <List Affected clients <client machine> and <client process ids> >
- **Change of the Availability state of an IPDServer process:** Activity ("IPDServer availability state"), <date>, <time>, <server machine>, <server process id>, <Availability state>, <List Affected clients <client machine> and <client process ids> >

The tags shown above are defined as follows:

- **<date>:** The date and time when the activity is logged
- **<time>:** The time when the activity is logged.
- **<server machine>:** The computer name on which the process started, terminated, or went in emergency mode.
- **<server process id>:** The process ID of the process that has been started, terminated, or went in emergency mode
- **<OS user name>:** The user account that the started process belongs to. Needed for evaluation of DCOM configuration.

- **<kind of termination>**: Regular or irregular (unexpected process termination).
- **<kind of emergency>**: The following emergency causes can be logged:
  - *"low memory"*: memory too low for the IPDServer process
  - *"critical c exception"*: critical exception of the Windows runtime lib
  - *"system critical exception"*: critical exception of the DPE server system
  - *"critical database exception"*: critical database exception
  - *"critical database abort exception"*: critical database exception forcing the failover abort procedure
  - *"RPC error update manager"*: DCOM RPC connection to the Master update manager process broken
  - *"RPC error lock manager"*: DCOM RPC connection to the Master lock manager process broken
  - *"IO error security logger"*: IO operation of the security logger process failed
  - *"RPC error security logger"*: DCOM RPC connection to the security logger process broken
  - *"system abort exception"*: critical exception of the DPE server system forcing the failover abort procedure
  - *"<-generic emergency->"*: numeric emergency failure code
- <lock cause>** The following causes can be logged:
  - *"locked by emergency handler, emergency state: <kind of emergency>"*: The IPDServer process was locked by the emergency handler – emergency cause see <kind of emergency> above
  - *"locked by running termination"*: The server termination is running and has locked the IPDServer access
  - *"locked by application request"*: The server is locked by an application request (API call)
- **<Kind of pool assigned to>**: The following pool assignments can be logged:
  - *"assigned to <description of pool>"*: The client is assigned to an IPDServer process of the Sub-ServerPool described by <description of pool>; the pool description may contain the connect option (exclusive, shared) and/or the pool label (defined by the application or the PoolingServer configuration) and/or the persistent pool flag and/or the connection mode (with or without data actualization).
  - *"assigned to pool of uninitialized invalid IPDServer processes - process is termination candidate"*: The IPDServer is moved to the pool of uninitialized server processes to be terminated by the IPDServer GarbageCollector; cause: the IPDServer process could not be initialized.

- *"Periodic IPDServer check: inactive process assigned to pool of unused IPDServer processes (client count: 0) - process is termination candidate"*  
The IPDServer has become inactive by an DCOM access timeout, access error or internal service shutdown.
  - *"Periodic IPDServer check: inactive process assigned to pool of unused IPDServer processes (availability state: 'Failover task completed', client count: <number of clients>) - process is termination candidate"*: The IPDServer is moved to the pool of unused server processes to be terminated by the IPDServer GarbageCollector; cause: after the server was set into the emergency mode the failover procedure has closed the server services; <number of clients> is the number of current clients still connected by DCOM and registered in the ServerPool.
  - *"IPDServer GC: active process assigned to pool of unused IPDServer processes"*: The IPDServer is moved to the pool of unused server processes to be terminated by the IPDServer GarbageCollector.
- <Availability state>**: The following availability states of the IPDServer process can be logged:
- *"Periodic IPDServer check: <state> - process is set into the inactive state and cannot be connected anymore."*: The availability state is logged if any kind of availability problems occur; the server availability state information is detailed by <state> with:
  - *"IPDServer connection is lost by RPC"*: The DCOM RPC connection to the IPDServer process is broken
  - *"accessing the ipdserver process has been timed out"*: Invoking the IPDServer process via DCOM has been timed out
  - *"IPDServer process termination is running"*: The IPDServer process is already terminating
  - *"IPDServer services are terminating"*: The IPDServer application services are already terminating
  - *"IPDServer emergency failover procedure has been completed"*: The IPDServer is removed from pool after internal emergency failover service shutdown
  - *"IPDServer COM connection is not initialized"*: The IPDServer cannot be accessed since the DCOM connection to that process was not initialized
  - *"unknown error occurred trying to access the ipdserver process"*: The IPDServer process could not be accessed caused by an unknown error
- <List Affected clients <client machine> and <client process ids>**: List of client processes that were affected by the server emergency state.

### 7.3.3 User Activity Logging

User activity logging retains information regarding user login and logout, opening and loading projects, and PackNGo activities. User activity logging is a prerequisite for "Delete non-leaf versions" (*Please refer to the [PPR Navigator Manual](#)*).



The resulting XML log file contains this information in the following format for the different activities:

- **User login:** Activity ("User login"), <user id>, <E5 user name>, <client type>, <client machine>, <client process id>, <OS user name>, <server machine>, <server process id>, <number of user logged on server process>, <number of **user** logged on server machine>, <execution time>
- **User logout:** Activity ("User logout"), <user id>, <E5 user name>, <client type>, <client **machine**>, <client process id>, <OS user name>, <server machine><server process id>, <number of user logged on server process>, <number of user logged on server machine>, <execution time>
- **Open Project:** Activity ("Open Project"), <user id>, <E5 user name>, <client type>, <client machine>, <client process id>, <OS user name>, <retrieved project name>, <execution time>
- **Load Project:** Activity ("LoadProject"), <date>, <time>, <user id>, <E5 user name>, <client type>, <client machine>, <client process id>, <OS user name>, <project **name**>, <project short name>, <project object id>, <node name>, <node short name>, <node object id>
- **Navig Load:** Activity ("NavigLoad"), <date>, <time>, <user id>, <E5 user name>, <**client** type>, <client machine>, <client process id>, <OS user name>, <project name>, <project short name>, <project object id>
- **Pack&Go:** Activity ("Pack&Go"), <user id>, <E5 user name>, <client type>, <client machine>, <client process id>, <OS user name>, <retrieved project name>, <retrieved node>, <execution time>

The tags shown above are defined as follows:

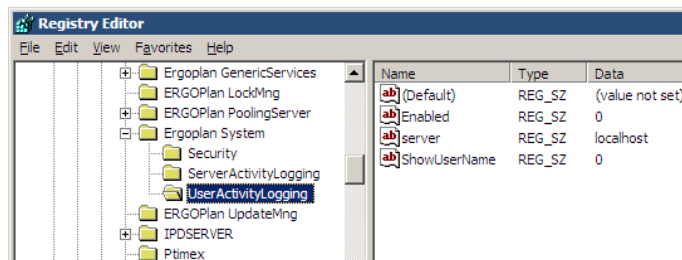
- **<date>:** The time when the activity is logged
- **<time>:** The time **when** the activity is logged
- **<user id>:** The unique **numeric** identifier for the E5 user account
- **<E5 user name>:** The **E5** user account of the connected user, as a string
- **<OS user name>:** The operating system user account of the connected user, as a string.
- **<client type>:** E5/V5, release, SP, hotfix
- **<client machine>:** The **machine** on which the connecting client application runs.
- **<client process ID>:** The **process** ID of client application
- **<server machine>:** The machine name of the server connected to by the client.
- **<server process ID>:** The **process** ID of the server process connected to by the client
- **<project name>:** The name of **the** project opened by the client or accessed for PackNGo
- **<project short name>:** Short name of the project opened by the client or accessed for PackNGo

- **<project object ID>**: Object id of project that has been opened by client or which is accessed for PackNGo.
- **<node name>**: The name of the process that is accessed for PackNGo or expanded by LoadProject
- **<node short name>**: The short name of process that is either accessed for PackNGo or expanded by LoadProject.
- **<node object ID>**: The **object** id of the process accessed for PackNGo or expanded by LoadProject
- **<number of user logged on server process>**: The number of users connected to the server process at the time of login or logout.
- **<number of user logged on server machine>**: The number of users connected to the server machine at the time of login or **logout**.

Data protection laws in some countries forbid the logging of user names (<E5 user name>, <OS user name>), therefore the logging of these is disabled by default. It can be enabled by the registry settings described below.

Three central settings are available on the master server under the following registry key:

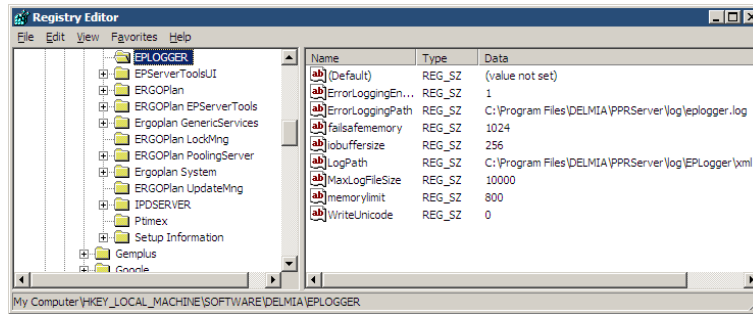
[HKEY\_LOCAL\_MACHINE\SOFTWARE\DELMIA\Ergoplan System\UserActivityLogging]



**Figure 187: Key for UserActivity Logging in the Registry**

- The “Enabled” setting is disabled by default (“0”). To enable User Activity Logging, set this value to “1”.
- The “server” value is set to “localhost” by default, where each server machine connects locally to the EPLlogger process and a separate log file is generated. A specific machine can be specified here so that logging is done in a single log file on az central machine.
- “ShowUserName” is set to “0” by default, which means that <E5 user name> and <OS user name> are not included in the generated log files. To include these in the log files, set this value to “1”.

An additional registry setting specifies the location for all log files written by the EPLlogger. This registry value is named “LogPath” and is stored under the following key: [HKEY\_LOCAL\_MACHINE\SOFTWARE\DELMIA\EPLLOGGER]



**Figure 188: Registry Key for Location of the Log Files**

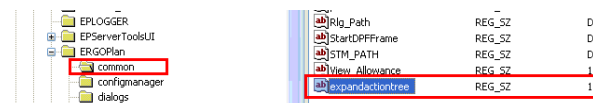
When changing the location of the generated log file, either one registry or several registries of several server machines must be changed, depending on whether the logging is done centrally or individually.

If User Activities are to be logged centrally, the DCOM configuration for the EPLogger process on that server machine must be configured so that it can be accessed remotely by all server processes.



For more information *Please refer to the* [DCOM Settings Manual](#).

## 7.4 Expand Action Tree



**Figure 189: Key for Expand Action Tree in the Registry**

The registry key “expandactiontree” is useful for the new Open Project dialog. If you open a MCM project, the Select Modification Statement dialog opens. With the default setting “expandactiontree”=0 the action tree will not expand. If the key value is set to “1”, the action tree will expand up to the last used Modification statement.

## 7.5 Automatic Termination for Lock and Update Managers

To help avoid excessive memory fragmentation for long processes, the Lock Manager and Update Manager can be terminated automatically when there are no IPD server connections. Once terminated, these processes are then restarted when the next IPD server process is started. This allows the memory for the Lock Manager and Update manager to be reinitialized without requiring any manual administrative intervention.

When Automatic Termination is enabled, it is triggered when there are no IPD server connections and either of the following conditions is true:

- Five Check Client Alive periods occur without any IPD Server connections. By default, the Check Alive period is two minutes, therefore ten minutes will have elapsed before terminating.

The default Check Client Alive period can be changed via the registry. For

the Update Manager, this is determined by the [HKEY\_LOCAL\_MACHINE\SOFTWARE\DELMIA\ERGOPlan UpdateMng\ErrorIterationPeriodCallbacks(sec)] registry value. For the Lock Manager, it is set via the [HKEY\_LOCAL\_MACHINE\SOFTWARE\DELMIA\ERGOPlan LockMng\ErrorIterationPeriodCallbacks(sec)] registry value.

- When a Check Client Alive period occurs and either of the defined memory usage limits is exceeded:
  - Virtual memory used of 500MB
  - Actual memory used of 300MB

### 7.5.1 Enabling Auto Termination

By default, Auto Termination is not enabled. It can be enabled via the following registry keys:

For the Update Manager, the registry value [HKEY\_LOCAL\_MACHINE\SOFTWARE\DELMIA\ERGOPlan UpdateMng\AutoTermination] must be "1".

For the Lock Manager, the registry value [HKEY\_LOCAL\_MACHINE\SOFTWARE\DELMIA\ERGOPlan LockMng\AutoTermination] must be "1".

## 7.6 Global Emergency Mode

Global Emergency Mode is raised when a master process runs into a critical error state that affects functionality used by other processes. Critical errors are then resolved automatically, such that Process Engineer does not require a manual restart.

## 8. Applying Time Analyses from an Oracle Database

### 8.1 General Information



Before applying time analyses from a DB-Ergotime database, all prerequisites described in the *PPR Navigator Manual* must be met:

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

The existence of a plantype set that provides the plantypes for the individual analysis methods of processes in the PPR-Navigator.

The attributes for the various time types and allowances must be configured.

- The extension of the plantype set is described for individual analysis methods in the section [Extending the Plantype Set](#).
- The required attributes are described in section [Attributes of the Process Analysis](#).
- The conversion of analysis data of the DB Ergotime database to the DB database is described in section [Importing Time analyses from an Oracle Database](#).

If you want to begin immediately with the creation and editing of process analyses, import the plantype set:

***plantypes\_ergoplan\_stm\_default.ini***. This plantype set is setup so that all plantypes and attributes that are required for editing process analyses are available and configured.

Create a new project with this plantype set. You can create process analyses in the new project.

The following sections will, however, involve existing projects with previously created time analyses. The plantype sets must be adjusted and the time analyses created in the projects must be converted and re-assigned to the respective process.

#### 8.1.1 Initial Situation

Two databases were used in version DPE 5.14:

- DB–Database
- DB-Ergotime

Processes were saved in the DB database, the time analyses in the DB-Ergotime database. To be able to process time analyses in the DELMIA Process Engineer® FinderControl was required.

##### 8.1.1.1 Objective

As of version DPE 5.15 there is only one database. For processing of time analyses FinderControl is no longer needed.

All time analyses should be available on the DB database after the upgrade from version DPE 5.14 to DPE 5.15.

The allocation of the time analysis data to the processes must also be retained.

## 8.1.2 Procedure

In the following, the steps necessary to transfer time analyses from the DB-Ergotime database to the DB database will be shown.

Execute the upgrade from version DPE 5.14 to DPE 5.15.

- 1) Adjust your plantype set.  
Extend the plantype set with the analysis methods that you want to use. You can use the plantype set ***plantypes\_ergoplan\_stm\_update.ini***. With it you can add all analysis methods to your existing plantype set as a new plantype. You can read more about this in the section [Extending the Plantype Set](#).
- 2) In order to edit the settings for the processes analyses, you must open the context menu in the plantype set of the system library and activate the entry *Settings for the STM – plantypes*.  
Check whether the settings can be called up on the plantype set of the system library. If this is not the case, the context menu call-up must be configured:  
Configuration Manager > Types > ergoplantypeset (plantype set) > Editors > EPSTMSettings. EPSTMSettingsManager:: ergoexplorer-library:: STM Plantype Settings > Single selection in browser to Yes.  
You can not create process analyses if you have not opened and **saved** the setting for editing at least once.
- 3) Check whether the datacards and datacard entries are available in the system library. If this is not the case, the datacards must be configured:  
Configuration Manager > Types > archivroot (Library) > parent-child relation > datacardfolder:: datacard > Parent-child information > tree view to Yes.  
Then the datacards and the datacard entries must be imported.



For more information on how datacards and datacard entries are imported, Please refer to the [PPR Navigator Manual](#).

- 4) Adjusting the attributes. In order to be able to make use of the functional diversity of the analysis methods, a few attributes need to be adjusted. You can read more about this in the section [Attributes of the Process Analysis](#).
- 5) You can transfer all of the analysis data from the DB-Ergotime database to the DB database by using the EPDBUpdater. You can read more about this in the section [Importing Time analyses from an Oracle Database](#).
- 6) The context menu entries for opening of FinderControl. If the context menu entries for opening the FinderControl are still active, you must switch them off or remove them from the configuration.

- 7) Execute STM Time Update.... You accomplish the Time Update through the menu *Tools/STM-Time Update*. You can accomplish the Time Update only if you activate the field *Allow Time Update* of the properties dialog of the dialog.

## 8.2 Extending the Plantype Set

Each analyses method is represented by a separate plantype:

**Table 37: Plantype Sets**

| Analysis Method            | Plantype                 |
|----------------------------|--------------------------|
| STM – UAS                  | ergocomptimeanalysisUAS  |
| STM – MTM-1                | ergocomptimeanalysisMTM1 |
| STM - MTM-2                | ergocomptimeanalysisMTM2 |
| STM – MEK                  | ergocomptimeanalysisMEK  |
| STM – STD                  | ergocomptimeanalysisSTD  |
| STM – SAM                  | ergocomptimeanalysisSAM  |
| STM - BasicMOST            | ergocomptimeanalysisMOST |
| STM - General Time Element | ergocomptimeanalysisAZB  |
| STM – Formula              | ergocomptimeanalysisFOR  |

Proceed as follows in order to extend a plantype set by the aforementioned analysis methods:

n the system library and select the plantype set to be extended.

- 1) Select *Import plantype set* in the context menu.
- 2) Select the plantype set ***plantypes\_ergoplan\_stm\_update.ini*** in the file selector that opens.
- 3) Activate *Keep customization* and *Create new plantypes* in the dialog that opens. This makes it possible to import new plantypes and keep their existing configuration as it is.

For more information on importing plantype sets, *Please refer to the [System Library Manual](#).*

All analysis methods are available to you after the import is successfully executed.



## 8.3 Attributes of the Process Analysis

All of the attributes that are used for editing process analyses are listed below. All new attributes in DPE 5.15 as well as the attributes from earlier versions are listed. Some of the earlier attributes are also used in version DPE 5.15, others are still available but inactive because of compatibility reasons.



Table 38: Editing Process Attributes

| Physical name              | Prompt                     | DPE 5.14 | New | DPE 5.15                 |                      |
|----------------------------|----------------------------|----------|-----|--------------------------|----------------------|
|                            |                            |          |     | Usage in:                |                      |
|                            |                            |          |     | ergocomp process default | ergocomp org process |
| <b>Basic times</b>         |                            |          |     |                          |                      |
| m_dblestimated_ttb         | Entered Time (ttb)         |          | x   | x                        | x                    |
| m_dblestimated_ttu         | Entered Process Time (ttu) |          | x   | x                        | x                    |
| m_dblestimated_tw          | Entered Waiting Time (tw)  |          | x   | x                        | x                    |
| m_dblestimated_trg         | Entered Set up Time (trg)  |          | x   | x                        | x                    |
| m_dblCalctime_ttb          | Calculated Time (TTB)      | x        |     | x                        |                      |
| m_dblCalctime_ttu          | Calculated Time (TTU)      | x        |     | x                        |                      |
| m_dblCalctime_tw           | Calculated Time (TW)       | x        |     | x                        |                      |
| m_dblCalctime_trg          | Calculated Time (TRG)      | x        |     | x                        |                      |
| m_dblCalctime_tg           | Calculated Time (TG)       | x        |     | x                        |                      |
| m_dblvalidtime_ttb         | Validtime (ttb)            |          | x   | x                        | x                    |
| m_dblvalidtime_ttu         | Validtime (ttu)            |          | x   | x                        | x                    |
| m_dblvalidtime_tw          | Validtime (tw)             |          | x   | x                        | x                    |
| m_dblvalidtime_tt          | Validtime (tt)             |          | x   | x                        | x                    |
| m_dblvalidtime_tg          | Validtime (tg)             |          | x   | x                        | x                    |
| m_dblvalidtime_trg         | Validtime (trg)            |          | x   | x                        | x                    |
| <b>Calculated times</b>    |                            |          |     |                          |                      |
| m_dblCalctime_te           | Calculated Time (TE)       | x        |     | x                        | x                    |
| m_dblCalctime_tr           | Calculated Time (TR)       | x        |     | x                        | x                    |
| m_dblCalctime_ta           | Calculated Time (TA)       | x        |     | x                        | x                    |
| m_dblCalctime_t            | Calculated Time (T)        | x        |     | x                        | x                    |
| m_strcalc_unit             | Calculation Unit           |          | x   | x                        | x                    |
| m_dblcalcquantity          | Calculation Quantity       |          | x   | x                        | x                    |
| <b>Allowances</b>          |                            |          |     |                          |                      |
| m_bCalculateAllowances     | Calculate Allowances       | x        |     | x                        | x                    |
| m_dblallowancetemp_1 to 3  | Allowancetemp 1 to 3       |          | x   | x                        |                      |
| m_dblallowancevalue_1 to 8 | Allowancevalue 1 to 8      |          | x   | x                        |                      |
| m_dblallowancetime_1 to 8  | Allowancetime 1 to 8       |          | x   | x                        |                      |
| <b>General attributes</b>  |                            |          |     |                          |                      |
| m_iValidTime               | Valid Time                 | x        |     | x                        | x                    |
| m_pAllowanceSet            | Allowance Set              | x        |     | x                        | x                    |
| m_strValueAdded            | Value Added                |          | x   | x                        |                      |

| Physical name                             | Prompt                                 | DPE 5.14 | New | DPE 5.15                 |                      |
|---|--|----------|-----|--------------------------|----------------------|
|   |  |          |     | Usage in:                |                      |
|   |  |          |     | ergocomp process default | ergocomp org process |
| m_dblValueAdding Percentage               | Value adding percentage                |          | x   | x                        |                      |
| m_AnalysisLines                           | Analysis Lines                         |          | x   | x                        |                      |
|   |  |          |     |                          |                      |
| <b>Attributes that are no longer used</b> |  |          |     |                          |                      |
| calcre-sult__sum_calculated_time          | Calculated Valid Times                 | x        |     | x                        | x                    |
| calcre-sult__sum_estimated_time           | Sum Estimated Times                    | x        |     | x                        | x                    |
| calcre-sult__sum_time1                    | Sum reference time                     | x        |     | x                        | x                    |
| calcre-sult__sum_time2                    | Sum 2nd time                           | x        |     | x                        | x                    |
| m_calctime                                | Calculated Time                        | x        |     | x                        | x                    |
| m_time                                    | Time = estimated Time                  | x        |     | x                        |                      |
| m_time1                                   | time1 = reference time                 | x        |     | x                        |                      |
| m_time2                                   | time2 = 2 <sup>nd</sup> estimated time | x        |     | x                        |                      |
| m_dblAttribute_1                          | Calculated Analysed Times              | x        |     | x                        |                      |
| m_dblValidTG                              | Valid TG                               | x        |     | x                        |                      |
| m_dblValidTRG                             | Valid TRG                              | x        |     | x                        |                      |
| m_dblTimePerUnit                          | Time per Unit                          | x        |     | x                        |                      |

The connections between the individual attributes are shown in the following figure. *Please refer to the [Figure 191](#).*

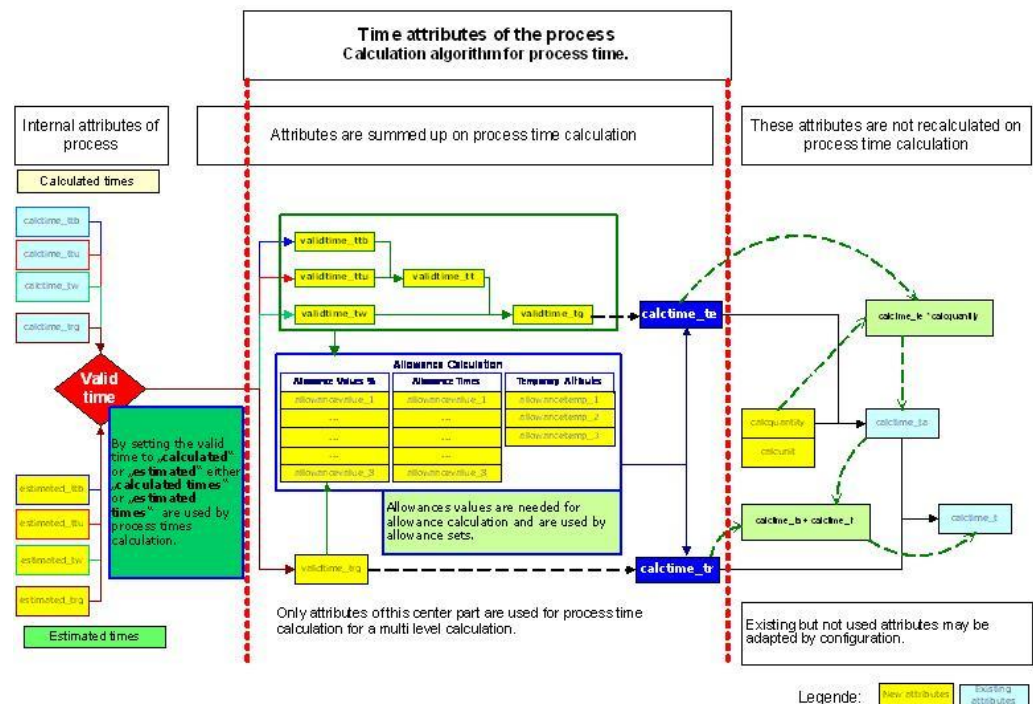


Figure 190: Open EP4 Database Conversion Importing Time analyses from an Oracle Database

### 8.3.1 Starting the Transfer

The import of ERGOTime analyses from an Oracle database is very similar to the conversion of an EP4 database and is therefore also executed via the EPDBUpdater.

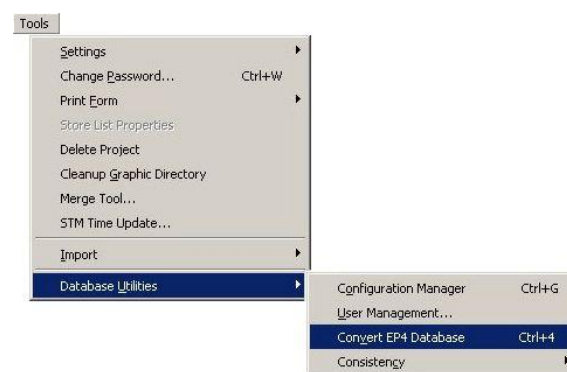


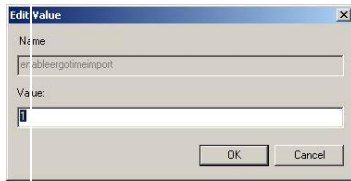
Figure 191: Open EP4 Database Conversion

### 8.3.2 Data Conversion Menu Item

After calling up the menu item "Convert EP4 database", a dialog with a tree view for selecting data appears. In order for the branch "ERGOTime work system" relevant to the ERGOTime import to appear, an entry must be made in the registration editor.

Under

```
HKEY_LOCAL_MACHINE\SOFTWARE\DELMIA\ergoplan\common\
"EnableERGOTimeImport"="
```



there must be a string value EnableERGOTimeImport with the value of 1. If this is not the case, a string value must be re-created. You can do this in the registry editor or via the menu **Tools < Settings < Maintenance**.

If you open the node "ERGOTime work systems", the connection parameters for access to the database are queried:

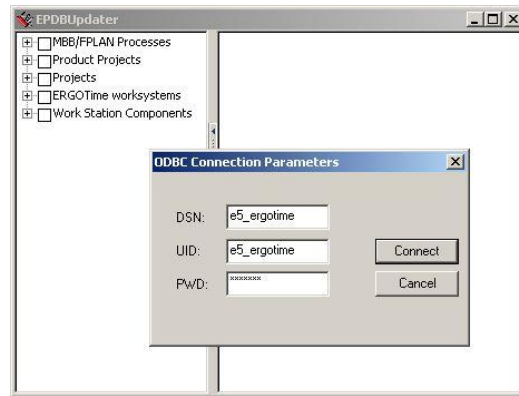


Figure 192: Open EP4 Database Conversion

### 8.3.2.1 Database Connection

#### DSN

Enter the ODBC name of the data source (DB Ergotime) under DSN.

#### UID

UID is the user name of the database

#### PWD

Enter the password under PWD.

#### Note

*Depending on the operating system used, you will find the relevant information under Settings < Control Panel < Administrative Tools < Data Sources (ODBC).*

*The ERGOTime database version that has been found is indicated in an information dialog.*



### 8.3.3 Starting the Import

The import of analyses from an Oracle database involves four steps.

- 1) Selecting the analyses and starting the conversion
- 2) Selecting the plantype set  
Selecting value added groups and entries (optional)
- 3) Selecting the target project
- 4) Selecting the conversion mode

### 8.3.3.1 Selecting the Analyses and Starting the Conversion

After connecting to the database, the tree view can be opened to the station level and the work systems and stations you want to import can be marked (individual analyses can not be selected).

You can start the import in one of three ways.

- 1) One option is via the EPDBUpdater menu. Select the menu item **Start Conversion**.
- 2) Another is via the EPDBUpdater menu bar. Select the corresponding button (Save icon).
- 3) The third option is via the context menu (right mouse click in the tree view). Select the menu item **Convert Selected Elements**.

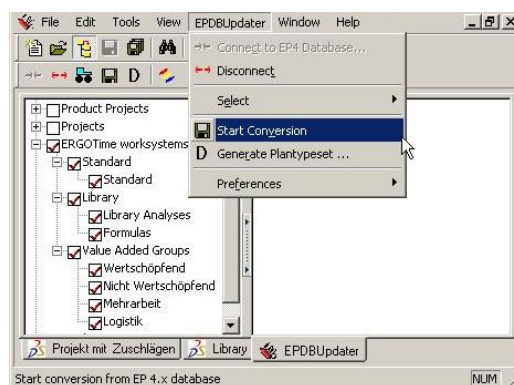


Figure 193: Starting the Conversion via the Menu

### 8.3.3.2 Select Plantype Set

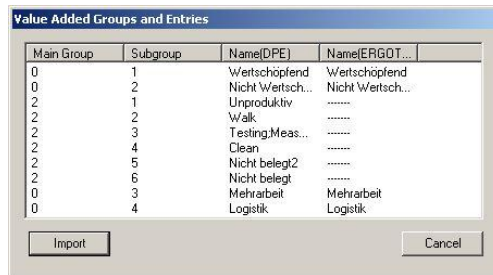
A dialog for selecting the plantype set is opened.

If one decides to import into a previously existing project, the project should of course have a suitable plantype set allocated to it. The plantype set does not need to cover all ERGOTime subsystems; Non-specific subsystems are then simply ignored during the import, which is noted in the modification log ([Log](#)). However inconsistencies could arise if referenced analyses are not imported because of a missing plantype set entry.



Figure 194: Data Import

The data import is started by confirming with 'OK'.



**Figure 195: Value Added Groups and Entries**

Select the value added groups and entries to be imported and activate the button *Import*. In the following dialog you will be asked to indicate the template. If you select *no* value added groups or entries, the import continues as usual.



**Figure 196: Choose a Template**



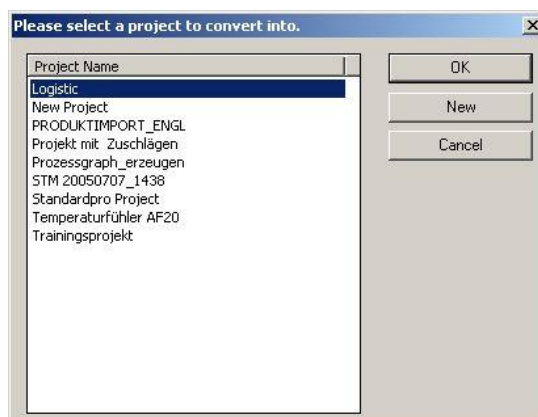
### Note

*All value added groups and entries are imported even if you have selected only individual entries.*

*Previously existing value added groups and entries are overwritten in the plantype set. This is because time analyses and previously set value adding percentages are transferred at a 1:1 ratio. Therefore the same requirements must apply to both the target system and the source system. Another reason is the compatibility between old and new analyses: they should all be based on the same value added definition*

### 8.3.3.3 Selecting the Target Project

In the next step, the target project for the data to be imported is specified. Ensure that you select a project of the previously selected plantype set.



**Figure 197: Select a Project**

Select a project or create new project using the "New" button.

If the plantype set of the selected target project does not contain any of the necessary types, the import is cancelled with a corresponding error message.

If you want to create a new target project, you must select a suitable plantype set. This plantype set can not be imported from here; it must already exist in the database.

### 8.3.3.4 Selecting the Conversion Mode

Set which analyses are to be converted in the process assignment.

- If **Import assigned analyses only** is active, only the analyses that are assigned to a process are imported.  
There is a condition: there must be processes in the project to which analyses were assigned.
- If **Import assigned analyses only** is not active, all analyses are imported into the project library.

**Exception:** The formulas are always imported into a template of the system library.

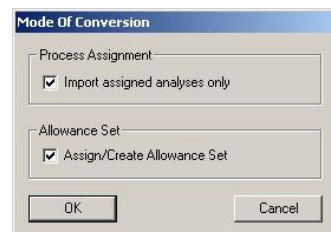


Figure 198: Conversion Mode

Allowance sets are not imported -- only the reference to a allowance set is, since allowances in the DELMIA Process Engineer® can be defined individually.

The data import is started by confirming with 'OK'.

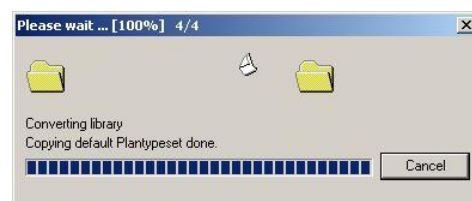
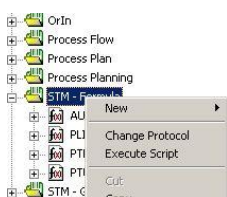


Figure 199: Data Import Started

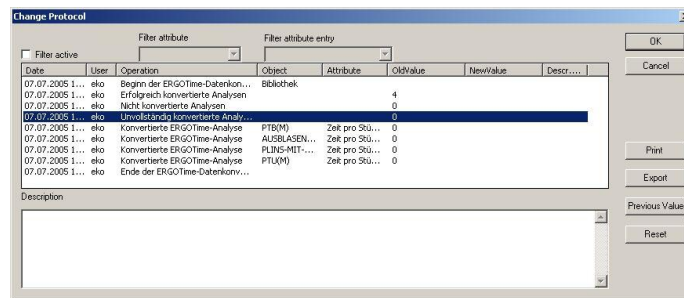
## 8.3.4 Log



A log is created for every imported analysis method. The log is opened via the pop-up menu of the analysis method:

Every imported analysis is logged with its code and its time per item. Furthermore, the analyses that can not be imported or can be imported only partially (without referenced analyses) are noted in the log.





**Change Protocol**

Filter attribute: [v] Filter attribute entry: [v]

☐ Filter active

| Date            | User | Operation                           | Object       | Attribute       | OldValue | NewValue | Descr.... |
|-----------------|------|-------------------------------------|--------------|-----------------|----------|----------|-----------|
| 07.07.2005 1... | eko  | Beginn der ERGOTime-Datenkon...     | Bibliothek   |                 |          |          |           |
| 07.07.2005 1... | eko  | Erfolgreich konvertierte Analysen   |              |                 | 4        |          |           |
| 07.07.2005 1... | eko  | Nicht konvertierte Analysen         |              |                 | 0        |          |           |
| 07.07.2005 1... | eko  | Unvollständig konvertierte Analy... |              |                 | 0        |          |           |
| 07.07.2005 1... | eko  | Konvertierte ERGOTime-Analyse       | PTB(M)       | Zeit pro Sku... | 0        |          |           |
| 07.07.2005 1... | eko  | Konvertierte ERGOTime-Analyse       | AUSBLASEN... | Zeit pro Sku... | 0        |          |           |
| 07.07.2005 1... | eko  | Konvertierte ERGOTime-Analyse       | PLING-MET... | Zeit pro Sku... | 0        |          |           |
| 07.07.2005 1... | eko  | Konvertierte ERGOTime-Analyse       | PTUM(M)      | Zeit pro Sku... | 0        |          |           |
| 07.07.2005 1... | eko  | Ende der ERGOTime-Datenkomm...      |              |                 |          |          |           |

Description: [v]

Buttons: OK, Cancel, Print, Export, Previous Value, Reset

**Figure 200: Log Details**

# B. Operating the Software

## 9. General Information

This chapter contains all the information required for the continuous productive operation of the software. This also includes the use of export and import tools for system elements and projects. It describes how the Server Tools can be used to monitor the current operation and how the Database Tools are employed to identify and resolve problems in the database.

### **Conversion of ERGOPlan 4 Databases**

#### **WSC import and export from E5 to E5**

- Please refer to the [Import and Export of Work System Elements and Projects Manual](#).

#### **Exporting and importing projects**

- Please refer to the [Import and Export of Work System Elements and Projects Manual](#).

#### **Cleanup Graphic Directory**

#### **Server Tools**

- Please refer to the [Server Tools Manual](#).

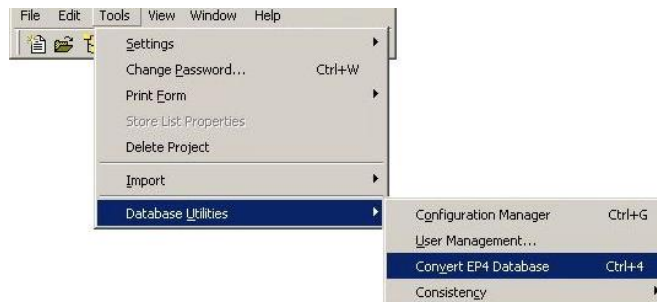
#### **Recurring Tasks**

#### **Database Tools**

## 10. Conversion of ERGOPlan 4 Databases

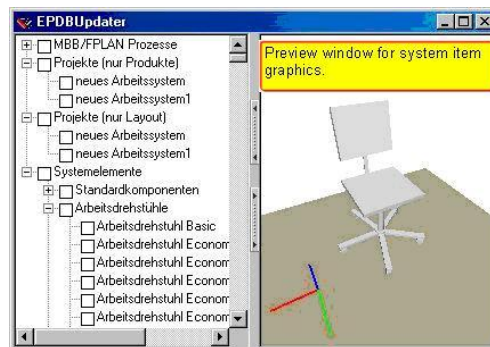
### 10.1 Convert Data

You can continue using your ERGOPlan 4 data in the DELMIA Process Engineer. However, to continue using your ERGOPlan 4 data you first have to convert the data into a DELMIA Process Engineer project.



**Figure 201: The “Convert EP4 database” Menu Item**

For this purpose the program „**EPDBUpdater**“ (ERGOPlan Database Updater) has been developed. The “EPDBUpdater” is started using the *Convert EP4 database* menu item in the “Tools” menu of the PPR Navigator. A window divided into two sections is opened.



**Figure 202: The “EPDBUpdater” window**

The following information is displayed on the left side:

- Processes (MBB/FPLAN processes)
- Projects (only products): only the product structure is read
- Projects (only layout): only the layout (as in E4 separated by systems/stations) can be read.
- System items

With products as well as with layouts on the highest level the “work systems” are always shown.

In the case of system items the corresponding graphics are shown as a 3D preview on the **right side**. By clicking in the preview window you have the

possibility, as with other graphic windows, to rotate or to translate the object in the view.

### 10.1.1 The “EPDBUpdater” Toolbar

In the toolbar of the PPR Navigator additional tools appear:

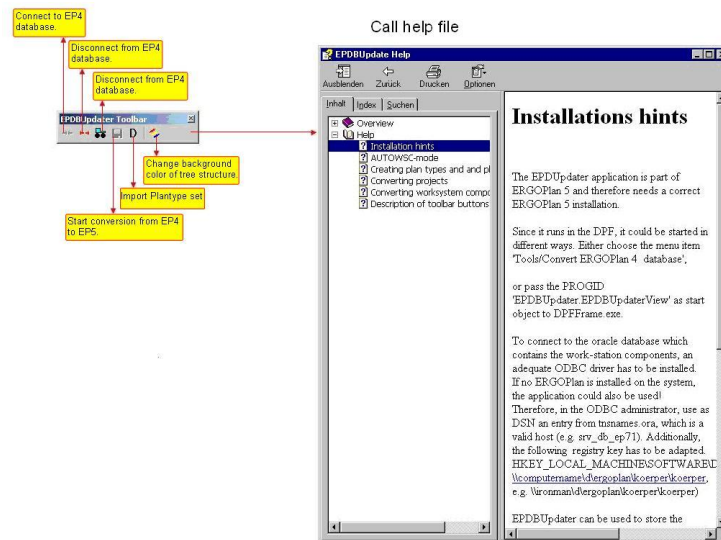


Figure 203: The “EPDBUpdater” Toolbar

### 10.1.2 The “EPDBUpdater” Menu Item

When using the “EPDBUpdater” the “EPDBUpdater” menu item is added in the PPR Navigator.

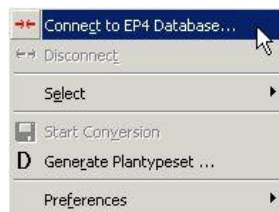


Figure 204: The “EPDBUpdater” Menu Item

The following menu items are available in the menu:

- **Connect to EP4 Database:** Using this menu item you can connect to an ERGOPlan 4 database.
- **Disconnect:** You can disconnect the connection to the ERGOPlan 4 database.
- **Select:** Using this menu item you can direct the selection in the left part of the window of the EPDBUpdater. The following sub-menu items are available:

#### Toggle Selection

You can change the selection of the currently marked tree element (mark/unmark).

### Workstation Components

You can select either all system items at once or you can select a certain system item group, for example, work chairs or desks. For this purpose, the

**None:**

**necessary options are available in an additional sub-menu**

No item is selected, a previously made selection is reversed.

- 1) Start Conversion
- 2) The conversion process is started

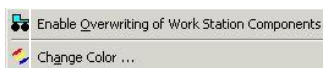
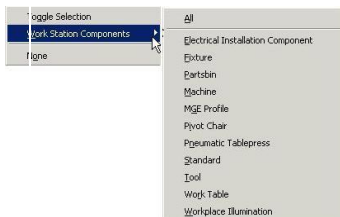
### Generate Plantypeset

A plantype set is loaded from an INI file. A file selection window opens for this purpose.

### Preferences

In the **Preferences** menu item you can make the following settings:

- **Enable Overwriting of Workstation Components:** If the system items already exist in the DELMIA Process Engineer database, the system items are overwritten without confirmation during the import process.
- **Change Color:** You can set the background color for the left window section here.
- **Help:** The Help function is called.



## 10.1.3 The EPDBUpdater Context Menu

If you right-click on an object in the left side of the EPDBUpdater, a context menu with the following menu items opens.



**Figure 205: The “EPDBUpdater” Context Menu**

- **Toggle Selection:** The selection in the left window section is reversed.
- **Select:** Functions for object selection are made available in the left window section.
- **All System Elements:** Select all system items.
- **All Siblings:** Select all siblings of the selected (active) element.
- **None:** No item is selected, a previously made selection is reversed.
- **Convert Selected Items:** The conversion process is started.

## 10.2 Database Settings

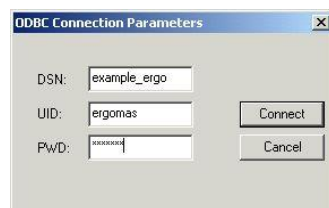
To convert data from the EP4 database, these data must first be displayed on the screen. For this purpose it is necessary to connect the EPDBUpdater with the EP4 database. First start the EPDBUpdater in the “Database Utilities”/EP4 database menu item. You can start the connection:



Using the Connect button in the EPDBUpdater toolbar (*Please refer to the [The “EPDBUpdater” Toolbar](#)*), or

By expanding the tree in the EPDBUpdater

The “ODBC Connection Parameters” dialog box opens.



**Figure 206: Connection Parameters**

## 11. Cleanup Graphic Directory

The graphics information is stored in the Process Engineer in a body directory. Redundant files can be removed from this directory with the function *Cleanup Graphic Directory*, which is included as of version PE 5.14.



### Note

*This function can be executed only by the administrator. Before you begin, ensure that all databases and applications for all normal users are closed. Before you begin, ensure that no graphics files can be created or deleted during the execution.*

### 11.1 General Comments

The function *Cleanup Graphic Directory* can be executed in three ways; in simulation mode, backup mode and in real mode.

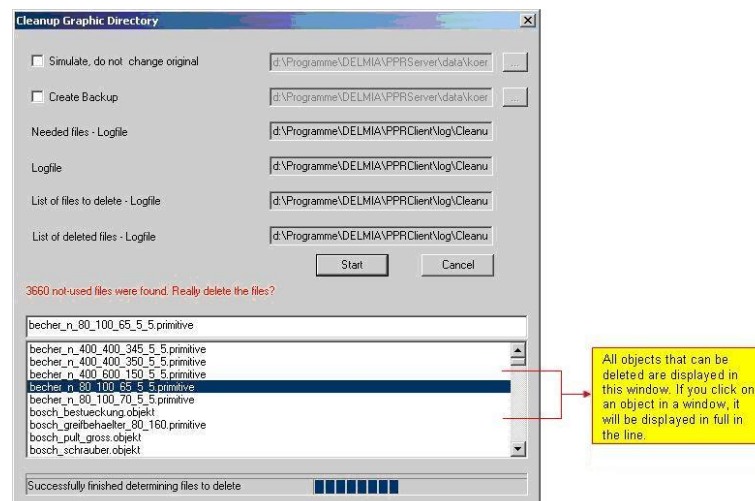


Figure 207: Cleanup Graphic Directory Dialog

#### 11.1.1 Information on Log Files

Log files are created for each mode of operation. The path of the log files are fixed and can not be edited. The standard location of the log files is the folder *PPRClient/log*.



### Note

*The path of both log files Required files – log file and log file are read from the folder of the registration editor. If this registration key has been deleted by accident, both paths are sent to the Temp folder of the local settings: for example C:\Documents and Settings/Username/Local Settings/Temp.*

### Needed Files – Logfile



A log file for required data which contains paths to all of the files in the body directory.

Needed files - Logfile `ELMIA\PPRClient\log\CleanupNeededFiles.log`

#### Figure 208: Log File for Required Files

The **log file for required files** contains all files used in graphics or layouts.

#### Logfile

A normal log file in which the entire process of the function *Cleanup Graphic Directory* is logged: backup mode and simulation mode.

Logfile `rogramme\DELMIA\PPRClient\log\Cleanup.log`

#### Figure 209: Log File for Statistical Data

All statistical data such as time, execution steps and number of objects are found in the **log file**. Problems which arise during the process are displayed and notes on remedying these problems are contained in this file.

#### List of the Files to Delete - Log File

This log file includes all objects that are to be deleted.

List of files to delete - Logfile `LMIA\PPRClient\log\CleanupFilesToDelete.log`

#### Figure 210: Path – Log File for Objects that are to be Deleted

#### List of Deleted Files - Log File

This log file includes all objects that have been deleted.

List of deleted files - Logfile `ELMIA\PPRClient\log\CleanupDeletedFiles.log`

#### Figure 211: Path – Log File for Objects that have been Deleted



#### Note

*Both log files are compared in order to determine whether all objects have been deleted. Both log files must as a rule correspond to one another.*

## 11.1.2 Learning about the Modes of Operation

### 11.1.2.1 Simulation Mode

In simulation mode you can simulate the cleaning of the body directory; previously existing files are not deleted in this mode. The result of the simulation is, just like in backup mode, the cleaned body directory with the files for which an operational reference to existing graphics and layouts exists.

In this procedure, the simulated files are copied to a simulation directory which you have created.

### 11.1.2.2 Backup Mode

In backup mode you can clean the body directory in real mode; in this mode unnecessary files are irrevocably deleted.

All files of the body directory are copied to a backup directory. In backup mode these copied files are checked and the redundant files are deleted from the body directory after the check.

### 11.1.2.3 Real Mode

In pure real mode the cleanup of the body directory is carried out without a **backup or simulations directory** being created.



#### Note

*If you have previously saved the directory, for example you have burned it to a CD or DVD, you do not need a backup directory. In this case you can execute the function Cleanup Graphic Directory without further consideration.*

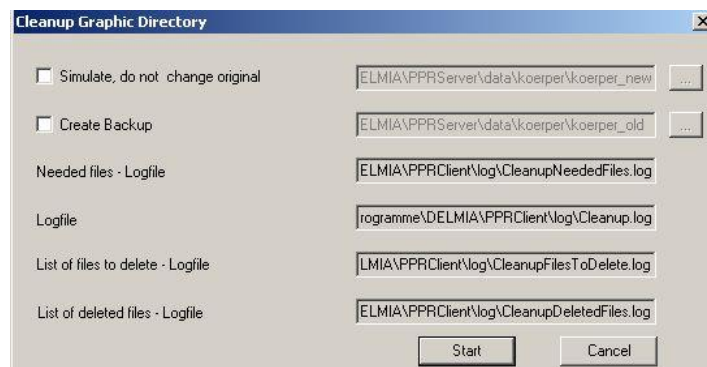


Figure 212: Neither Backup nor Simulation Activated

### 11.1.3 Important Notes for the Execution

Before starting *Cleanup Graphic Directory*, check to ensure that the following prerequisites for execution have been fulfilled:

- The directory for the log files must exist and must not be write-protected (full access is required).
- Full access to the body directory is required
- Full access to the simulation directory is required (if a simulation is to take place)
- There must be enough space in the backup directory for the complete body directory (with subdirectories), if "Create Backup" is marked). Full access to the directory is required.
- The path to the body directory must be set correctly in the registry.

## 11.2 Starting Cleanup Graphic Directory

Before executing *Cleanup Graphic Directory*, ensure again that the prerequisites listed under *Important Notes on the Execution* have been fulfilled.

This function is executed above all in order to delete files that are no longer required. In this way you can ensure that the current graphics files are always located in the body directory, and that the body directory does not become too large, thus preventing disk space from being used unnecessarily.

The following is checked:

- First, all links from graphics objects to the database are checked.
- Files for graphics or layouts which no longer exist in the application are deleted from the body directory after execution in backup mode and real mode.



### Caution

*Windows functions and dialogs are used during the execution for deleting and copying files. Most of these dialogs have an Abort button. You must never click this button, otherwise the entire process will be aborted. This behavior is Windows-specific and cannot be changed.*

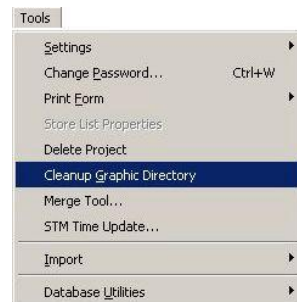


Figure 213: Starting *Cleanup Graphic Directory* via Tools Menu

## 11.2.1 Opening the Cleanup Graphic Directory Dialog

- 1) Select *Cleanup Graphic Directory* in the Tools menu. Please refer to the [Figure 213](#).
- 2) In order to activate the simulation mode, place a checkmark in the *Simulate, do not change original* field. **Create Backup must not** be activated in this mode.
- 3) In order to activate backup mode, place a checkmark in the **Create Backup** field. *Simulate, do not change original must not* be activated in this mode.

☒ Simulate, do not change original

☒ Create Backup



### Caution

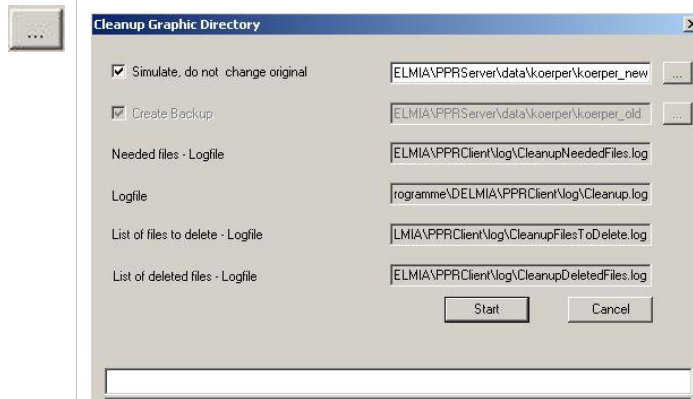
*Before starting always check whether there is enough storage space for the simulation or backup mode.*

#### 11.2.1.1 Working in Simulation Mode

It is advisable to first simulate the cleanup redundant graphics files in the body directory before the execution in backup mode in order to first get a result for the cleaned-up body directory. If the result is satisfactory, you do not have to

worry about the implementation in real mode and you can execute the function in backup mode; you already know the result from the simulation.

- 1) This button is used to set the simulation directory.



**Figure 214: Activate Simulation in the Dialog**

- 2) Click on *Start* in order to start the simulation. The current information on the process is always displayed in the status line (*Ready* in the figure).

- The new simulation directory is created according to the rights of the administrator. All needed files are copied from the body directory into this simulation directory.
- The process is logged in both log files (*Logfile and Needed file – Logfile*); if errors occur during the process, you will be given important notes via the log files as to how you can remedy the errors.

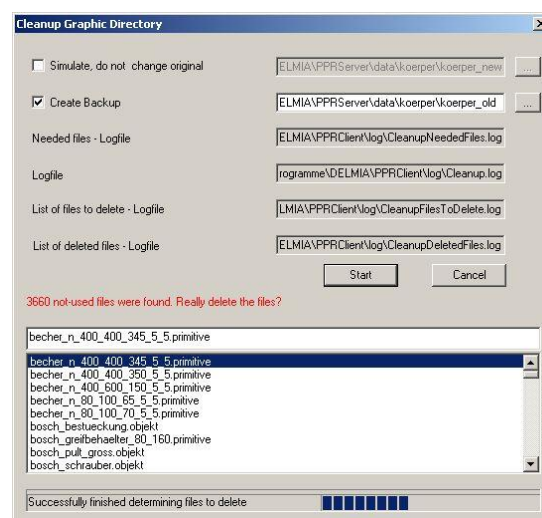
Cancel

After finishing the process, you can leave the dialog by clicking *Cancel*. Please refer to the [Figure 214](#).


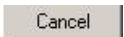
### 11.2.1.2 Working in Backup Mode



Set the backup directory using the button.



**Figure 215: Activating Backup in the Dialog**

- 
- 
- 3) Click *Start* in order to start the backup.
  - 4) The new backup directory is created according to the rights of the administrator. All files are copied from the body directory into this backup directory.
    - Only the valid files are copied (remain) to the body directory.
  - 5) The process is logged in both log files; if errors occur during the process, you will be given important notes via the log files as to how you can remedy the errors.
  - 6) Confirm the query as to whether files really should be deleted with *OK*. The entire process will be aborted if you click on *Abort*.
  - 7) After finishing the process, you can leave the dialog by clicking *Cancel*.

## 11.3 Server Tools



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

## 12. Recurring Tasks

To ensure optimum performance of the database despite growing data volume, the statistics must be recomputed regularly (daily or at least weekly, but especially after creation or change of larger data volumes). It is not necessary to stop the PPR server to do so. If the statistics are not recomputed regularly, a performance drop is likely to occur.

On all database users execute the following command (example for e5\_database)

```
Exec dbms_stats.gather_schema_stats('E5_DATABASE',
                                     estimate_percent => 90,
                                     granularity => 'ALL',
                                     method_opt   => 'FOR ALL INDEXED COLUMNS SIZE 1',
                                     cascade=> TRUE)
```

1) In ORACLE 9.2.0 the following command can be used

```
Exec                                dbms_stats.gather_schema_stats('E5_DATABASE',
                                     estimate_percent   => dbms_stats.auto_sample_size,
                                     granularity         => 'ALL',
                                     method_opt         => 'FOR ALL INDEXED COLUMNS SIZE 1',
                                     cascade=> TRUE)
```

### 12.1 Database Tools



For more information, *Please refer to the* [DBAnalyser Manual](#) and [PlanType-Set Switch Manual](#).

### 12.2 Terminating the PPR Server Processes

The PPR server processes (*IPDServer.exe* and *EPSTools.exe* on all server machines) use up operating system resources (e.g. main storage and handles). In time these resources are fragmented and cannot be reused. On the one hand this can lead to a performance drop, on the other hand to depletion of system resources. A known limitation in Windows 2000 Professional Server, for instance, is that no more than 2 GB of main storage can be used per process. You can avoid such problems in production by regular restart of the processes. To do that, all users must be logged off from the respective server.



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

Termination of the PPR server processes also leads to release of all ORACLE sessions of the server. Thus, these resources are also re-released on the database level.

When any type of problem occurs with the PPR Server, this method should also be followed if that seems required upon analysis of the log files. Thereby the continuous operation of the other PPR servers and users can be guaranteed.

## 12.3 Terminating the PPR Master

Sometimes it is also required to terminate all processes. Then, upon termination of all PPR server processes, the PPR services (UpdateMg.exe, LockMng.exe and EPPoolingServer.exe) must be terminated. This is required, for instance, before you export or import a database dump.



# Appendix

## 1. General Information

This appendix contains a description of the data types and the control types. Additionally, you will find the data model for Automatic Line Balancing.

### 1.2 Data Types

The data types you can use are described in the following. Data types that are necessary for internal calculations are not listed here.

**Double** data type (double value with double precision) are saved as 64 bit double values (8 bytes) ranging from: **-1.79769313486231E308** to **-4.94065645841247E-324** for negative values and from **4.94065645841247E-324** to **1.79769313486232E308** for positive values.

**Single** data type (double value with single precision) are saved as 32 bit-double values (4 bytes) ranging from: **-3.402823E38** to **-1.401298E-45** for negative values and from **1.401298E-45** to **3.402823E38** for positive values.

**Integer** data type are saved as 16 bit numbers (2 bytes) ranging from – 32,768 to 32,767. With **Integer** data types variables you can also display list values. A list value consists of an endless quantity of unambiguous integers; each integer has a special meaning in the used context.

**String** Data type: **There** are two types of string values:

Fixed-length string values and variable-length string values.

- Fixed-length string values, the 'normal strings' can contain anywhere from 1 to around 1000 characters. The maximum number depends on the language settings used.

There are individual fields which can accommodate up to 4000 characters e.g. code rules.

Fields with 4000 characters:

```
XDOCoderule    m_strCoderule
XDOConfiguredObject m_strExtEffectivity
XDOConfiguredObject m_strCoderuleCopyNew
XDOConfiguredObject m_strCoderulePlanningCodeCopyNew
XDODefaultImplm_note
XDOErgoCompBase m_strVPMID
```

The same applies to the attribute in all derived classes.

These are filed in the ORACLE database as a VARCHAR2 data type. The database administrator is capable of configuring attributes with more than 1000 and up to 4000 characters. If the value of the attribute is longer than the defined number, all characters over this number are not saved.

Variable-length strings are BLOBs (binary large objects). These are filed in the ORACLE database as a LONG RAW data type.

Variable-length string values can contain up to 2 billion (or 2<sup>31</sup>) characters.

**Date** data types are saved as 64 bit double values (8 bytes) and can save a date ranging from January 01<sup>st</sup>, 100 to December 31<sup>st</sup>, 9999 and a time ranging from 0:00:00 to 23:59:59.

**Currency** data type: a data type with a value ranging from 922,337,203,685,477.5808 to 922,337,203,685,477.5807. Use this data type for calculating currencies and for calculations with fixed decimals, where precision is decisive. In Visual Basic the type identifier At (@) represents a value of the **Currency** data type.

**Float** data type = **Double** data type

**Bool** data type: Variables of the **Bool** data type are saved as 16 bit numbers (2 bytes); they can only assume the values **True** or **False**. Variables of the **Bool** data type are output as True or False. When translating other *numeric data types* into values of the **Bool** data type, 0 becomes **False** and all other values become **True**. When translating values of the **Bool** data type into other data types, **False** becomes 0 and **True** becomes -1.

**MetaFile** data type: This data type is necessary for printing graphics.

**Callback** data type: This data type is necessary for printing.

## 2. Control Type

Prompt only

### PromptOnly

No input or output field is created, but only the name is displayed. Can be used as caption or group separator.

### ComboBox

A ComboBox combines the characteristics of a text field (Edit) and of a list box (ListBox). Like in a list box the user can select an existing value.

### Edit

A text field that is mostly used for displaying (or entering) information entered by the user. The input length is dependent on the number of characters defined.

### Multiline Edit

Similar to Edit except for the fact that the input field can consist of multiple lines.

### Checkbox

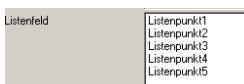
Shows whether an item has been selected or not. The checkbox is used to give the user a choice between two values such as *Yes/No*, *True/False* or *On/Off*.

### RadioButton, OptionButton

Shows whether or not an item has been selected in a group of selection possibilities. The option field is used to show whether or not an item has been selected within a group. Bear in mind that the option fields within a group are mutually exclusive.

### List box

Shows a list of values from which you can select a value. If the list field (list box) is bound to a data source, it shows the values of the data source.





## SelectFile


Direct input of a file or direct opening of a file selector to enter the selected **RTFEdit**

Input field (similar to multiline edit) with the possibility to edit texts.

If this control type is used in a series with other control types, the RTF field is displayed together with the formatting parameters when printing. To avoid this effect use the RTFEdit control type alone on one side.



## FileViewer

After selecting , you can open the selected file or application by using the *Open* icon. A selected image file or HTML page is displayed in the top part of the field. All picture formats that can be displayed by the Internet Explorer are displayed. Whether the properties dialog is open or not, the display can be used interactively if the selected file (or URL, e.g. [www.delmia.de](http://www.delmia.de)) has HTML formatting. To display the file in the Internet Explorer, click on the Internet Explorer icon.



## SelectAndOpenFile

After selection the selected file or application can be opened by using the *Open* icon.



## Bitmap

Similar to the FileViewer. However, only 2D graphic files with the extensions \*.jpg, \*.bmp and \*.gif are displayed; these cannot be selected freely. Only one file may be assigned. Thereafter manual change is not possible.

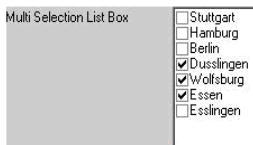
Assignments can be made by a script. To be able to make an attribute more flexible using this type of control, you can assign a script assignment to the attribute; it could for instance open a file selector and then enter path and file-name.



## Multi Selection List Box

Several entries can be selected in a list field. The *Multi Selection List Box* combines a list with several check boxes.

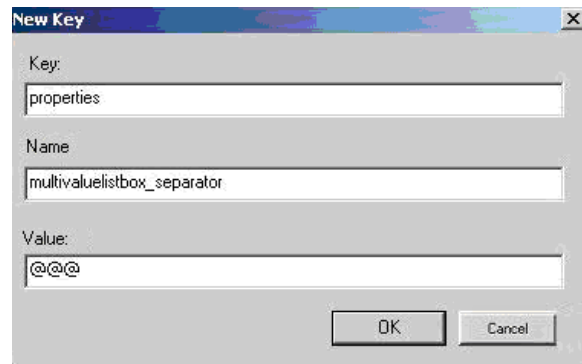
If the control type Multi Selection List Box was selected from an attribute, the corresponding file type is string (Length: approx. 1000 ANSI characters, extendable to 4000 characters) and a value list must be created.



Further, the key Properties under **Tools > Settings > Maintenance** on the Global screen panel must not have an empty value for the entry *multivaluelist-box\_separator*. The key must be generated if it does not exist. Follow these instructions to generate the key:

- 1) Open the maintenance tool via the menu *Tools > Settings > Maintenance*.
- 2) On *Global* screen panel select the entry *ErgoPlan* and generate a new key.



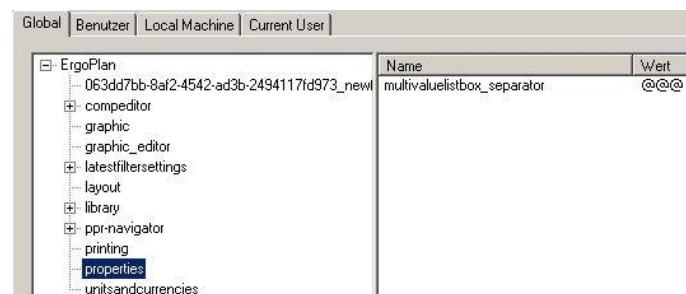


A dialog box titled "New Key" with three input fields: "Key:" containing "properties", "Name:" containing "multivaluelistbox\_separator", and "Value:" containing "@@@".

Figure 216: New Key

The value must be entered. By default the field contains the characters @ @ @.

- The key should look as shown below.



A dialog box titled "Key Value" with a tree view on the left and a table on the right. The tree view shows a hierarchy under "ErgoPlan" with "properties" selected. The table has two columns: "Name" and "Wert".

| Name                        | Wert |
|-----------------------------|------|
| multivaluelistbox_separator | @@@" |

Figure 217: Key Value



### Caution

*ProcessFilter, ProductFilter, ResourceFilter, EffectivityFilter, and LabelFilter. LineFilter should only be used on the top node (project node) If used on all other nodes, this results in error messages.*

## 3. Data model for Automatic Line Balancing

### Description of Plantypes with Attributes

Table 39: Manufacturing Concept

| Attribute        | Meaning  | Type     | Value Range | Comment |
|------------------|--|----------|-------------|---------|
| name             | Name   | String   |             |         |
| modificationdate | Date   | DateTime |             |         |
| cycletime        | Cycle time (s)                                       | Float    | 0 -         |         |
| waytime          | Way time [s]   | Float    | 0 -         |         |
| waytimereduced   | Reduced way time [s]                                 | Float    | 0 -         |         |
| maxcapacity      | Max. number of workplaces in the case of MultiCycles | Integer  | 0 -         |         |

| Attribute               | Meaning   | Type    | Value Range | Comment   |
|-------------------------|---|---------|-------------|---|
| processselectionuserule | Selection rule for processes in the case of automatic balancing | Integer | 0 – 3       | 0 = Undefined<br>1 = Sum of successors<br>2 = Max. of successors<br>3 = Concurrence |
| stationselectionuserule | Planning of station areas in the case of automatic balancing    | Integer | 0 – 2       | 0 = Undefined<br>1 = Work place with min. usage<br>2 = Concurrence                  |
| note                    | No. of 1. station   | String  |             |   |
| stationlength           | Station length  | Float   |             |   |
| stationwidth            | Station width   | Float   |             |   |
| materialareawidth       | Width of the material supply area                               | Float   |             |   |
| calculationperiod_alb   | Basic calculation periods                                       | Float   | 0 -         |   |

**Table 40: Line ergocomplantdefault**

| Attribute        | Meaning            | Type    | Value Range  | Comment  |
|------------------|--------------------|---------|--------------|--|
| setuptime        | Setup time         | Float   | 0 -          |  |
| setuptimereduced | Reduced setup time | Float   | 0 -          |  |
| bodydirection    | Body direction     | Integer | 0 – 3        | 0 = Lengthwise forwards<br>1 = Lengthwise backwards<br>2 = Crosswise to the right<br>3 = Crosswise to the left |
| viale_version    | Version-No.        | Integer | 0 -          | Not configurable   |
| cycletime        | Cycletime          | Float   | 0 -          | Default for new balancing  |
| alb_fix_layout   | Layout fixed       | Boolean | True / False |  |

**Table 41: Station ergocomplantdefault\* (\*was ergocomplanstation)**

| Attribute                     | Meaning                         | Type    | Value Range | Comment  |
|-------------------------------|---------------------------------|---------|-------------|--|
| name                          | Name                            | String  |             |  |
| nameshort                     | 1-based index                   | String  |             |  |
| stationtype                   | Station type                    | Integer | 0 – 2       | 0 = Undefined<br>1 = Manual station<br>2 = Automatic station |
| cycletime                     | Cycle time [s]                  | Float   | 0 -         |  |
| materialarea_locklength_left  | Locked material area left [mm]  | Integer | 0 -         |  |
| materialarea_locklength_right | Locked material area right [mm] | Integer | 0 -         |  |
| stationheight                 | Stationheight                   | Float   | 0 -         | 0 =  |

| Attribute | Meaning     | Type  | Value Range | Comment                                 |
|-----------|-------------|-------|-------------|---|
|           |             |       |             | SH_NOTHING<br>1 = SH_LOW<br>2 = SH_HIGH |
| width     | Width [mm]  | Float | 0 -         |   |
| length    | Length [mm] | Float | 0 -         |   |

**Table 42: Workplace ergocomplantdefault\* (\*was ergocomplanstation)**

| Attribute         | Meaning               | Type    | Value Range | Comment  |
|-------------------|-----------------------|---------|-------------|--|
| name              | Name                  | String  |             |  |
| nameshort         | Number                | String  |             |  |
| carbodyposition   | Workstation area      | Integer | 0 – 2048    | Binary coded car body positions                            |
| lockstatus        | Lock status           | Integer | 0 -         |  |
| multicyclecount   | Multi cycle count     | Integer | 0 -         |  |
| multicyclelimit   | Multi cycle limit     | Integer | 0 -         | Max. number of workstations in the case of multi cyclers.  |
| containerareaside | Side of material area | Integer | 0 – 3       | 0 = Undefined<br>1 = Left<br>2 = Right<br>3 = Left + Right |
| workarea_color    | Color of work area    | Integer | 0 -         |  |
| workarea_name     | Name of work area     | Integer | 0 -         |  |

**Table 43: Worker: ergocomplantworker**

| Attribute | Meaning | Type   | Default Value | Comment                                      |
|-----------|---------|--------|---------------|--|
| name      | name    | string | worker        | Attribute currently unimportant for planning |

**Table 44: Behavior: ergocomprocessbehavior**

| Attribute | Meaning | Type   | Default Value | Comment  |
|-----------|---------|--------|---------------|--|
| name      | name    | string | DNBBehavior   | Type must not be changed. Attribute currently unimportant for planning |

**Table 45: Logical Part bin: ergocomplantdefault**

| Attribute | Meaning | Type    | Value Range | Default Value        |
|-----------|---------|---------|-------------|----------------------|
| length    | Length  | Integer |             |                      |
| width     | Width   | Integer |             |                      |
| height    | Height  | Integer |             |                      |
| name      | Name    | String  |             | ALB Logical Part Bin |
| nameshort | Number  | String  |             |                      |

| Attribute   | Meaning              | Type   | Value Range | Default Value |
|-------------|----------------------|--------|-------------|---------------|
| Processlist | Name of process list | String |             |               |
| TSAlist     | Name of TSA list     | String |             |               |
| partID      | Part ID              | String |             |               |
| partbinID   | Part bin ID          | String |             |               |

**Table 46: Material Area: ergocomplantarea**

| Attribute                     | Meaning                   | Type   | Value Range | Comment   |
|-------------------------------|---------------------------|--------|-------------|---|
| name                          | name                      | String |             |   |
| nameshort                     | Number                    | String |             |   |
| containerareaside             | Side of the material area | String | 0 – 2       | 0 = not defined<br>1 = left<br>2 = right              |
| length                        | Length [mm]               | Float  |             |   |
| width                         | Width [mm]                | Float  |             |   |
| materiala-rea_locklength_left | Restricted area           | Float  |             | Meaning should be "locklength"                        |
| alb_lockedareas_strean        | Locked Areas in ALB       | String |             | All start and end values of the LockedAreas queued up |

**Table 47: Process: Ergocompprocessdefault**

| Attribute             | Meaning                               | Type    | Value Range | Comment  |
|-----------------------|---------------------------------------|---------|-------------|--|
| name                  | Name                                  | String  |             | True = Automatic Process<br>False = Manual Process         |
| nameshort             | Number                                | String  |             |  |
| validtime             | Valid Time [s]                        | Float   | 0 -         |  |
| time                  | Time [s]                              | Float   | 0 -         |  |
| calctime              | Calculated Time [s]                   | Float   | 0 -         |  |
| coderulefrequency     | Frequency                             | Float   | 0 -         |  |
| valueaddingpercentage | Value adding percentage [%]           | Float   | 0 -         |  |
| attribute_viale_1     | Automatic process                     | Bool    |             | 0 = Undefined<br>1 = Left<br>2 = Right<br>3 = Left + Right |
| carbodyposition       | Carbodyposition                       | Integer | 0 -         |  |
| stationlinkagefrom    | Station linkage from                  | Integer | 0 -         |  |
| stationlinkageto      | Station linkage to                    | Integer | 0 -         |  |
| coderulestring        | Code Rule                             | String  |             |  |
| attribute_14          | Seite der zugeordneten Materialfläche | Integer | 0 – 3       |  |
| workheight            | Work height                           | Integer | 0 – 5       | 0 = Undefined<br>1 = very low<br>2 = low                   |



| Attribute      | Meaning         | Type   | Value Range | Comment                                     |
|----------------|-----------------|--------|-------------|---|
|                |                 |        |             | 3 = medium<br>4 = height<br>5 = very height |
| maxcar         |                 | Bool   | 0/-1        |   |
| productionline | Line identifier | String |             |   |

**Table 48: TSA (analog to process): ergocompprocessdefault**

| Attribute             | Meaning                            | Type    | Value Range | Comment   |
|-----------------------|------------------------------------|---------|-------------|---|
| name                  | Name                               | String  |             |   |
| nameshort             | Number                             | String  |             |   |
| validtime             | Vaild time                         | Integer |             | Unimportant in V5 environment..   |
| time                  | Time [s]                           | Float   |             | Used in case the attribute calctime is not set.                                     |
| calctime              | Calculated Time [s]                | Float   | 0 -         | Used in case the attribute calctime is set.   |
| coderulefrequency     | Code rule frequency                | Float   | 0 -         |   |
| valueaddingpercentage | Percentage of Value Adding[%]      | Float   | 0 -         |   |
| attribute_viale_1     | Is Automatic process               | Bool    |             | true = automatic process<br>false = manual process                                  |
| carbodyposition       | Arbeitsbereich                     | String  | 0 -         |   |
| stationlinkagefrom    | Station linkage from               | Integer | 0 -         |   |
| stationlinkageto      | Station linkage to                 | Integer | 0 -         |   |
| coderulestring        | Code Rule                          | String  |             |   |
| attribute_14          | Side of the assigned Material Area | String  | 0 – 3       | 0 = Undefined<br>1 = Left<br>2 = Right<br>3 = Left + Right                          |
| workheight            | Work height                        | Integer | 0 – 5       | 0 = Undefined<br>1 = Very low<br>2 = Low<br>3 = Medium<br>4 = High<br>5 = Very high |
| maxcar                | Belongs to Max. Car                | Bool    | 0/-1        | Process counts for Max. Car   |
| productionline        | Production Line for ALB            | String  |             |   |

**Table 49: Part: ergocompproductdefault**

| Attribute | Meaning | Type   | Value Range | Comment |
|-----------|---------|--------|-------------|---------|
| name      | Name    | String |             |         |

| Attribute | Meaning | Type   | Value Range | Comment |
|-----------|---------|--------|-------------|---------|
| nameshort | Number  | String |             |         |

**Table 50: Parts bin: ergocomplantdefault or of this derived**

| Attribute                       | Meaning   | Type    | Value Range | Comment  |
|---------------------------------|---|---------|-------------|--|
| name                            | Name  | String  |             |  |
| nameshort                       | Number  | String  |             |  |
| length                          | Length [mm]                                     | Float   | 0 -         |  |
| width                           | Width [mm]                                      | Float   | 0 -         |  |
| height                          | Height [mm]                                     | Float   | 0 -         |  |
| alb_shelfplanningat<br>tributes | Possible shelves                                | String  | 0 – 63      | The six least significant bits indicate whether the bin fits in shelves 1 – 6. |
| numberoflevels                  | Number of levels of the possible standard shelf | Integer | 0 – 6       |  |

**Table 51: Resource: ergocomplantstation**

| Attribute | Meaning | Type   | Value Range | Comment |
|-----------|---------|--------|-------------|---------|
| name      | Name    | String |             |         |
| nameshort | Number  | String |             |         |

**Table 52: Shelf: ergocomplantdefault**

| Attribute         | Meaning                           | Type   | Range | Comment                                      |
|-------------------|-----------------------------------|--------|-------|--|
| name              | Name                              | String |       |  |
| nameshort         | Number                            | String |       |  |
| length            | Length [mm]                       | Float  | 0 -   |  |
| width             | Width [mm]                        | Float  | 0 -   |  |
| height            | Height [mm]                       | Float  | 0 -   |  |
| isdefaultshelf    | Standard shelf                    | bool   |       |  |
| innerlength       | Inside length of shelf level [mm] | Float  | 0 -   | innerwidth and innerheight still missing     |
| floorheight       | Floor height                      | Float  | 0 -   | Distance of the lowest level from the ground |
| numberoflevels    | Number of levels                  | String | 0 – 6 |  |
| shelflevelheight1 | Height from shelf board 1         | Float  |       |  |
| Shelflevelheight2 | Height from shelf board 2         | Float  |       |  |
| Shelflevelheight3 | Height from shelf board 3         | Float  |       |  |
| Shelflevelheight4 | Height from shelf board 4         | Float  |       |  |
| Shelflevelheight5 | Height from shelf board 5         | Float  |       |  |

| Attribute          | Meaning                   | Type  | Range | Comment |
|--------------------|---------------------------|-------|-------|---------|
| Shelflevelheight6  | Height from shelf board 6 | Float |       |         |
| shelflevelangle1   | Angle of shelf board 1    | Float |       |         |
| Shelflevelangle2   | Angle of shelf board 2    | Float |       |         |
| Shelflevelangle3   | Angle of shelf board 3    | Float |       |         |
| Shelflevelangle4   | Angle of shelf board 4    | Float |       |         |
| Shelflevelangle5   | Angle of shelf board 5    | Float |       |         |
| Shelflevelangle6   | Angle of shelf board 6    | Float |       |         |
| shelflevelprotude1 | Overhang of shelf board 1 | Float |       |         |
| Shelflevelprotude2 | Overhang of shelf board 2 | Float |       |         |
| Shelflevelprotude3 | Overhang of shelf board 3 | Float |       |         |
| Shelflevelprotude4 | Overhang of shelf board 4 | Float |       |         |
| Shelflevelprotude5 | Overhang of shelf board 5 | Float |       |         |
| Shelflevelprotude6 | Overhang of shelf board 6 | Float |       |         |

## 4. Description of Relation with Attributes

**Table 53:** `proc_runningon_plant` Relation between Process and Workstation.

| Attribute              | Meaning   | Type    | Value Range | Comment  |
|------------------------|---|---------|-------------|--|
| nameshort              | Start time   Prozesszeit  | String  |             |  |
| sortindex              | sequence  | Integer | 0 – 999     | in PE 5.7:<br>m_strExternalId  |
| internalbalancing-data | Data stream separated by:<br><br>1. <i>internal model number</i><br><br>2. <i>bIsMasterProcess</i><br><br>3. <i>bUseMasterProcess</i> |         |             | bIsMasterProcess:<br><br><i>IsMasterProcess for Material Planning</i><br><br>bUseMasterProcess:<br><br><i>UseMasterProcess for Material Planning</i> |

**Table 54:** `balancing_proc_isbalancedby_resource` Relation between Process plan and Line

Attention: This relation replaces the relation mc\_balances\_pg

| Attribute              | Meaning   | Type   | Value Range | Comment   |
|------------------------|---|--------|-------------|---|
| internalbalancing-data | Several values are strung together::<br>1. <i>activemodel</i><br>2. <i>mainmodel</i><br>3. <i>modelweight</i> | String |             | This relation replaces the relation mc_balances_pg. |

| Attribute | Meaning   | Type | Value Range | Comment |
|-----------|---|------|-------------|---------|
|           | 4. internal model number<br>5. proc graph check sum |      |             |         |

**Table 55:** process\_mustprecede\_process Precedence relation between processes

| Attribute              | Meaning      | Type    | Value Range | Comment |
|------------------------|--------------|---------|-------------|---------|
| connectionpoint-source | Begin or End | Integer | 0 -         |         |
| connectionpointtarget  | Begin or End | Integer | 0 -         |         |

**Table 56:** process\_mustprecede\_process Precedence relation between TSAs

| Attribute              | Meaning      | Type    | Value Range | Comment |
|------------------------|--------------|---------|-------------|---------|
| connectionpoint-source | Begin or End | Integer | 0 -         |         |
| connectionpointtarget  | Begin or End | Integer | 0 -         |         |

**Table 57:** process\_runsbefore\_process Sequence relation between TSAs

| Attribute              | Meaning      | Type    | Value Range | Comment |
|------------------------|--------------|---------|-------------|---------|
| connectionpoint-source | Begin or End | Integer | 0 -         |         |
| connectionpointtarget  | Begin or End | Integer | 0 -         |         |

**Table 58:** proc\_firstprocesses\_prod Relation between process and plant

| Attribute | Meaning  | Type   | Value Range | Comment                          |
|-----------|----------|--------|-------------|----------------------------------|
| nameshort | Quantity | String |             | in PE 5.7:<br>dbl_defattribute_1 |

**Table 59:** proc\_firstprocesses\_prod Relation between TSA and Part

| Attribute | Meaning  | Type   | Value Range | Comment                                      |
|-----------|----------|--------|-------------|--|
| nameshort | Quantity | String |             | See also: Relation between Process and Part. |

**Table 60:** proc\_usescontainerfrom\_prod Relation between process and container

| Attribute              | Meaning                        | Type   | Value Range | Comment |
|------------------------|--------------------------------|--------|-------------|---------|
| note                   | Different values strung        | String |             |         |
| internalbalancing-data | Number of the product variants | String |             |         |

Table 61: **plant\_provides\_prod** Relation between product and container

| Attribute                | Meaning                 | Type    | Value Range | Comment  |
|--------------------------|-------------------------|---------|-------------|--|
| numberofitems            | quantity                | Integer | 0 -         | Number, part per parts bin   |
| cust_chargepart          | Customer charge part    | Boolean |             |  |
| cust_colourcode          | Color code              | String  |             | Text; must be unique when several relations with different colors are created. |
| cust_containerprinciple  | Container principle     | Integer | 0 -         |  |
| cust_leadpart            | Lead part               | Boolean |             |  |
| cust_numbercontainerpart | Number of bins per part | Integer | 0 -         |  |
| cust_sbf                 | Standard supply form    | String  |             | Example: JIS, JIT, warehouse, or similar                                       |
| cust_tfnumber            | Part family number      | String  |             |  |
| productionline           | Line identifier         | String  |             |  |

Table 62: **proc\_uses\_plant** Relation between process and resource

| Attribute | Meaning | Type   | Value Range | Comment |
|-----------|---------|--------|-------------|---------|
| name      | Name    | String |             |         |
| nameshort | Number  | String |             |         |

Table 63: **proc\_alike\_proc** Relation between processes in various process graphs.

| Attribute             | Meaning               | Type   | Range | Comment  |
|-----------------------|-----------------------|--------|-------|--|
| internalbalancingdata | Various values strung | String |       | This attribute is in all relations in PE 5.12! |

Table 64: **proc\_isalternative\_proc** Relation between alternative processes in the same process graph

| Attribute                | Meaning | Type | Range | Comment |
|--------------------------|---------|------|-------|---------|
| Relation already exists. |         |      |       |         |

Table 65: **albspec\_partsbin\_positioned\_in\_shelf** Relation between a container and its shelf

| Attribute              | Meaning  | Type   | Range | Comment |
|------------------------|--|--------|-------|---------|
| internalbalancing-data | Reference to the internal number of the relation proc_usescontainerfrom _prod, used to unmistakably assign the container | String | 0 -   |         |

For the shelves two new plantypes of the type resource must be created: **Shelf** and **Shelf Library**. **Shelf** must be an allowed child element of **Shelf Li-**

brary and of the plantype for the stations in ALB ("Machine"). The **Container** plantype must be a child of the **Station**.

**Table 66:** balancing\_resource\_consistsof\_resource Relation between Material Area and Container or between Shelf and Container

| Attribute | Meaning                 | Type | Range | Comment |
|-----------|-------------------------|------|-------|---------|
| grapfinfo | Position in the graphic |      |       |         |

For this relation "Display in graphic" must be activated.

### **Saving Layout Relevant Data (Position, Rotation)**

1) Graphic information for display in the layout is saved in the "relationship\_nodes" of the following objects:

Line, Station, Parts bin, and Shelf

2) Graphic info is saved to the following relation: balancing\_resource\_consistsof\_resourceUpdated Plan Types and Properties

## 5. Release and Location in Documentation

**Table 67: Plan Types**

The table below lists new plan types. (*Please refer to the Migration Documentation and Checklist.*)

| Plan Type Name        | Derived From (E5 Base Type/Plan Type) | Nameshort            | Attribute Names Prompt (Attribute name) | Release Created |
|-----------------------|---------------------------------------|----------------------|---|-----------------|
| Manufacturing Program | Process Component                     | manufacturingprogram | Link To Drawing (link_to_drawing)       | 18              |
|                       |                                       |                      | NC Program (mfg_output_file)            | 18              |
| Manufacturing Setup   | Process Component                     | manufacturingsetup   | Link To Drawing (link_to_drawing)       | 18              |
| Assembly Station      | Process Component                     | assemblystation      | Link To Drawing (link_to_drawing)       | 18              |

**Table 68: Properties**

The table below lists plan type set properties and the location within the documentation of their description.

| Item      | Property                | Release Created | Location in Documentation   |
|-----------|-------------------------|-----------------|---|
| Attribute | Own rights              | 16              | <i>Please refer to the <a href="#">Unique Keys, Own Rights</a></i>                |
|           | Can have macro assigned | 16              | <i>Please refer to the <a href="#">Unique Keys, Can have macro assigned</a></i>   |
|           | Position                | 16              | <i>Please refer to the <a href="#">Unique Keys, Position</a></i>                  |
|           | Extended copy           | 18              | <i>Please refer to the <a href="#">Unique Keys, Extended Copy</a></i>             |
|           | Exclude from MCM        | 18              | <i>Please refer to the <a href="#">Manufacturing Change Management Manual</a></i> |
|           | Hide in 3DLive          | 18              | <i>Please refer to the 3DLive documentation</i>                                   |



| Item  | Property                               | Release Created | Location in Documentation  |
|-------|--|-----------------|--|
|       | Fixed height                           | 19              | <i>Please refer to <a href="#">Unique Keys, Fixed height</a></i>                           |
|       | Extended Copy for Supplier Integration | 19              | <i>Please refer to <a href="#">Unique Keys, Extended Copy for Supplier Integration</a></i> |
| Group | Attributes per row                     | 16              | <i>Please refer to the <a href="#">Creating a new group, Attributes per row</a></i>        |
|       | Group                                  | 16              | <i>Please refer to the <a href="#">Creating a new group, Group</a></i>                     |
|       | Own rights                             | 16              | <i>Please refer to the <a href="#">Creating a new group, Own rights</a></i>                |
|       | Can have macro assigned                | 16              | <i>Please refer to the <a href="#">Creating a new group, Can have macro assigned</a></i>   |
|       | Position                               | 16              | <i>Please refer to the <a href="#">Creating a new group, Position</a></i>                  |
|       | Alignment                              | 16              | <i>Please refer to the <a href="#">Creating a new group, Alignment</a></i>                 |
| Page  | Own rights                             | 16              | <i>Please refer to the <a href="#">Overwriting pages, Own rights</a></i>                   |
|       | Can have macro assigned                | 16              | <i>Please refer to the <a href="#">Overwriting pages, Can have macro assigned</a></i>      |
|       | Position                               | 16              | <i>Please refer to the <a href="#">Overwriting pages, Position</a></i>                     |
|       | Alignment                              | 16              | <i>Please refer to the <a href="#">Overwriting pages, Alignment</a></i>                    |
|       | Is configured                          | 16sp6           | <i>Please refer to the <a href="#">Editing the properties of types, Is configured.</a></i> |
|       | Exclude from MCM and CCZ               | 16sp6           | <i>Please refer to the <a href="#">Manufacturing Change</a></i>                            |

| Item                  | Property                        | Release Created | Location in Documentation   |
|-----------------------|---------------------------------|-----------------|---|
|                       |                                 |                 | Management Manual   |
|                       | Consider for activation by link | 16              | Please refer to the <a href="#">Properties of parent-child relation</a> , consider for activation by link |
|                       | Loop check group                | 16              | Please refer to the <a href="#">Properties of parent-child relation</a> , Loop check group                |
| Type                  | CCZ Membership                  | 17              | Please refer to the <a href="#">Editing the properties of types</a> , CCZ Membership.                     |
|                       | Log at Project                  | 18              |   |
| Plantype              | CCZ Membership                  | 17              | Please refer to the <a href="#">Editing the properties of types</a> , CCZ Membership.                     |
|                       | Log at Project                  | 18              |   |
| Format                | Name                            | 16              | Please refer to the <a href="#">Properties of Fonts and Colors</a> , Name                                 |
|                       | Size                            | 16              | Please refer to the <a href="#">Properties of Fonts and Colors</a> , Size                                 |
|                       | Style                           | 16              | Please refer to the <a href="#">Properties of Fonts and Colors</a> , Style                                |
|                       | Color                           | 16              | Please refer to the <a href="#">Properties of Fonts and Colors</a> , Color                                |
|                       | Background color                | 16              | Please refer to the <a href="#">Properties of Fonts and Colors</a> , Background color                     |
| Globalattribute-group | Attribute Property ID           | 16              | Please refer to the <a href="#">Global layout design</a>  |
|                       | Priority                        | 16              | Please refer to the <a href="#">Global layout design</a>  |

# List of Figures

|   |    |
|---|----|
| Figure 1: Project Library Section.....                                    | 5  |
| Figure 2: Relation between ergocompbase and subcompitem .....             | 7  |
| Figure 3: Relations in List View.....                                     | 8  |
| Figure 4: Simplified data structure in the DELMIA Process Engineer® ..... | 9  |
| Figure 5: Registration File with Six Values .....                         | 10 |
| Figure 6: Example for the Structure of an RLG File.....                   | 10 |
| Figure 7: German Language Settings – Checkbox Selection .....             | 15 |
| Figure 8: English Language Settings - Checkbox Selection .....            | 15 |
| Figure 9: Global Tab .....  | 18 |
| Figure 10: New Key .....  | 18 |
| Figure 11: New Value .....  | 19 |
| Figure 12: New Value .....  | 19 |
| Figure 13: UserProfileSDMASyncWriteInterval Setting.....                  | 20 |
| Figure 14: User Tab.....  | 20 |
| Figure 15: UserProfileSDMASyncWriteInterval Settings.....                 | 21 |
| Figure 16: Types Schematic .....  | 22 |
| Figure 17: Dialog Layout using Pages and Groups and Attributes .....      | 22 |
| Figure 18: Types and Plantypes in the DELMIA Process Engineer® .....      | 23 |
| Figure 19: Creating a New Plantype Set for System Elements.....           | 26 |
| Figure 20: Creating a New Plantype Set .....                              | 26 |
| Figure 21: Plantype Set Properties .....                                  | 27 |
| Figure 22: Creating a Resource Plantype .....                             | 27 |
| Figure 23: Plantype Properties (Technical Resource) .....                 | 29 |
| Figure 24: Editing a Plantype .....                                       | 32 |
| Figure 25: Example for a Hierarchical Structure.....                      | 33 |
| Figure 26: Linking Plantypes using “Drag and Drop“ .....                  | 34 |
| Figure 27: The Completed Plantype Set Example.....                        | 34 |
| Figure 28: Plantype Set Selection .....                                   | 35 |
| Figure 29: Calling the Plantype Editor in the General Library .....       | 35 |
| Figure 30: Configuration Tool Dialog with Selected Plantype Set.....      | 36 |
| Figure 31: Creating a New Attribute via the Context Menu.....             | 38 |
| Figure 32: New Folder for Attributes .....                                | 38 |
| Figure 33: Plant Properties Example.....                                  | 39 |
| Figure 34: Editing the Plant plantype .....                               | 39 |
| Figure 35: Configuration Tool Dialog.....                                 | 40 |

|   |    |
|---|----|
| Figure 36: Creating a New Attribute .....                                 | 40 |
| Figure 37: Properties Dialog with “Example_Site“ Input Field.....         | 41 |
| Figure 38: Creating a New Page via the Context Menu.....                  | 41 |
| Figure 39: Properties of the “Site” page .....                            | 42 |
| Figure 40: Creating a New Group via the Context Menu .....                | 42 |
| Figure 41: Properties of the “Site” Group .....                           | 42 |
| Figure 42: Assigning the Group/Page to the Attribute Site .....           | 42 |
| Figure 43: The “Plant” Properties Dialog with the “Site” Register .....   | 43 |
| Figure 44: Open the Configuration Tool on the Planttype Factory .....     | 44 |
| Figure 45: Planttype Hierarchy.....                                       | 44 |
| Figure 46: Properties of Attribute Name .....                             | 45 |
| Figure 47: Configuration Tool.....  | 46 |
| Figure 48: Opening the Configuration Manager .....                        | 47 |
| Figure 49: Context Menu .....   | 51 |
| Figure 50: Compare Dialog when Opened for the First Time .....            | 53 |
| Figure 51: Compare Dialog.....  | 54 |
| Figure 52: Context Menu “Sort Types By” .....                             | 55 |
| Figure 53: Context Menu for Types.....                                    | 55 |
| Figure 54: List of Types .....  | 56 |
| Figure 55: Select Type.....   | 56 |
| Figure 56: The Properties of a Type.....                                  | 59 |
| Figure 57: Select Bitmap Dialog.....                                      | 62 |
| Figure 58: Importing a Bitmap.....  | 63 |
| Figure 59: EPBitmap Changer .....   | 64 |
| Figure 60: Properties of the Editors for Context Menu Items .....         | 65 |
| Figure 61: Configuration List of Attributes .....                         | 67 |
| Figure 62: Properties of Attributes – Select Control Type.....            | 77 |
| Figure 63: Create Value List Entry via the Context Menu .....             | 77 |
| Figure 64: Properties for a New Value List Entry.....                     | 77 |
| Figure 65: Properties of Empty Value List Entry – Display in Browser..... | 78 |
| Figure 66: Empty Field for Display of an Attribute in a Combobox.....     | 79 |
| Figure 67: Select Copy in the Context Menu .....                          | 79 |
| Figure 68: Attribute .....  | 79 |
| Figure 69: New Key .....  | 80 |
| Figure 70: Press F1 for More Information.....                             | 80 |
| Figure 71: Creating a New Page via the Context Menu.....                  | 81 |
| Figure 72: Overwriting Pages – Context Menu.....                          | 82 |
| Figure 73: The Properties of a Page .....                                 | 82 |

|  |     |
|--|-----|
| Figure 74: Creating a New Group for a Page .....                                   | 83  |
| Figure 75: The Properties of a Group.....  | 84  |
| Figure 76: Group in a Group .....  | 85  |
| Figure 77: Example for Grouping of Groups.....                                     | 85  |
| Figure 78: Assigning an Attribute to a New Group .....                             | 86  |
| Figure 79: Example.....  | 88  |
| Figure 80: Example of a Configured Attribute .....                                 | 90  |
| Figure 81: Global Attributes .....   | 90  |
| Figure 82: List of Attributes .....  | 90  |
| Figure 83: New Group Entry .....   | 91  |
| Figure 84: Specify the Priority .....  | 91  |
| Figure 85: Parent-Child Relations .....  | 92  |
| Figure 86: New Parent-Child Relations .....  | 92  |
| Figure 87: Selecting a Parent-Child Relation .....                                 | 93  |
| Figure 88: Properties of a Parent-Child Relation .....                             | 93  |
| Figure 89: Create Information for Parent-Child Relation via the Context Menu ..... | 99  |
| Figure 90: Selecting the Parent-Child Information in the Browser .....             | 99  |
| Figure 91: Parent-Child Information .....  | 100 |
| Figure 92: Delete Parent-Child Relations.....                                      | 101 |
| Figure 93: Autorelation.....   | 103 |
| Figure 94: Chain Structure .....   | 103 |
| Figure 95: Autocorrelations Configuration .....                                    | 104 |
| Figure 96: The Type "ergocompproductdefault" .....                                 | 105 |
| Figure 97: Properties of an Assembly before the Changes .....                      | 106 |
| Figure 98: Properties of the Group "Designation and Number" .....                  | 106 |
| Figure 99: The Input Field "Group".....  | 107 |
| Figure 100: The Group "Designation and Number" .....                               | 107 |
| Figure 101: The Group "Designation and Number" .....                               | 108 |
| Figure 102: The Attribute "nameshort" .....  | 109 |
| Figure 103: The Free Attribute "attribute_10" .....                                | 110 |
| Figure 104: Properties of the Attribute "Responsible Planner".....                 | 110 |
| Figure 105: Resource Properties .....  | 111 |
| Figure 106: The Result of the Configuration.....                                   | 112 |
| Figure 107: Display Folder in Project Library .....                                | 113 |
| Figure 108: Display a Context Menu Entry.....                                      | 114 |
| Figure 109: Example of "Inherit Effectivity"; Case 1 .....                         | 116 |
| Figure 110: Example of "Inherit Effectivity"; Case 2 .....                         | 116 |
| Figure 111: Example of "Inherit Effectivity"; Case 3 .....                         | 117 |

|  |     |
|--|-----|
| Figure 112: Example of "Inherit Effectivity"; Case 4 .....   | 117 |
| Figure 113: Open Balancing Configuration .....   | 118 |
| Figure 114: Configuration Tool: Opening the Balancing Process .....  | 119 |
| Figure 115: Open Balancing Configuration .....   | 120 |
| Figure 116: WLB Balancing – Configuration Manager.....   | 120 |
| Figure 117: Work Load Balancing Configuration .....  | 120 |
| Figure 118: Export WLB Configuration.....  | 121 |
| Figure 119: Automatic Synchronization of the Autorelations .....   | 122 |
| Figure 120: Planttype ergocompplantdefault.....  | 122 |
| Figure 121: Organizational Structure for 3 Level Work Load Balancing.....  | 123 |
| Figure 122: Organizational Plantype for the Workplaces.....  | 124 |
| Figure 123: Organizational Node OrgLine .....  | 124 |
| Figure 124: Three Hierarchical Level Structure .....   | 126 |
| Figure 125: Balancing Configuration .....  | 127 |
| Figure 126: Configuration Tool: Editor for Opening the Work Load Balancing in the<br>Manufacturing Concept ..... | 129 |
| Figure 127: Enable Show Tools and Parts .....  | 129 |
| Figure 128: Balancing Configuration .....  | 131 |
| Figure 129: Balancing Configuration, Tab Plantype Forms .....  | 132 |
| Figure 130: Attributes of the Station Resource Type .....  | 133 |
| Figure 131: Process Attributes.....  | 133 |
| Figure 132: New Attribute .....  | 134 |
| Figure 133: Attribute identifierwlb.....   | 136 |
| Figure 134: Attribute Updating Color.....  | 137 |
| Figure 135: Attribute 'balancing_alternating_identifier' .....   | 138 |
| Figure 136: Dialog for Specifying the Number of Workplaces .....   | 139 |
| Figure 137: Message whenever no cycle time has been specified .....  | 140 |
| Figure 138: Attribute Configuration .....  | 140 |
| Figure 139: Balancing - Configuration; Tab Relations .....   | 141 |
| Figure 140: Message if an Error Appears in the Configuration .....   | 142 |
| Figure 141: Work Load Balancing Configuration .....  | 143 |
| Figure 142: Level 1 Work Load Balancing Configuration A .....  | 143 |
| Figure 143: Level 1 Work Load Balancing Configuration B .....  | 144 |
| Figure 144: Level 1 Work Load Balancing Configuration C .....  | 144 |
| Figure 145: Work Load Balancing Started.....   | 144 |
| Figure 146: Choose Configuration .....   | 145 |
| Figure 147: Four Views of Work Load Balancing .....  | 145 |
| Figure 148: Work Load Balancing Structure.....   | 145 |

|   |     |
|---|-----|
| Figure 149: BOM Entry .....   | 146 |
| Figure 150: Link stations with the Subline .....                                | 146 |
| Figure 151: Level 2 Work Load Balancing.....                                    | 146 |
| Figure 152: Level 2 Work Load Balancing Customization A.....                    | 147 |
| Figure 153: Level 2 Work Load Balancing Customization B.....                    | 147 |
| Figure 154: Level 2 Work Load Balancing Customization C.....                    | 148 |
| Figure 155: Level 2 Work Load Balancing for Processes .....                     | 148 |
| Figure 156: Error Message .....   | 149 |
| Figure 157: Create a New Work Load Balancing .....                              | 149 |
| Figure 158: Level 3 Work Load Balancing Customization A.....                    | 149 |
| Figure 159: Level 3 Work Load Balancing Customization B.....                    | 150 |
| Figure 160: Level 3 Work Load Balancing Customization C.....                    | 150 |
| Figure 161: Value list of the attribute carbodyposition .....                   | 153 |
| Figure 162: New Plantype Shelf.....   | 154 |
| Figure 163: Error Message .....   | 155 |
| Figure 164: R18 Data Model – Relations .....                                    | 162 |
| Figure 165: Parent/child-relation for the Workplace.....                        | 163 |
| Figure 166: Activate the Owns Behavior Option.....                              | 163 |
| Figure 167: DNBBehavior – TSA .....   | 163 |
| Figure 168: Logical Part Bin.....   | 163 |
| Figure 169: Logical Part Bin – Shelf.....                                       | 163 |
| Figure 170: Part Bin – Logical Part Bin .....                                   | 163 |
| Figure 171: DBAssistant Tools.....  | 165 |
| Figure 172: Automatic Line Balancing Customization .....                        | 166 |
| Figure 173: Open Balancing Configuration .....                                  | 166 |
| Figure 174: Configuration of Automatic Line Balancing; tab 3 .....              | 167 |
| Figure 175: Schematic Diagram.....  | 168 |
| Figure 176: Automatic Synchronization of the Autorelations .....                | 169 |
| Figure 177: Configuration of Automatic Line Balancing; tab Plantype Forms ..... | 170 |
| Figure 178: Open Balancing Configuration .....                                  | 170 |
| Figure 179: Dialog Configure ALB Attribute .....                                | 171 |
| Figure 180: Deleting a Predefined Attribute .....                               | 171 |
| Figure 181: New Attribute .....   | 172 |
| Figure 182: Attribute Identifieralb .....                                       | 174 |
| Figure 183: Configuration of Automatic Line Balancing; Tab Relations .....      | 174 |
| Figure 184: Default values for Automatic Line Balancing .....                   | 175 |
| Figure 185: Script Action Properties.....                                       | 176 |
| Figure 186: ErrorLogSeverity” Registry Key.....                                 | 178 |



|   |     |
|---|-----|
| Figure 187: Key for UserActivity Logging in the Registry.....                                 | 183 |
| Figure 188: Registry Key for Location of the Log Files .....                                  | 184 |
| Figure 189: Key for Expand Action Tree in the Registry .....                                  | 184 |
| Figure 190: Open EP4 Database Conversion Importing Time analyses from an Oracle Database..... | 191 |
| Figure 191: Open EP4 Database Conversion .....  | 191 |
| Figure 192: Open EP4 Database Conversion .....  | 192 |
| Figure 193: Starting the Conversion via the Menu .....  | 193 |
| Figure 194: Data Import .....   | 193 |
| Figure 195: Value Added Groups and Entries.....   | 194 |
| Figure 196: Choose a Template.....  | 194 |
| Figure 197: Select a Project.....   | 194 |
| Figure 198: Conversion Mode.....  | 195 |
| Figure 199: Data Import Started.....  | 195 |
| Figure 200: Log Details.....  | 196 |
| Figure 201: The “Convert EP4 database” Menu Item.....   | 198 |
| Figure 202: The “EPDBUpdater” window .....  | 198 |
| Figure 203: The “EPDBUpdater” Toolbar .....   | 199 |
| Figure 204: The “EPDBUpdater” Menu Item .....   | 199 |
| Figure 205: The “EPDBUpdater” Context Menu .....  | 200 |
| Figure 206: Connection Parameters .....   | 201 |
| Figure 207: Cleanup Graphic Directory Dialog.....   | 202 |
| Figure 208: Log File for Required Files .....   | 203 |
| Figure 209: Log File for Statistical Data .....   | 203 |
| Figure 210: Path – Log File for Objects that are to be Deleted.....                           | 203 |
| Figure 211: Path – Log File for Objects that have been Deleted .....                          | 203 |
| Figure 212: Neither Backup nor Simulation Activated .....                                     | 204 |
| Figure 213: Starting <i>Cleanup Graphic Directory</i> via Tools Menu.....                     | 205 |
| Figure 214: Activate Simulation in the Dialog.....  | 206 |
| Figure 215: Activating Backup in the Dialog.....  | 206 |
| Figure 216: New Key .....   | 213 |
| Figure 217: Key Value .....   | 213 |

# List of Tables

|  |     |
|--|-----|
| Table 1: Attributes and Information .....                          | 5   |
| Table 2: Table of Values for Language Codes .....                  | 11  |
| Table 3: Minimum Access Rights for Users.....                      | 16  |
| Table 4: Plantype Overview .....                                   | 28  |
| Table 5: Plantype Properties.....                                  | 29  |
| Table 6: Plantype Properties.....                                  | 32  |
| Table 7: Plantype Links.....                                       | 34  |
| Table 8: Tool Bar Description.....                                 | 48  |
| Table 9: Editable Process Engineer Items .....                     | 57  |
| Table 10: Description of Type Properties .....                     | 59  |
| Table 11: Example for Transparent Colors.....                      | 63  |
| Table 12: Context Menu Properties.....                             | 65  |
| Table 13: Properties of an Attribute.....                          | 70  |
| Table 14: Properties of an Attribute.....                          | 72  |
| Table 15: Description of the Properties of a Value List Entry..... | 77  |
| Table 16: Description of the Properties of a New Page .....        | 82  |
| Table 17: Description of Group Properties .....                    | 84  |
| Table 18: Properties of Fonts and Color.....                       | 89  |
| Table 19: Properties of a Parent-Child Relation .....              | 94  |
| Table 20: Description of Properties.....                           | 100 |
| Table 21: Schematic for the Definition of the Display Order .....  | 108 |
| Table 22: Value List Entries .....                                 | 111 |
| Table 23: Subline.....   | 125 |
| Table 24: Usage Folder .....                                       | 125 |
| Table 25: New Attribute Dialog Description.....                    | 134 |
| Table 26: Station Attributes.....                                  | 139 |
| Table 27: Car Body Positions.....                                  | 151 |
| Table 28: New Values.....  | 152 |
| Table 29: Value Calculation for Shelf Levels.....                  | 156 |
| Table 30: Value List.....  | 157 |
| Table 31: Table of New Plantypes .....                             | 159 |
| Table 32: Reference and Instance Objects .....                     | 159 |
| Table 33: Plantypes for processes.....                             | 159 |
| Table 34: Worker – Logical Partsbin .....                          | 160 |
| Table 35: Table of New Attributes.....                             | 161 |

|   |     |
|---|-----|
| Table 36: New Attributes Dialog Description .....   | 172 |
| Table 37: Plantype Sets .....   | 188 |
| Table 38: Editing Process Attributes .....  | 189 |
| Table 39: Manufacturing Concept .....   | 213 |
| Table 40: Line ergocomplantdefault .....  | 214 |
| Table 41: Station ergocomplantdefault* (*was ergocomplanstation).....   | 214 |
| Table 42: Workplace ergocomplantdefault* (*was ergocomplanstation).....   | 215 |
| Table 43: Worker: ergocomplantworker .....  | 215 |
| Table 44: Behavior: ergocompprocessbehavior .....   | 215 |
| Table 45: Logical Part bin: ergocomplantdefault .....   | 215 |
| Table 46: Material Area: ergocomplantarea .....   | 216 |
| Table 47: Process: Ergocompprocessdefault.....  | 216 |
| Table 48: TSA (analog to process): ergocompprocessdefault .....   | 217 |
| Table 49: Part: ergocompproductdefault .....  | 217 |
| Table 50: Parts bin: ergocomplantdefault or of this derived.....  | 218 |
| Table 51: Resource: ergocomplantstation.....  | 218 |
| Table 52: Shelf: ergocomplantdefault.....   | 218 |
| Table 53: <a href="#">proc_runningon_plant</a> Relation between Process and Workstation.....  | 219 |
| Table 54: <a href="#">balancing_proc_isbalancedby_resource</a> Relation between Process plan and<br>Line .....  | 219 |
| Table 55: <a href="#">process_mustprecede_process</a> Precedence relation between processes.  | 220 |
| Table 56: <a href="#">process_mustprecede_process</a> Precedence relation between TSAs .....  | 220 |
| Table 57: <a href="#">process_runsbefore_process</a> Sequence relation between TSAs .....   | 220 |
| Table 58: <a href="#">proc_firstprocesses_prod</a> Relation between process and plant.....  | 220 |
| Table 59: <a href="#">proc_firstprocesses_prod</a> Relation between TSA and Part .....  | 220 |
| Table 60: <a href="#">proc_usescontainerfrom_prod</a> Relation between process and container..  | 220 |
| Table 61: <a href="#">plant_provides_prod</a> Relation between product and container. ....  | 221 |
| Table 62: <a href="#">proc_uses_plant</a> Relation between process and resource .....   | 221 |
| Table 63: <a href="#">proc_alike_proc</a> Relation between processes in various process graphs. .   | 221 |
| Table 64: <a href="#">proc_isalternative_proc</a> Relation between alternative processes in the same<br>process graph .....                           | 221 |
| Table 65: <a href="#">albspec_partsbin_positioned_in_shelf</a> Relation between a container and its<br>shelf .....                                    | 221 |
| Table 66: <a href="#">balancing_resource_consistsof_resource</a> Relation between Material Area<br>and Container or between Shelf and Container ..... | 222 |
| Table 67: Plan Types.....   | 223 |
| Table 68: Properties .....  | 223 |

# Index

## A

|                                 |     |
|---------------------------------|-----|
| ALB                             |     |
| Configuring .....               | 151 |
| Analysis Method .....           | 188 |
| Assigning Bitmaps .....         | 63  |
| Attribute                       |     |
| Generic .....                   | 6   |
| Non-Generic .....               | 6   |
| Attribute Name .....            | 44  |
| Attributes                      |     |
| New attribute .....             | 38  |
| Automatic Line Balancing        |     |
| Carbodyposition .....           | 152 |
| Export .....                    | 175 |
| <b>Import</b> .....             | 175 |
| Loading Standard Values .....   | 174 |
| Material Area .....             | 154 |
| Opening the Configuration ..... | 166 |
| The General Settings Tab .....  | 167 |
| Automotive Line Balancing ..... | 151 |
| Autorelation .....              | 130 |

## B

|                          |    |
|--------------------------|----|
| Bitmaps                  |    |
| 24 Bit Bitmaps .....     | 64 |
| Transparent Colors ..... | 64 |
| Buffer .....             | 29 |

## C

|   |         |
|---|---------|
| Change Management .....                     | 62, 97  |
| Circulation .....                           | 29      |
| Compare .....                               | 54      |
| Starting the Comparison .....               | 54      |
| Configuration Manager .....                 | 48      |
| Update Mode .....                           | 53      |
| Update mode Customer Update .....           | 54      |
| Update Mode Customer Update - Make Entries  |         |
| Invisible .....                             | 54      |
| Update Mode Update All .....                | 54      |
| Configuration Tool                          |         |
| Sort according to Names and Prompts .....   | 56      |
| Configuring Types .....                     | 48      |
| Context Menu                                |         |
| Sort Types By .....                         | 56      |
| Context Menu Editors                        |         |
| Consider for Activation by Link .....       | 98      |
| Defined by .....                            | 98, 100 |
| Description .....                           | 100     |
| Is a Simple Relationship .....              | 98      |
| Must have Unexposed Object for Source ..... | 98      |
| Must have Unexposed Object for Target ..... | 98      |
| Owner Type .....                            | 98      |

|                                     |     |
|-------------------------------------|-----|
| Convert Data .....                  | 198 |
| Creating Customer Data .....        | 28  |
| Creating Plantype Hierarchies ..... | 34  |
| Cycle Time .....                    | 140 |

## D

|                                |     |
|--------------------------------|-----|
| Data Type .....                | 44  |
| Database                       |     |
| DB-Database .....              | 186 |
| DB-Ergotime .....              | 186 |
| Database Settings to EP4 ..... | 201 |
| Defined by .....               | 98  |

## E

|   |        |
|---|--------|
| Editing Plantype .....                          | 32     |
| Editing Types                                   |        |
| Assign Attribute to a New Group .....           | 87     |
| Create New Groups .....                         | 84, 86 |
| Define Properties of a New Page .....           | 83     |
| Defining a Value List Entry to Display an Empty |        |
| Field .....                                     | 79     |
| Deleting Types .....                            | 57     |
| Description of Properties of Editors .....      | 66     |
| Description of Properties of Types .....        | 60     |
| Edit Group Properties .....                     | 85     |
| Edit Page Layout and Groups .....               | 58     |
| Editing Groups of a Type .....                  | 84     |
| Editing Pages of a Type .....                   | 81     |
| Editing the Properties of Types .....           | 59     |
| Editors and List of Menu Items .....            | 57     |
| Editors for the Context Menu .....              | 66     |
| Free Attributes .....                           | 69     |
| List of Attributes .....                        | 57     |
| Overwriting Pages .....                         | 82     |
| Overwriting Types .....                         | 59     |
| Parent-Child Relations .....                    | 58     |
| Types, Editing .....                            | 57     |
| Using Base Types .....                          | 57     |
| Value List for Attributes .....                 | 77     |
| EPBitmapChanger                                 |        |
| Change .....                                    | 65     |
| Prerequisites .....                             | 65     |
| State .....                                     | 65     |
| EPBitmapChanger .....                           | 64     |
| EPBitmapChanger                                 |        |
| Close .....                                     | 66     |
| EPDBUpdater Context Menu .....                  | 200    |
| Ergocompbase .....                              | 8      |
| Examples for the Configuration of               |        |
| Plantypes .....                                 | 39     |
| Examples for the Configuration of Types         |        |
| .....   | 105    |
| Exporting the Database .....                    | 52     |
| Extended <i>Properties</i> of an Attribute      |        |

|                              |    |
|------------------------------|----|
| Change in Release State..... | 77 |
| Target.....                  | 76 |

## F

|                     |          |
|---------------------|----------|
| Fastener .....      | 29       |
| FinderControl ..... | 186, 187 |

## G

|                                       |     |
|---------------------------------------|-----|
| General Finder .....                  | 1   |
| General Properties of a Plantype..... | 30  |
| Graphic Directory                     |     |
| Backup Mode.....                      | 204 |
| General Information .....             | 202 |
| Notes on the Execution .....          | 204 |
| Real Mode .....                       | 204 |
| Simulation Mode .....                 | 203 |
| Staring Function.....                 | 205 |

## I

|                                       |          |
|---------------------------------------|----------|
| <i>Importieren the Database</i> ..... | 53       |
| Input Folder .....                    | 123, 125 |
| Interlaced Transport .....            | 29       |

## L

|  |     |
|--|-----|
| List of parent/Child Relations               |     |
| Is enabled.....                              | 96  |
| List of Parent/Child Relations               |     |
| Is Configured .....                          | 97  |
| List of Parent/Child Relations .....         | 93  |
| Color Selection .....                        | 95  |
| Copy Link to Child.....                      | 97  |
| Copy Link to Child Object (Change Mangement) |     |
| .....  | 97  |
| Copy Link to Child Object (Versioning).....  | 97  |
| Description .....                            | 95  |
| Information .....                            | 97  |
| Inherit Effectivity.....                     | 97  |
| Inheritance.....                             | 95  |
| Is Autorelationpath.....                     | 97  |
| Is Unique for Parent .....                   | 97  |
| Is Unique for Parent and Child.....          | 97  |
| Loopcheck Group.....                         | 95  |
| PPR Relation .....                           | 95  |
| Relation Class.....                          | 95  |
| Show in Graphic.....                         | 97  |
| List of Parent/Chsild Relations              |     |
| Is Relevant .....                            | 96  |
| List of Parent-Child Relations               |     |
| Change in Integrate State.....               | 96  |
| Change in Release State.....                 | 97  |
| Change Protocol .....                        | 100 |
| Child List.....                              | 95  |
| Child Parent Relation .....                  | 95  |
| Child Set.....                               | 95  |
| Child Type .....                             | 95  |
| is Autorelation.....                         | 100 |
| Is Autorelation.....                         | 96  |
| is Unique .....                              | 100 |
| Loop Check .....                             | 100 |
| Parent Type.....                             | 95  |

|             |    |
|-------------|----|
| Prompt..... | 95 |
|-------------|----|

## M

|                                      |        |
|--------------------------------------|--------|
| Manufacturing Change Management .... | 61, 98 |
|--------------------------------------|--------|

## N

|                          |    |
|--------------------------|----|
| New Functions            |    |
| General Information..... | 2  |
| Nonliability .....       | ii |

## O

|                                     |             |
|-------------------------------------|-------------|
| Organizational Object .....         | 24          |
| Organizational Structure Item ..... | 24          |
| OrgProcess.....                     | 29          |
| OrgProduct .....                    | 29          |
| Output Folder.....                  | 123, 125    |
| Overwrite .....                     | 38, 59, 119 |

## P

|  |        |
|--|--------|
| Permissions .....                        | 34     |
| Physical Name.....                       | 44     |
| Plantype.....                            | 5      |
| Plantype Attribute.....                  | 36     |
| Plantype Context Menu                    |        |
| Copy .....                               | 33     |
| Delete .....                             | 34     |
| Edit.....                                | 33     |
| Paste Reference .....                    | 34     |
| Permissions .....                        | 33, 34 |
| Print.....                               | 34     |
| Properties .....                         | 34     |
| Refresh Sort Index .....                 | 34     |
| Reload .....                             | 34     |
| Set Insert Point.....                    | 34     |
| Plantype Properties                      |        |
| Abbreviation.....                        | 30     |
| Bitmap for Graph.....                    | 31     |
| Recursive.....                           | 31     |
| Show in Graph .....                      | 31     |
| Sort Number .....                        | 31     |
| Sorting Criterium .....                  | 31     |
| Top Level.....                           | 31     |
| Type Name .....                          | 30     |
| Use for Work System Components .....     | 31     |
| Plantype Set                             |        |
| Extending.....                           | 188    |
| plantypes_ergoplan_stm_default.ini ..... | 186    |
| Plantype Set                             |        |
| Creating Structures .....                | 27     |
| Plantype Set                             |        |
| plantypes_ergoplan_stm_update.ini .....  | 187    |
| Plantype Set                             |        |
| plantypes_ergoplan_stm_update.ini .....  | 188    |
| Plantypes.....                           | 22     |
| Process Plantypes.....                   | 28     |
| Product Plantypes .....                  | 28     |
| Resource Plantypes .....                 | 28     |

|  |          |
|--|----------|
| Plantypes Area .....                             | 28       |
| Preparing  |          |
| Adjusting ALB for R18 .....                      | 160      |
| Data Model .....                                 | 161      |
| Glossary .....                                   | 159      |
| New Plantypes .....                              | 158      |
| Platype Relationships .....                      | 162      |
| Preparing ALB Platype Sets .....                 | 158      |
| Process .....                                    | 29       |
| Process Platype .....                            | 24       |
| Product .....                                    | 29       |
| Product Platype .....                            | 24       |
| <i>Properties of an Attribute</i>                |          |
| Constraint (09AZ) .....                          | 77       |
| is Generated Key .....                           | 77       |
| is Key .....                                     | 77       |
| Properties of types                              |          |
| Name .....                                       | 62       |
| Properties of Types                              |          |
| Attributes per Row .....                         | 60       |
| Base Class Type Name .....                       | 60       |
| Bitmap .....                                     | 60       |
| Caption Plural .....                             | 60       |
| Caption Singular .....                           | 60       |
| Defined by .....                                 | 61       |
| Depends on Values .....                          | 61       |
| Description .....                                | 61       |
| Dialog Type .....                                | 61       |
| Display Parent/Child Relations in Dbeditor ..... | 61       |
| Has Attachments .....                            | 61       |
| Has Versions .....                               | 61       |
| is Abstract .....                                | 62       |
| is Configured .....                              | 62       |
| is Relevant .....                                | 62       |
| is Searchable .....                              | 62       |
| Own Rights .....                                 | 62       |
| Physical Name .....                              | 63       |
| Special Attribute .....                          | 63       |
| <b>R</b>   |          |
| Resource .....                                   | 29       |
| Resource Platype .....                           | 24       |
| <b>S</b>   |          |
| Specifying Relations .....                       | 58, 93   |
| Station .....                                    | 29       |
| Subcompitem .....                                | 7        |
| Subline .....                                    | 123, 124 |

**T**

|                                      |     |
|--------------------------------------|-----|
| Technical Object .....               | 24  |
| Technical Structure Item .....       | 24  |
| Terminate PPR Master .....           | 209 |
| Terminate PPR Server Processes ..... | 208 |
| The EPDBUpdater Menu Item            |     |
| None .....                           | 200 |
| Toggle Selection .....               | 200 |
| Workstation Components .....         | 200 |
| Time Analyses .....                  | 186 |
| Transport .....                      | 29  |

**U**

|                    |          |
|--------------------|----------|
| Update Mode .....  | 53       |
| Usage Folder ..... | 123, 125 |

**V**

|  |     |
|--|-----|
| Views in the Work Load Balancing ..... | 130 |
|--|-----|

**W**

## WLB

|  |     |
|--|-----|
| – Configuration Dialog                             |     |
| Enable Locking .....                               | 130 |
| Enable Show Parts .....                            | 129 |
| Enable Show Partsbinmanagement .....               | 130 |
| Enable Show Tools .....                            | 129 |
| Enable Toolmanagement .....                        | 130 |
| Attribute List .....                               | 171 |
| Available Configuration .....                      | 120 |
| work load balancing                                |     |
| Attribute <i>balancing_alternating_color</i> ..... | 137 |
| Attribute <i>balancing_update_color</i> .....      | 136 |
| Work Load Balancing                                |     |
| Number of Workplaces .....                         | 138 |
| Work Load Balancing                                |     |
| Configuring .....                                  | 118 |
| Configuring .....                                  | 128 |
| Work Load Balancing                                |     |
| Attribute <i>balancing_number_of_workplaces</i> .. | 138 |
| Work System View .....                             | 50  |