



HOME

User Manual

DELMIA Process Engineer®

Printing Manual



Foreword

This manual provides an introduction to the print functions in the Process Engineer.

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

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

No Liability or Guarantee

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

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

Copyright

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

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

Some parts of this software are owned by Unigraphics Solutions Inc. and are copyrighted © 2011. All rights reserved.

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

Modifications

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

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

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

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

© 2001-2011 Dassault Systèmes - All Rights Reserved

Table of Contents

1.	Introduction	1
1.1	How to Use this Manual	1
1.2	Documentation Conventions and Symbols	1
1.3	New Functions in Printing Manual	2
2.	Introduction to Printing	3
2.1	Printing Procedures	3
2.1.1	Start Printing Procedures via File Menu	3
2.1.2	Starting Printing Procedures via Contextual Menu	4
2.1.3	Starting Printing Procedures via Properties Dialog	4
2.2	Printing Types	5
2.2.1	Printing Lists	6
2.2.2	Print Selected Objects	11
2.2.3	Print Graphic Window	14
2.3	Printing Options	21
2.3.1	Using File Menu Options	22
2.3.2	Printing of Tree Structures	26
3.	Introduction to Print Forms	28
3.1	Generic Print Forms	28
3.1.1	Enhancement when Printing Lists	28
3.2	Types of Print Forms	29
3.2.1	Creating Print Forms for Objects	30
3.3	Print Form Designer	31
3.3.1	Editing Print Forms in the Designer	31
3.3.2	User-Defined Variables	33
3.3.3	Tools for Creating Print Forms	33
3.4	Valuating Variables and Fields in Print Forms	40
3.4.1	Defining a Sum Variable	40
3.4.2	Inserting Sum Variables in Tables	41
3.5	Script Variables for Printing	43
3.5.1	Creating Script Variables/Fields/Tables	43
3.5.2	Using Script Variables	44
3.5.3	Using Script Fields	45
3.5.4	Using Script Tables	46

3.6	Script Variables for Print Form	47
3.6.1	Internal Table Set-up for Scripts	48
3.6.2	Table Structuring for Printouts	48
3.6.3	Creating Table for the Designer Printout	49
3.6.4	Graphics Printing in Tables with Script Variables	53
3.6.5	DIN A0 Graphics Printing	55
3.6.6	Scaled Graphics Printing	56
3.7	Printing Setting	57
3.7.1	Page Format Settings	57
3.7.2	Printing Hierarchical PPR Structures	58
3.7.3	Inserting Logo and Copyrights in Print Forms	60
4.	Creating Queries from External Databases	61
4.1	Creating Queries via External Data Sources Function	61
4.1.1	Creating New Queries	63
4.1.2	Determining Type of Report	64
4.1.3	Assigning Data for PrintOut	65
4.1.4	Using Syntax for Design	66
4.1.5	Editing or Deleting Queries	70
4.2	Reporting on Data Structures - Printing Tables	71
4.2.1	Object Wizard	72
4.2.2	Author Print Object View	73
4.2.3	Starting the Object Wizard function	78
4.2.4	Designing the Pages of the Data Structure in the Input Window	80
4.2.5	Showing Relationship in Object Wizard	84
4.2.6	Creating a Print Form for the Report	87
	List of Figures	92
	List of Tables	96
	Index	97

1. Introduction

This manual explains how to use the Process Engineer Print functions for your planning purposes.

1.1 How to Use this Manual

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

- How to create and edit print forms
- How to use the designer for print forms
- How to start print functions and use script variables



Note

When handling the basic printing 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.

-  This symbol marks the individual operational steps involved in a particular operating instruction. Operating instructions describe operational steps, for example, how to open a menu or execute a function.
-  This symbol marks listed subjects. The symbol for listed subjects can be either used to structure a continuous text or to list main subject keywords.
-  This symbol marks list inside a bulleted or numbered list.
-  This symbol marks cross reference information that is available in another manual.

1.3 New Functions in Printing Manual

No new functionality has been added for this release.

2. Introduction to Printing

The following section explains how to start the printing process and which options are available for editing the print forms in the DELMIA Process Engineer.

A Designer helps in layout of forms that enables easy editing of the forms.

Print Forms

You can call the Designer in the DELMIA Process Engineer using the menu **Tools < Print forms**. There you can select the type of form you need as well as the respective editing mode (create a new form or open an existing form). When working with objects, you can query for a print context, which more closely defines the print attributes.

The Designer interface displays a window with the **List of variables**. Here you can distinguish **Variables** and **Fields**. **Variables** are data fields that are filled be filled once per page. **Fields** are filled several times on one page with various types of information.

Printing

Forms can either be send directly to the printer or displayed as a preview by the DELMIA Process Engineer®. There is also an option to send a form by E-mail from the preview mode. The recipient must have the respective viewer installed, however.

The printouts can also be exported (HTML, JPG, PDF...). Simply select the appropriate export filter from the print settings.

2.1 Printing Procedures

There are three different ways to start a printing procedure in the DELMIA Process Engineer.

2.1.1 Start Printing Procedures via File Menu

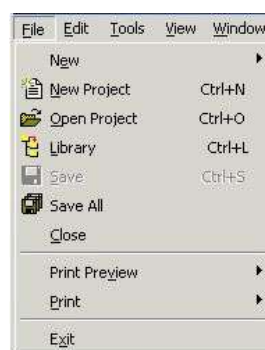
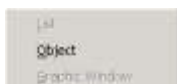


Figure 1: Print using the File Menu

Print Preview

In the **Print Preview** menu, you can perform the following actions, depending on your selection:

- **List:** Display a list (list view) in a preview window
- **Object:** Display an object in the preview window
- **Graphic Window:** Display a graphic window in a preview window



Print

The following options are available in the **Print** menu:



- **List:** Printing a list (list view)
- **Object:** Printing an object
- **Graphic Window:** Printing a **Graphic** window

2.1.2 Starting Printing Procedures via Contextual Menu

Objects can be printed in the DELMIA Process Engineer using the contextual menu.

- 1) Right-click object and select **Print** menu item from the contextual menu.
- 2) You can open Print Preview using the **Preview** menu item.
- 3) You can send your printouts to a printer using the **Print** menu item.

A print form must first be created in order to print these objects. Otherwise, you can get a message stating that the print forms for this object are not available.

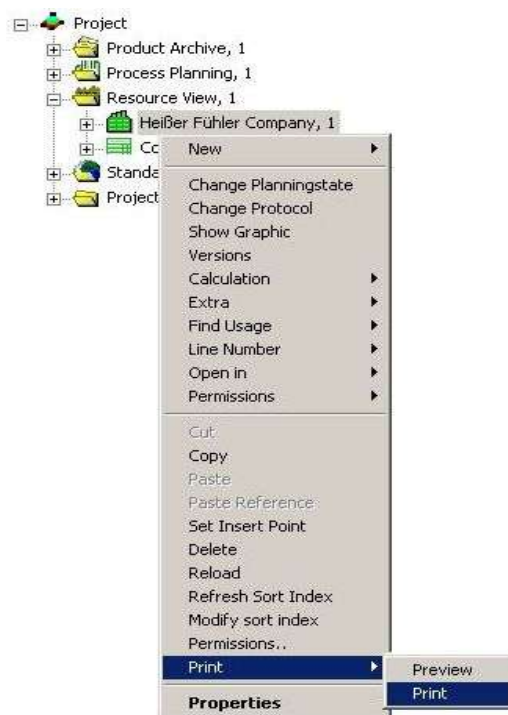


Figure 2: Print Contextual Menu

2.1.3 Starting Printing Procedures via Properties Dialog

When you open **Properties** dialog of a process the **Preview** and **Print** buttons become available. Using this button, you can print the content of a dialog.

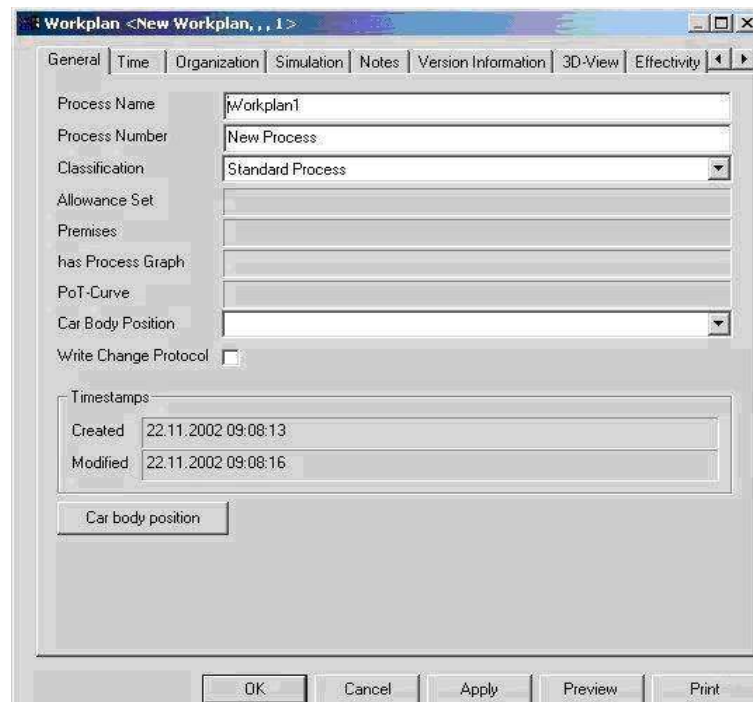


Figure 3: Properties Dialog of a Process

The (Figure 4) **Generic List Settings** Dialog opens – where you can specify the pages (Register) of the dialog as well as the individual fields of the page. This dialog is only opened if you enable the **Display printing generic list settings** entry in the **Tools < Settings < Change < Printing** tab. If this entry is not enabled, you receive a print preview or a printout of the corresponding data in the **Properties** dialog. Using **GenericList Settings** you can individually design your printout. Please refer to the Figure 6.

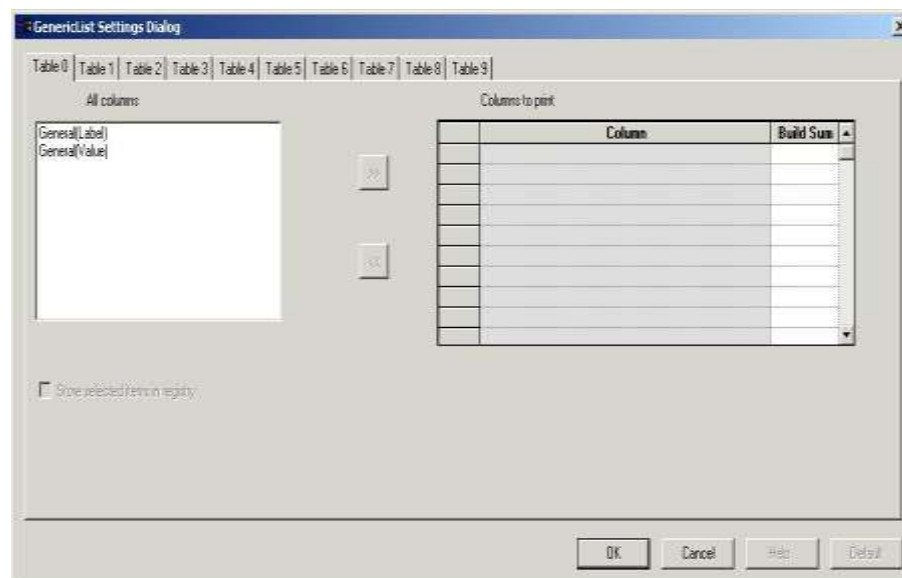


Figure 4: GenericList Settings Dialog

2.2 Printing Types

There are various options in the DELMIA Process Engineer for printing components and lists.

- [Print Graphic Window](#)

- Printing Lists
- Print Selected Objects
- Print Graphic Window

2.2.1 Printing Lists

You can print out lists in the DELMIA Process Engineer®.

- 1) Select a list entry with the left mouse button in the left window (object list) or in Finder. *Please refer to the Figure 5.* The **Print Preview** or **Print** functions for printing lists under the **File** menu appears. If no list entry is enabled, the menu items appears grey and you cannot print out any lists.

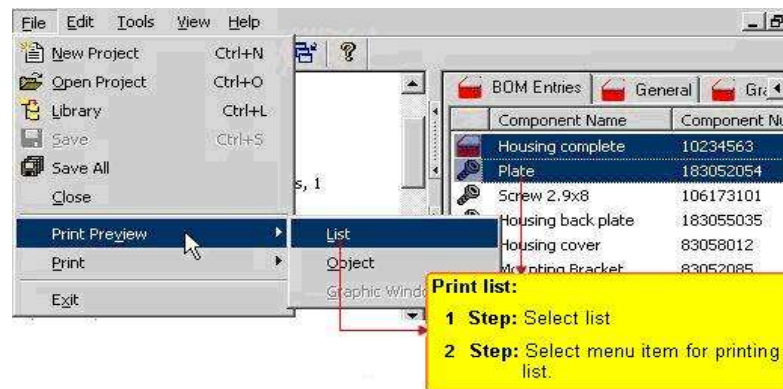


Figure 5: Printing a List

- 2) The **Generic List Settings** dialog opens. Here you can specify your output options. *Please refer to the Figure 6.*



Note

*If you select **Print** or **Print Preview** in the contextual menu, this is interpreted as printing of an object. To print lists, you must always call the menu **File < Print** or **Print Preview**.*

On the left side of the **Generic List Settings** dialog, under **All Columns**, you can find all column names of the object list.

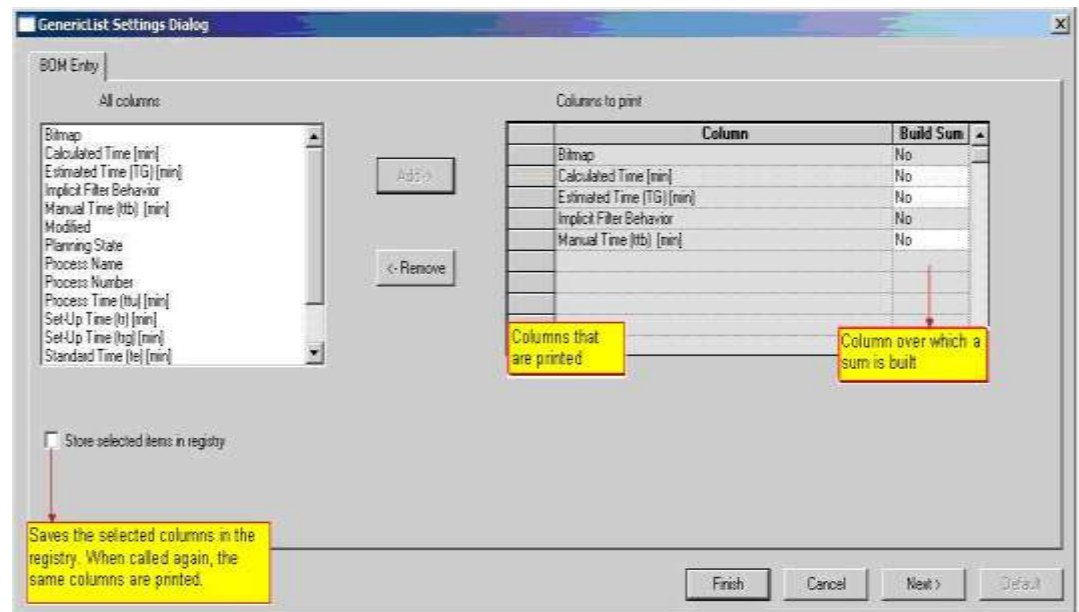


Figure 6: Dialog for Generating Generic List Forms

3) Highlight the column names you want to see on your printout. You can also make a multiple selection.

Add >

4) Enable the button **Add** with the right-pointing arrow. All selected column name entries moves to the right **Columns to Print** table. Only these columns get printed.

< Remove

5) To remove column names from the **Columns to Print** table, enable the **Remove** button with left-pointing arrow.

You can freely define the order of the columns.

→ Bitmap

6) Select a line by choosing the desired line in the first column of the **Columns to Print** table (a black arrow appears).

No
Yes

7) If the last column **Build Sum** of the **Columns to Print** table is not highlighted in grey, the selected column entry is a numeric value that can be used for building sums. If you want to create a sum, open the selection dialog and choose **Yes**. The default does not include the build sum option.

Column width and column sort are found in the window where the list printing gets started. This gives you the option of making changes to the layout of the list printout. *Please refer to the Figure 8.*

8) You can start printing with the **Finish** button. If, however, you want the print-out to be sorted, click **Next** button. A dialog for sorting the list opens.

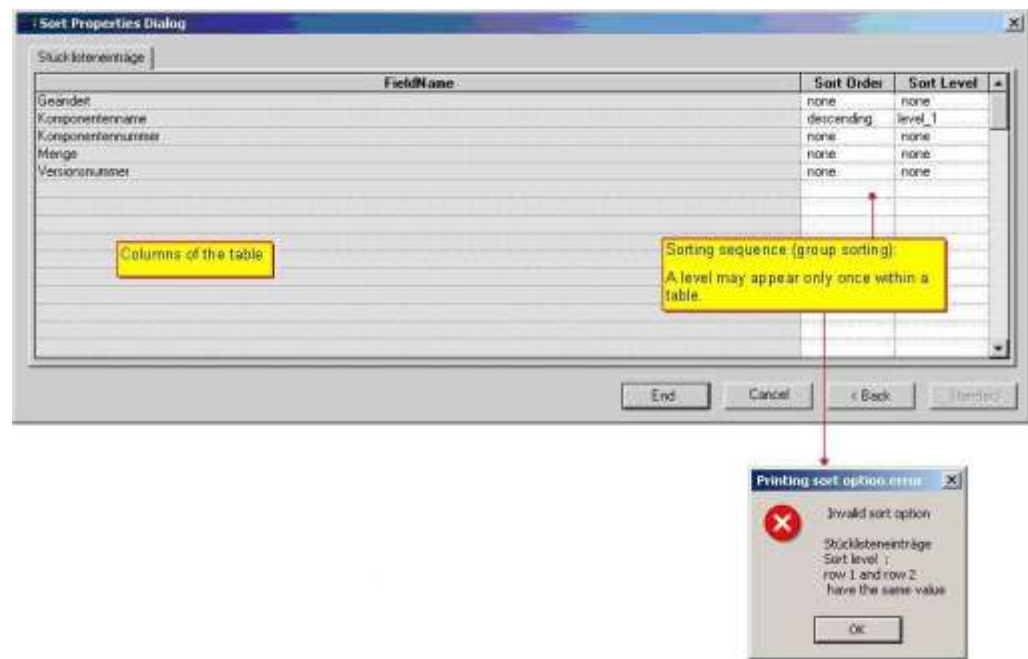


Figure 7: Sorting Table Columns

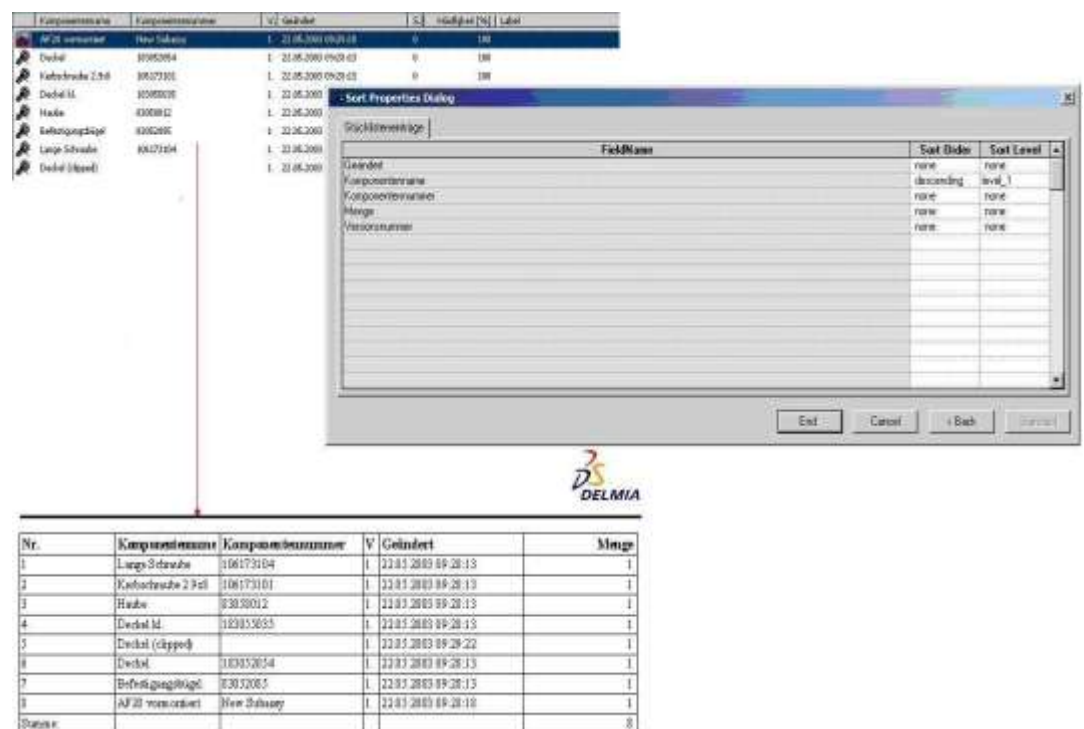


Figure 8: Configuration of List Printing

- 9) Left click between two column titles and hold the mouse button down. The column width can be changed by moving the mouse. This column width is used when the list is printed.
- 10) Columns can also be completely hidden in the same way as just described. The columns are printed according to the following mechanism:

Table 1: Columns Printing

List Printing: Display Column Selection Dialog	Save Selected Entries in the Registry	Response
0	0	Genericlist Settings dialog is not displayed All columns are printed No sum calculation
0	1	GenericList Settings dialog is not displayed If a configuration is found in the settings dialog, this is used for the printout.
1	0	Genericlist Settings dialog is displayed Selected configuration is used for this printout
1	1	Genericlist Settings dialog is displayed Selected configuration is saved and used for other printouts

2.2.1.1 Define Path for Printing of Lists

Using Path Settings you can directly reference a directory and select a print form from this directory or pre-select a print form from this directory as the default.

As standard, the path for printing of lists is set by default. On the basis of this standard setting you can print lists for PPR components without having to define a path or to select a print form. Print immediately starts with the standard form referenced.

- 1) In the **Tools < Settings < Maintenance > Global < Printing < genform-**name you may set a path for printing of lists that deviates from the standard settings.

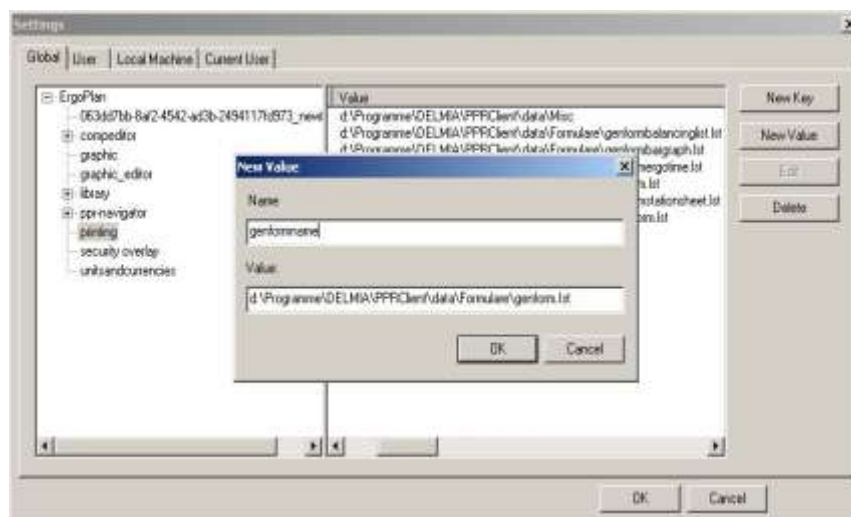


Figure 9: Settings for Printing of Lists – Preset Path

There are two ways of setting your own path:

Set Path for Directory

Used to set a path to the directory for forms to be used for printing of lists.

With the aid of this setting you can define the path to your own directory with print forms. Upon start of a new print transaction, the directory defined opens to allow selecting the print form.



- 2) To set a path, select **genformname** entry.
- 3) Click **Edit** button.
- 4) Enter the path **Value** in the **Edit Value** dialog.
- 5) Confirm the entry with **OK**.

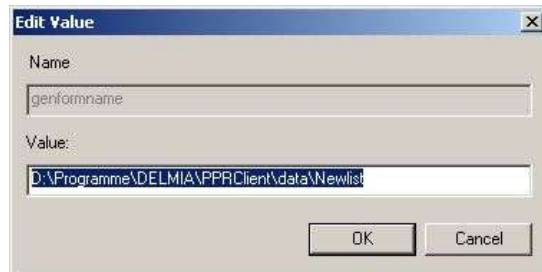


Figure 10: Set Directory Path for Print Forms

- 6) Start the print transaction. The directory set opens. Select and open the print form.

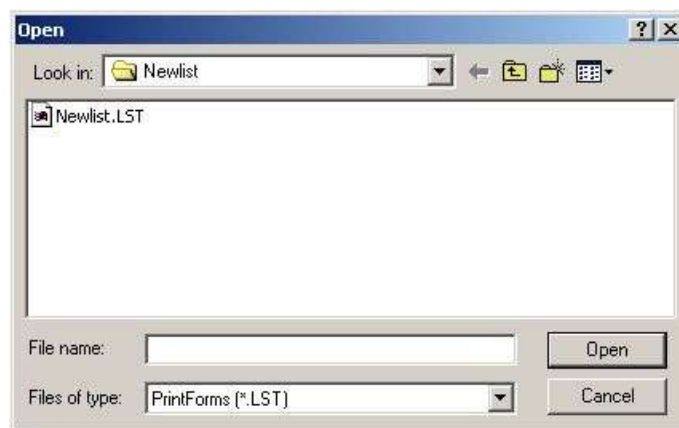


Figure 11: Set Path for Print Forms



Note

If you enter a path that does not exist, a message appears upon start of the print transaction.

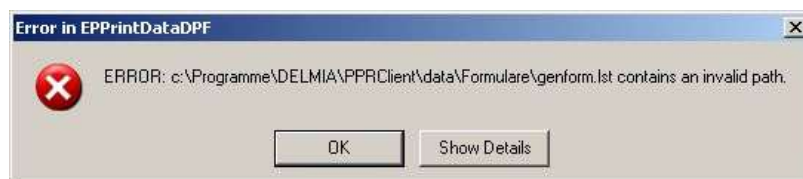


Figure 12: Message –Path Set does not Exist

Set Path and Pre-select Print Form

This is used to set a path to the directory for forms to be used for printing of lists and to pre-select a print form. With the aid of this setting you can define the path to your own directory with print forms and additionally pre-select a print form.

Upon start of the print transaction a directory is not opened, printing of the list starts immediately. This corresponds to the standard setting; however, you have defined a particular print form before.

To change the path settings proceed as discussed in [Set Path for Directory](#). However, you need to specify the print form in addition to the path. This setting allows immediate start of printing without opening a directory.

No Path Set

If you have not set a path under **genformname**, the dialog is opened in the standard directory (PPRClient\bin).

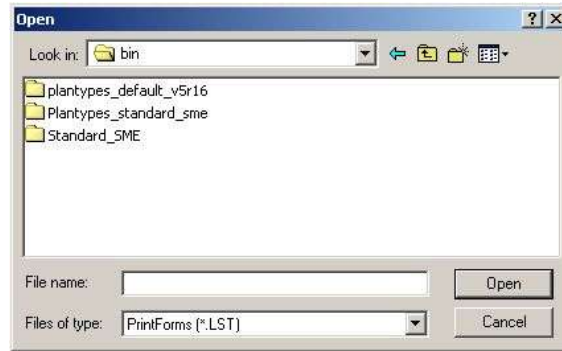


Figure 13: No Path Set



Note

If you do not set a path, the list entry **genformname** is deleted. To re-create the entry for **genformname**, create a new value in the directory **Global < printing:** Name **equal to** **genformname** and as value enter the path to your print forms. Please refer to the [Figure 9](#).

2.2.2 Print Selected Objects

So far it was only possible to print complete list views. It was not possible to print selectively, i.e. to print only selected objects or the list view of the finder.

With the aid of the **ListPrintSelectedObjects** setting in the global settings, you can select printing of objects selected in the listview or printing of the list-view in the finder.

- 1) Make settings under **Tools < Settings < Maintenance < Global < Ergoplan < Printing < ListPrintSelectedObjects**.

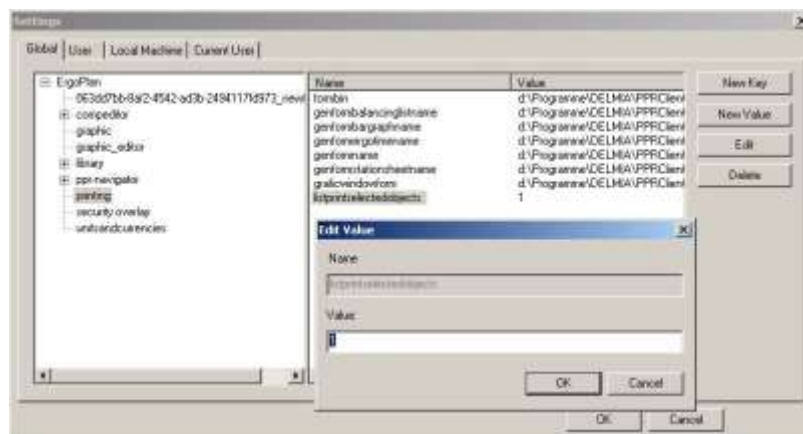


Figure 14: Settings for Selective Printing of Lists

- 2) Enter the value for selective printing of lists.

Table 2: Values for Selective Printing of Lists

Value	Description
1	When the value is set to 1, selected objects may be printed.
0	When the value is set to 0, the list is printed completely.

2.2.2.1 Example for Selective Printing of Lists in Listview

To print lists selectively, select the objects to be printed in the list. Selective printing of lists results only if the value in the global settings has been set to one. *Please refer to the [Table 2](#).*

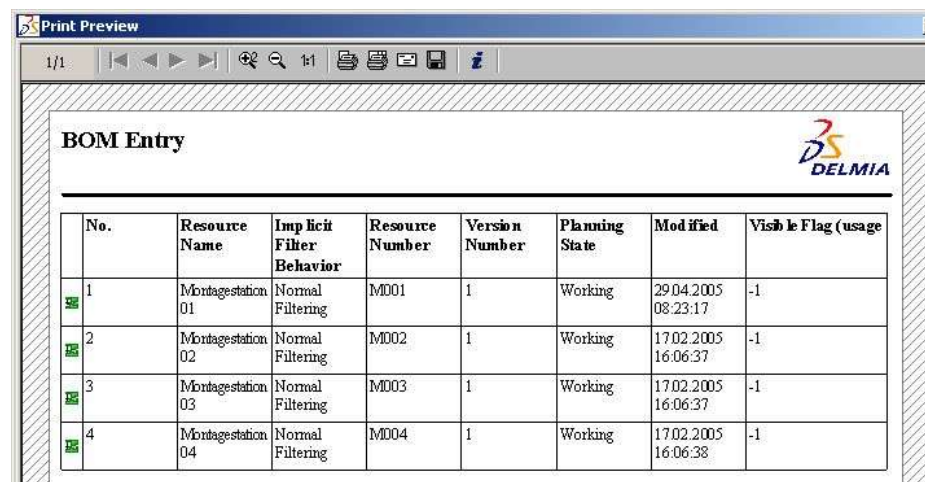
- 1) Select objects in the Listview.



Resource Name	Implicit Filter Behavior	Resource Number	Version Number
Montagestation 01	Normal Filtering	M001	1
Montagestation 02	Normal Filtering	M002	1
Montagestation 03	Normal Filtering	M003	1
Montagestation 04	Normal Filtering	M004	1
Montagestation 05	Normal Filtering	M005	1
Montagestation 06	Normal Filtering	M006	1

Figure 15: Select Objects in the Listview

- 2) Start the print transaction - print preview shown in [Figure 16](#). The print-out corresponds to the selections made in the Listview.



No.	Resource Name	Implicit Filter Behavior	Resource Number	Version Number	Planning State	Modified	Visible Flag (usage)
1	Montagestation 01	Normal Filtering	M001	1	Working	29.04.2005 08:23:17	-1
2	Montagestation 02	Normal Filtering	M002	1	Working	17.02.2005 16:06:37	-1
3	Montagestation 03	Normal Filtering	M003	1	Working	17.02.2005 16:06:37	-1
4	Montagestation 04	Normal Filtering	M004	1	Working	17.02.2005 16:06:38	-1

Figure 16: Selected Objects are Printed – Example Listview

2.2.2.2 Example – Selecting Objects to be Printed in the List View of the Finder

- 1) Select the objects in the list view of the finder – [Figure 17](#) shows the project finder.



Figure 17: Make Selections in the List View of the Finder

- 2) Start the print transaction - print preview shown in Figure 18. The print-out corresponds to the selection made in the list view of the finder.

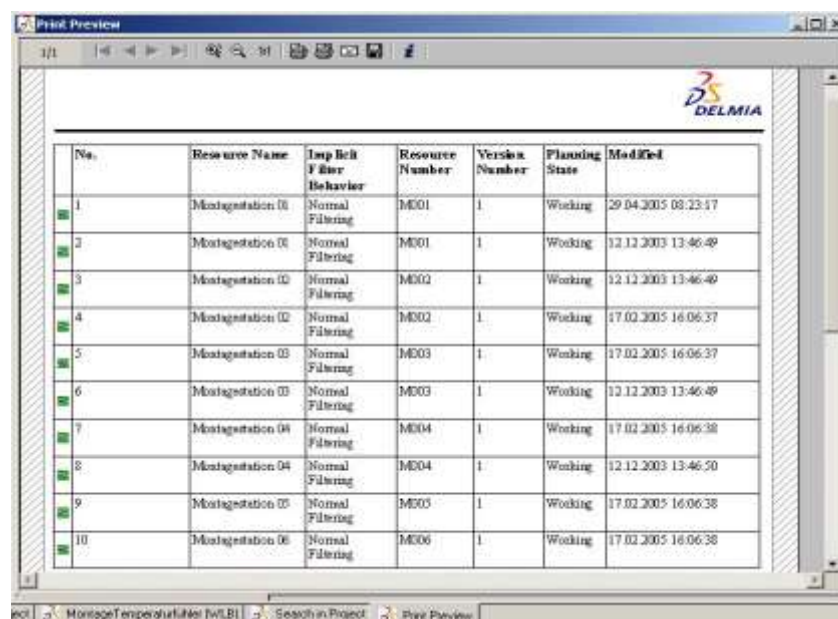


Figure 18: Selected Objects are Printed – Example Finder

2.2.2.3 Path Settings

There are two ways of setting your own path:

Set Path for Directory

Used to set a path to the directory for forms to be used for printing of graphics.

With the aid of this setting you can define the path to your own directory with print forms. Upon start of a new print transaction, the directory defined will open to allow selecting the print form.



Note

If you enter a path that does not exist, a message will call attention to this fact upon start of the print transaction.



Figure 19: Message – Displayed when a Path Set does not Exist

Set Path and Pre-Select Print Form

Used to set a path to the directory for forms to be used for printing of graphics and to pre-select a print form.

With the aid of this setting you can define the path to your own directory with print forms and additionally pre-select a print form.

Upon start of the print transaction a directory is not opened, printing of the list starts immediately. This corresponds to the standard setting, however, you have defined a particular print form before. *Please refer to the [Examples for Set Path for Printing of Graphics](#).*

No Path Set

- If you have not set a path under **graficwindowform**, the dialog is opened in the standard directory (PPRClient\bin).

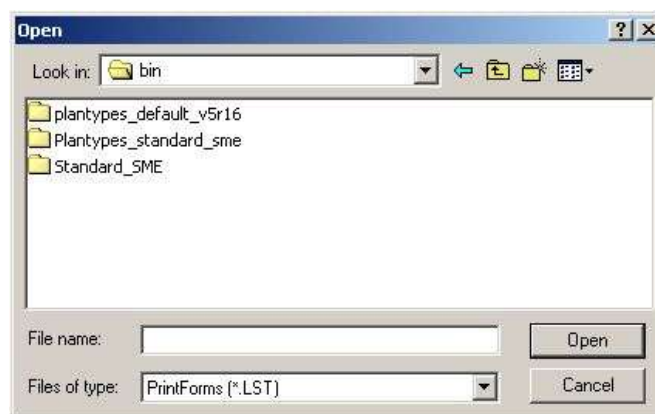


Figure 20: No Path Set

Note

*If you do not set a path, the list entry **graficwindowform** is deleted. To re-create the entry for **graficwindowform**, create a new value in the directory **Global < printing**: Name **equal to** **graficwindowform** and as value enter the path to your print forms.*

2.2.3 Print Graphic Window

No print form needs to be selected in order to print a graphic window. A default form is used in the Process Engineer and its path is set as the default. If printed forms for enabling a different type of display are used, then these forms must be created in the system library for the respective plantype in the Plan Type Set. The path must also refer to the directory in which the print forms are prepared in the **Tools < Settings < Print**.

Note

When making a printout for a graphic window, settings such as those made and used when printing an object for a graphic printout are not needed. The display for the graphic window printout can be done directly in the graphic if

the graphic window was opened using Edit graphics. No changes can be made using display graphics.

2.2.3.1 Using Edit Graphics

The graphic window must be open to generate a printout for a graphic window. The **Edit graphics** function is available for objects of the resource structure and for system elements. In addition to pure graphics using the graphic window function you can also print graphics, for the Process Graph and the Manufacturing Concept.

Change from the PPR Navigator to the resource view. Open the graphics window via the contextual menu **Edit graphics**. In the graphics window, you can change object color or hide floor.



For more information on graphic functions for editing, *please refer to the [Graphic Tools Manual](#)*. For information on Editing graphics for system elements, *Please refer to the [System Library Manual](#)*.

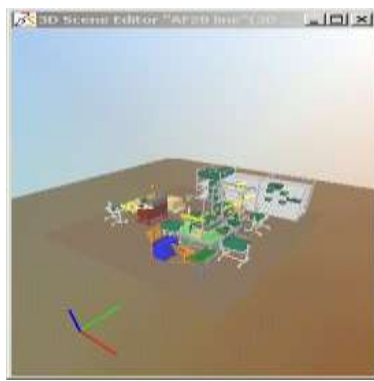


Figure 21: Graphic Window with Adjusted Settings

2.2.3.2 Printing Graphics

- 1) Click on the open graphic window and select either **Print preview** or **Print** < **Graphic window** from the **File** menu. If several print forms should be available, a directory opens from which the respective print form can be selected.

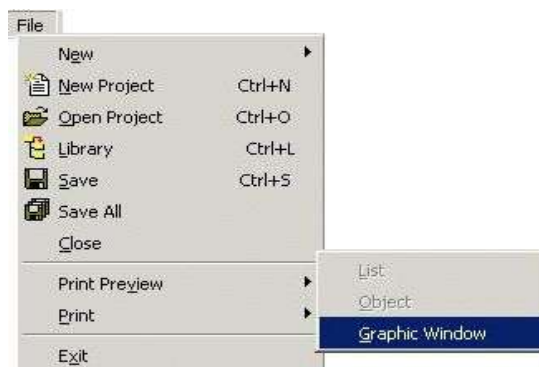


Figure 22: File – Print Menu

- 2) The graphic gets printed according to your selection.

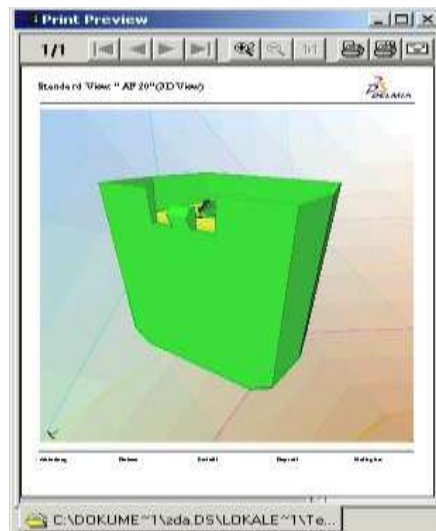


Figure 23: Example of a Graphic Window Printout – Print Preview

2.2.3.3 Examples for Set Path for Printing of Graphics Example - Enter Directory Path

- 1) To set a path, select **graficwindowform** entry.
- 2) Click **Edit** button.
- 3) In the **Edit Value** dialog enter the path **Value**. Confirm the entry with **OK**.

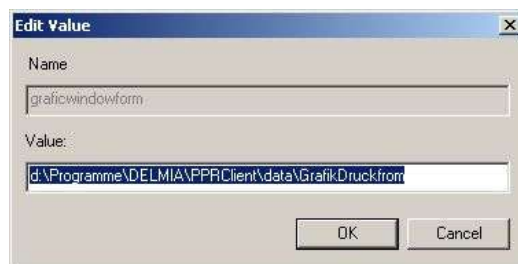
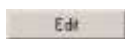


Figure 24: Set Path for Directory

- 4) Start the print transaction. The directory set opens. Select the print form and confirm the selection with **OK**.

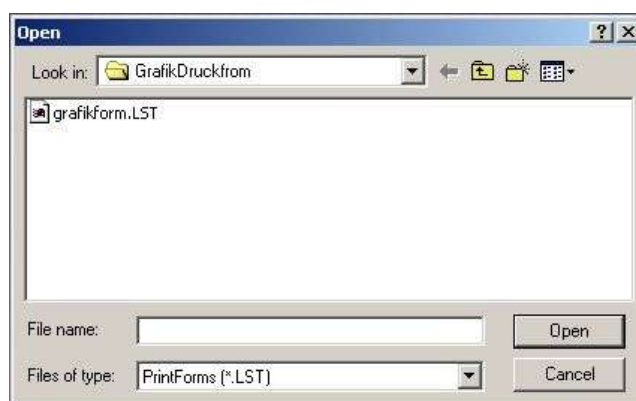


Figure 25: Set Path for Print Forms

Example - Set Directory and Pre-Select Print Form

To change the path setting proceed as outlined in the first example. However, you need to specify the print form in addition to the path. This setting allows immediate start of printing without opening a directory.

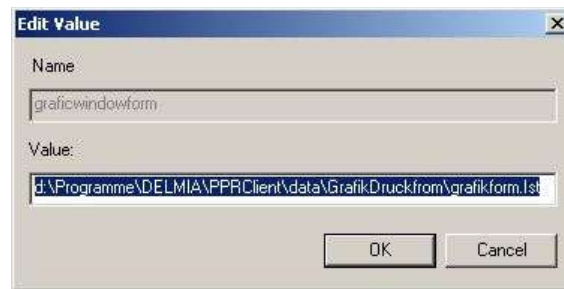


Figure 26: Path Setting with Pre-Selected Print Form

2.2.3.4 Printing Object Graphics

The printing process is started based on the default settings for printing graphics. User-defined settings can be derived from default settings, which you can use individually. Before printing, you can check the printout and correct it in the print preview.

- 1) To start Print preview or Print, select the object and then select either **Print preview** or **Print** via the contextual menu or **Print preview or Print < Object** via the **File** menu.
- 2) If several print forms are assigned to one plantype, you must select the respective print form from the directory. After selecting the print form, click **Open**.
- 3) The directory only appears if several print forms have been assigned to a plantype.

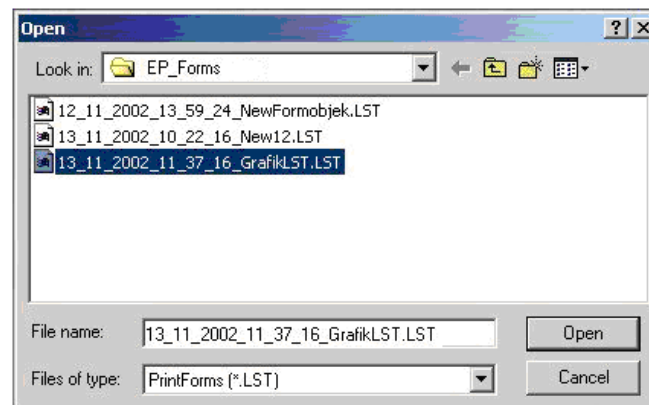


Figure 27: Select Directory for Print Form

- 4) In the **Print Settings** dialog you can either select a default setting for displaying the printout or generate a user-defined setting before a default setting. *Please refer to the [To Make User-Defined Settings](#)*

To Make User-Defined Settings

The **Print Settings** dialog appears after the print preview or the printing process has been started. Select the printer before starting the printing process.

- 5) Select default settings.
- 6) Create user-defined settings and select them for printing graphics.
- 7) Use **Delete** to delete the settings after you have selected the corresponding setting. Use **OK** to start the print preview or the printing process for default and/or user-defined settings.

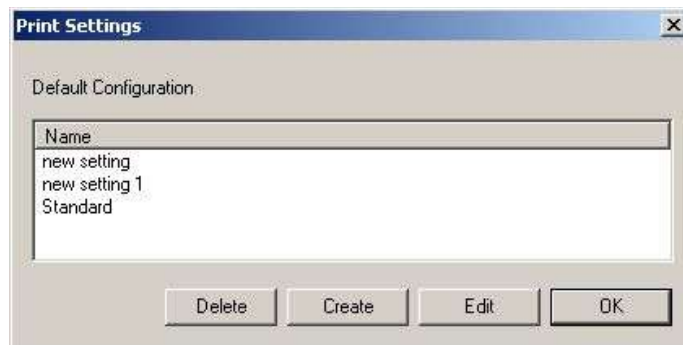


Figure 28: Select Setting for the Printout

To Create User-Defined Settings

- 1) To be able to create your own individual settings and to use them for printing graphics, you have to start either the print preview or the printing process. User-defined settings are always derived from a default setting.
- 2) Select a default setting in the **Print Settings** dialog.
- 3) Click **Edit**. Determine the settings for the printout in the **Print Properties** dialog.
- 4) User-defined settings can be changed at any time. Enable the **Print Properties** dialog using **Edit**.
- 5) Confirm the entries for the user-defined settings with **OK**. The user-defined settings are shown under **Name** in the **Print Settings** dialog.
- 6) For the printout: Afterwards select the user-defined setting and click **OK**. The print preview or the printing process is started.

2.2.3.5 Printing Several Graphics in a Table

Example



Example for printing out an object – selected print form with assigned graphics in a table.

Note

*If a print form contains a table to which several graphics which are to be assigned for print out, then a selection from the **Print settings** dialog must be made for each individual graphic. Again, default settings and user-defined settings can be used.*

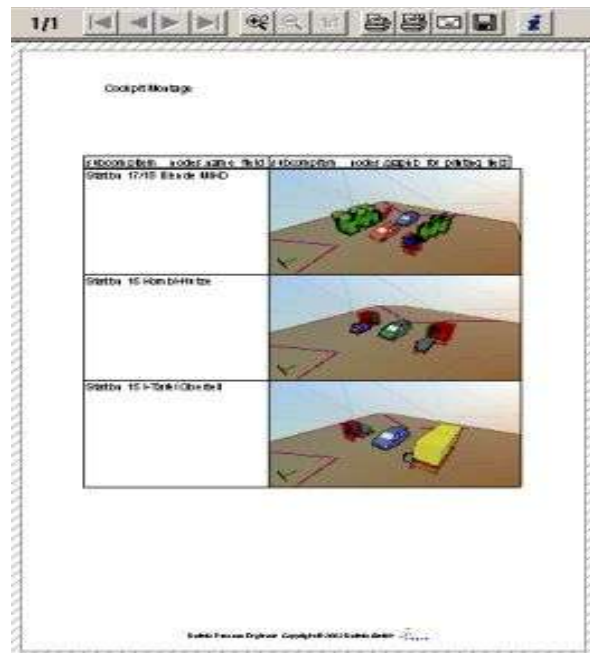


Figure 29: Object Printing – Several Graphics in a Table

2.2.3.6 Printing Graphics of Selected Resources from the List-view

If you wish to get a quick overview of the graphics in the structure that are assigned to a resource, you can apply the menu item **Graphics of selected objects** to the respective parent node and thereby display the graphics selected in the listview. This function is available for selected resources only.

The resources are selected in the listview:

- Listview of the PPR Navigator
- Listview or the resource view if you change to it

To print graphics of the parent node for selected resources, create a print form for object printing for the respective parent node in List and Label; it must allow printing of graphics.

- 1) To be able to print graphics of selected resources, first select the parent node - Assemblyline in the example. Please refer to the [Figure 30](#).
- 2) Select the resources in the listview for which you wish to display graphics – Resources M1 and M2 in the example. Please refer to the [Figure 30](#).

Example

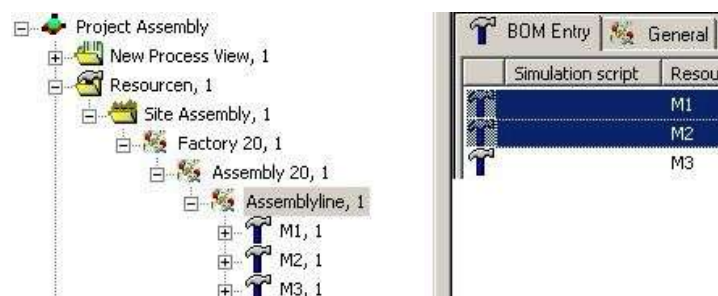


Figure 30: Select Resources in the Listview

- 3) The menu item **Graphics of selected objects** allows to print the graphics of the selected resource; the print preview allows to view the graphics before printing.

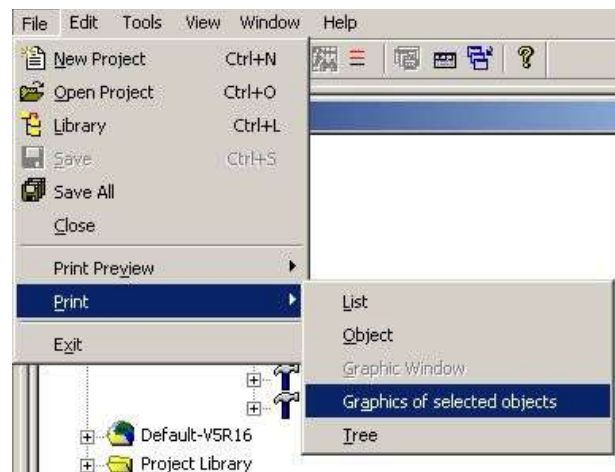


Figure 31: Print Graphics of Selected Resources

- 4) The **Print Settings** dialog allows to select the view – **3d View** in the [Figure 32](#).

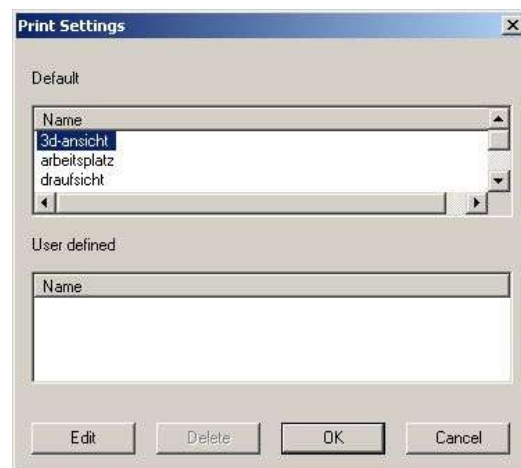


Figure 32: Select View for the Printout

- 5) The print preview displays the graphics accordingly. *Please refer to the [Figure 33](#).*

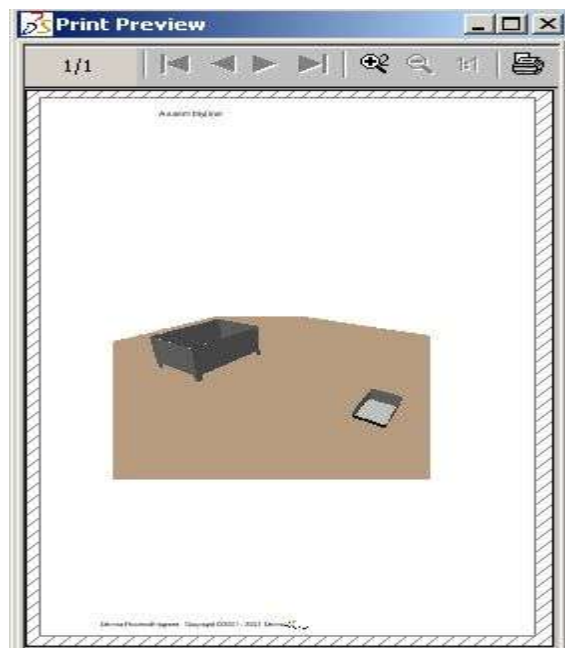


Figure 33: Example - Graphics for Selected Resources

2.3 Printing Options

From PE 5.16. SP6 version, you can print any tree structures from the PPR-Navigator by using the menu item **Print < Tree**. You can start the printing process on any node in the PPR-Navigator. You can print PPR-Navigator structures in two different modes:

- Using List Layout (List and Label)
- Printing Image of Tree Structure

The PPR-Navigator can print only structures. These options do not allow to print PPR structures or PPR components from the Project Library.

The options for the printout are configured in the dialog **Select expand level**. Both printing modes have these options available. The dialog is opened by selecting either the menu item **Print** or **Print Preview**.

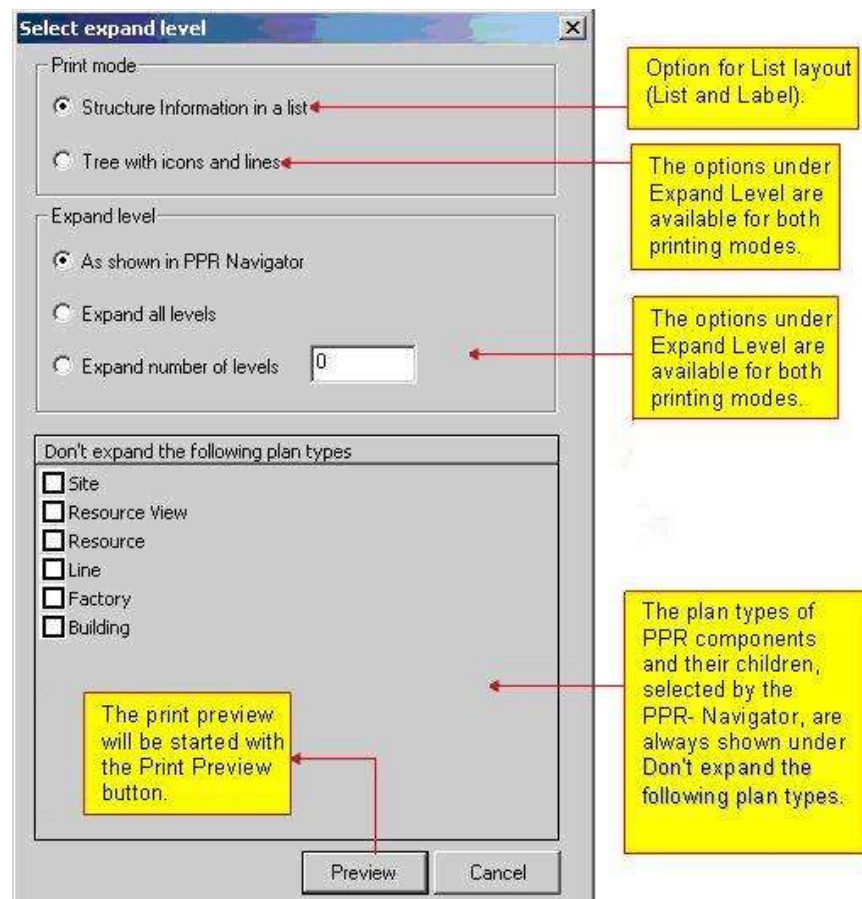


Figure 34: Dialog – Print Option Tree

2.3.1 Using File Menu Options

The dialog **Select expand level** is available under both menu items **Print** and **Print Preview**. Tree structures can be printed directly, using the menu item **Print**, or it can be followed by the **Print Preview**, either way this dialog gets opened.

Printing with the Menu Item Print

If you select the menu item **Print**, printing starts immediately after selecting the corresponding options for the printout in the dialog.

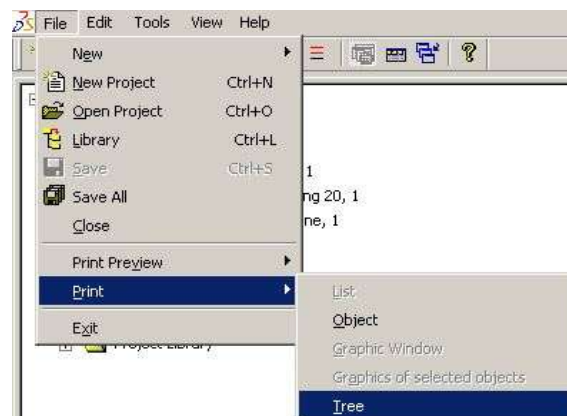


Figure 35: Menu Item Print

Printing with the Menu Item Print Preview

If you select the menu item **Print Preview**, you can select between the options and view the print layout in the Print Preview. The Print Preview shows an individual page for each selected option. The currently selected page can be printed from the Print Preview.

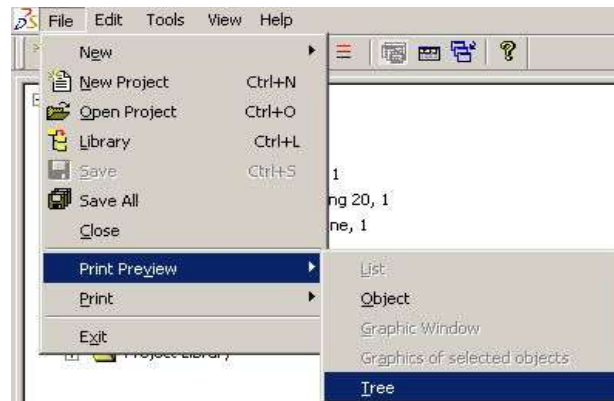


Figure 36: Menu Item Print Preview

2.3.1.1 Select Expand Level Dialog Description

Print Modes

The mode for the print out is selected under the **Print mode** option.. Please refer to the [Figure 34](#).

- **Structure Information in a List Option:** If you select this option, by standard a List layout formula created in List and Label is used for the printout.

Example for List layout with the option **Expand all levels**.

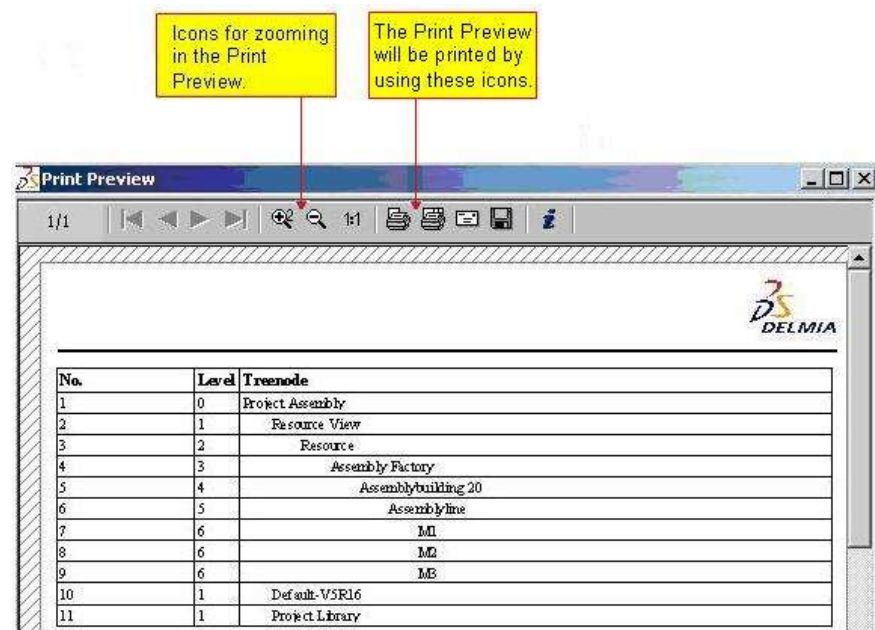


Figure 37: Example for List Layout

Using your own LIST Layout Forms

To use your own List layout forms you have to enter a unique path under **Settings < Maintenance < Global < Printing** in the key **genformtreeprint**, from where your own List layout forms can be selected. Please refer to the [Define Path for Printing of Lists](#).

- **Tree with Icons and Lines Option:** This option copies the printout of the Tree structure of the PPR- Navigator.

Example of using the **Expand all levels** option.

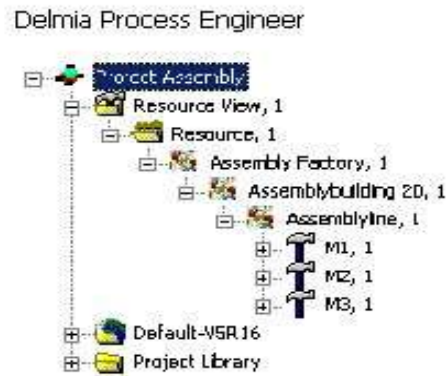



Figure 38: Example of Tree Structure in PPR-Navigator

Expand Level Option

All options listed under **Expand level** are available for both printing modes.

- **As shown in PPR Navigator Option:** If you activate this option, you receive the exact tree structure shown currently in the PPR-Navigator. All options listed under **Do not expand the following plan types** though, cannot be available with the option Expand level.



No.	Level	Treenode
1	0	Project Assembly
2	1	Resource View, 1
3	1	Default-VSR16
4	1	Project Library

Figure 39: Tree structure as shown in the PPR-Navigator

- **Expand all Levels Options:** If you activate this option, all structures from the selected PPR components are opened and shown in the Print Preview. The dialog shows the plan types that need to be considered for the printout.

In example, [Figure 40](#), the PPR component Assemblybuilding is selected (plantype Building). The two plan types Resource and Line belong to the structure of this plantype. All PPR components of these two plan types, which belong to the structure of the Assemblybuilding, are used for the printout.

You can exclude plan types from the printout by combining the options under **Don't expand following plan types**.

For example, if you want only lines to be displayed, activate additionally the option Line: in our example the resources M1, M2, and M3 would not be considered for the printout. It would make sense to activate this option, if a bigger number of lines are assigned to the Assemblybuilding.

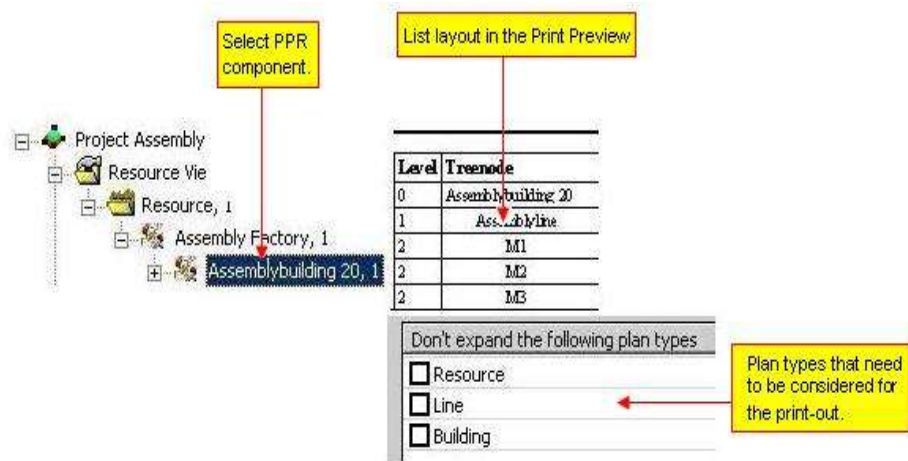


Figure 40: Example for Expand all Levels - List Layout

Expand Number of Levels Option

If you activate this option, all hierarchical levels of a structure that have been entered as values in this option, are considered for the printout – in this example up to the third hierarchical level. The level zero is always equivalent to the selected PPR component in the PPR-Navigator.

The following also applies to this option: You can exclude plan types from the printout by combining the options under **Do not expand following plan types**.

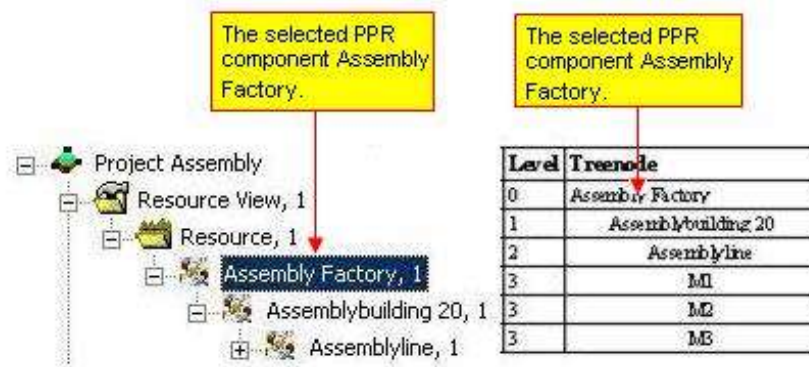


Figure 41: Example Expand Number of Levels

Do not Expand the following Plan Types Option

You can use these options only in combination with the options **Expand all levels** and **Expand number of levels**.

You can exclude plan types from the printout with these options.

In the example the project node is selected - plan types used in the project are shown in the dialog. According to the selected PPR components, plan types can be used for the printout.

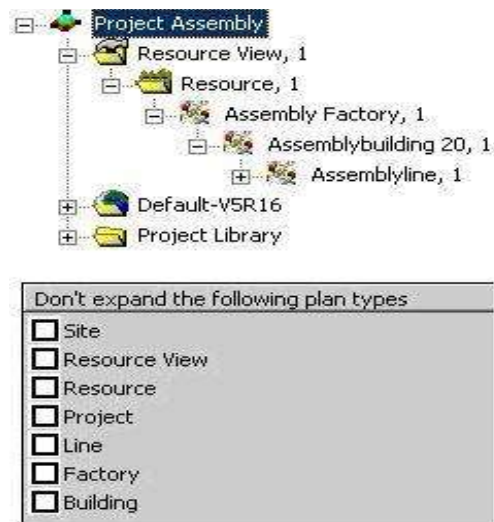


Figure 42: Option for Excluding Plan Types

For example, if you would select all plan types from the dialog, only the project node would be printed/shown in the Print Preview.

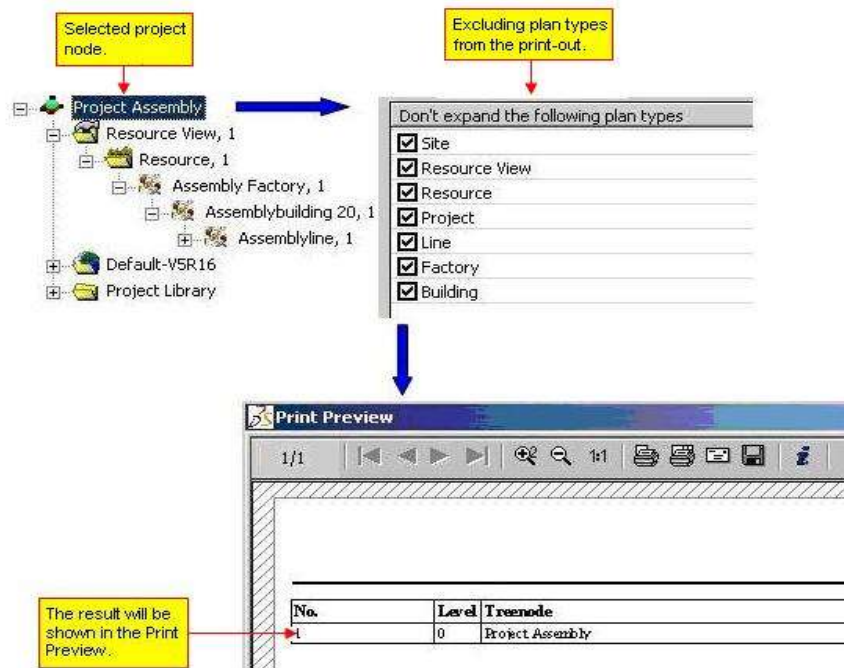


Figure 43: Examples for Excluding Plan Types from the Print Out

2.3.2 Printing of Tree Structures

- 1) To print a structure from the PPR-Navigator, select the PPR component of the structure that you want to print from the PPR- Navigator - in this example it is the project node.
- 2) Select **File < Print < Tree**.

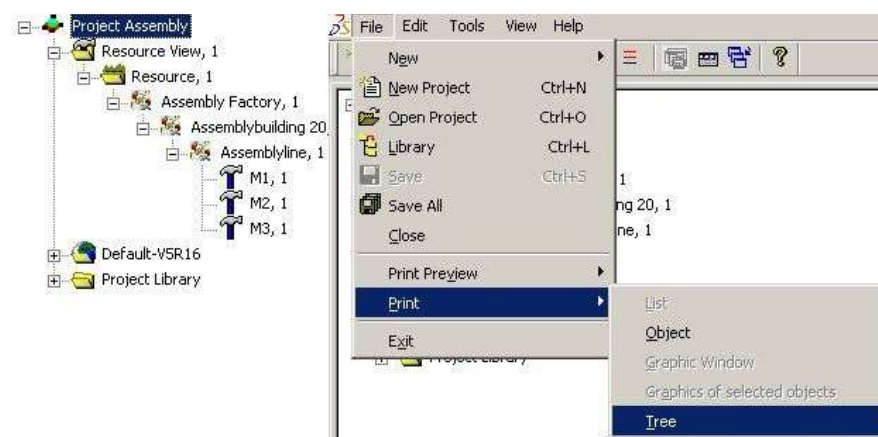


Figure 44: Contextual Menu for Print and Print Preview

- 3) Select the option for the printout in the **Select expand level** dialog.
Please refer to the [Using File Menu Options](#) and [Figure 34](#).

3.Introduction to Print Forms

The **Print form** menu item can be found under the **Tools** menu item. The following functionalities for editing or creating print forms are offered in this menu item.

Table 3: Functionalities for Editing or Creating Print Forms

Menu Items	Description
New list	Creating a new form for printing lists
New object	Creating a new form for printing objects
New graphic window	Creating a new form for printing graphics
Edit list	Edit form for printing lists
Edit object	Edit form for printing objects
Edit graphic window	Edit form for printing graphics
Configuration of graphic	Here you can determine the appearance of the graphic printouts. Depending on the object, you can specify various graphic print configurations
Delete object	All print forms of a selected object are deleted
Script variables	Definition of script variables

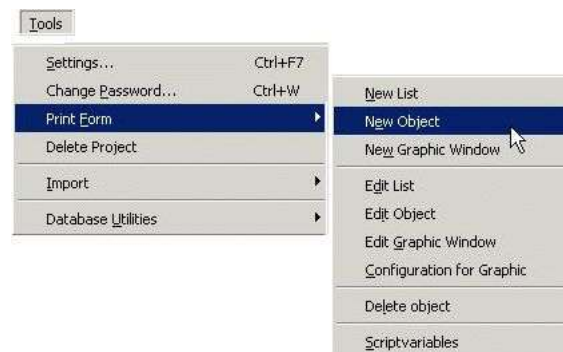


Figure 45: The Print Form Menu

3.1 Generic Print Forms

Generic print forms are forms which, depending on the object selected, are already created temporarily in the DELMIA Process Engineer.

3.1.1 Enhancement when Printing Lists

The columns to be printed can be selected during printing the lists. It is also possible to add a “build sum” for numeric columns. To enable these functions, go to the **Tools < Settings < Printing** and activate the control box **Print list: Dialog for displaying column selection** and **Print record count columns on printing generic lists**.

Variables used from Program Modules

Program modules could previously freely assign the following variables:

var0, var1,... var9.

These variables were permanently registered, the quantity was limited to 10 variables and the variable names could not be changed.

The number of “freely-definable” variables is no longer limited in Version PE 5.10 and the variable names can be freely assigned.



Note

*The variables are read from lists from the "printgenfreevarnames.txt" file when printing. The file can be found in the \DELMIA\PPRClient\data\Misc.directory. In order for the Process Engineer to find the file, the entry **formbin** must also be set accordingly in the Settings/Maintenance Tool dialog. If this entry is not set there, the following error message appears when starting the Process Engineer.*

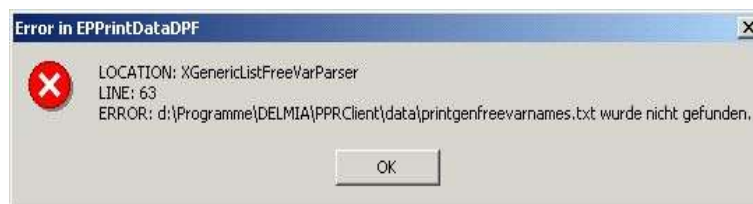


Figure 46: Error Message for Incorrect Path Declaration of Form Binaries

3.2 Types of Print Forms

Just like in the case of the printing types, there are also three types of forms for which a new form can be created.

- **Form for printing types, plantypes, and objects:** A form can be created for each selected object (station), so that the associated attributes can be printed.
- **Generic form for printing lists:** This form is used for printing lists and can be regarded as a form template, from which the program automatically produces a temporary form suited to the list currently selected. The “gen-form.lst” form supplied by DELMIA should be edited to create a form. All parts of this form can be edited, except for the set-up of the table. Only the size and position of the table can be changed.
When you want to create a new list print form, you can import the file „gen-form.lst” into the Designer and make the necessary changes. The file itself remains unedited.
- **Form for printing graphics:** This type of form allows you to print the currently selected view of a graphic.
As in Item 2, the “grafikform.lst” file can be imported and modified when creating graphic print forms.

Modifying Print Forms

This function can be executed only by the Administrator.

If you receive an error message when modifying print forms with this text:

'Can't write form XXX.LST file into DB'

you have to increase the maximum size for the print form objects. It is set by default to 256KB. You should set it to 512KB.

- 1) You can set the value with this SQL script at the user e5_database.

```
delete from OPTIONREGISTRY where classname = 'XDOPrintForm' and membername =  
'm_blobForm' and optionname = 'BLOBSIZE';  
  
insert into OPTIONREGISTRY values ( 'XDOPrintForm', 'm_blobForm', 'BLOBSIZE',  
'524288' );  
  
commit work;
```

Figure 47: SQL Script

- 2) Replace 524288 by an appropriate value until you receive no more errors.
- 3) After setting the size, restart the PPR Server.

3.2.1 Creating Print Forms for Objects

Enhancement for Object Print Forms

The print forms for printing objects were previously stored in the configuration database.

Since Version PE 5.10, the print forms have been stored in the “Data database”. The following basic changes thus arise:

- Print forms are always assigned to a plantype.
- Print forms can now only be created in the system library. A plantype needs to be selected.
- No print forms can be created, edited, or deleted anymore in the Plan Type Editor or in the Configuration Manager.
- The following print-specific entries in the configuration manager are now ineffective or not applicable:
PrintContext
PrintInfo

To Create Print Forms for Objects

Print forms are created in the system library for the Plan Type Set used in the project. Print forms for objects are created for resources. A print form can be created for every plantype of the resource structure in the Plan Type Set.

- 1) Select the resource for which the print form is to be created in the **System library < Planning Type Sets < Plan Type Set < Plan Type Resources**.
- 2) Select **Tools < Print form < New object**.
- 3) Save the new print form in the dialog under a name. After saving the print form, the Designer opens.

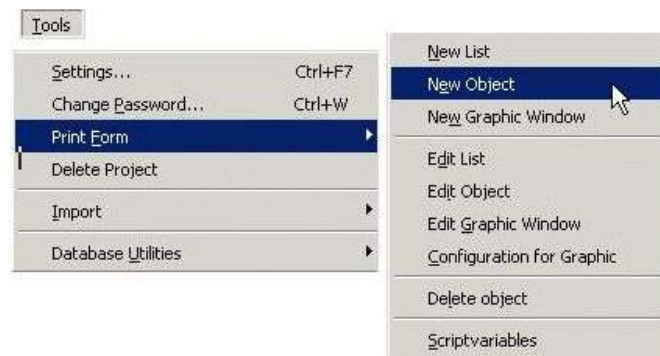


Figure 48: Create a New Print Form for Objects in the System Library

3.3 Print Form Designer

3.3.1 Editing Print Forms in the Designer

The **Designer** is used for form layout. You can call the Designer from the DELMIA Process Engineer using the menu **Tools < Print form**.

There you can select the type of form you need as well as the respective editing mode (create a new form or open an existing form). When working with objects, you can be queried about a print context, which simply defines the print attributes more closely. You can be queried if there is more than one print context. There is only one print context in the current version.

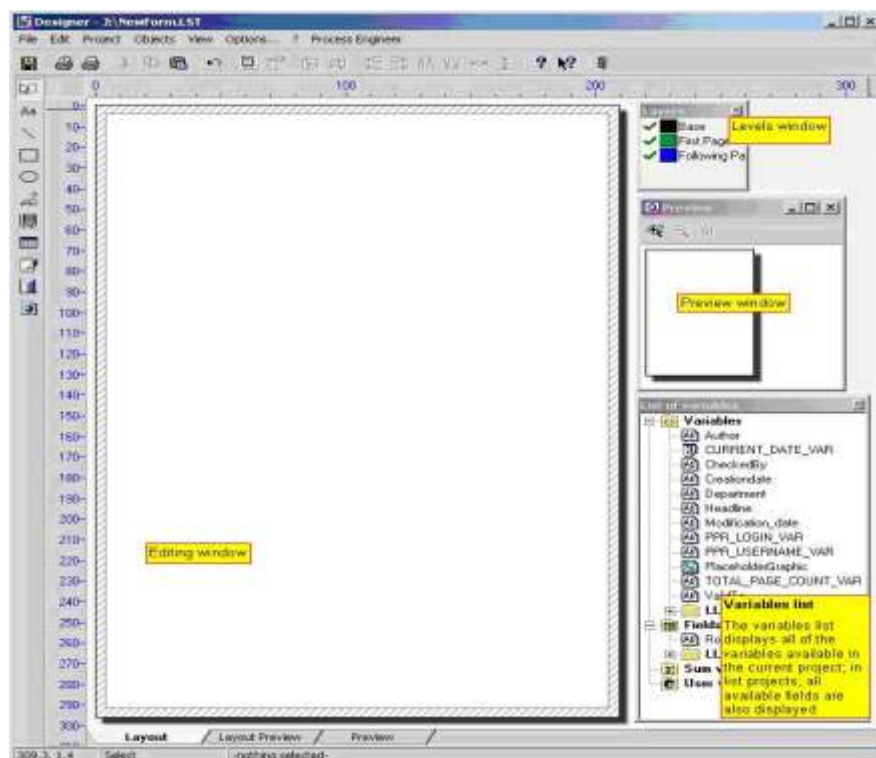


Figure 49: Overview – List & Label – Window

The List of Variables

The Designer interface displays a window with the **list of variables**. This distinguishes between **Variables** and **Fields**.

- **Variables:** The **variables** are data fields that are only filled in once per page.



- **Fields:** There are those **Fields**, which are filled in several times on one page with various types of information.

Caution

Fields can only be in table objects.

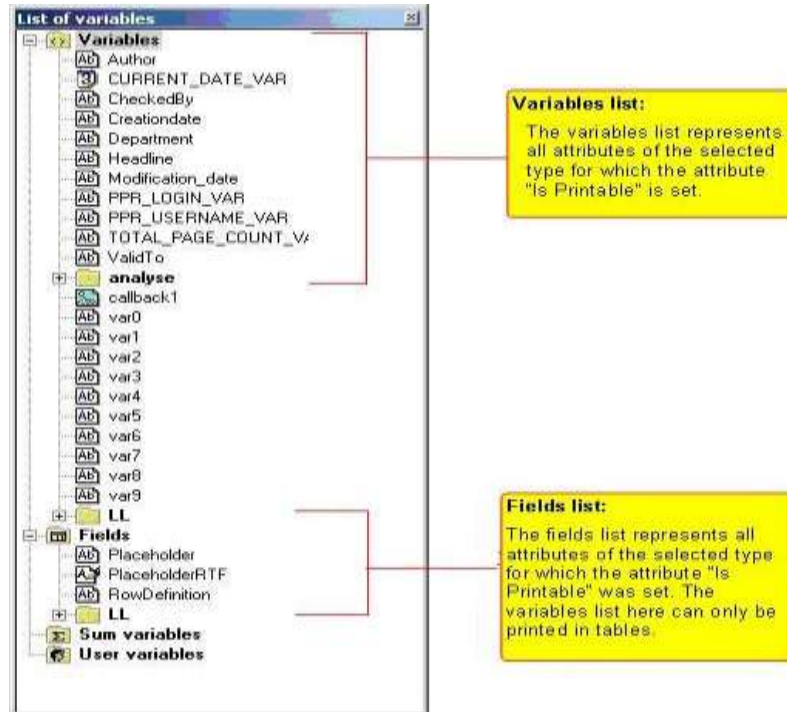


Figure 50: List of Variables

Variables or fields are displayed differently depending on data type. The DELMIA Process Engineer currently supports the following types for printing:

Text

Output of character combinations. One thing to watch out for here is the line wrap. Only whole words can wrap.

Numeric

The output of numeric data is always printed as a double value by default. But the Designer provides formatting so that the data can also be printed out as a whole number (integer). The values surrounded by units are printed out in the configuration as an interface-unit selected unit.

Date

Output of date values The Designer prepares various formats for this purpose (For example, date with day of week and time of day).

Boolean

Displays true or false. Boolean variables and fields cannot be placed directly onto a form. A User-variable must be created to print a Boolean variable.

RTF

Printing out RTF formatted texts as variables and in tables.

DELMIA Graphics

Printing out DELMIA graphics that are attached to an object. DELMIA graphics cannot be printed in tables.

3.3.2 User-Defined Variables

User variables can be defined in the Designer. One thing to keep in mind here is that all variable names that are drawn from the field or variable lists must be placed in brackets.

Example

RIGHT:

(ergocomprocessdefault_proc_running_plant.writechangeprotocol_field) + 1

WRONG

ergocomprocessdefault_proc_running_plant.writechangeprotocol_field + 1

3.3.3 Tools for Creating Print Forms

The following chapter helps you to create your own print forms using the Designer layout tools.

- [Change Page Layout](#)
- [Create Tables](#)
- [Sort Tables](#)
- [Create Text Fields](#)
- [Insert Drawings](#)
- [Create Diagrams](#)

To Change the Page Layout

You can set the page layout (for example, DIN A4, landscape format) in the Designer. To change the page layout, left click **Page layout** in the **Project** menu. The **Layout** window opens, where you can set the paper format in the **Printer selection** group.

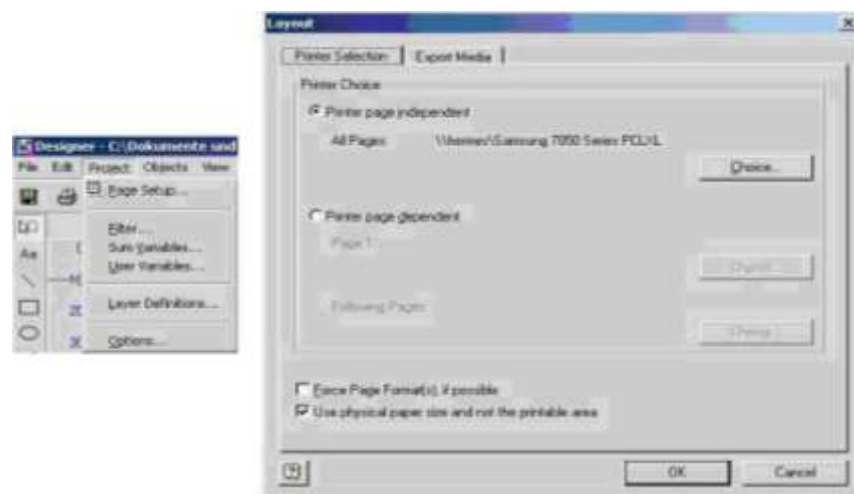


Figure 51: Setting the Page Layout

The following tools are available in the Designer on the left side of the toolbar.

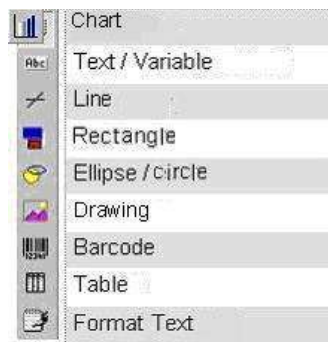



Figure 52: Designer Toolbar

To Create Tables

You need the Tables object to create lists, tables, and reports. It enables you to display your data in the form of a list or table. A table consists of the following elements:

- Headers
- Lines of data
- Footers
- Groups of lines

The quantity, width, and content of the columns can be freely defined in the individual line types.

- 1) To create a table, left click  button in the Designer toolbar.
- 2) The cursor assumes the shape of a cross.
- 3) Now left click in the editing area of the Designer and draw a frame. Use the frame to determine the size of the table.

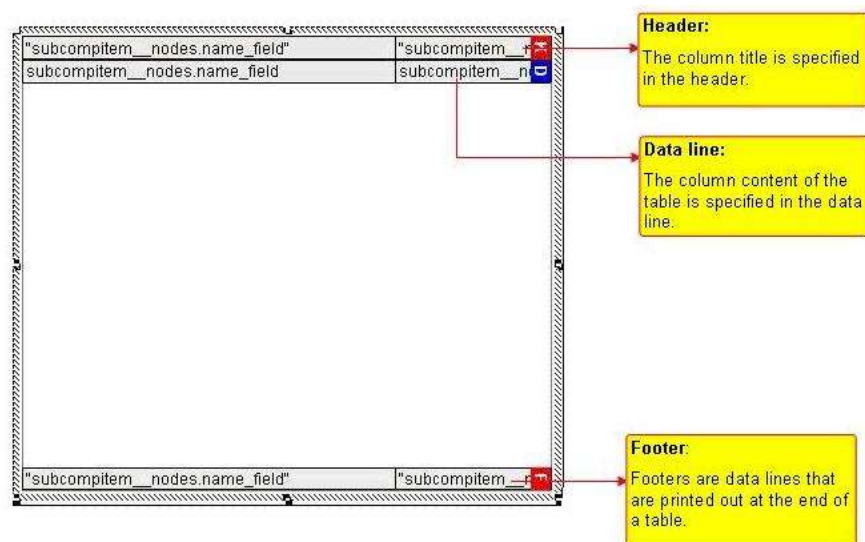


Figure 53: The Table Object

Creating and editing lines of data is shown [Figure 54](#).

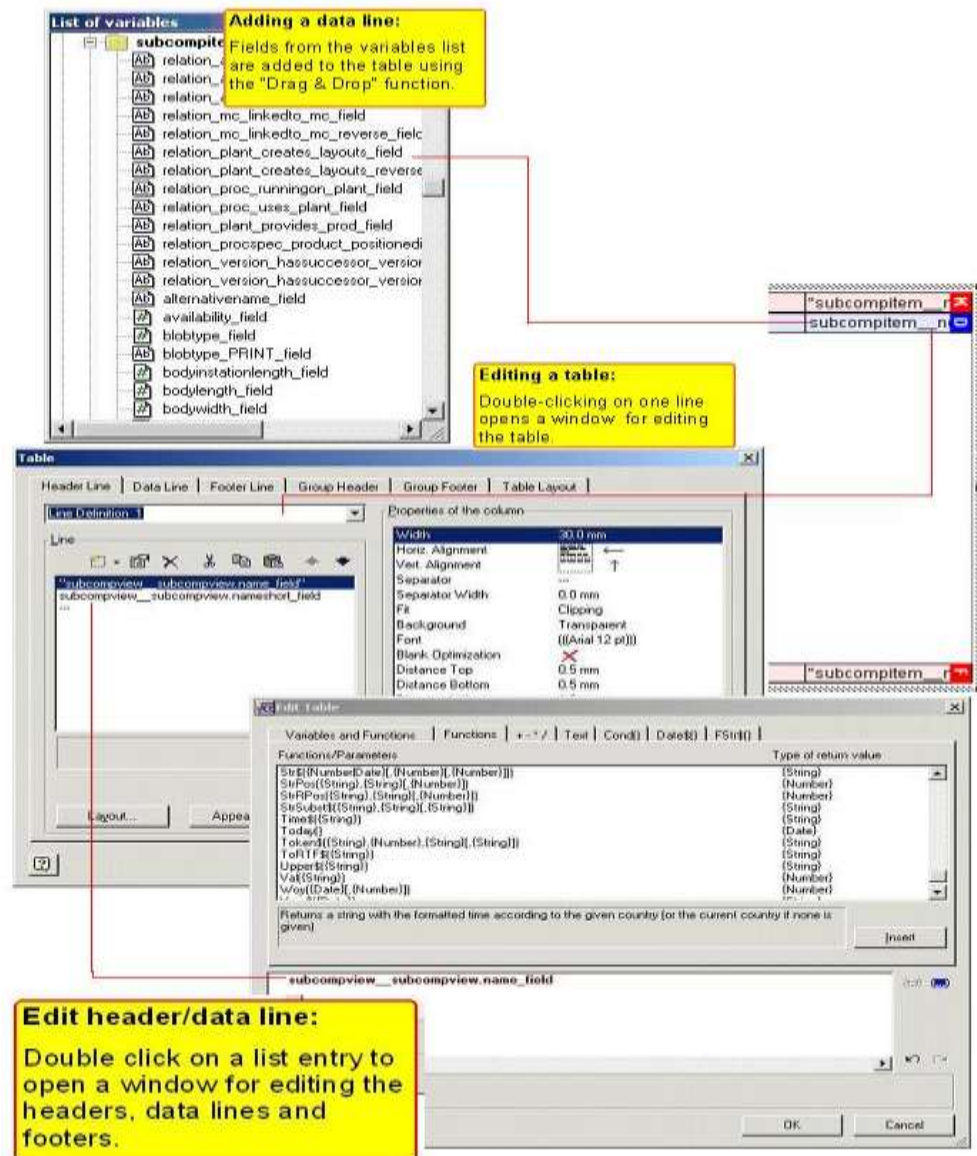


Figure 54: Creating and Editing Headers, Lines of Data and Footers To Sort Table Columns

Table columns can be sorted in Version PE 5.10.

- 1) Create the entire table in Designer.
- 2) Save the form. The form **must be saved**, or else the table cannot be displayed.
- 3) Using the **Tasks < Sort Tables** menu, you can now call a **Sort Table** dialog to be used for setting up the sort function for each table.



Figure 55: Additional Menu Point in Designer for Sorting Table Columns

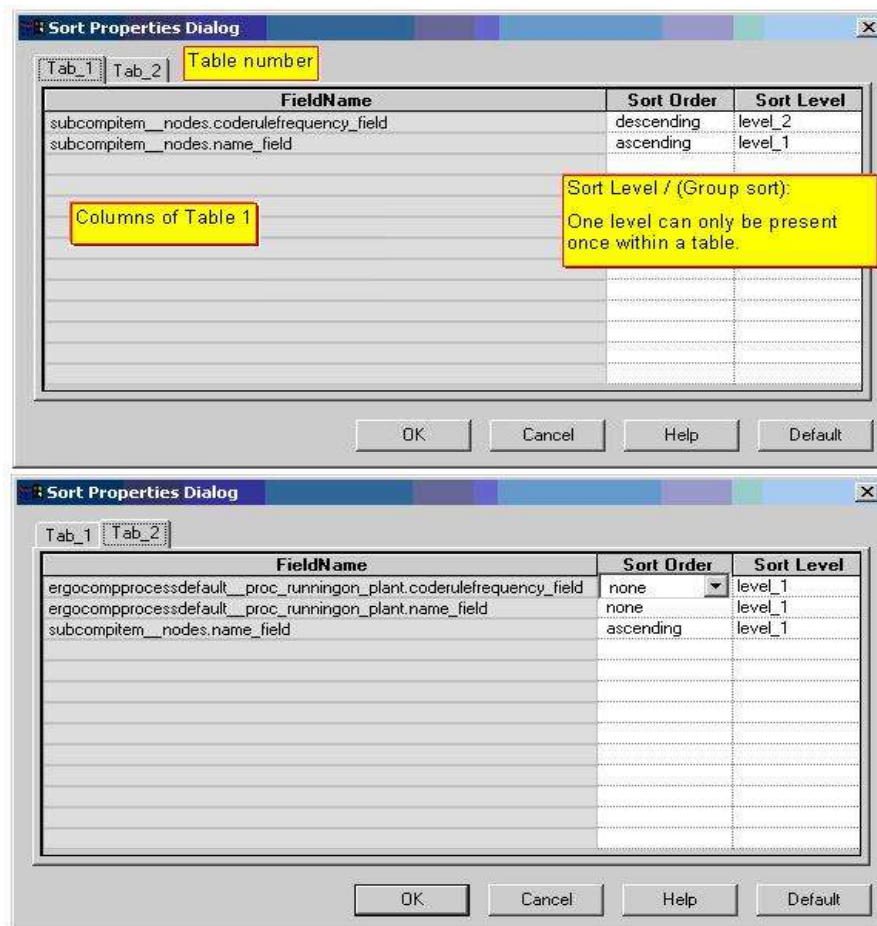


Figure 56: Sorting Table Columns

With the settings as shown in the illustration, you can the printout displayed below.

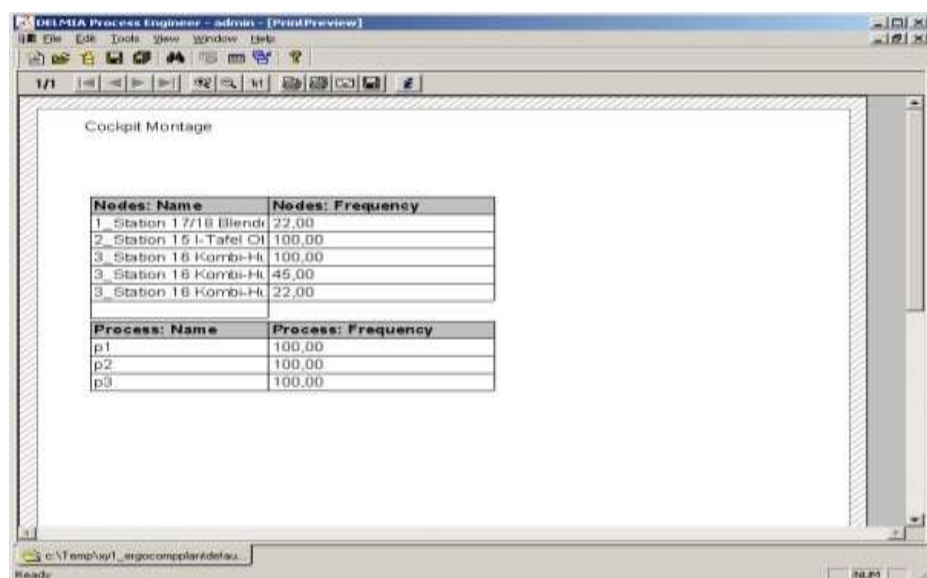



Figure 57: Sorting Table Columns

To Create Text Fields

A text field must be created in order to print out variable content or simple texts. To do so, left click  button in the Designer toolbar.

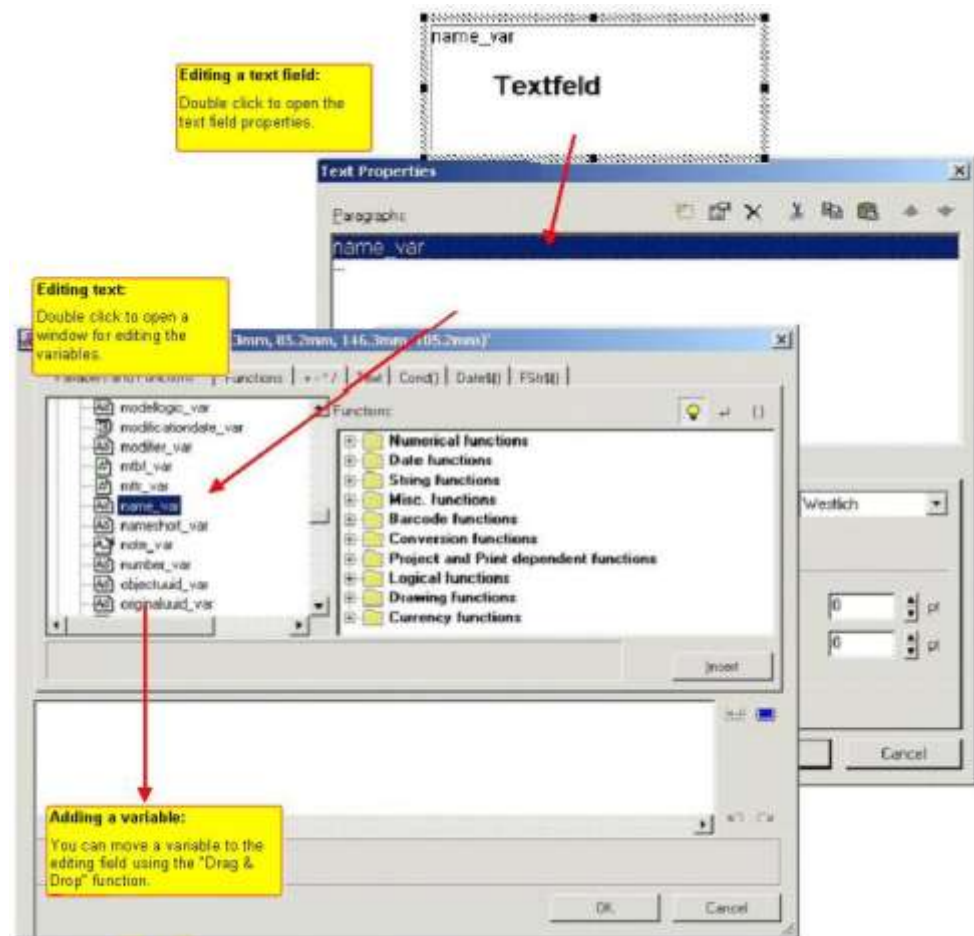



Figure 58: Editing a Text Field

To Insert Drawings

- 1) To integrate a drawing into the printout, left click  button.
- 2) Now use the left mouse button to draw a frame in the editing area where the graphic needs to be placed.

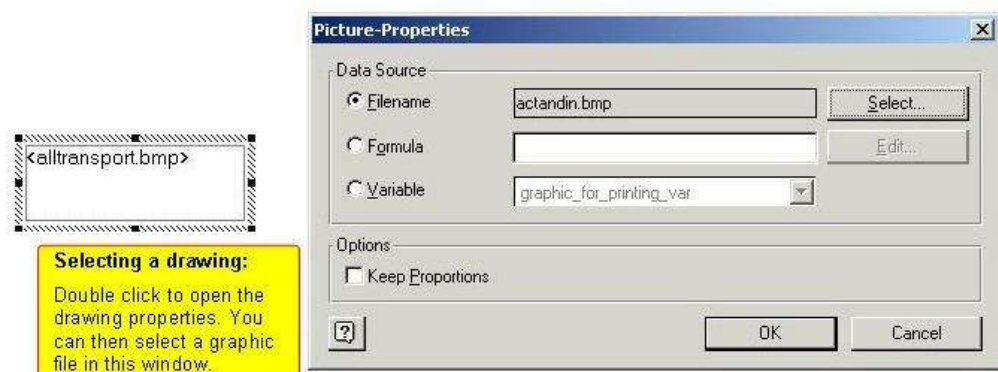


Figure 59: Inserting a Drawing

To Create a Diagram


- 1) You create a diagram area with the  chart pictogram. Then you can position this area below the table.
- 2) Double click the diagram area or use the contextual menu (Properties entry) to open the **Chart Properties** dialog. There, you specify the diagram type. In the example given, the **Pie** diagram has been selected.



Figure 60: Diagram Properties Dialog

You now have to select the data source that must be included in the table: as in the example “**coderulefrequency**”.

- 3) Click **OK** button to confirm.
- 4) Select **Accumulation of values** entry in the **Type of calculation with same data value** selection field in the **Type of Calculation** tab.
- 5) Enter the size on which your data source depends in the **Source for segment size** input field in the same tab. In the example, this is the length of the station: **subcompitem_nodes.length_field**.
- 6) Now enter a name for the diagram. Apply the same procedure as when naming the table.

Linking the Chart to the Table

It is important to name the diagram and the table in order to link both objects together.

- 7) Select the table and Objects list entry in the **Objects** menu to link the chart to the table. Select the name of the table in the objects list, which you have just assigned.
- 8) Click **Link to...** button to link the table to the chart object.
- 9) Click **OK** button to confirm.
- 10) Save your print object and return to Delmia Process Engineer®. Your printout should resemble [Figure 61](#).

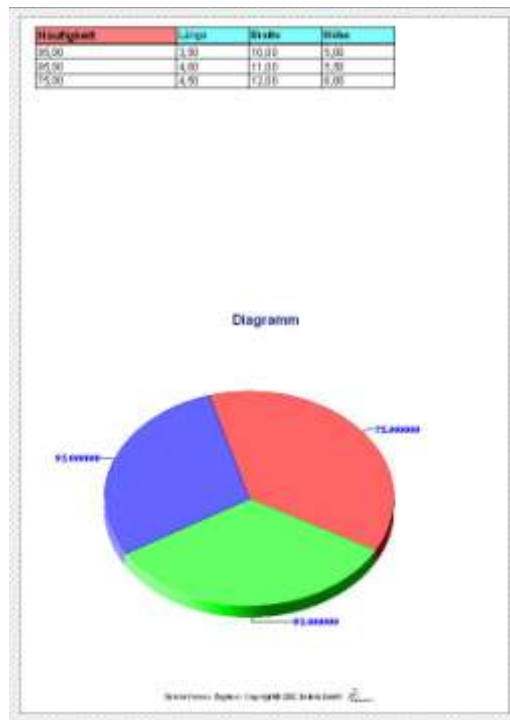


Figure 61: Example of a Pie Diagram

Printing Diagrams

The following rules should be observed when printing diagrams:

- After a diagram has been created, it must be linked to a table.
- Diagrams can only display data that is also included in the linked table.
- Several diagrams can be linked to one table.

The following example demonstrates how to create diagrams:

Example



Initial Situation

A plant consists of stations 1, 2, and 3. Each station has a different “Frequency” and different dimensions, such as “length” in the example

Station Frequency Length

Station - 1 75% 10

Station - 2 85% 11

Station - 3 95% 12

- 1) After opening List and Label. Create a New Table.
- 2) Move the coderulefrequency field and the length from the **subcompitem_nodes** folder into the table.
- 3) Enter a Name for the Table.
- 2) Select the table and press <Ctrl + Shift + Enter>.
- 3) Alternatively, you can enable the **Object** menu item in the menu bar and select **Name** entry.



Figure 62: Enter Name for Table

3.4 Valuating Variables and Fields in Print Forms

You can evaluate variables and fields in the print forms using mathematical operations (i.e. build sum). The steps needed for creating valuations are explained in the following chapters.

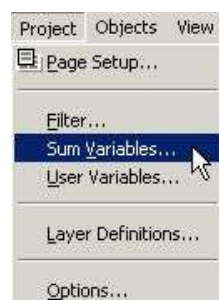


Figure 63: Sum Variables Menu

3.4.1 Defining a Sum Variable

When using forms for printing objects, you can define one or several sum variables. If you want to place the sum variable outside of the table object, you must create a dependency to the table object (Link objects).

If the sum variable is located within the table object, as a footer, then this dependency exists automatically.



Caution

Please note when linking objects that you can never link several table objects to one another.

If you want to print several tables on one form and each table has its own sum variables or if one table has several sum variables, you must create the tables using multiple line definitions, or else the sums are calculated incorrectly.

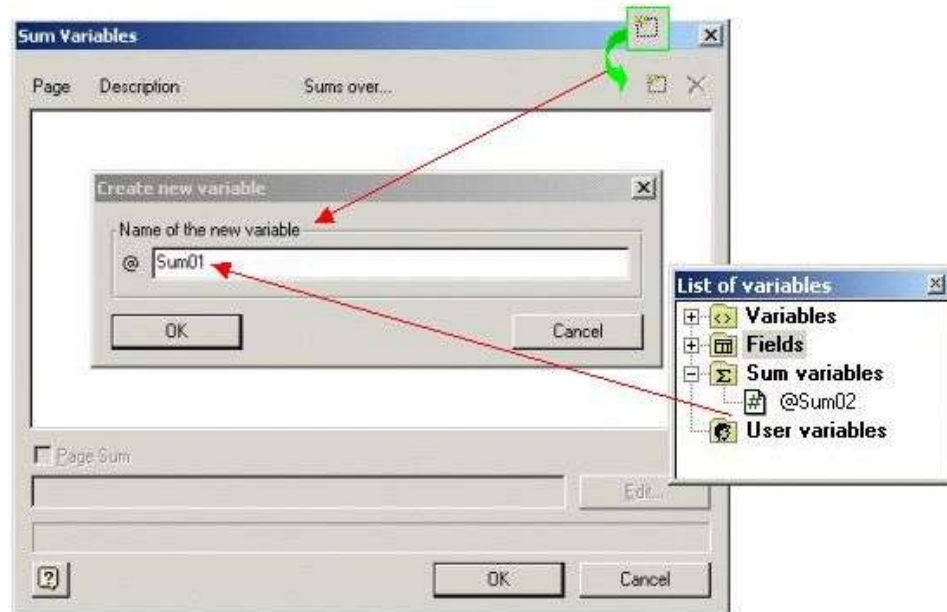


Figure 64: Creating a Sum Variable

3.4.2 Inserting Sum Variables in Tables

You can insert sum variables in footers. Double-click in a table that you previously created and enable the “footer” register. Now add a sum variable to the list of footer variables from the variables window using the “Drag and Drop” function.

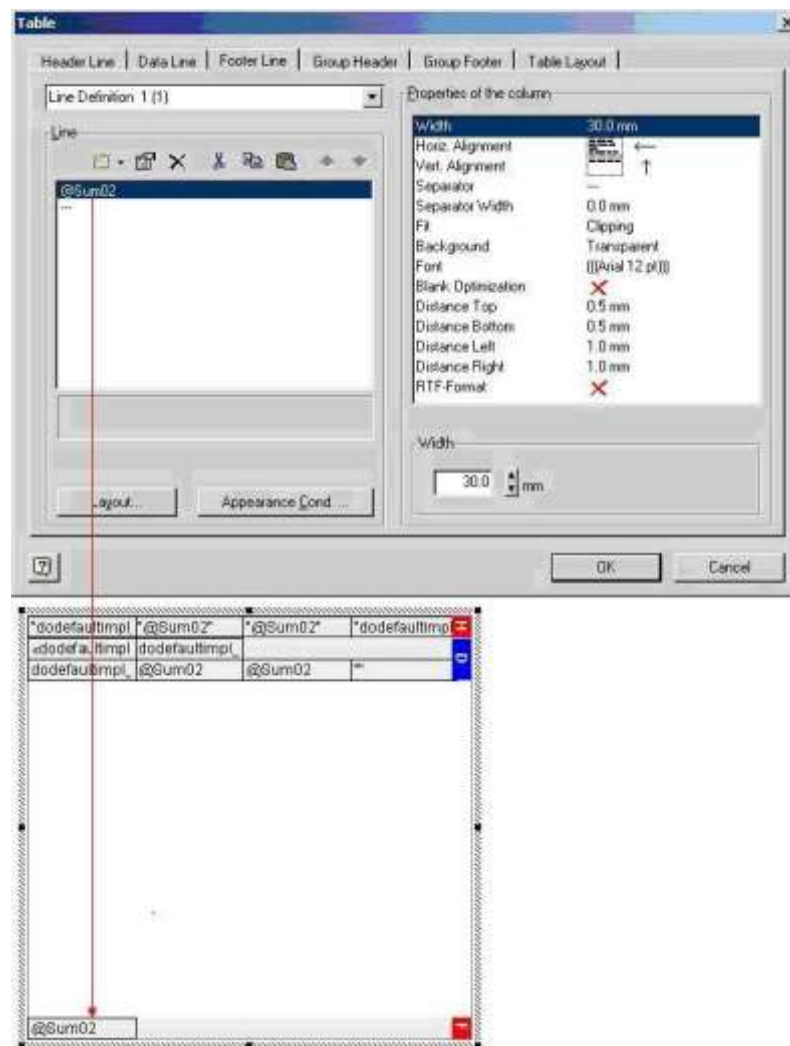


Figure 65: Inserting a Footer

A sum variable must always be linked to a table. If a sum variable is within the table (footer), the dependency exists automatically. If the sum variable is placed outside the table (e.g. in a text field), then it must be linked to the table (*Please refer to the [Figure 66](#)*).

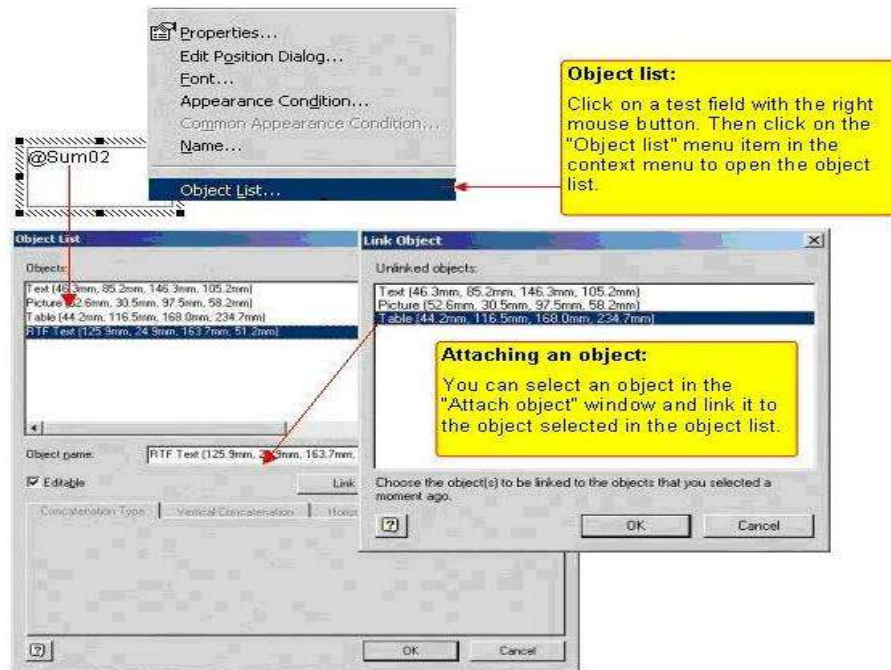


Figure 66: Linking a Text Field to a Table

3.5 Script Variables for Printing

Using script variables, you can print all of the information included in the Data database that is used in a script. Scripts are used to make the data from the Process Engineer available for printing. This data is printed out using the print form created in the Designer.

Script variables are treated the same as normal variables and fields by the print module. The only difference is the way the data is read from the database.

3.5.1 Creating Script Variables/Fields/Tables

Script variables/fields/tables are created using the **Tools < Print forms < Script Variables** menu.

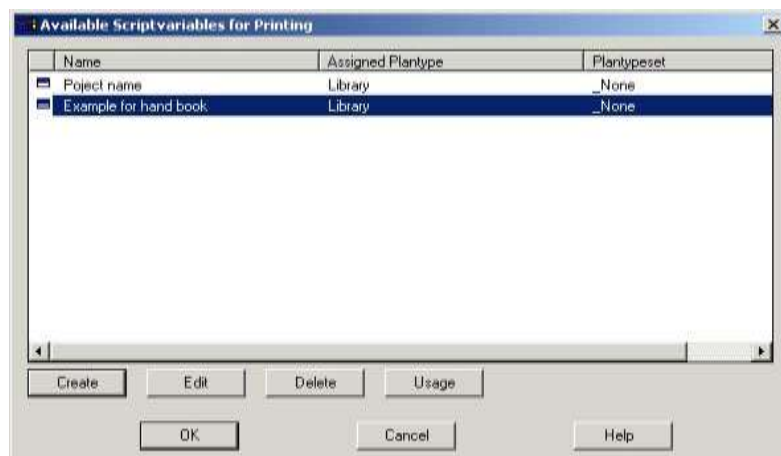


Figure 67: Defining Script Variables

Usage

Displaying Usage

When working with script variables, all print forms in which the script variables are being used are displayed via **Usage**.

Create

Creating Script Variables

Script variables are created using **Create**. Creating a new script variable. *Please refer to the Figure 67.*

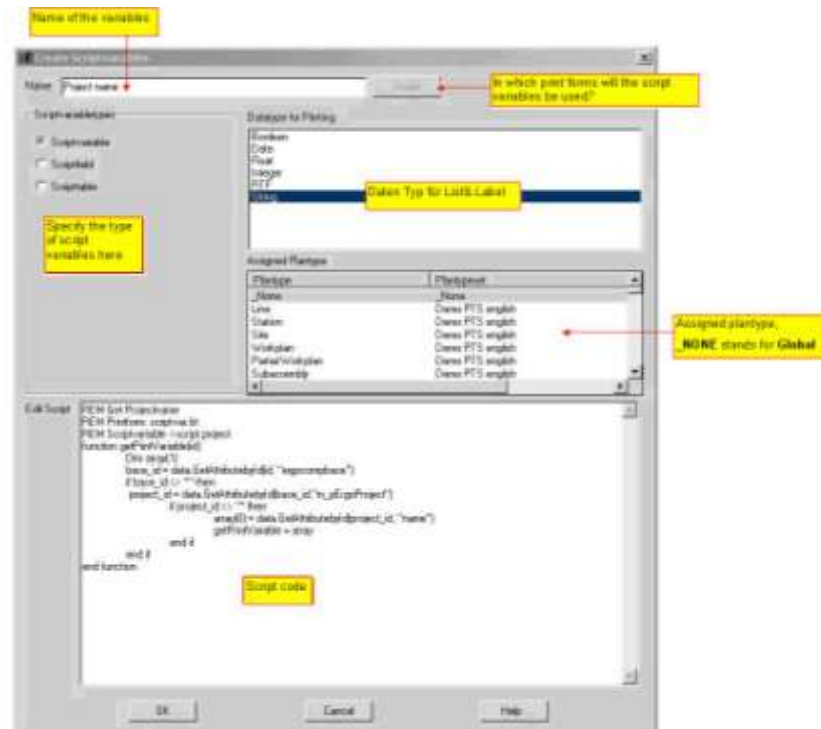


Figure 68: Defining Script Variables

Edit

Editing Script Variables

Script variables can be opened and edited using *Edit*. Script variable types cannot be defined when editing script variables.

Delete

Deleting Script Variables

Script variables can be deleted using **Delete**. You can use three types of script variables for printing lists: Script variables, Script fields, and Script tables.

Assigning Plantype

Script variables/fields/tables can only be assigned directly to one plantype or globally to all plantypes. Script variables/fields/tables that are available globally can be found in all projects and for all plantypes. By selecting **None**, you are assigning globally.

3.5.2 Using Script Variables

Script variables are registered as a variable in List and Label (Designer – Script variables dialog) and can be recognized by the `–scriptvar` extension.

The respective script must have the `function getPrintVariable(id)` function and return a one-dimensional field (array) with an entry.

Example of a Script for Script Variables

Example

```
REM Get Projectname
REM Printform: scriptvar.lst
```

```

REM Scriptvariable ->script.project
function getPrintVariable(id)
  Dim array(1)
  base_id = data.GetAttributebyId(id, "ergocompbase")
  if base_id <> "" then
    project_id = data.GetAttributebyId(base_id, "m_pErgoProject")
    if project_id <> "" then
      array(0) = data.GetAttributebyId(project_id, "name")
      getPrintVariable = array
    end if
  end if
end function

```

Figure 69: Example – Script for a Script Variable

3.5.3 Using Script Fields

Script fields are registered as fields in List and Label and can be recognized by the **–scriptfield** extension.

The respective script must have the **function getPrintField(parent_id)** function and return a one-dimensional field (array) with *n* entries.

Example

Example of a Script for Script Fields

```

REM Extends the "usual" GetChildren routine:
REM a) the children list is ordered (ascending) by name
REM b) before retrieving the children, the number of children
REM is displayed
REM PrintForm childnames.lst

function getPrintField(parent_id)
  call Data.SetOrderAttribute("name", "ASC")
  count=Data.GetChildrenCount(parent_id, "nodes")
  Dim array(1)
  Redim array(count)
  child_id = data.GetFirstChild(parent_id, "nodes")
  base_id = GetBase(child_id)
  if base_id <> "" then
    array(iChildCount) = data.GetAttributebyId(base_id, "name")
    iChildCount = 1

    Do while child_id <> ""
      value = data.GetAttributebyId(child_id, "name")
      child_id = data.GetNextChild(parent_id, "nodes")
      if child_id <> "" then
        base_id = GetBase(child_id)
        if base_id <> "" then
          array(iChildCount) = data.GetAttributebyId(base_id,
"name")
          iChildCount = iChildCount + 1
        end if
      end if
    Loop
    getPrintField = array
  end if
end function

function GetBase(sci_id)
  GetBase = Data.GetAttributebyId(sci_id, "ergocompbase")
end function

```

Figure 70: Example – Script for Script Fields

3.5.4 Using Script Tables

Script tables are registered as fields in List and Label and can be recognized by the *-scriptable* extension.

- In addition, for each script table 50 fields of the type **string**, **bool**, **rtf**, **float**, **integer**, and **date** are registered.
- The names for these fields are made up as follows:
<Scripttablename>.<Datentyp>.Col1-50_scriptablefield
- The respective script must have the **function getPrintTable (parent_id)** function and return a two-dimensional array with *n* entries, with the content of the first column corresponding to the name of the scripttablefield-variables used on the form. The columns correspond to the data.

Example

Example of a Script for Script Tables

```
REM Print data printarray
REM PrintForm is scripttable.lst
REM Scriptname: scripttable_print.txt
Redim printarray(4,1)
printarray(0,0) = "script.line1.string.col1"
printarray(1,0) = "script.line1.string.col2"
printarray(2,0) = "script.line1.string.col3"
printarray(3,0) = "script.line1.integer.col1"
REM childlistnames
Dim cln(2)
cln(0) = "nodes"
cln(1) = "plant_provides_prod"
cln(2) = "nodes"
REM column count
cc = 1
REM actual column
actcol = 0
function getPrintTable(parent_id)
    rem call Data.CacheChildIds(False)
    level = -1
    call GetChildren("", parent_id, level)
    getPrintTable = printarray
end function

function GetBase(sci_id, the_other_parent)
REM check if subcompitem or relationship
bIsDerived = Data.IsDerivedFromClass(sci_id, "XDOErgoCompBase")
if bIsDerived = true then
    GetBase = sci_id
else
    bIsDerived = Data.IsDerivedFromClass(sci_id, "XDOSubCompItem")
    if bIsDerived = true then
        GetBase = Data.GetAttributebyId(sci_id, "ergocompbase")
    else
        bIsDerived = Data.IsDerivedFromClass(sci_id, "XDORelation-
ship")
        if bIsDerived = true AND the_other_parent <> "" then
            GetBase = Data.GetLinkedObjectAttributebyId(sci_id,
the_other_parent, "oid")
        else
            MsgBox("Unknown base class")
        end if
    end if
end if
end function
```

Example

```

function GetName(id)
    name = Data.GetAttributebyId(id, "name")
    MsgBox(name)
    GetName = name
end function

sub GetChildren(old_parent, new_parent, level)
    level_in = level + 1
    if level_in <= 2 then
        parent_base_id = GetBase(new_parent, old_parent)
        if parent_base_id <> "" then
            cc = cc + Data.GetChildrenCount(parent_base_id, cln(level_in))
            Redim Preserve printarray(4, cc)
            child_id = Data.GetFirstChild(parent_base_id, cln(level_in))
            Do while child_id <> ""
                child_base_id = GetBase(child_id, parent_base_id)
                name = GetName(child_base_id)
                call WriteDataToArray(name, level_in)
                call GetChildren(parent_base_id, child_id, level_in)
                child_id = Data.GetNextChild(parent_base_id,
            cln(level_in))
            Loop
            REM This call is important (is required to reset the pointer to
            REM the beginning of the already fetched list)
            call Data.ResetIterator(parent_base_id, cln(level_in))
        end if
    end if
end sub

sub WriteDataToArray(name, level)
    actcol = actcol + 1
    For y = 0 to 3
        if y = level then
            printarray(y,actcol) = name
        else
            if y = 3 then
                printarray(y, actcol) = level
            else
                printarray(y, actcol) = ""
            end if
        end if
    Next
end sub

```

Figure 71: Example – Script for Script Tables

3.6 Script Variables for Print Form

The script variables are used to set up the print forms in the Designer (software for creating print forms). The structured display of the print forms is defined by the script variables, which are inserted into a table in the Designer that corresponds to the hierarchy being designed.

Script variables are used for the group header and for data lines:

- The names for the printout are determined in the group header, such as Station, Product, and Part.
- The data from the Process Engineer is displayed in the data lines of the printout.
- The data from the Process Engineer is made available for the printout by means of the scripts. Thus this data is always displayed internally in a ta-

ble and the structure of the table is arranged horizontally. *Please refer to the Table 4.*

- The table structure for the printout in the Designer is arranged vertically and corresponds to the printout to be created. *Please refer to the Table 4.*

Print forms can be created for all of the structures that can be created in the Process Engineer. One requirement for this is that a suitable script is created.

Based on the example of a structure with the three hierarchy levels for the print form Station, Product and Parts used the basic procedure for the application of script variables and the arrangement in the Designer is to be shown.

Example

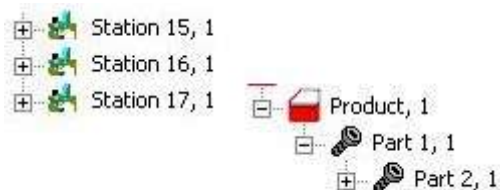


Figure 72: Example – Structure in the PPR Navigator Resources, Product, Parts

3.6.1 Internal Table Set-up for Scripts

The Table 4 shows the basic set-up of how the data that is acquired by means of the table script is processed internally in the table. The set-up of the table corresponds in principle to the structure that is to be displayed on the print form, except that the structuring is in a horizontal direction.

Table 4: Script Variables – Structured Table Set-up for Scripts

Using script variables for the printout	Specifying data for stations, products and linked parts using script variables.							
Using a script variable for stations: <code>script.line1.String.Col1_scripttablefield</code>	Station 15	""	""	""	Station 16	""	""	Station 17
Using a script variable for products linked to stations: <code>script.line1.String.Col2_scripttablefield</code>	""	Product 1	""	""	""	Product 2	""	""
Using a script variable for parts linked to products: <code>script.line1.String.Col3_scripttablefield</code>	""	""	Part 1	Part 2	""	""	Part 1	""
Using a script variable for change and hierarchical displays for table lines: <code>script.line1.Integer.Col1_scripttablefield</code>	0	1	2	2	0	1	2	0

3.6.2 Table Structuring for Printouts

The basic set-up for the presentation of a printout is shown in the Table 5.

Table 5: Script variables – Structured Table set-up for the Print Form

Names of the script table fields being used		
script.line1.String.Col1_scripttablefield (All stations)		
Station 15		
	script.line1.String.Col2_scripttablefield (All products linked to the station)	
""	Product X	
		script.line1.String.Col3_scripttablefield (All parts linked to the product)
""	""	Part A
script.line1.Integer.Col1_scripttablefield (Change and presentation requirement for the table lines)		
0	1	2

The display corresponds to the three hierarchy levels Station, Product, and Parts used, and shows the structured set-up of the table in a vertical direction.

- Quotation marks in a cell mean that this cell should be blank in the print-out.
- With a line break, the hierarchy levels are displayed as structured in the printout. Zero stands for the highest hierarchy level, which is Station in the example. One stands for the next highest hierarchy level, which is the Product in the example.
- A line break must be generated for each hierarchy level in the group header. If no line break is entered, there cannot be a structured display.

3.6.3 Creating Table for the Designer Printout

The print form for the structured printout is created using a table in the Designer, to which the script variables are assigned.

The brief description of the most important steps in creating a print form is presented in this section. *Please refer to the [Print Form Designer](#).*

- 1) Open the Designer via **Tools < Print forms < New object** or **Edit object** (then select print forms).
- 2) Select the table, then open **Properties** by right clicking using the contextual menu. A table must first be added to get a new form.

To display a print form, first create a group header, which is formed from the name and the respective script variables. The second step involves inserting the script variables for the data lines into the print form.

The line break for a structured printout should be inserted right away if all of the hierarchy levels for the group header of the print form have been created.

In addition, another script variable must be inserted in the table for the registration in the Designer. The **script variable** for the registration is created at the same time that a script variable is created.

Creating the Group Header

Group Header

- 3) Click **Group header** tab in the dialog. You then need to specify the line definition for each line of the group header. In the example this entails the three line definitions for Station, Product, and Parts usage.



Note

Please ensure that the script variable also corresponds to the respective hierarchy level for each line definition.

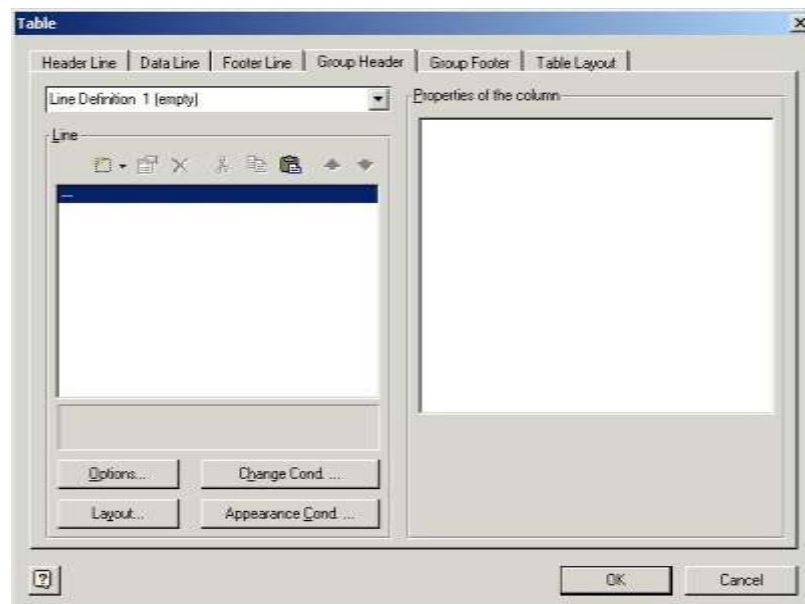


Figure 73: Dialog – for Creating Group Header and Data Lines

Data Line

Specifying Data Lines

- 4) Click **Data line** tab in the dialog. You then need to specify the line definition for each data line. In the example this entails the three line definitions for Station, Product, and Parts usage.

Change Cond. ...

Determining Change Requirements for the Printout

- 5) Click **Group header** tab in the dialog and then on **Change requirements/functions**.
- 6) Under the **Functions** tab, select the string function to which the script variable for the change requirement is assigned. The string function with the assigned script variable is shown in the dialog. *Please refer to the [Figure 74](#).*

The printout is displayed as structured if the change requirements are inserted in the table. A change requirement must be created for each hierarchy level in the group header.

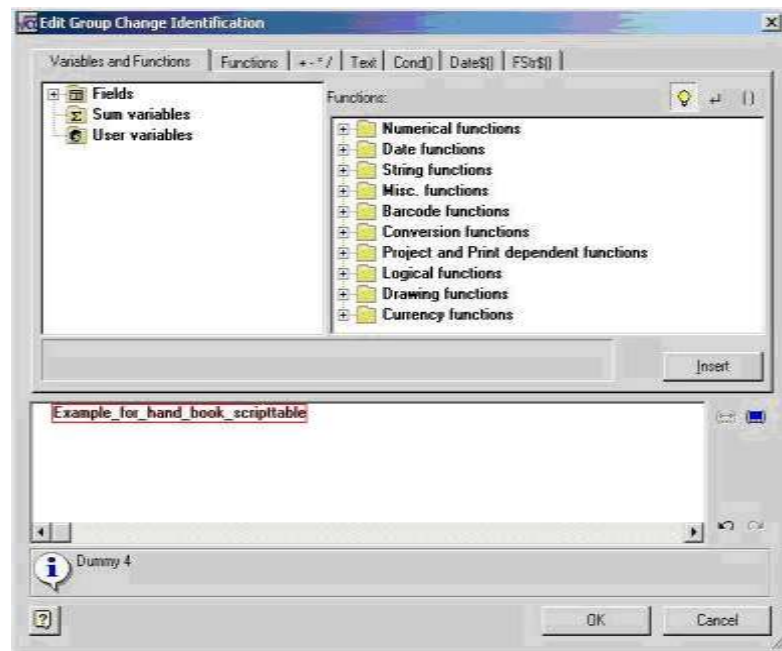


Figure 74: String Function with Assigned Script Variable

Inserting Script Variables for the Registration

This script variable enables the registration in the Designer.

In versions previous to PE 5.12 the script variable always had to appear in the table for logging on in the Designer (the example shows the script variable Example manual...). However, it did not have to be visible. You no longer need to add the script variable in version PE 5.12.

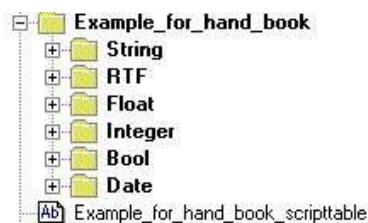


Figure 75: Script Variable for the Registration in the Designer

- 7) If you do not want the script variable to be displayed in the printout, set the **Width** to zero.

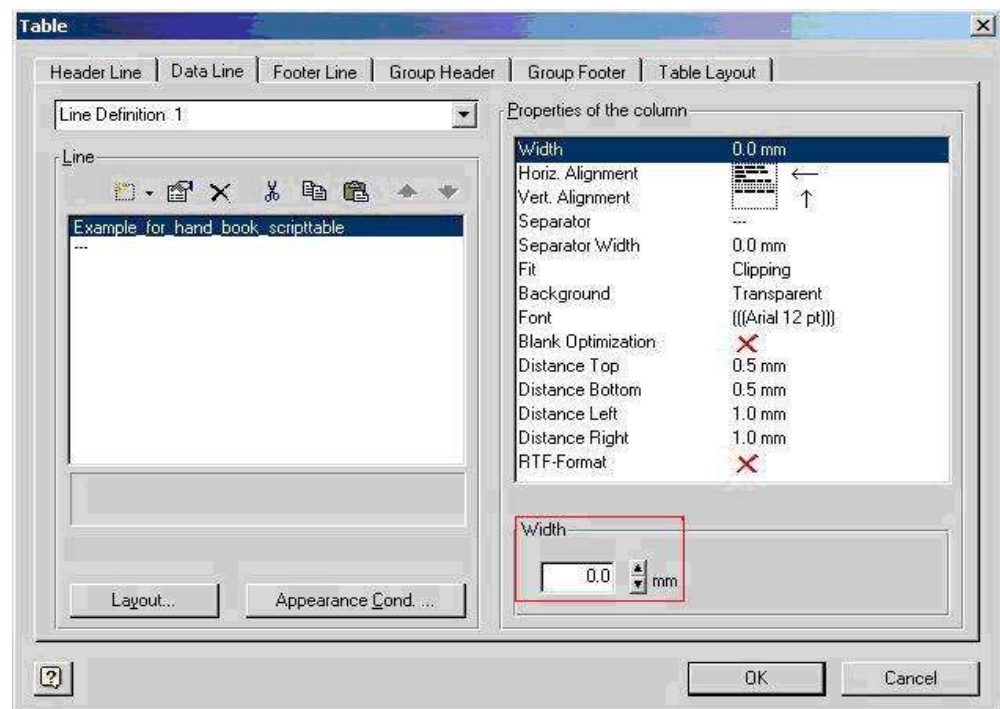


Figure 76: Set the Width for the Script Variable to Zero

Example

Example for structuring a print form in the Designer.

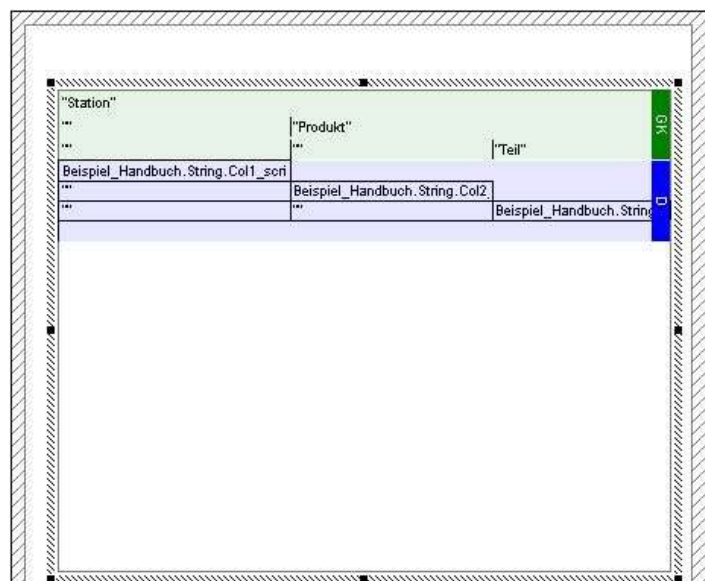


Figure 77: Example for Structuring in the Designer

Example

Example for a Printout

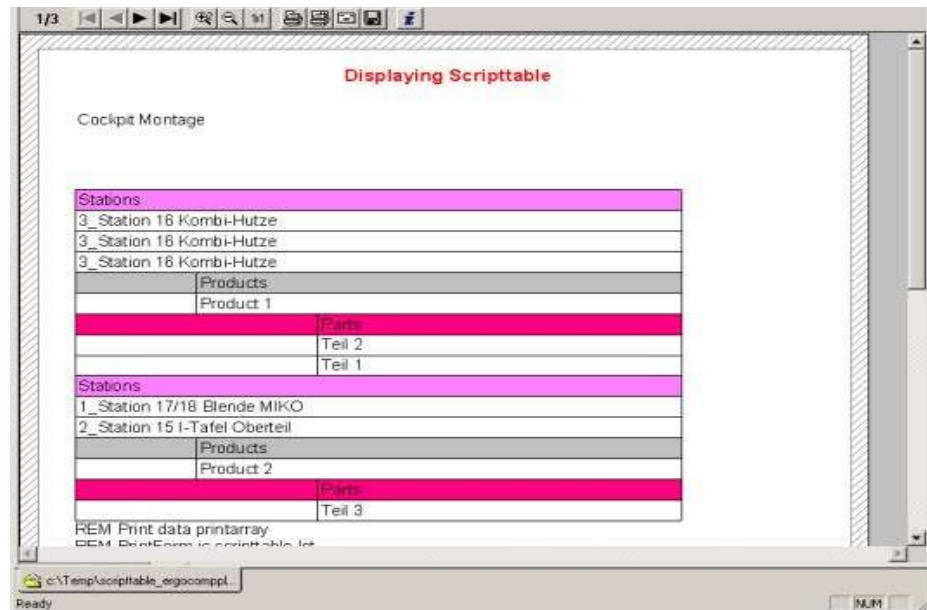


Figure 78: Example for a Printout

3.6.4 Graphics Printing in Tables with Script Variables

- 1) Generate a new script variable with the name "script.line1" of the type "scriptable". Use the script below.

Note

When naming script variables please ensure that the same variable name is used in the script itself. If the script variable is named HUGO, the variables in the script should be names "HUGO.graphic.col1", "HUGO.string.col2" etc.

- 2) Generate a printing form for object printing on the plantypes line < system.
- 3) Import the template **grafscripttable.LST** in List and Label. Save the form (you may need to match the variables to your script variables).
- 4) Start the print preview (object print) on a line.
- 5) All stations (depending on your plantype set) are displayed as graphics in a table. If you have already generated print settings with Graphics configuration, you can select an individual print setting for each graphic. In the example, two different print settings are used. A 3D view and a plan view. The plan view can be printed to scale (*Please refer to the [Scaled Graphics Printing](#)*).

```
REM Print data print array
REM PrintForm is scripttable.lst
REM Scriptname: scripttable_print.txt
Redim printarray(4,1)
printarray(0,0) = "script.line1.string.col1"
printarray(1,0) = "script.line1.string.col2"
printarray(2,0) = "script.line1.string.col3"
printarray(3,0) = "script.line1.integer.col1"
REM childlistnames
Dim cln(2)
cln(0) = "nodes"
cln(1) = "plant_provides_prod"
cln(2) = "nodes"
REM column count
cc = 1
REM actual column
actcol = 0
```

```

function getPrintTable(parent_id)
    level = -1
    call GetChildren("", parent_id, level)
    getPrintTable = printarray
end function

function GetBase(sci_id, the_other_parent)
    REM check if subcompitem or relationship
    bIsDerived = Data.IsDerivedFromClass(sci_id, "XDOErgoCompBase")
    if bIsDerived = true then
        GetBase = sci_id
    else
        bIsDerived = Data.IsDerivedFromClass(sci_id, "XDOSubCompItem")
        if bIsDerived = true then
            GetBase = Data.GetAttributebyId(sci_id, "ergocompbase")
        else
            bIsDerived = Data.IsDerivedFromClass(sci_id, "XDORelation-
ship")
            if bIsDerived = true AND the other parent <> "" then
                GetBase = Data.GetLinkedObjectAttributebyId(sci_id,
the_other_parent, "oid")
            else
                MsgBox("Unknown base class")
            end if
        end if
    end if
end function

function GetName(id)
    name = Data.GetAttributebyId(id, "name")
    REM MsgBox(name)
    GetName = name
end function

sub GetChildren(old_parent, new_parent, level)
    level_in = level + 1
    if level_in <= 2 then
        parent_base_id = GetBase(new_parent, old_parent)
        if parent_base_id <> "" then
            cc = cc + Data.GetChildrenCount(parent_base_id,
cln(level_in))
            Redim Preserve printarray(4, cc)
            child_id = Data.GetFirstChild(parent_base_id, cln(level_in))
            Do while child_id <> ""
                child_base_id = GetBase(child_id, parent_base_id)
                name = GetName(child_base_id)
                if level_in = 0 then
                    call WriteDataToArray(child_base_id, level_in)
                else
                    call WriteDataToArray(name, level_in)
                end if
                call GetChildren(parent_base_id, child_id, level_in)
                child_id = Data.GetNextChild(parent_base_id, cln(level_in))
            Loop
        end if
    end if
end sub

sub WriteDataToArray(name, level)
    actcol = actcol + 1
    For y = 0 to 3
        if y = level then
            printarray(y, actcol) = name
        else
            y = 3 then
                printarray(y, actcol) = level
            else
                printarray(y, actcol) = ""
            end if
        end if
    end if
end sub

```

```

end if
Next
end sub

```

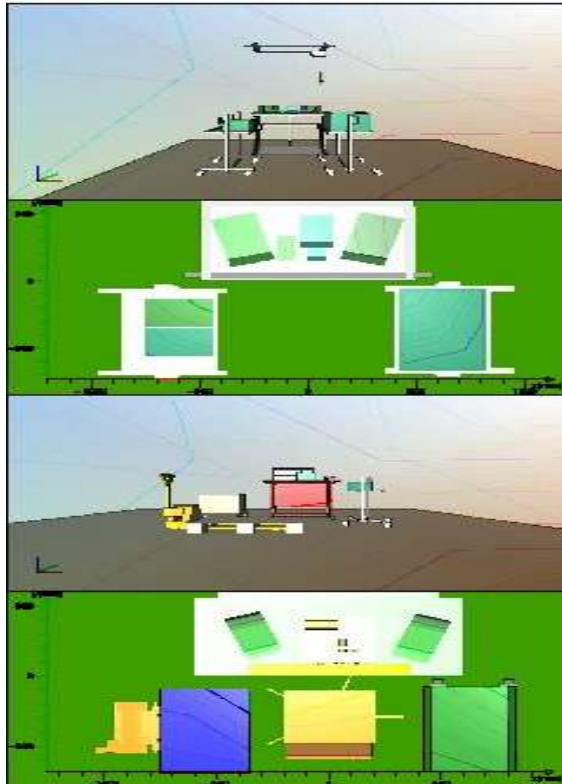


Figure 79: Displaying Scripttable

3.6.5 DIN A0 Graphics Printing

In this example, a printing form for DIN A0 printing is generated. Depending on your installation, you can find the templates for this example in \\DELMIA\PPRCClient\data\Formulare\examples. Both forms "Key_a0_1.lst" and "Key_a0_2.lst" can be used for the example.

In order to be able to work with these templates, the connected printer or plotter must be able to take paper formats of this size. Therefore, the correct printer should be selected before printing (also as a preview).

If you want to view this example in the preview, it may take more time for the image to appear (depending on the hardware – equipment). In order to avoid this, you can reduce the size of the graphics in the printing form.

You have to generate a script variable in order to assure that the printing form works properly:

- 1) Generate a global script variable with the name "script.project_scriptvar" of the type String. Use the script below for this.

```

REM Get Projectname
REM Printform: scriptvar.lst
REM Scriptvariable ->script.project

```

```

function getPrintVariable(id)
  Dim array(1)
  if id <> "" then
    project_id=data.getattributebyid(id,"ergoproject")
    if project_id <> "" then

```

```

        array(0) = data.GetAttributeById(project_id, "name")
        getPrintVariable = array
    end if
end if
end function

```

- 2) In the settings, deactivate the option which calls the standard form for graphics printing.
- 3) Open a station graphic.
- 4) Activate the graphics print preview.
- A file selection list opens in which you can select a template or another printing form.
- 5) The preview setup begins after a selection is made.

3.6.6 Scaled Graphics Printing

There is a new additional variable in object and graphics printing. This variable is named **PPR_GRAPHICSCALE_VAR**.

The variable **PPR_GRAPHICSCALE_VAR** stores the actual scaling of the graphics. It is initialized while the graphics are being printed.

The global script variables can also be used to enter additional information.

The following variables appear in List and Label:

- The variable "PlaceholderGraphic"
- The variables **PPR_GRAPHICSCALE_VAR**
- The standard variables user names, login names, present date
- All global script variables

This makes printing scaled graphics very simple. Proceed as follows:

- 1) Generate a new printing form for objects.
- 2) In List and Label use the variable **PlaceholderGraphic** for the ##graphic (drag the variable directly from the list to the form). Align the size of the variables in the form. In the contextual menu of the variables select the entry Name and give the name Graphic to the variables.
- 3) Set the variable **PPR_GRAPHICSCALE_VAR**. Give this variable the name *Scale*. Now both variables need only to be linked to one another.
- Since both variables have a unique name, they can simply be linked to one another.
- 4) Select the object list in the **Objects** menu.
- 5) Look for the object **Graphic** in the objects list. Activate it.
- 6) Click **link to...** button.
- 7) You can find the variable **PPR_GRAPHICSCALE_VAR** in the selection list under the name **Scale**. Link this variable to the graphic.
- 8) Save the form. You can now leave the designer.
- The scaling is also displayed the next time a graphic is printed.

3.7 Printing Setting

For any type of print form you can select the print orientation of the page format. Corresponding to the settings, a print form can either be printed in the landscape or portrait format. The settings for the print form are saved in the database and are automatically used for printing with the respective print form.

3.7.1 Page Format Settings

Set Portrait or Landscape Format for a Print Form

The settings for page orientation are either defined while creating the print form or are made upon opening of the print form.

The settings are made in the Designer program module under **Project < Page Setup < Print Setup**.

- 1) Select **Project < Page Setup** in the Designer program module.

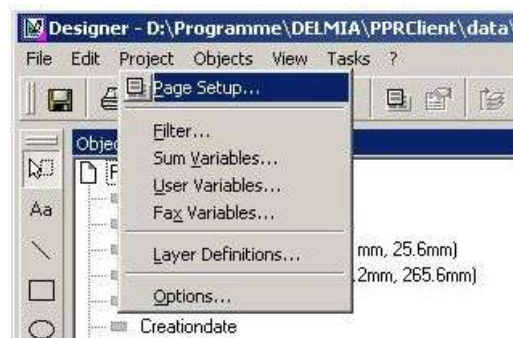


Figure 80: Select Page Setup Menu Item

- 2) Click **Select** to open the print setup dialog.
- 3) To set the orientation of the page format either select **Portrait** or **Landscape** – in the example Portrait has been selected.



Figure 81: Set Page Orientation

- 4) Confirm the selection with **OK**. When the print form is saved, the setting is saved along with it.
- When print with this print form, the print is made with the set orientation.

3.7.2 Printing Hierarchical PPR Structures

As of release PE 5.16.SP6 you can print entire hierarchical structures of PPR objects with attributes, including the project node, with the aid of print forms.

The option **Use parent node during print form design** can be used, for instance, to get definite information about the hierarchical structures in which PPR components are used – for example, the stations ABC are part of line A in factory B etc.

Only if this option is activated, the Parent directory is available in List and Label. This directory makes the attributes available that can be used to design the print form. Depending on the PPR component for which you wish to design a print form, the associated parent nodes and attributes selectable for design of the form are displayed.



For more information on printing turn to the, *Please refer to the Printing Tab in Settings Manual.*

To be able to print a hierarchical structure, you need to activate the option **Tools < Settings < Change < Printiing < Use parent nodes during print form design.**

Use parent nodes during print form design ☒

Figure 82: Hierarchical Structures for Print Forms

Note

Activate the option “Use parent nodes during print form design” only when designing a print form for hierarchical structures. If you fail to deactivate the option after completing design of the print from, this may lead to long response times during opening of List and Label.

Create print forms for printing of objects of PPR components in the system library in the directory of the plantype set of the respective PPR component.

Parent Directory

The Parent directory offers the attributes for design of hierarchical structures (parent node) – the example shows the parent nodes of the resource PPR component Site.

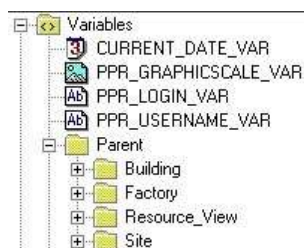


Figure 83: Parent Directory in List and Label

Ergoproject Directory

This directory lists the attributes for the project node, such as the attribute for the project name in the General directory.

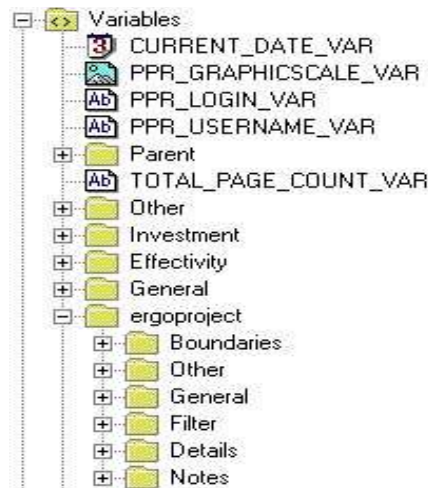


Figure 84: Ergoproject Directory

Example

In this example, the print form has been created with the aid of the attributes from the Parent directory in List and Label for the resource PPR component Site.

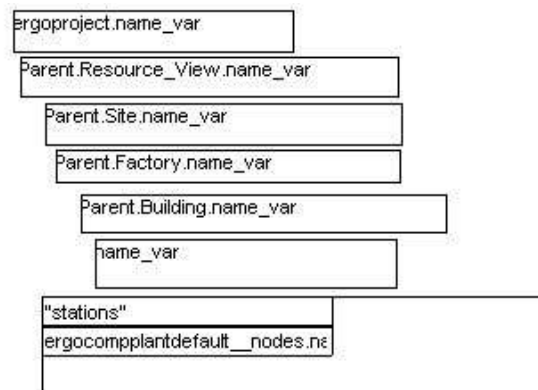
Example

Figure 85: Example - Site Print Form

- 1) After creating the print form open the contextual menu – on Assemblyline in the example.
- 2) First select the Preview entry to view the result before the actual printing.



Figure 86: Select Print Preview

The result of the print preview corresponds to the print form used – in the example all parent nodes of Assemblyline plus the stations of Assemblyline listed in the PPR navigator are displayed.

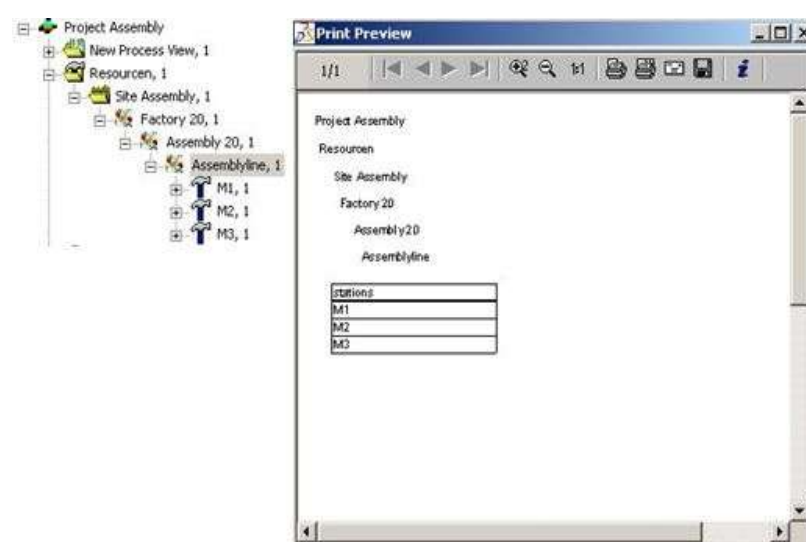


Figure 87: Print Preview of Hierarchical Structure

3.7.3 Inserting Logo and Copyrights in Print Forms

Reports created with forms delivered in DPE setup always contain a DELMIA logo and a copyright notice. Most probably this originates from DPE Rapid, where modifications of print forms were not allowed. But now it is possible to change the forms. Perform the following steps to do this.

- 1) Select the checkbox, "Insert DELMIA logo and copy rights in print forms", **Tools < Settings < Change < Printing < Insert DELMIA logo and copy rights in print forms**. By default this checkbox is selected.
- 2) Create a print form/report on any Plan type. Drag and drop the required attributes and save the print form/report.
- 3) Exit the designer.
- 4) Select Print/Preview on the Plan Type in the project to see the desired print form/report with logo and copyrights.
 - On creating a report/form in the designer for the first time, the DELMIA logo and copy rights are not displayed, while on editing the form, the logo and copyrights are displayed.
 - Once the print form/report is created with the checkbox selected, Print/Preview of the report always contains DELMIA Logo and copyright irrespective of the checkbox selected/deselected during the preview.
 - The DELMIA logo can be replaced with different logo and copyrights and can be edited during editing the print form/report.
 - On editing the print form/report with the checkbox selected, even if the logo and copy rights are deleted, the logo and copyrights still exist in the Print/Preview.
- 5) Deselect the checkbox, "Insert DELMIA logo and copy rights in print forms" to get print form without logo and copyrights.
 - The created print form/report can be edited with the checkbox selected to get the logo and copyrights.

4. Creating Queries from External Databases

An ODBC-interface (Standard language to access external data sources Open Database Connectivity) is used to make connection to external databases.

Several queries can be created, normally in tabular form, using ODBC-interfaces. Following this, the **External data sources** function can be used to prepare the queries for printing.

The actual query to retrieve data from an external database is written in the SQL programming language (Standard language to search in external data bases - Structured Query Language). A connection to the external databases must be made in order to get access to data.

Queries and connections to external databases should only be made by an administrator or an authorized employee.

The query can be created with the assistance of the **External data sources** function. The corresponding print forms on which the queries are to be printed are created in List and Label.

The **External data sources** function allows you to use and print unlimited data resources from databases, providing additional information on DPE data which can be used in the current DPE planning.

Access to other Data Sources Dialog

This dialog enables you to do two things: first, create a connection to an external database, and second, begin formatting the data for printing.

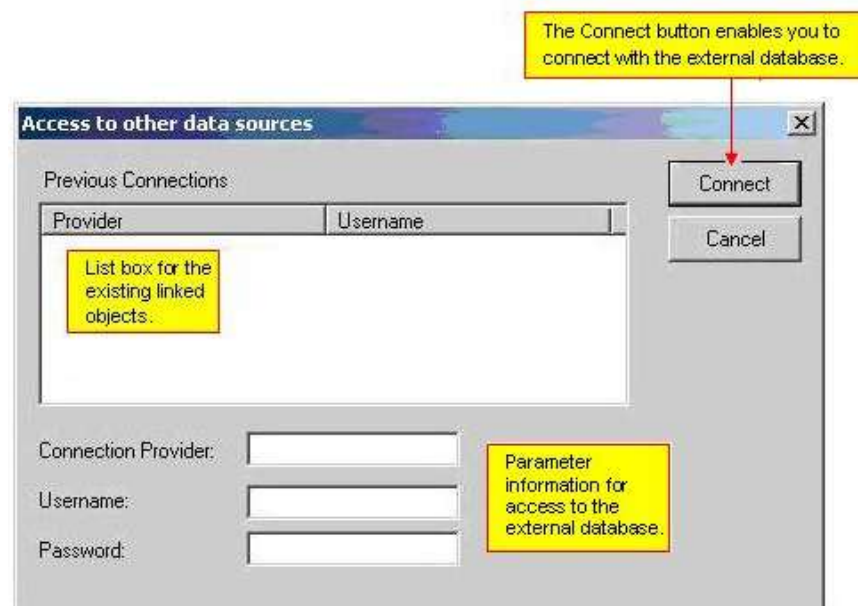


Figure 88: Access to other Data Sources Dialog

4.1 Creating Queries via External Data Sources Function

The parameters that need to be entered in order to make a connection with an external database using the **External data sources** function are set out in the ODBC connection. Several different queries can be created in the dialog of each individual connection that is made.

Parameter Information

- Connection Provider
- Username
- Password

To Start the External Data Sources Function

To use the **External data sources** function in the Tools menu, the system library must be opened first.

- 1) To start the External data sources function, select **Tools < Print Form < External Data Sources**.

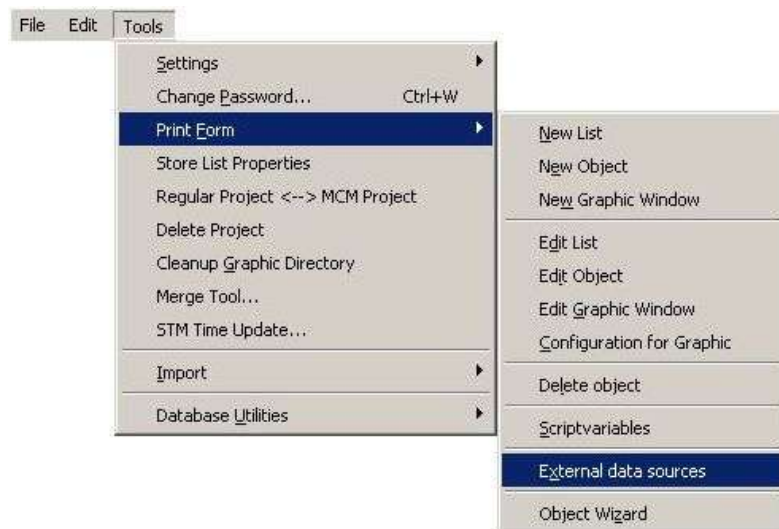


Figure 89: Starting the Connection to the External Database

To Make a Connection

There are two ways to make a connection to an external database:

- Entering all parameters
 - Selecting an existing connection
- 2) The list box is empty as long as there is no connection to an external database. In this case all parameters (Connection Provider, Username and Password) must be entered.

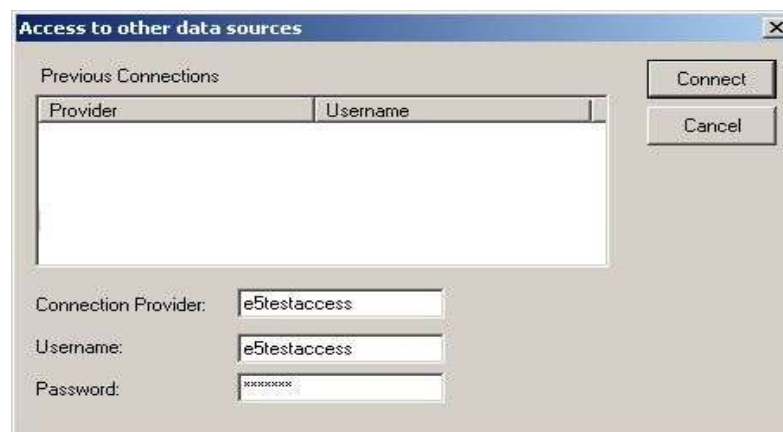


Figure 90: Entering Parameters in Access to Other Data Source Dialog

When connections are made, they get displayed in the list box.

- 3) In this case the connection can be selected from the list box. Just the password needs to be entered.

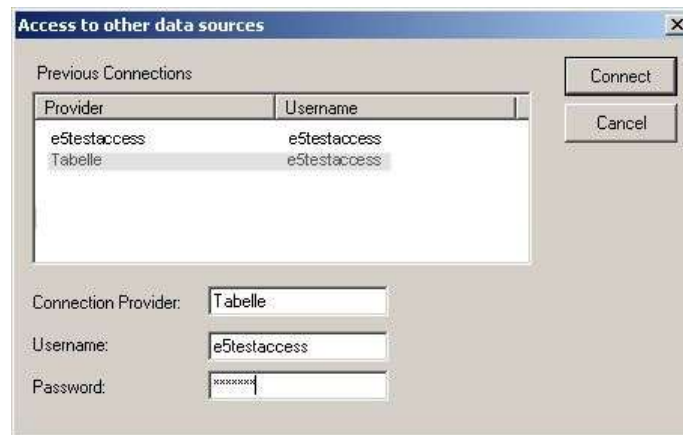


Figure 91: Click Cancel Button

- 4) To make the connection click **Connect** button. You can cancel the process using the **Cancel** button.

4.1.1 Creating New Queries

The sections described below helps you get to know the requirements for creating queries that can be printed out in List and Label. The print form is created in List and Label, which is then used according to the design.

Just a Brief Note on the Subject of Design

Use **Create** button to design a new query. The **Edit** and **Delete** buttons are only active after a query has been created and selected.

After creating a query, this dialog displays all existing queries. The **Edit** and **Delete** buttons enable you to edit or delete queries.

First, we want to create a new query. The different possibilities for queries need to be taken into consideration, when designing a query:

- A single query can be valid for all projects.
- A query is only valid for a specific plantype set.
- A query in a project is only valid for a particular plantype in a plantype set.

To start creating a query, click **Create** button to open the **Author Query** view. Please refer to the [Figure 99](#).

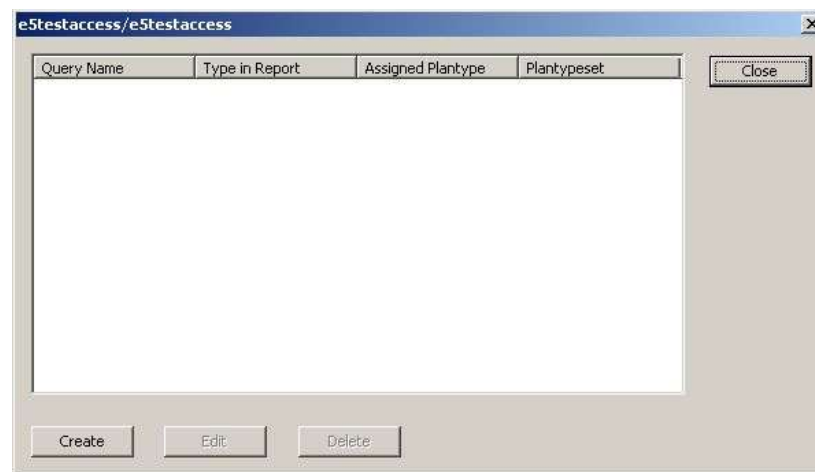


Figure 92: Dialog where Created Queries are Displayed

4.1.2 Determining Type of Report

There are two possible variables that can be used from the **Type in Report** field for creating a report.



Figure 93: Type in Report

Using the Variable Type

If you choose the **Variable** type, the report is be in the form of continuous text.

If you just want to keep the first line of an access connection for a database, selecting **Variable** in the **Type in Report** is sufficient. Regardless of the amount of data received, only the first line of the data appears on the report if this selection is used. This variable is displayed in List and Label under the directory **Variables < external_data_source**. Use this variable to create the print form.

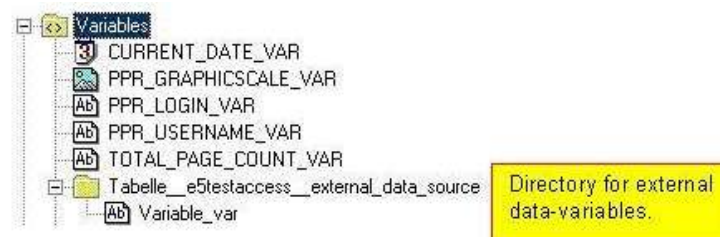


Figure94: The Directory for the Variable Type is in List & Label

Using the Field Type

If you choose the **Field** type, the report is in the form of a table. The **Field** selection is the most common form used for creating reports in List and Label. The **Field** type enables you to organize the data into different tables. These variables are displayed in List and Label under the directory **Field < external_data_source**. Use this variable to create the print form.

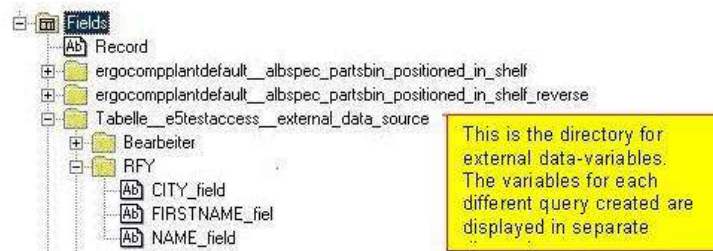


Figure 95: Directory in List and Label for Field Type

Use drag and drop to move these variables to the print form being created.

4.1.3 Assigning Data for PrintOut

Depending on the design of the query, use the syntax and any possible attributes for the query. The following example shows the basic procedure used for this.

You can determine the structure of the query and how the different data should be used in the **Author Query** view.

Using the Variable Type for Queries

We will just briefly touch on the **Variable** type, since the table form created using the **Field** variable is more commonly used from our view point.

In **Variable** type, only the first line, which is set in the syntax under **Query Statement** is printed: in our example it is the **Name** attribute. The next examples show how to create the syntax for the query.

Query Name

The name of the query is always entered under **Query Name**. This name must always be unique.

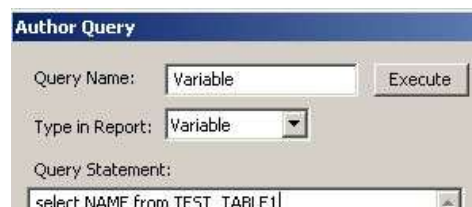


Figure 96: Enter the Name for Query

In List and Label the variable is displayed under this name in the directory of the external data: In the example, the Variables directory:

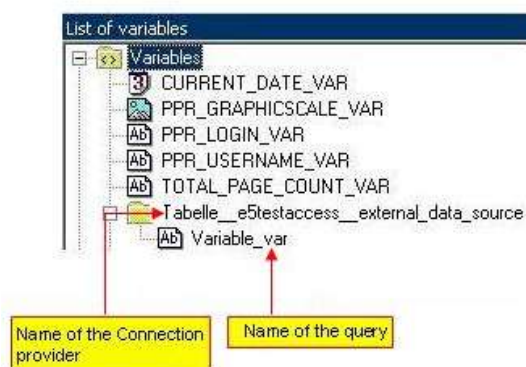


Figure 97: Show Variable in List and Label



Using the Execute Button

The **Execute** button enables you to see the results before the query has been saved or the print form has been created in List and Label – in this example the name displayed is from the external test-database *Test_Table 1*.

After clicking **OK**, you go back to the **Author Query** view. Here the query can be edited at any time before it is finally closed and saved. This dialog enables you to check the appearance and content of the print form before it is created.

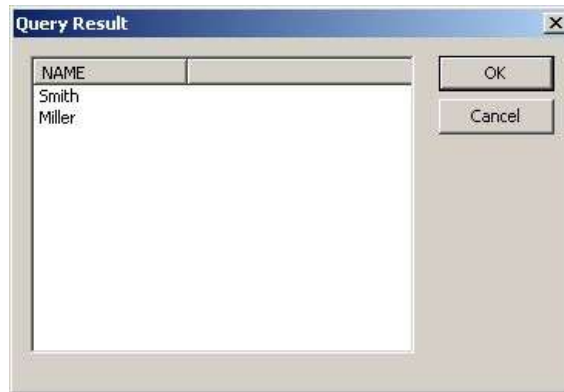


Figure 98: Displaying Results

4.1.4 Using Syntax for Design

As was mentioned earlier, queries can be created for all projects, for a specific plantype set, as well as for a specific plantype within a plantype set.

The selection you make under **Assigned Plantype** is an important one: This selection determines the application area for the query being made. For instance, if you select the Line plantype, only the data assigned to the Line plantype can be queried.

- If you want the query to be valid for all projects, select **None**.
- If a plantype set is selected, the query is only valid for projects that are based on this plantype set.
- Selecting a specific plantype results in the query being valid only for the plantypes, which are created in the project using this plantype set.

The following examples show how the dialog works, and second, how to create queries using the attributes of a plantype. The first example shows the procedure to be used for both the None (all projects) application area and the specific plantype set application area, with the only difference between the two being their application area. *Please refer to the [Project and Plantype Set Application Areas](#)*. The second example shows how to determine the application area for the attributes of a plantype. *Please refer to the [Creating a Query for Plantype Attributes](#)*.

Project and Plantype Set Application Areas

In this example a query is created that is applied to all projects. In line with this, the same procedure can be used for queries that only apply to the projects of a specific plantype set. Instead of **None** being entered, select the appropriate plantype set under **Assigned Plantype**.

Example of the Field type, all projects application area, and selected plantype set.

4.1.4.1 Assigning Attributes for Syntax in the Query Statement Field

- 1) Enter the name of the query next to Query Name - Table in the example.
- 2) Type in Report = Field, this is the type to be selected.

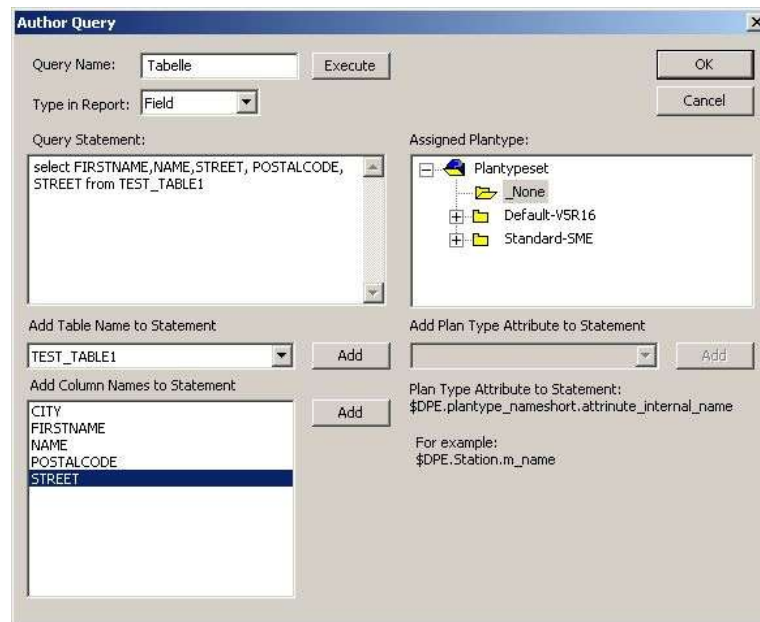


Figure 99: Example – All projects Application Area

Add Table Name to Statement

All external database tables are displayed in this area. Select the table from the combobox. Use the **Add** button to assign it to the **Query Statement** area.

Add Column Names to Statement

All columns belonging to the selected table are shown in this area. Use the *Add-button* to the right, to assign the selected column to the *Query Statement* area. Each column must be assigned individually.

Query Statement

Enter the syntax for the query in this area. To position data, place the cursor in the appropriate position. The input of the attribute always follows the exact position of the cursor. The data from the **Add Table Name to Statement**, **Add Column Names to Statement** and **Add Plantype Attribute to Statement** areas can be entered manually or assigned using the Add button.

The operands such as, select, from, where, equal (=) or unequal (<>) must be typed directly into the area. Pay attention to the spacing between operands and the data. A blank space must be entered between operators and data.

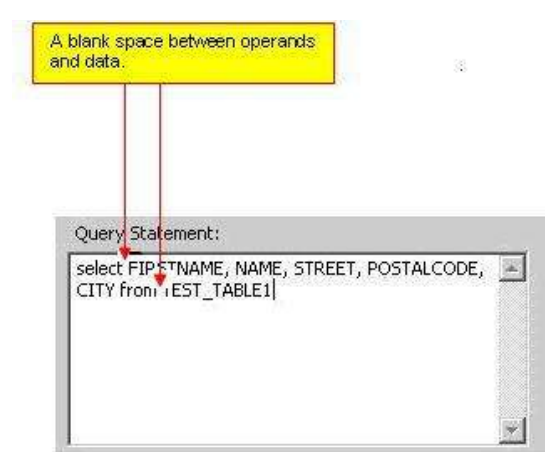


Figure 100: Syntax for a Query

3) After entering the data click **Execute** button to display the result.

The result can be viewed in the **Query Result** view.

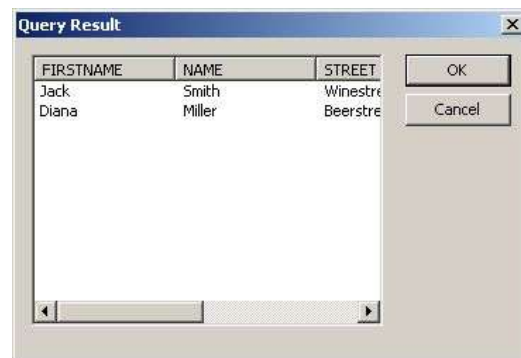


Figure 101: Display of the Results

4) Confirm the selection with **OK**.

Once again, each time you access data on an external database using ODBC, several different queries can be created. Saved queries are accessible and can be opened for editing. The access name of the external database and the user name are displayed in the title.

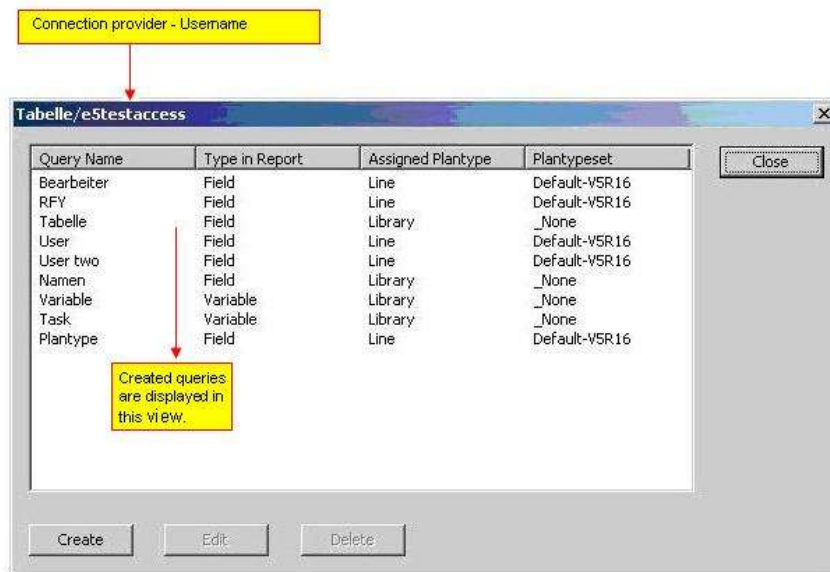


Figure 102: Database Table

- 5) Close the dialog using the **Close** button and then save the query.
- 6) Create the matching print form in List and Label to print out the query.

Creating a Query for Plantype Attributes

Once again, in this example a unique name for the query must be determined and a selection must be made from the **Type in Report** for the query design to be used. In this example, the **Field** type is selected again.

The attributes under **Add Plan Type Attribute to Statement** are only available if a plantype is also selected, under a selected plantype set. The available attributes match the selected plantype.

This method enables you to create customized queries for specific plantypes. The attributes placed in the Query Statement area using the **Add** button, and linked with the appropriate operands and data are queried.

In this example, you should find all of the location names and first names from the external database which do not match the **nameshort** attribute.

- 1) To assign the attributes of a plantype you must first select the plantype set and then select the plantype – the **Line** plantype in this example.
- 2) All of the selected attributes that are assigned to a plantype are displayed in the **Add Plan Type Attribute to Statement** area.
- 3) Select the attribute, and use the Add-button to assign the attribute – the **nameshort** attribute in this example.

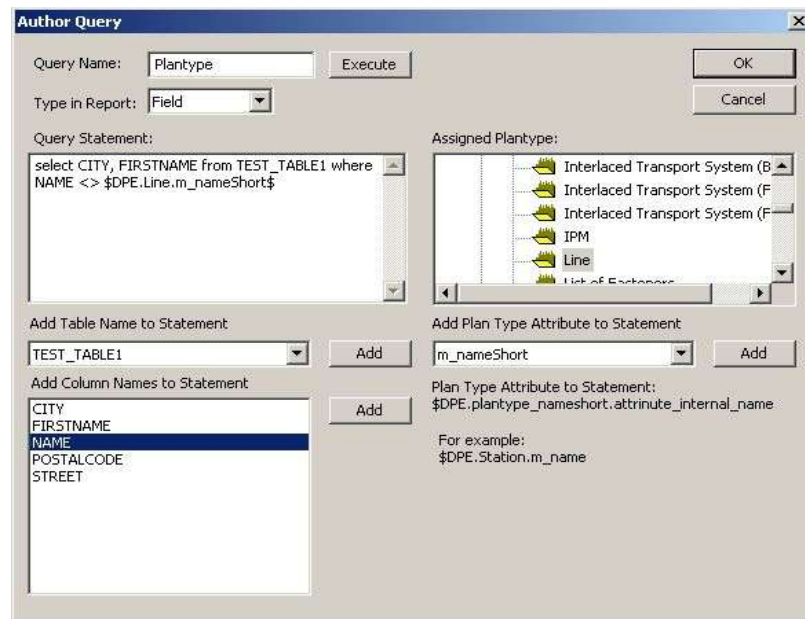


Figure 103: Assigning the Attribute of a Plantype

4) After entering the data, click **Execute** button to display the result.

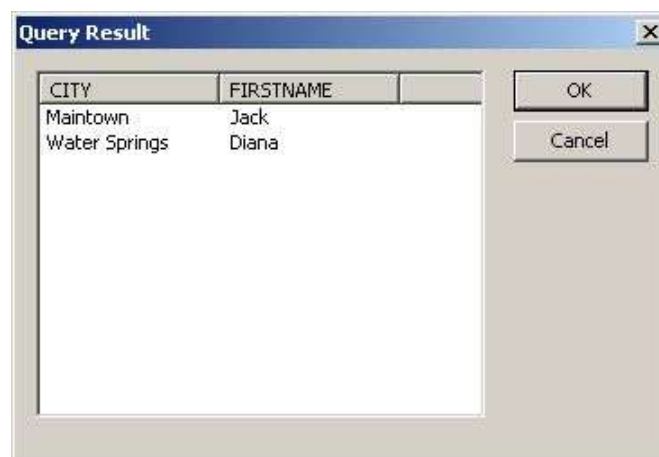


Figure 104: Result of a Query Containing an Attribute

5) Close the dialog and save the result.

4.1.5 Editing or Deleting Queries

- 1) To edit a query, select the query and then click **Edit** button.
- 2) To delete a query, select the query and click **Delete** button.

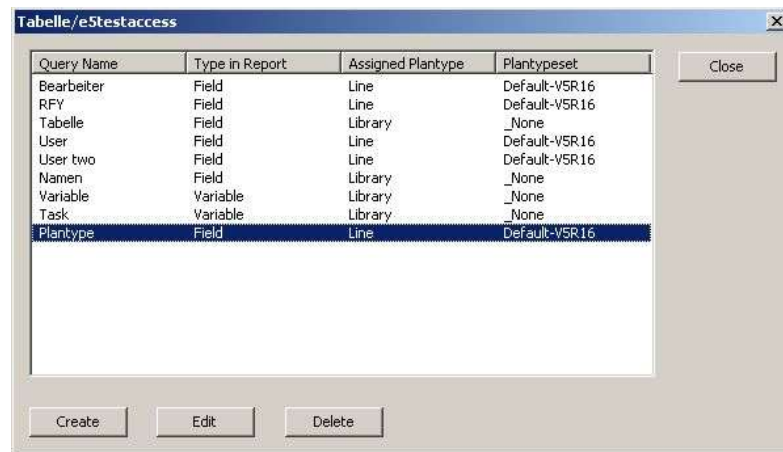


Figure 105: Selecting the Query for Editing or Deleting

Editing a Query

After opening the query using the **Edit** button, certain areas of the query can be edited.

- You can edit the database access (table) in the **Add Table Name to Statement** area. For instance, if you would like to use the same syntax to access other data.
- You can assign the query (**Query Statement**) other columns in the **Add Column Names to Statement** area.
- All areas or objects highlighted grey, such as the name, type of query, and the selection for plantypes cannot be edited.

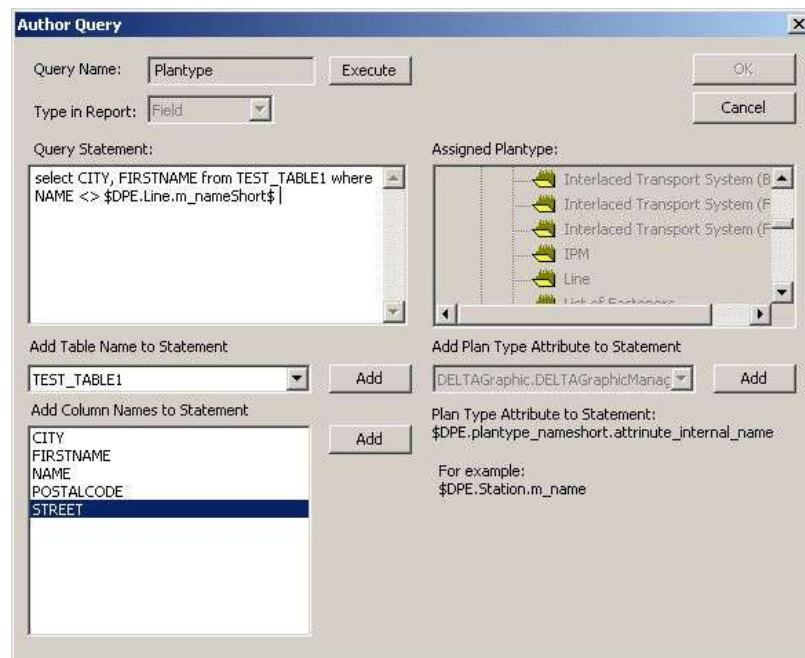


Figure 106: Editing a Query

4.2 Reporting on Data Structures - Printing Tables

The new **Object Wizard** function can be used to create a practically endless variety of reports on PPR-Navigator data structures. The matching print forms

that can be used to print the data report in the form of a table, are created in List and Label.

The reports being created are always based on a specific plantype in the project. Reports can be created for each plantype in a plantype set used in the project - this means reports can be designed and created for product, resource and process components.

Of course every table printout has limitations due to the size of the paper being used – for instance, due to the amount of data being reported on, there might not be enough space in the header for the amount of columns that should be displayed in a table.

By designing the data structure for the report in a relational manner using the **Object Wizard** function, these limitations can be avoided.

The reports and print forms in List and Label should be designed solely by an administrator or an authorized employee.

To use the **Object Wizard** function found in the **Tools** menu, the system library must be open and the specific plantype must be selected in the plantype set.

4.2.1 Object Wizard

Example

Example - reporting on data structures with the assistance of the Object Wizard.

This simple example will show the results achievable by using the **Object Wizard**.

In this example data structures from the **Line** plantype are to be reported on. The goal is to display all of the line processes used in those stations – for example a report on an assembly line with stations. The **plantypes Line < Station < Workplan < Operation** have been used for this example.

To save this report, first create the individual pages of the plantype that are to be considered in the report using the **Object Wizard** function. Second, create the appropriate print form in List & Label.

The following page explains the process of creating reports and the corresponding print form; this is based on the plantype used in the example.

Stations	Processname	Processtime
Station 0101		
	P1	0.000000
	P2	45.000000
	P3	60.000000
	P4	25.000000
Station 0102		
	P2-1	10.000000
	P2-2	12.000000
	P2-3	21.000000
Station 0103		
	P1	0.000000
	P2	45.000000
	P3	60.000000
	P4	25.000000
Station 0104		
	P2-1	10.000000
	P2-2	12.000000
	P2-3	21.000000

Figure 107: Print the Data Structure Report

4.2.2 Author Print Object View

A report can be designed in the **Author Print Object** dialog. Additional report conditions can be entered by opening further input windows using the **Next Level** button.

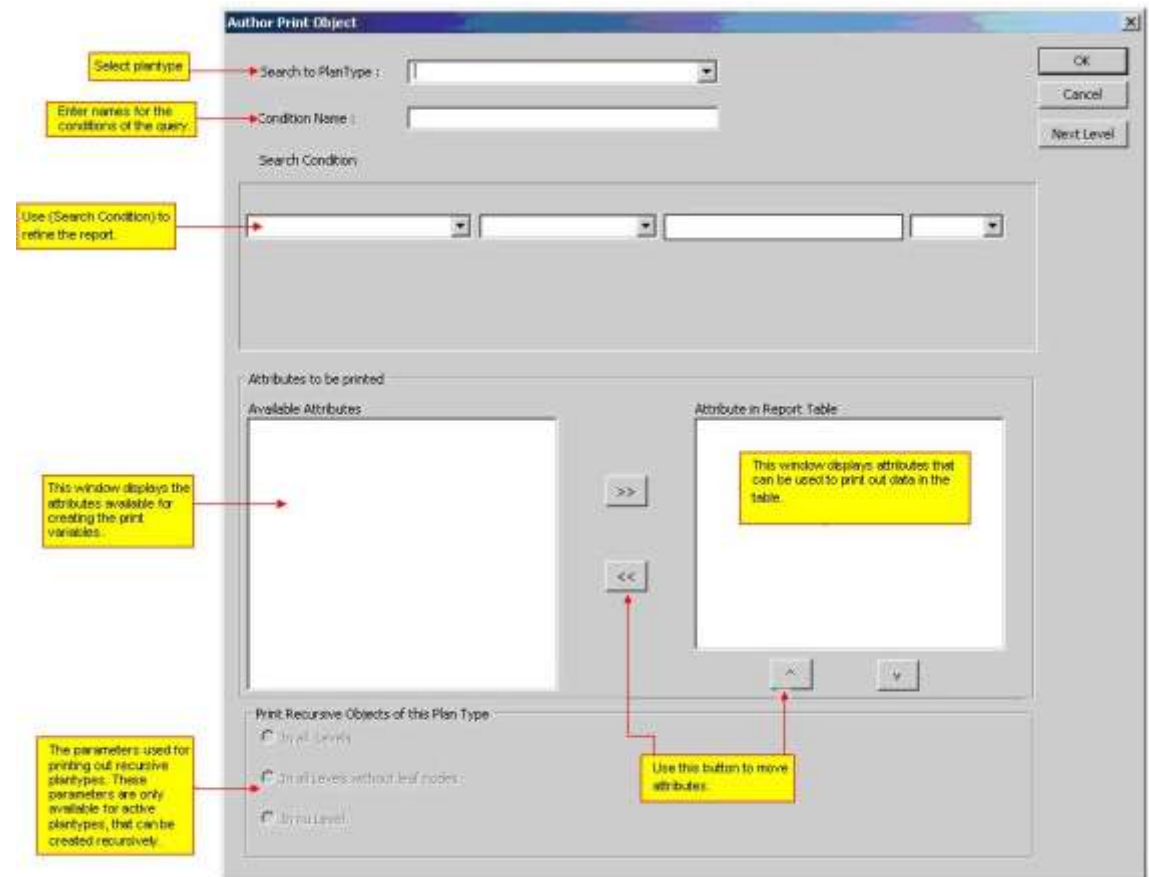


Figure 108: View of the Author Print Object dialog

4.2.2.1 Entering Report Data in the Author Print Object Dialog

The reports created depend on the plantypes selected. To begin the report, design the individual pages of the report using the **Next Level** button.

These individual pages make up the structure that is used for the table printout. This fixed structure is always bound to the plantypes. Keep in mind the following when designing the print form in List and Label: The print form created in List and Label must always be for the same plantype being used in the report.



Note

*If the foregoing points are disregarded, the table printout can be distorted. The data structure selected using the **Object Wizard** and the print form made in **List and label** must be a 100% match, otherwise problems will arise.*

4.2.2.2 Author Print Object Dialog Description

Search to Plantype

One page of the report can be created for each plantype you select from the **Search to Plantype** combobox. The availability of the attributes used for creating the print variables is dependent on the plantype selected.

As mentioned before, reports can be structured relationally using the **Object Wizard**. The word relational in this case describes a relationship between PPR components, for example a parent-child relationship in a hierarchical structure or between PPR components with different plantypes.

The plantypes with which a relation can be made are displayed in the combo-box and can also be used for designing individual pages of the report. So if a connection is made in a page of the **Author Print Object** dialog between plantypes by way of a relation, this is called a relational report structure.

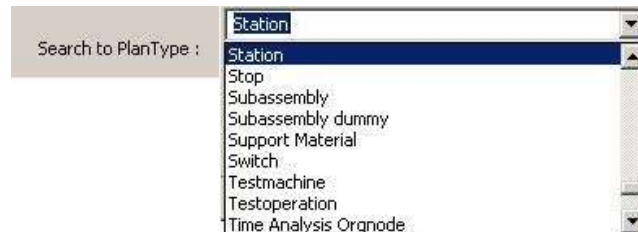


Figure 109: Selecting a Plantype for a Page

Condition Name

The name selected for the page should be related to the data being reported on in that page – for example if stations are going to be included in the report of the plantype, select a name like **Show Stations**. This name is only related to this particular page in the report. The informal character of this entry is important.



Figure 110: Entering the Name of each Page of the Report

Search Conditions

By refining the parameters of a report, the conditions for the search of the data can be entered at **Search Condition**. The plantype that is selected at **Search to Plantype** determines the amount of available attributes. The logical operator makes extending conditions and linking conditions with further report parameters possible. A new line is created for each logical operator selected. A new line can only be created if one or both of the two logical operators, **And/Or** are selected.

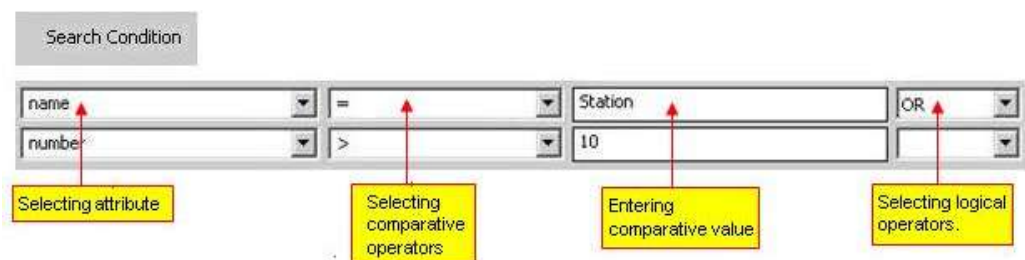


Figure 111: Entering the Report Parameters to Search for the Data

- **Selecting the Attribute:** Select the attribute for the condition from this combobox. The selection of available attributes is dependent on the plantype selected at **Search to Plan Type**.



Figure 112: Selecting the Attribute

- **Selecting Comparative Operators:** From this combobox, select the comparative operator to be used for comparing of the attribute selected with the comparable value.

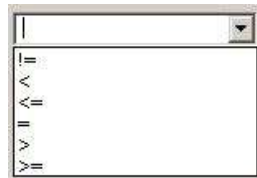


Figure 113: Selecting the Comparative Operators

- **Comparative Value:** The comparative value can be used to select the comparative condition to be compared with the attribute. Numbers, letters, and alphanumeric symbols can be used for the entry. The comparative value must be typed in the field.



Figure 114: Enter Comparative Value

- **Logical Operators:** The logical operators **AND** and **OR** enable the entering of further report conditions. *Please refer to the [Figure 111](#).*



Figure 115: Logical Operators

Attributes to be Printed – Available Attributes

The selection of available attributes is dependent on the plantype that has been selected at **Search to Plan Type**.



- 1) Use the (double arrow to the right) button to move the selected attribute into the **Attribute in Report Table** window.



- 2) Use the (double arrow to the left) button to remove the attribute from the **Attribute in Report Table** window.

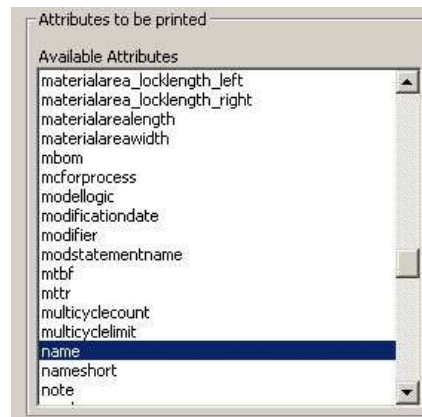


Figure 116: Selecting the Attribute for Printing

Attributes to be Printed –Attributes in Report Table

It is through these attributes that the printing of the table actually takes place. An attribute must be provided in this window to print data for the selected plantypes. These attributes are available as print variables in List and Label. These print variables are used to create the table in List and Label. These buttons can be used to change the order:

- 1) The (up arrow) button can be used to move the selected attribute further upward.
- 2) Use the (down arrow) button to move the selected attribute further down.

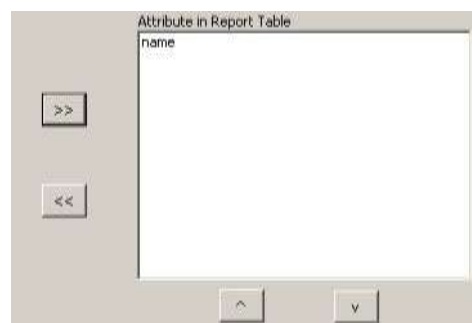


Figure 117: Attribute for the Table Printout

Parameters for Recursive Reports

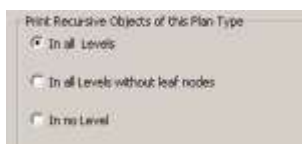
A recursive plantype must be selected to use the three parameters for reports. This option is not available for all other plantypes and is therefore displayed in grey. A so-called recursive plantype is a plantype that can recreate its own plantype structure: for example, the **Operation** process plantype.

Recursive Plantype

A recursive plantype makes it possible to create a large hierarchical structure of any size using the same plantype. Each plantype created recursively has that same plantype as parent.

Example - the Operation Plantype is a Recursive Plantype

The example shows what exactly is meant by a recursively created plantype. Please refer to the [Using Recursive Plantypes for the Report](#).



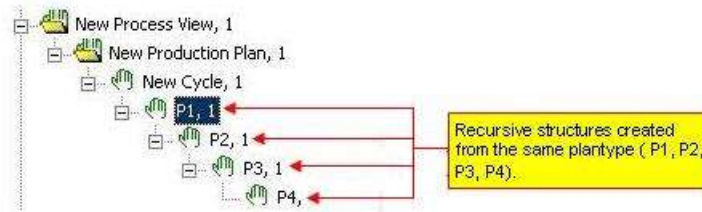


Figure 118: Example of Structures Created Recursively

Print Recursive Objects of this Plantype

These three options enable the recursive data structures (product, process, and resource structures) to be included in the report and the table printout. In the example recursive plantypes are used to create a process structure.

This would also apply if recursive plantypes from the product or resource structure are used in the data structure report.

▪ The In all Levels Option

All recursive plantypes within the hierarchy of the data structure are included in the report.

☒ In all Levels

- 1) By activating the **In all Levels** options all recursive plantypes are included in the report. The plantypes P1, P2, P3, and P4 in this example.



Figure 119: Plantypes

▪ The In all Levels without Leaf Nodes Option

This option makes it possible to refine the report to recursive plantypes. In a structure that is based on a recursive, the lowest level in the Hierarchy is excluded from the report.

☐ In all Levels without leaf nodes

- 1) If the **In all Levels without leaf nodes** option is activated, the last level of the recursive plantype being used is excluded from the report. It is the P4 process in the example.

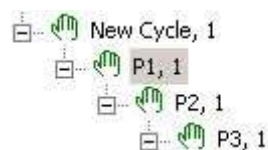


Figure 120: P4 Process

▪ The In no Level Option

This option simplifies the report even further. In this case, only the first recursive plantype of a hierarchy is included in the structure – any children are excluded from the report with this option.

☐ In no Level

- 1) If the **In no Level** option is activated, only the first level of the recursive plantype is used in the report. In the example only the P1 process is included in the report.



Figure 121: P1 Process

4.2.3 Starting the Object Wizard function

- 1) Open the system library, select the planttype from the planttype set for which the report is to be created.



Note

Keep in mind that the print form created in List and Label must be for the same planttype.

- 2) Select **Tools < Print Form < Object Wizard** from the menu bar. The **Print Object** dialog opens.

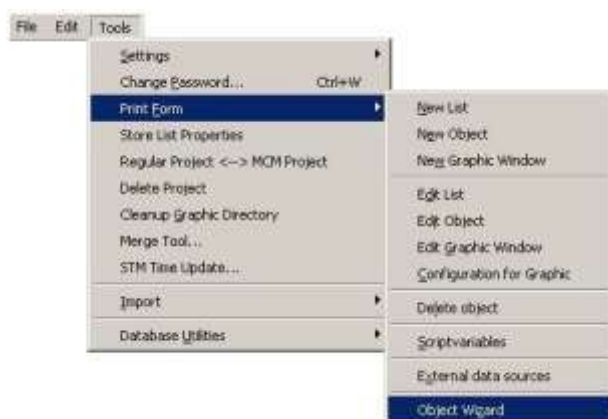
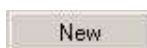


Figure 122: Starting the Object Wizard Function

- 3) The planttype selected in the system library is displayed in the dialog.
- 4) To create a new report, click **New** button. After clicking the New button, the next input window opens, *Please refer to the [Figure 125](#).*



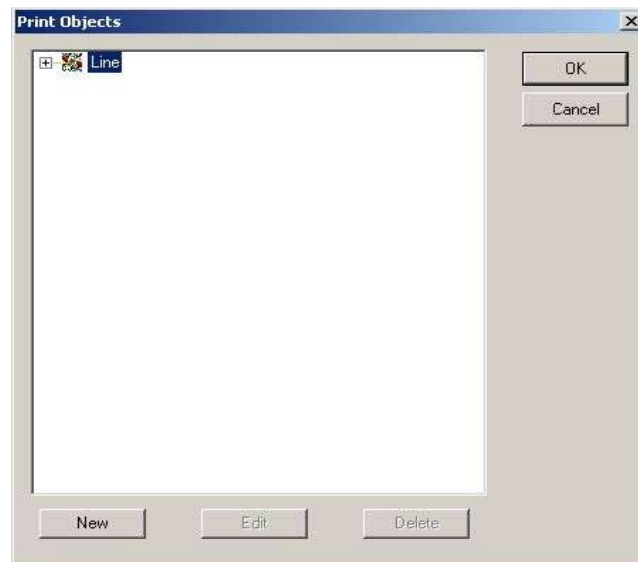


Figure 123: Print Objects Dialog

If other reports for this plantype have been created, they can be displayed in this dialog by clicking on the plus sign.

- Any existing reports can be edited or deleted using the **Edit** and **Delete** button respectively.
 - To edit or delete a report, the report must be selected in the dialog.
- 4) Starting with the first page of a report, the **Edit** button is active. To edit a report click plus sign of the report until the page to be edited appears.

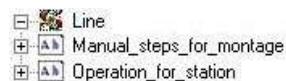


Figure 124: Reports that have been created are Shown

4.2.3.1 Entering the Name of the Report

- 1) The name of the report should be entered here. The report can be displayed and opened for editing under this name.
 - The name used for a report must be unique. If a name is entered that already exists, a message appears stating this.
- 2) After confirming the entry with **OK**, the **Author Print Object** dialog is opened. This dialog enables the designing of each page of the report. *Please refer to the [Designing the Pages of the Data Structure in the Input Window](#).*

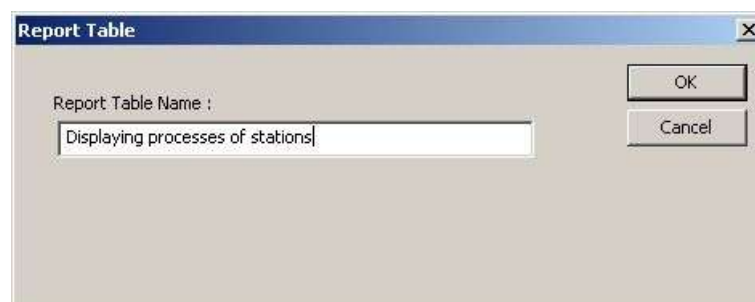


Figure 125: Entering the Name of the Report

4.2.4 Designing the Pages of the Data Structure in the Input Window

After the **Object Wizard** function has been started, it is time to make some of the key decisions for a data structure report. For instance, for which plantypes should the report be and which object data should be reported on and displayed in the report.

Due to the fact that a report can be created for a variety of different plantypes, a simple example will be used to help users get to know the procedure basics. The procedure demonstrated here can be used for any other reports, created using the **Object Wizard** function. Keep in mind that an individual page should be created in the **Author Print Object** dialog, for each plantype selected to be included in the report.

As a basic rule: All possible relations between plantypes are included in the report. This includes but is not limited to the parent-child relationships of a hierarchical structure or the relations between different plantypes that are created with the assistance of a relation, such as those between resource and process plantypes.

Relation Scheme and Report

The scheme below shows the basic relations that are possible in the report. The direction of the arrows shows the order in which the query and report I take place. According to the same method, an extended version of the same scheme should be used for larger structures. A scheme for the printout should always be created before the function is started. Otherwise, the possibility exists that the wrong data could be supplied to the data structure being created.

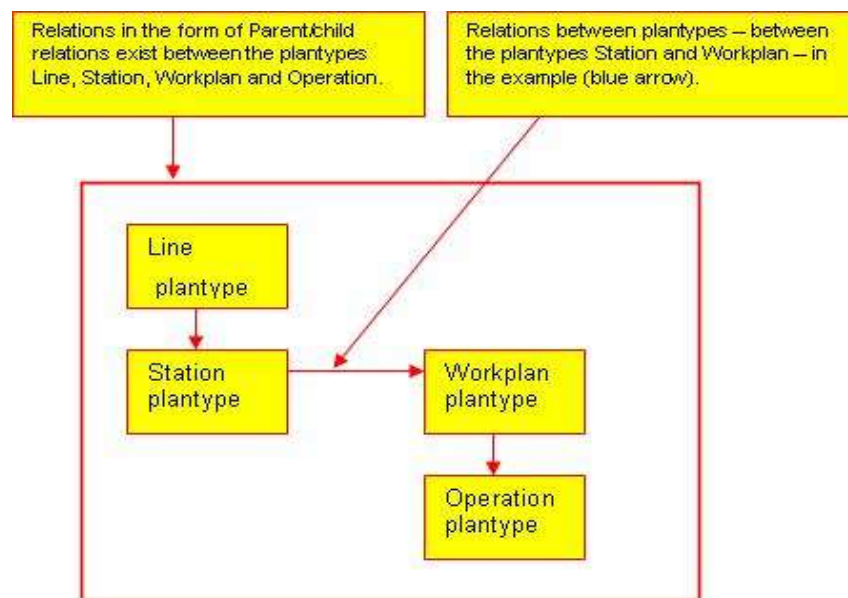


Figure 126: Relations Scheme

4.2.4.1 Designing the First Page of the Report – Display Stations

The design of the individual pages of the report must follow the scheme created before starting – all stations and processes of a line are to be displayed in this example.

The first step is to find all of the stations in a line:

- 1) After the **Object Wizard** function has been started for the **Line** plantype, select Station from **Search to Plantype**.

- 2) Enter the name of the page in **Condition Name**. If the search is to be refined even further, enter the conditions under **Search Condition**.
- 3) For this example only the names of the stations found are to be displayed; the *name* attribute should be added to the **Report Table** window.
- 4) Click **Next Level** button to design the next page.

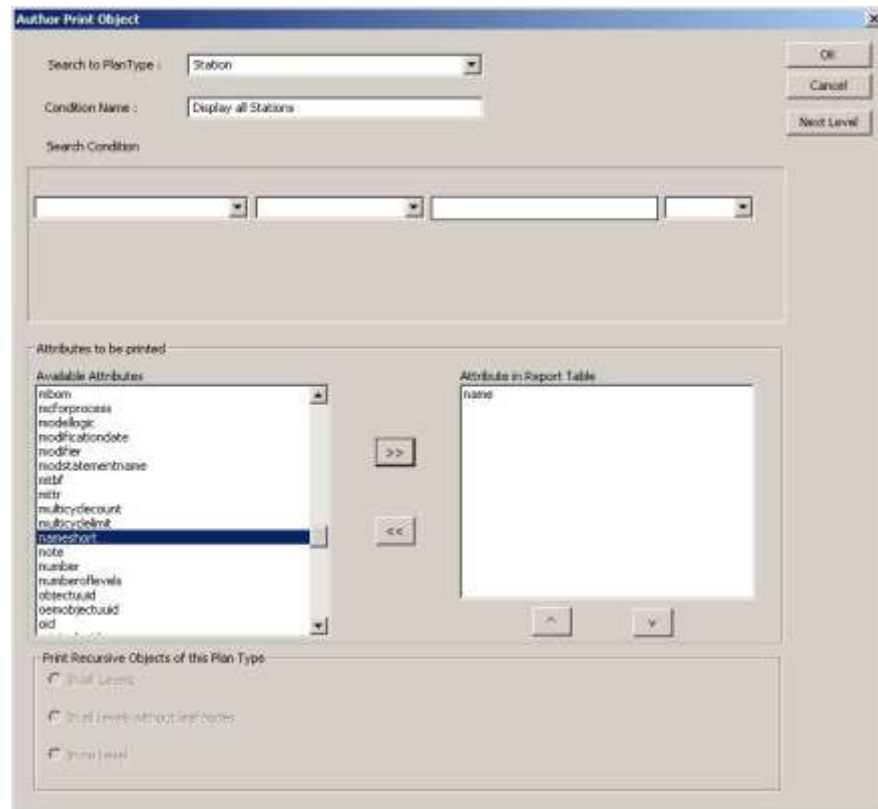


Figure 127: Example of the First Page – Search for Stations

4.2.4.2 Designing the Second Page of the Report – Creating Links to Workplan

To find all of the processes that are used on the stations of a line, the relationship to the process structure must be made here on the second page. This must be an actual relation also found in the PPR-Navigator.

In the example the processes are assigned to the **Workplan** plantype.

- 1) Select Workplan from **Search to Plantype** and enter the name under **Condition name**. This is the final step in this example. This page is finished, since the only items displayed in this example, are the processes. This of course means that the Workplan plantype are not displayed in the printout.
- 2) Click **Next Level** button to create the next page.

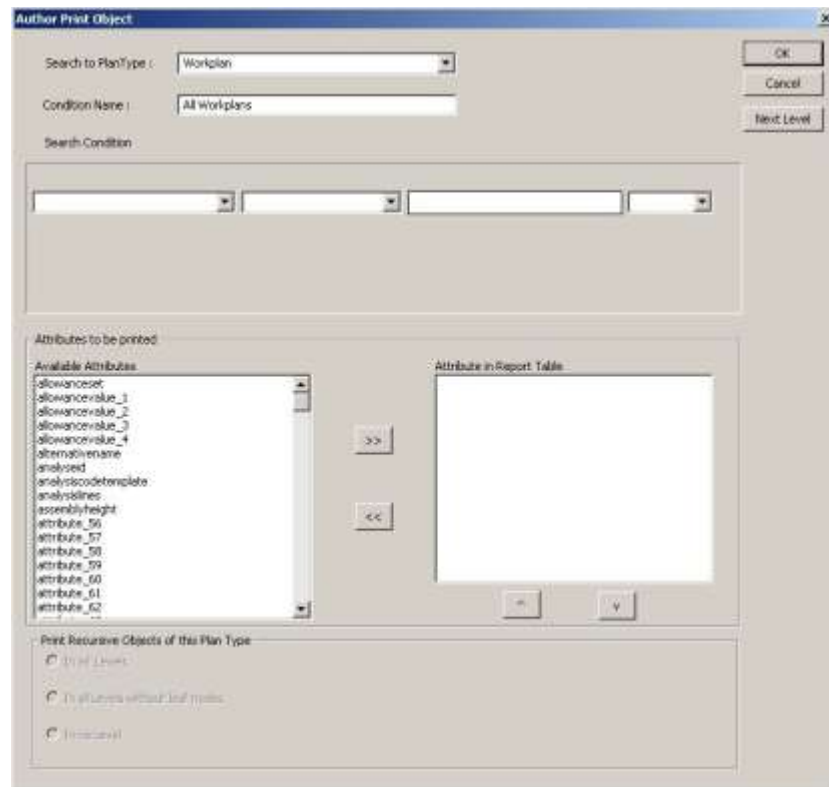


Figure 128: Example of the Second Page – Creating Relations to Workplan

4.2.4.3 Designing the Third Page of the Report – Display Processes

First of all, a small summary of the steps that have already been carried out:

- The function has been started for the Line plantype.
- The first page for the **Station** plantype was made, thereby creating the parent/child relationship between the plantypes **Line** and **Station**.
- On the second page, the link to the process structure was created and therefore also the relation between the plantype **Station** (resource plantype) and the plantype **Workplan** (process plantype).

The third and final page of our example also involves a parent/child relationship. All of the processes that are used in the stations by way of the **Station/Workplan** relation are to be found. The processes are from the **Operation** plantype.

The process name and process time of the process found are to be displayed.

- 1) Select the **Operation** plantype.
- 2) Add the **name** and **time** attributes to the Report Table to display the time and name.
- 3) Click **OK** to finalize the report. Save the report.

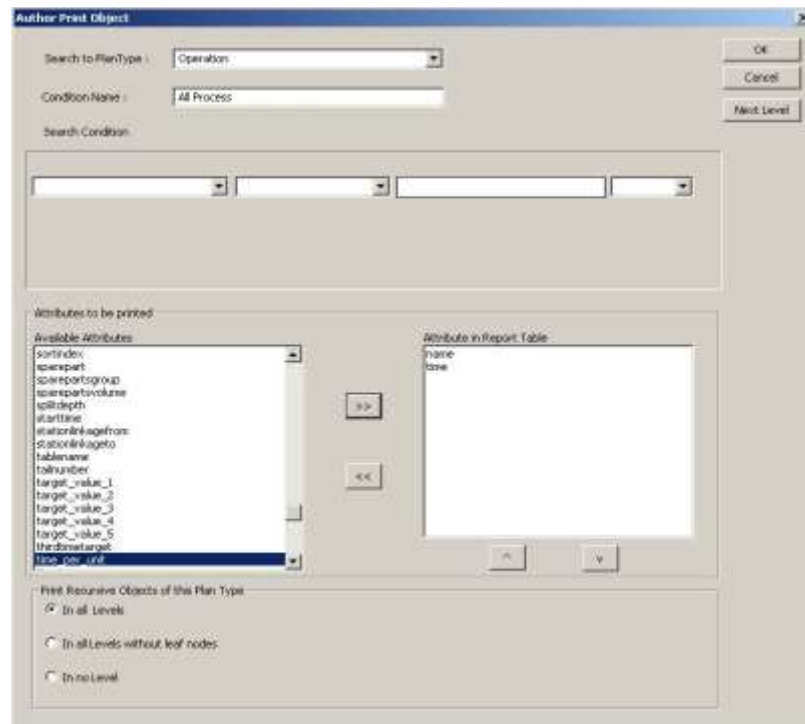


Figure 129: Example of the third Page – Display Processes

4.2.4.4 Display Result of the Report in the Print Object Dialog

- 1) The report is displayed in the **Print Object** dialog. To have all of the created pages displayed, click plus sign. The hierarchical structure of the opened report reflects the previously created scheme, and therefore the procedure for creating the pages also.

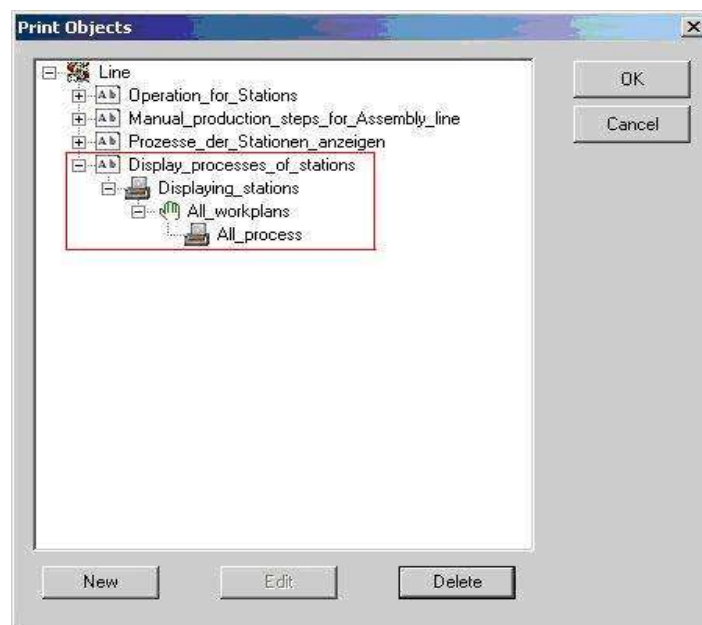


Figure 130: Created Reports are Displayed

Result of the Report in Printed Form

The print form is based on the report that is considered in the foregoing pages.

- 2) To save this printout, open the print form on the **Line** plantype in the PPR-Navigator.

Stations		
	Processname	Processtime
Station 0101		
	P1	0.000000
	P2	45.000000
	P3	60.000000
	P4	25.000000
Station 0102		
	P2-1	10.000000
	P2-2	12.000000
	P2-3	21.000000
Station 0103		
	P1	0.000000
	P2	45.000000
	P3	60.000000
	P4	25.000000
Station 0104		
	P2-1	10.000000
	P2-2	12.000000
	P2-3	21.000000

Figure 131: Example of a Report – Line with Stations

4.2.5 Showing Relationship in Object Wizard

Perform the following steps to specify the desired relation in **Search to Plan Type** field in Object wizard;

- 1) Select the checkbox "Show relations in object wizard", **Tools < Settings < Change < Printing < Show relations in object wizard.**
- 2) Open Author print object view by selecting desired plantype in system library and clicking **Tools < PrintForm < Objectwizard.**
- 3) The relation combobox is enabled in the **Author Print Object** dialog and it lists the possible relations for the selected plan type, for example, target plan types based upon the relation are listed in the "Search to Plan Type" at Next Level.

- 3) Under WorkPlan 1.3 the station filed is blank, denotes that there are no stations related to WorkPlan 1.3 with relation “process uses resource”.
- 4) On Process Plan, Create access Objects and create a new report table, via **Tools < PrintForm < ObjectWizard**.

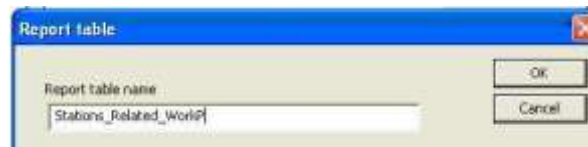


Figure 134: Report Table

- 5) Fill the **Author View** dialog, [Figure 132](#). Select WorkPlan in **Search to Plan type**, as soon as you select the plan type the **Select Relationship** combo is enabled which list all the possible relations that can be associated with Work Plans.
Select **Process uses Resource** as relation.
- 6) Fill the other details in the dialog and click **Next Level**.
- 7) In Next Level dialog select the desired Target Plan Type in **Search to Plantype**. Select **Station** as Plan Type and fill the attributes to be printed. In this example if the relation between WorkPlan and Station is not selected then, all the Stations under WorkPlan irrespective of the relations are shown.
- 8) Create a new Print Form on Process plan. Select Process plan in the Library and click **Tools < PrintForm < New Object**.
- 9) The designer pops up. Make sure that access object created is available in the designer. Drag and drop the required attributes. Save the form and exit the designer.

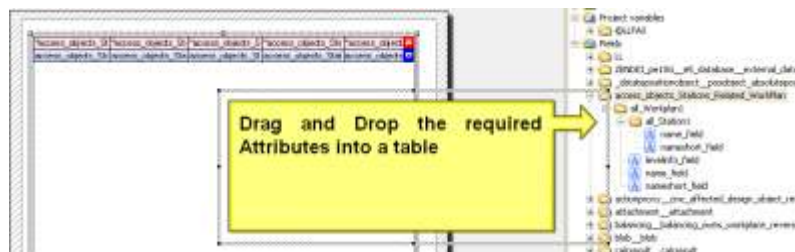


Figure 135: Dearg and Drop in Designer

- 10) Select Print/Preview on the Process Plan in the project.
- 11) The resultant report is as shown below.

access_objects_Stz	access_objects_Sta	access_objects_St	access_objects_Stat	access_objects_S
Workplan1.1	New Process			1_Workplan
		Station1.1		2_Station
Workplan1.2				1_Workplan
		Station2		2_Station
Workplan1.3				1_Workplan
Workplan1.4				1_Workplan
		Station4		2_Station

Figure 136: Final Report View

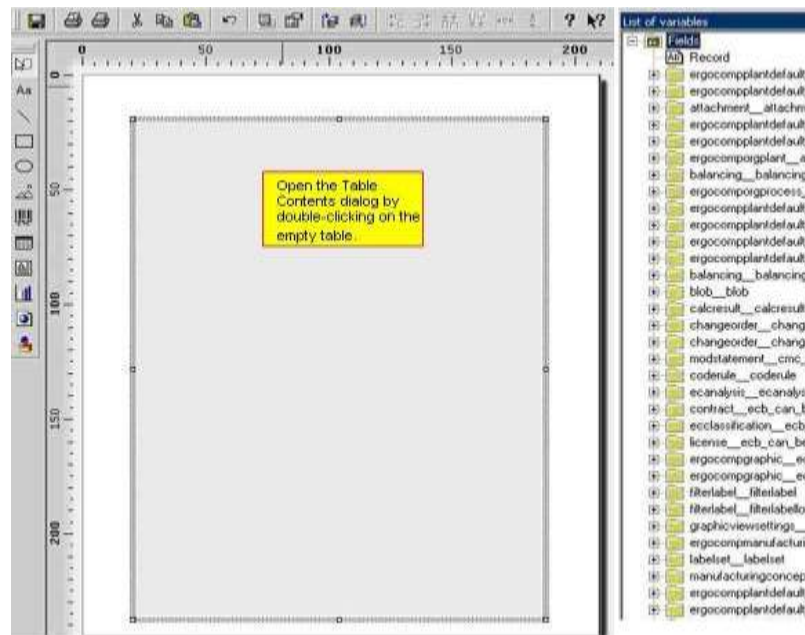


Figure 138: Empty Table in the Preview Window of List and Label

The **name** print variable is used in this example for the report of the stations.

The two print variables **name** and **time** are used for printing the process.

The print variables are assigned to the table from the corresponding directory.

4.2.6.3 Assigning Print Variables from Directories

- 1) Select the print variable from the directory and position it in the table using drag and drop.

The directory displayed reflects the structure of the report created.

- The print variables created for stations can be found under **Display all stations - the name_field** print variable.
- The print variables created for processes can be found under **All processes - the name_field** and **time_field** print variables.

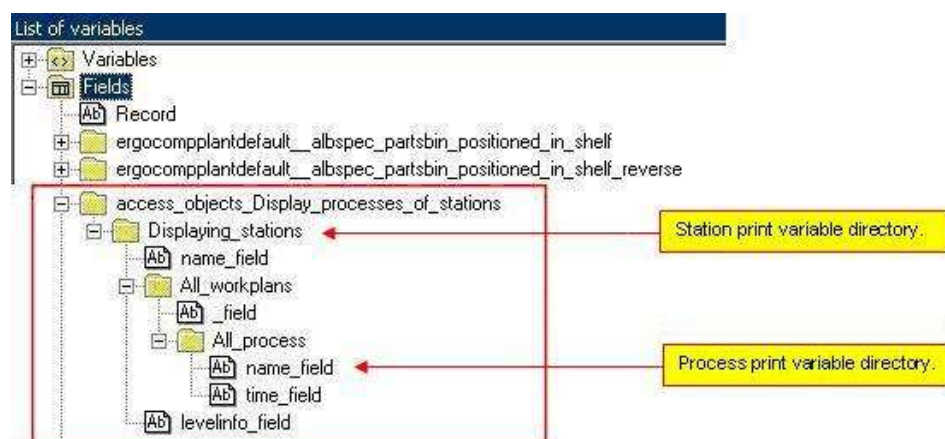


Figure 139: Assigning Print Variables to the Table using Drag & Drop

4.2.6.4 Assigning Print Variables from the Dialog

- 1) Open the **Table Contents** dialog by double-clicking on the empty table or on a cell already found there.

- 2) The directory for print variables is displayed under **Edit Table**.
 - 3) Once again double-click on the empty line displayed in the **Line** window, to open the **Edit Table** dialog.
- The dialog can also be opened for editing by double-clicking on any definitions for data lines or header lines that are present.

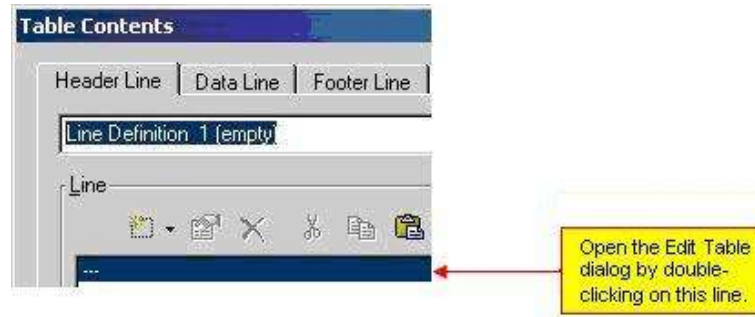


Figure 140: Table Contents Dialog

- 4) Open directory under the **Variables and Functions** tab and insert the print variables into the table.

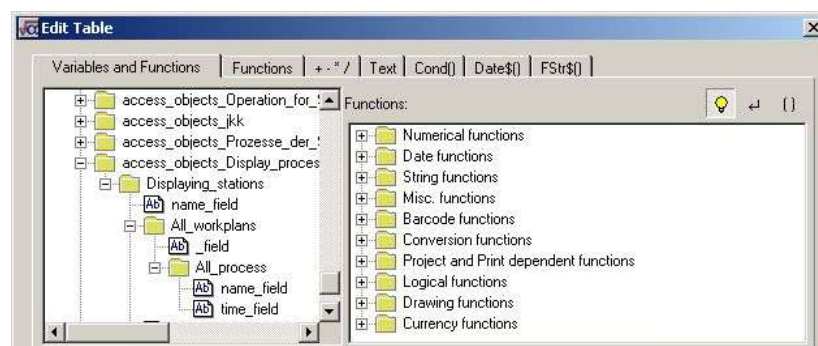


Figure 141: Inserting the Print Variables into the Table

4.2.6.5 Creating Line Definitions

The following example shows the settings for defining header lines and data lines. For this example, three header lines are created for Station names, Processname, and Processtime, followed by the corresponding data lines to show the results of the report on the table. A print form must always conform to the data structure of the report.

Setting Blank Spaces

Give the report a well structured design by using the definition for empty cells to create indentations for the header and data lines.

Stations		
	Processname	Processtime

Line definition two contains empty cells for the Processname and Processtime headers in the example.

Figure 142: Empty Cells

Empty cells are created using two quotation marks. As is the case with other data lines, the width of an empty cell in the table can be set by adjusting the field width or by sliding the columns.

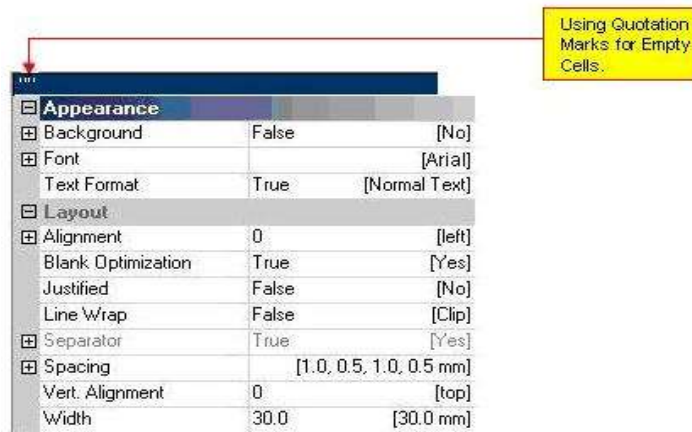


Figure 143: Setting Empty Cells

4.2.6.6 Determining the Line Definition for Stations

The header lines are fixed in size and are used for each report. The size of data lines vary and are changeable. Variable sizes thus depend on the area for which the report, and therefore the printout, is to be generated. The report in this example should be on the data structures in the **Line** plantype.

The variable sizes is shown in the following example:

- Line A has 10 stations, so 10 stations appears on the printout.
- Line B, by contrast, has only five stations, so only five appears on the printout.

The Figure 144 shows the defined header line for this example:

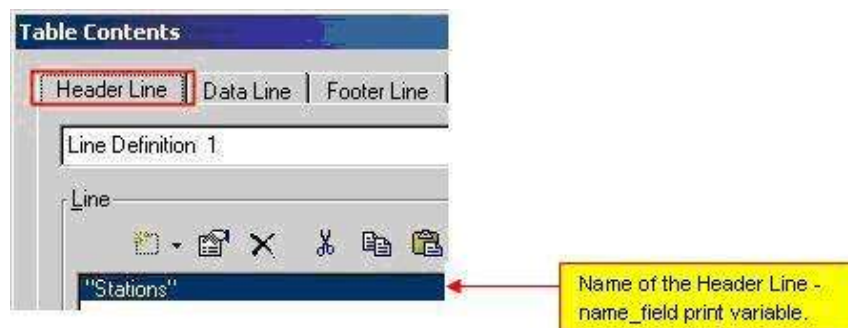


Figure 144: Example – Definition for the Stations header line

The Figure 145 shows the defined data line for this example:

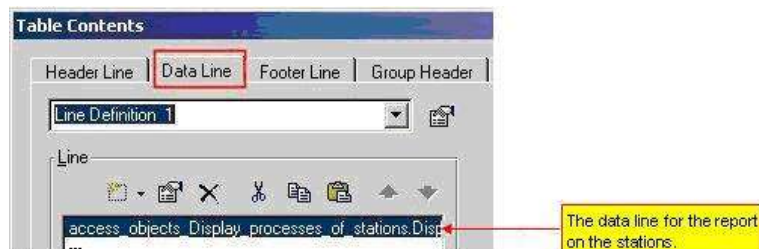


Figure 145: Example – Definition for the Stations Data Line

4.2.6.7 Determining the Line Definition for Processes

Use the Line definition 2 to create the definition for the header lines and data lines of processes. The [Figure 146](#) shows the defined header line for this example.

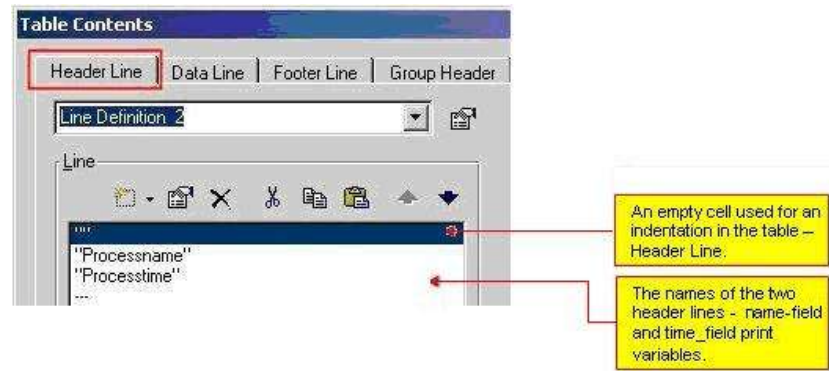


Figure 146: Example – Definition for the Processes Header line

The [Figure 147](#) shows the defined data lines for this example.

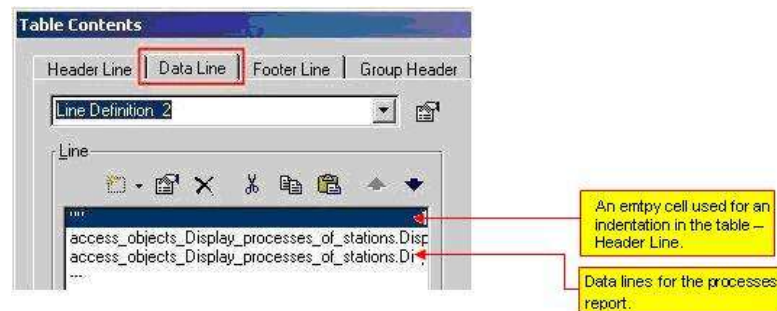


Figure 147: Example – Definition for the Processes Data Line

List of Figures

Figure 1: Print using the File Menu.....	3
Figure 2: Print Contextual Menu	4
Figure 3: Properties Dialog of a Process.....	5
Figure 4: GenericList Settings Dialog	5
Figure 5: Printing a List	6
Figure 6: Dialog for Generating Generic List Forms	7
Figure 7: Sorting Table Columns.....	8
Figure 8: Configuration of List Printing	8
Figure 9: Settings for Printing of Lists – Preset Path	9
Figure 10: Set Directory Path for Print Forms	10
Figure 11: Set Path for Print Forms.....	10
Figure 12: Message –Path Set does not Exist	10
Figure 13: No Path Set	11
Figure 14: Settings for Selective Printing of Lists.....	11
Figure 15: Select Objects in the Listview	12
Figure 16: Selected Objects are Printed – Example Listview	12
Figure 17: Make Selections in the List View of the Finder	13
Figure 18: Selected Objects are Printed – Example Finder	13
Figure 19: Message – Displayed when a Path Set does not Exist	14
Figure 20: No Path Set	14
Figure 21: Graphic Window with Adjusted Settings	15
Figure 22: File – Print Menu	15
Figure 23: Example of a Graphic Window Printout – Print Preview	16
Figure 24: Set Path for Directory	16
Figure 25: Set Path for Print Forms.....	16
Figure 26: Path Setting with Pre-Selected Print Form	17
Figure 27: Select Directory for Print Form.....	17
Figure 28: Select Setting for the Printout	18
Figure 29: Object Printing – Several Graphics in a Table.....	19
Figure 30: Select Resources in the Listview	19
Figure 31: Print Graphics of Selected Resources.....	20
Figure 32: Select View for the Printout	20
Figure 33: Example - Graphics for Selected Resources.....	21
Figure 34: Dialog – Print Option Tree.....	22
Figure 35: Menu Item Print	22
Figure 36: Menu Item Print Preview	23

Figure 37: Example for List Layout	23
Figure 38: Example of Tree Structure in PPR-Navigator	24
Figure 39: Tree structure as shown in the PPR-Navigator	24
Figure 40: Example for Expand all Levels - List Layout.....	25
Figure 41: Example Expand Number of Levels.....	25
Figure 42: Option for Excluding Plan Types	26
Figure 43: Examples for Excluding Plan Types from the Print Out	26
Figure 44: Contextual Menu for Print and Print Preview	27
Figure 45: The Print Form Menu	28
Figure 46: Error Message for Incorrect Path Declaration of Form Binaries	29
Figure 47: SQL Script	30
Figure 48: Create a New Print Form for Objects in the System Library	31
Figure 49: Overview – List & Label – Window	31
Figure 50: List of Variables	32
Figure 51: Setting the Page Layout	33
Figure 52: Designer Toolbar	34
Figure 53: The Table Object	34
Figure 54: Creating and Editing Headers, Lines of Data and Footers	35
Figure 55: Additional Menu Point in Designer for Sorting Table Columns	35
Figure 56: Sorting Table Columns.....	36
Figure 57: Sorting Table Columns.....	36
Figure 58: Editing a Text Field.....	37
Figure 59: Inserting a Drawing	37
Figure 60: Diagram Properties Dialog	38
Figure 61: Example of a Pie Diagram	39
Figure 62: Enter Name for Table	40
Figure 63: Sum Variables Menu	40
Figure 64: Creating a Sum Variable	41
Figure 65: Inserting a Footer	42
Figure 66: Linking a Text Field to a Table	43
Figure 67: Defining Script Variables	43
Figure 68: Defining Script Variables	44
Figure 69: Example – Script for a Script Variable	45
Figure 70: Example – Script for Script Fields	45
Figure 71: Example – Script for Script Tables.....	47
Figure 72: Example – Structure in the PPR Navigator Resources, Product, Parts.....	48
Figure 73: Dialog – for Creating Group Header and Data Lines	50
Figure 74: String Function with Assigned Script Variable	51

Figure 75: Script Variable for the Registration in the Designer.....	51
Figure 76: Set the Width for the Script Variable to Zero	52
Figure 77: Example for Structuring in the Designer	52
Figure 78: Example for a Printout	53
Figure 79: Displaying Scripttable	55
Figure 80: Select Page Setup Menu Item	57
Figure 81: Set Page Orientation	57
Figure 82: Hierarchical Structures for Print Forms.....	58
Figure 83: Parent Directory in List and Label	58
Figure 84: Ergoproject Directory.....	59
Figure 85: Example - Site Print Form	59
Figure 86: Select Print Preview	59
Figure 87: Print Preview of Hierarchical Structure	60
Figure 88: Access to other Data Sources Dialog	61
Figure 89: Starting the Connection to the External Database	62
Figure 90: Entering Parameters in Access to Other Data Source Dialog.....	62
Figure 91: Click Cancel Button	63
Figure 92: Dialog where Created Queries are Displayed	64
Figure 93: Type in Report	64
Figure94: The Directory for the Variable Type is in List & Label	64
Figure 95: Directory in List and Label for Field Type	65
Figure 96: Enter the Name for Query	65
Figure 97: Show Variable in List and Label.....	65
Figure 98: Displaying Results	66
Figure 99: Example – All projects Application Area	67
Figure 100: Syntax for a Query	68
Figure 101: Display of the Results.....	68
Figure 102: Database Table	69
Figure 103: Assigning the Attribute of a Plantype	70
Figure 104: Result of a Query Containing an Attribute	70
Figure 105: Selecting the Query for Editing or Deleting.....	71
Figure 106: Editing a Query	71
Figure 107: Print the Data Structure Report.....	72
Figure 108: View of the Author Print Object dialog	73
Figure 109: Selecting a Plantype for a Page.....	74
Figure 110: Entering the Name of each Page of the Report.....	74
Figure 111: Entering the Report Parameters to Search for the Data	74
Figure 112: Selecting the Attribute	75

Figure 113: Selecting the Comparative Operators.....	75
Figure 114: Enter Comparative Value	75
Figure 115: Logical Operators	75
Figure 116: Selecting the Attribute for Printing	76
Figure 117: Attribute for the Table Printout	76
Figure 118: Example of Structures Created Recursively	77
Figure 119: Plantypes	77
Figure 120: P4 Process	77
Figure 121: P1 Process	78
Figure 122: Starting the Object Wizard Function	78
Figure 123: Print Objects Dialog.....	79
Figure 124: Reports that have been Created are Shown	79
Figure 125: Entering the Name of the Report	79
Figure 126: Relations Scheme	80
Figure 127: Example of the First Page – Search for Stations.....	81
Figure 128: Example of the Second Page – Creating Relations to Workplan.....	82
Figure 129: Example of the third Page – Display Processes.....	83
Figure 130: Created Reports are Displayed.....	83
Figure 131: Example of a Report – Line with Stations	84
Figure 132: Print Form Table.....	87
Figure 133: Empty Table in the Preview Window of List and Label	88
Figure 134: Assigning Print Variables to the Table using Drag & Drop.....	88
Figure 135: Table Contents Dialog	89
Figure 136: Inserting the Print Variables into the Table.....	89
Figure 137: Empty Cells	89
Figure 138: Setting Empty Cells	90
Figure 139: Example – Definition for the Stations header line.....	90
Figure 140: Example – Definition for the Stations Data Line	90
Figure 141: Example – Definition for the Processes Header line	91
Figure 142: Example – Definition for the Processes Data Line	91

List of Tables

Table 1: Columns Printing	9
Table 2: Values for Selective Printing of Lists	12
Table 3: Functionalities for Editing or Creating Print Forms	28
Table 4: Script Variables – Structured Table Set-up for Scripts	48
Table 5: Script variables – Structured Table set-up for the Print Form	48

Index

C

Creating a Table in the Designer	49
Creating Print Forms	29
Creating Tables	34
Creating Text Fields	37

D

Druckformulare auswerten.....	40
-------------------------------	----

H

Hierarchical Structures	
Creating Print Forms	58
Ergoproject Directory	59
Example:	59
Parent Directory	58
Printing of PPR Structures	58
Using the Option	58

I

Insert Drawings	37
-----------------------	----

N

Nonliability	ii
--------------------	----

O

Object Wizard	
Designing the Report	81
Example of a Printout.....	73
General Information.....	72
Getting to Know the Author Print Object View	74
Selecting Plantypes	74
Starting the Object Wizard	79
Using Parameters for Recursive Planning.....	77

P

Page Format for Print Forms	
General Information.....	57
Set Portrait or Landscape Format.....	57
Path for Printing of Graphics	
Making Settings.....	13
Path for Printing of Lists	
General Information.....	9
Making Settings.....	9
Print Forms	
Creating Print Forms	3
Editing Print Forms in the Designer.....	31

Generic Print Forms.....	28
Modifying Print Forms	29
Print Forms for Objects	30
Printing Print Forms	3
Print Forms – Editing Print Forms.....	28
Print Graphic Window.....	14
Print Graphics of Selected Resources	
Display Print Preview	20
General Information.....	19
Printing Graphics.....	20
Select Resources	19
Print Lists of Selected Objects	
Examples:.....	12
General Information.....	11
Printing Lists in the DELMIA Process	
Engineer.....	6
Printing Object Graphic	17
Printing Tree Structures	
Do not Expand Plan Types Option.....	25
Expand Level Option	24
Print.....	22
Print Mode Option	23
Print Preview	23
Printing Options	21
Printing Tree Structures	26
Using Options	22

Q

Queries for External Databases	
All Projects Application Area	68
Assigning Data	66
Create Link.....	62
Creating New Queries	64
Determining the Type of Report.....	65
Editing or Deleting Queries.....	71
General Information.....	62
Plantype Application Area.....	70
Using Syntax.....	67

S

Script Variables for Printing.....	43
------------------------------------	----

T

Tables for Script Variables	48
Tools for Print Forms.....	33
Types of Printing in Process Engineer.....	5

U

Using Script Variables	44
------------------------------	----