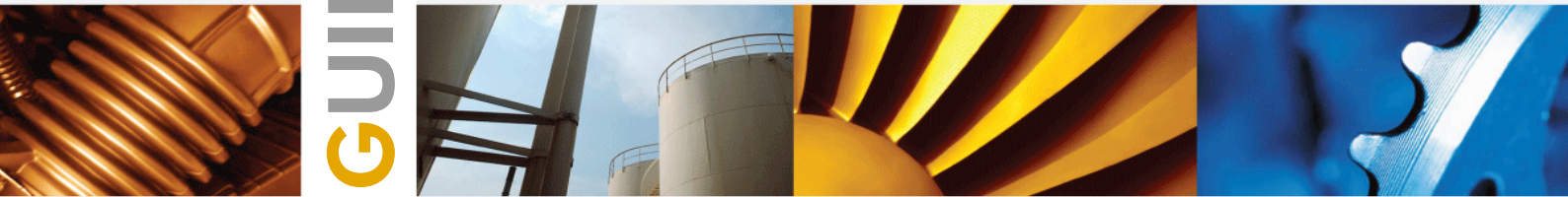


MEDUSA⁴

VERSION 5.2

Administration

GUIDE



All rights reserved. No part of this documentation may be reproduced in any manner (print, photocopy or other) without the written permission of CAD Schroer GmbH.

CAD Schroer GmbH has made its best effort to ensure that the information in this document is accurate and reliable, but cannot guarantee the accuracy, timeliness, reliability or completeness of any of the information contained herein. CAD Schroer GmbH will not make any warranty nor accept legal responsibility or liability of any kind for consequences resulting from errors or omissions.

Registered Trademarks of CAD Schroer GmbH:
MEDUSA, STHENO

Trademarks of CAD Schroer GmbH:
MEDUSA₄, STHENO/PRO, MEDEA, MPDS

Third-Party Products and Trademarks:
Pro/ENGINEER, Pro/DETAIL and Pro/TOOLKIT are registered trademarks of Parametric Technology Corporation.

All other brand or product names are trademarks or registered trademarks of their respective owners.

April 2013

Copyright © CAD Schroer GmbH

Germany

CAD Schroer GmbH
Fritz-Peters-Str. 26-30
47447 Moers

Tel. +49 2841 91 84 - 0
Fax +49 2841 91 84 - 44
e-mail: info@cad-schroer.de
www.cad-schroer.de

TABLE OF CONTENTS

Preface	5
Installation	7
Installing MEDUSA.	8
Uninstalling.	10
Configuring MEDUSA	11
Description of a Project	12
Description of MEDCONFIG	13
Using MEDCONFIG.	16
Overview of Configuring a Project	20
Before You Start Configuration	22
The Product List.	25
Step 1 - Creating a Product List.	26
Step 2 - Configuring the MEDUSA Products	29
Administration	35
Administrator Mode	36
Creating and Editing MEDUSA Styles	37
Setting up Standard Colors	43
Setting up Standard Boldness (Thickness) Values	46
Setting up Boldness for Plotting Dimensions	47
Setting up Standard Sheet Filename	48
Setting up the Page Label	51
Setting up the Sheet Header Editor	52
Setting up Symbol Manager.	62
Layer Set Management	66

Customize Layers	70
Customize Layer Sets	71
User-Defined Dashboards	74
Dashboard Element Selection Limits	83
Create Balloon Text Symbols	85
Define Used Codepage	91
Index	93

PREFACE

Book Conventions

The following table illustrates and explains conventions used in writing about MEDUSA applications.

Convention	Example	Explanation
Menu	Choose <i>Zoom</i> from the View menu Add button Choose the tool <i>Creates thin solid lines.</i>	Indicates a command, function or button that you can choose from a menu, dialog or tooltray.
Syntax	<code>acos 0.345</code> The <code>ciaddobj</code> command Return or Control-g	User input, commands, keywords and keys to press on a keyboard.
SyntaxBold	Enter command> plot_config	Where system output and user input are mixed, user input is in bold.
<i>SyntaxItalic</i>	<code>tar -cvf /dev/rst0 filename</code>	Supply an appropriate substitute for each variable; for the given example replace <i>filename</i> with an actual file name.
<i>Filename&path</i>	<code>medusa\med2d\m2d\src\</code>	Shows path and filenames.
UPPERCASE	MEDUSA or CADCONVERT	Names of products.
<i>italic</i>	<i>left mouse button</i> <i>Drafting User Guide</i>	Indicates the buttons to press on a mouse and names of books.
bold	A temporary group is a collection of ...	Emphasize text.

Please note: Illustrations showing menus and forms are taken from a window system. The display for other platforms can differ slightly.

Online User Documentation (HTML)

Online documentation for each book is provided in HTML format. You can view this online documentation in the MEDUSA installation directory and directly by calling it up within the MEDUSA user interface.

Installation Directory

1. Navigate to the directory where MEDUSA is installed.
<MEDUSA installation directory>/meddoc/doc/<language>/ (Unix)
<MEDUSA installation directory>\meddoc\doc\<language>\ (Windows)
where <language> is either *english*, *german* or *french*.
2. Click on the file *mainmenu.htm*.
3. Click the book title you want to view.

MEDUSA Interface

1. Click left on the entry *Help* inside the main menu.
2. Choose MEDUSA Documentation from the pulldown menu.
A browser opens showing the *mainmenu.htm* listing all available documents.

Printing Version of the Documentation (PDF)

A PDF (Portable Document Format) file is included for each online book. You must have Acrobat Reader installed to view and print PDF files. If you don't have the Acrobat Reader, you can download it for free from the Adobe homepage:

<http://www.adobe.com/products/acrobat/readstep.html>

In order to search for a keyword across PDF files you can use **Acrobat Reader**. Therefore **Acrobat Reader Version 6.0** or higher has to be installed. The Reader provides a multiple search function, i.e. you can specify complete directories containing several PDF files for searching.

INSTALLATION

This chapter describes how to install MEDUSA.

- [Installing MEDUSA](#) 8
- [Uninstalling](#) 10

Installing MEDUSA

This section gives some hints which are helpful for the installation of MEDUSA with and without the MEDUSA Information System (MEDInfo).

Before you start the installation on a Windows system, please close all applications.

Please note: The installation has to be executed by a user with administrator rights.

The installation is done by the installation assistant which will guide you through the setup process. If errors occur during installation, these are reported in the file *log.txt* in the installation path.

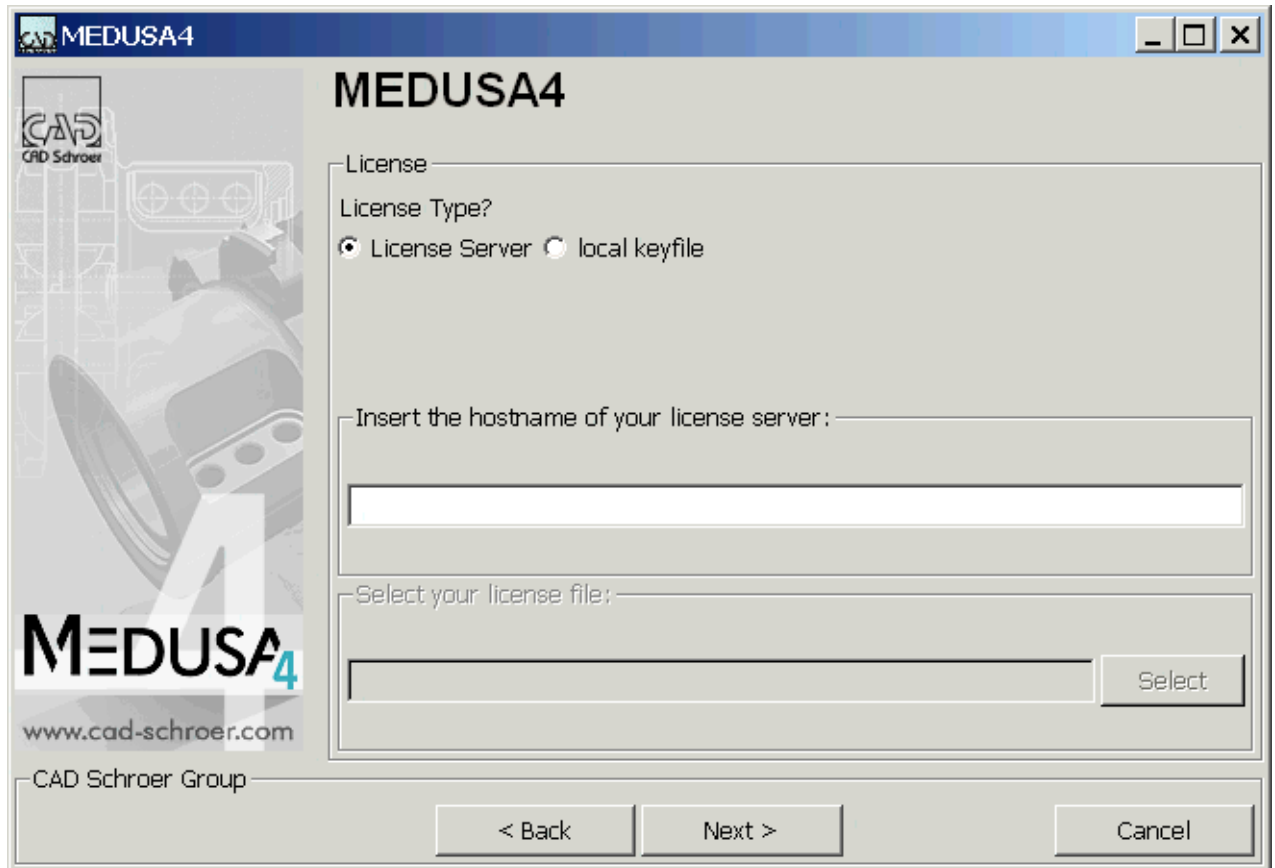
Installation without MEDInfo

After you have started the installation, follow the steps of the installation assistant. During the installation you will be prompted for relevant information, for example, language and installation directory. Most of the dialogs are self-explanatory but you should consider the following:

- The installation path may not contain spaces.
- You can decide between a local license file and a license server.
You choose `local keyfile` for license files which do not contain a `SERVER` and `DAEMON` line. If you choose a `License Server`, consider that it has to be configured separately. For information about the installation of the license server see the *License Management Guide, chapter Installation*.

With the definition of the license the specifications for any projects, are set. That is, the assignment of the variable `MEDUSA_LICENSE_PATH` will be defined (see the *login* file within the MEDUSA project).

Figure 1 Kind of License



Installation with MEDInfo

Precondition for MEDInfo is the installation of the Apache Web Server including the PHP environment. If there is no Apache web server installed and you want to use MEDInfo, go to the apache directory on the installation disc and launch *install.bat*.

Please note: MEDUSA and MEDInfo have to be installed on local harddisks, because MEDInfo uses services, which are available locally only. This is valid for the installation on a client as well as on a server.

MEDInfo can be installed any time after MEDUSA has been installed. To do this you only have to launch the installation executable from the installation disc and to choose the MEDInfo products.

Uninstalling

Windows

To uninstall the software open in your start menu the options

`Programs -> MEDUSA ->uninstall MEDUSA`

The deinstallation assistant is started which leads you through the deinstallation process. All components of the application and the whole MEDUSA product will be removed.

You also can uninstall using the options `Settings -> Control Panel` from your start menu and then choose the entry `Software`.

UNIX

Under UNIX run the tool `<installation path>/_uninstall/uninst.csh` to remove all MEDUSA components.

CONFIGURING MEDUSA

This chapter describes how you configure a MEDUSA project using the MEDUSA configuration utility called MEDCONFIG. You must have completed the installation procedures before working through this chapter.

In this chapter, you learn about the following subjects:

- Description of a Project..... 12
- Description of MEDCONFIG 13
- Using MEDCONFIG 16
- Overview of Configuring a Project 20
- Before You Start Configuration..... 22
- The Product List..... 25
- Step 1 - Creating a Product List 26
- Step 2 - Configuring the MEDUSA Products..... 29

Description of a Project

When you have finished installing MEDUSA (see “[Installation](#)” on page 7), you configure a project. A project provides the link between the installed MEDUSA products, and is one definition of how you want to run MEDUSA. This definition includes, for example, the choice of products you want to use, and which the description of workstations you want MEDUSA to run on.

As a reference you can use the project `<medusa>\master_project`. Do not change this project.

This following table shows the products of a MEDUSA installation in different projects.

Table 1 MEDUSA User Project Examples

Product	proj1	proj2	proj3
medsys	X	X	X
med2d	X	X	X
med3d			X
med3dprop			X
med3dshrink			X
med3dtran			X
med3dui			X
medshade			X
meddoc	X	X	X
medmech		X	

Please note: In case of a localized version, for example german, additional products like `medchangeger`, `medch3duiger`, `medchmechger` are necessary. The language is defined by the last characters of the products, `ger` for german. For other languages the suffix is appropriate, `fra` for französisch/french, `ja` for japanese and `it` for italian.

Each project shown in the previous table, `proj1` through `proj3`, has a different combination of MEDUSA products. Each project is independent, but the one thing they have in common is that they use the same source files within MEDUSA. For example, all three projects use the same `med2d` product but refer to it separately.

You can configure workstations to suit the hardware and MEDUSA products you intend to use with each project.

Description of MEDCONFIG

MEDCONFIG is a MEDUSA utility you use to configure a project with the MEDUSA products you have installed, as well as any Customization you have made to MEDUSA.

How MEDCONFIG Works

MEDCONFIG runs under the MEDUSA Executive, and uses information provided within the MEDUSA products to configure a project. When you start up MEDCONFIG, you give it the path-name of the project directory where it can write its information. MEDCONFIG then checks that the directory exists and is accessible, and then looks to see if the project has been configured already.

The following files are maintained by MEDCONFIG in the project directory:

login.bat	Is a shell script that sets up the search paths for the MEDUSA program startup files, and the project directory.
mxedef.bin mxedef.fasl	These files contain the binary definition of the product environment. For example, they contain information on which MEDUSA products are configured in the project, and where the plot queues and temporary workspace directories are located.

If these files are not present, then the project is new and needs to be configured as a new project. This is called an initial configuration and you are asked for some information by MEDCONFIG automatically.

If these files exist, MEDCONFIG recognizes a project as already configured, and therefore MEDCONFIG starts.

The Initial Configuration

If MEDCONFIG carries out an initial configuration, it asks you for these types of information:

- Name of the project directory
- List of products to include in the project (if a `product_list.dat` file does not exist)
- Temporary workspace
- Text string used for saving sheets and models

There are examples of running MEDCONFIG throughout this chapter where you can see exactly how MEDCONFIG prompts you for this information.

After an initial configuration, MEDCONFIG puts you at the interactive level. At this point, you may decide to quit from MEDCONFIG and save your configuration environment. This is advisable since any hardware or operating system crashes may cause you to lose all of the configuration session so far. You can then re-enter MEDCONFIG and configure workstations and other settings.

Completing a Project Configuration

When you have finished configuring the project, you quit from MEDCONFIG. This saves all the information you supplied into the files called `mgedef.bin`, `mgedef.fasl` and `login.bat`, which MEDCONFIG writes to your project directory. After this, you are ready to use MEDUSA.

After Configuration

After configuration perhaps you have to make same settings again, e.g. for online help .

Online Help - Documentation

The user guide and online help of MEDUSA is available in the languages English, German and French.

After finished configuring a project only the English online help is available. If you want to use a localized online help, you have to enter the following line into the `login.bat`:

```
set MED_LANG=<lang>
```

`<lang>` can be either `eng` (english), `ger` (german) or `fra` (french).

Please note: Reconfiguring a project (see [“Reconfigure a Project” on page 21](#)) also writes new the `login.bat`. Therefore you have to add the variable definition again after reconfiguration.

Environment Variables

Separators of Decimal Values: In order to be able to enter both as separator of decimal values, comma (, is default) or point (.), you have to set the following environment variables within the `login.bat` of the user project (`<medusa>\<userproject>\login.bat`):

```
set UIE_COMMA_IS_DECIMAL=1  
set MED_COMMA_IS_DEZIMAL=1
```

This effects, that a separator is always displayed as comma (,) in an input field, regardless of inserting comma or point.

Unicode Filename Support: To enable the extended unicode filename support on Windows the environment variable `MED_UNICODE_FILENAMES=1` must be set.

Please note: Reconfiguring a project (see [“Reconfigure a Project” on page 21](#)) also writes new the `login.bat`. Therefore you have to add the variable definition again after reconfiguration.

Using MEDCONFIG

This is a brief description of how you use MEDCONFIG. More detail is given with the individual steps of the configuration procedure.

Syntax

You enter the MEDCONFIG commands as shown below:

```
medconfig options
```

where *options* refers to the list below:

Option	Description
<code>project</code>	Starts MEDCONFIG in PROJECT mode, where <code>project</code> is the full pathname of the project.
<code>-admin</code>	Starts MEDCONFIG in ADMIN mode. This gives you the complete range of MEDCONFIG commands you use to compile fonts and menus, for example.
<code>-printdef filename</code>	Writes the definition of the MEDUSA environment into the file you specify in <code>filename</code> . This information includes the products configured in the project, and the variables used for configuration.
<code>-batch</code>	Starts MEDCONFIG as a batch job. Use this option with the <code>reconfigure</code> option. This option enables MEDCONFIG to run without user intervention as it assumes defaults for all configurable actions. This is not recommended for initial project configuration.
<code>-reconfigure</code>	Starts MEDCONFIG and allows you to reconfigure the project. This option uses existing configuration settings as the defaults and leaves you in ADMIN mode.
<code>-force</code>	Forces all configurations to recreate their associated binary files, whether or not the binary files are up to date. This option is used with <code>reconfigure</code> after you have added new products to an existing product list.
<code>-help</code>	Displays the available MEDCONFIG command line options.

MEDCONFIG Modes

MEDCONFIG runs in two modes, PROJECT and ADMIN, which give you access to sets of commands.

- PROJECT mode is the default mode; you enter this mode if you start MEDCONFIG without entering any options at all.
- ADMIN mode gives you additional MEDCONFIG commands which enable you to make changes to the compiled files within MEDCONFIG.

The table below shows which commands are available (Y) in these modes. The total set of commands available reflects the products you include in the project. For example, if you are not using MEDUSA DB, you do not see the `mdba_application_control` command.

Table 2 MEDCONFIG Commands

Command	Description	ADMIN	PROJ
<code>bold</code>	Builds the merged boldness.map	Y	
<code>code</code>	Builds the code binary	Y	
<code>ddl</code>	Builds the ddl binary	Y	
<code>defaults</code>	Builds the user interface defaults binary	Y	
<code>dfont</code>	Builds a decortext character font	Y	
<code>fits</code>	Builds the fits binary	Y	
<code>font</code>	Builds alternative character font	Y	
<code>mdba_application_control</code>	Accesses the MEDUSA DB Application Control Menu	Y	
<code>mdbc_rebuild_control</code>	Accesses the MEDUSA DB Customization Control Menu	Y	
<code>message</code>	Builds the message binary files	Y	
<code>patts</code>	Builds the patts binary	Y	
<code>plot_config</code>	Configure this project's use of plotting	Y	
<code>plotting_dev_kit</code>	Define new custom plotting products	-	-
<code>prims</code>	Builds the prims binary	Y	
<code>psyms</code>	Builds the psyms binary	Y	
<code>raster_path</code>	Resets or checks Raster Backdrop search path	Y	
<code>styles</code>	Create XML Style file	Y	
<code>ttable</code>	Builds the translation table binary	Y	Y
<code>uistrings</code>	Builds the user interface string binaries	Y	

Table 2 MEDCONFIG Commands

Command	Description	ADMIN	PROJ
wsconfig	Configures a workstation	Y	Y
xresources (UNIX only)	Builds the xresources directory	Y	
reconfigure	Reconfigures the project	Y	
bacis	Enters interactive Bacis2	Y	
quit	Quits the session	Y	Y
help or ?	Prints this list	Y	Y

Conventions Inside MEDCONFIG

MEDCONFIG asks a number of questions while it is running. Each question is followed by a default answer in angle brackets. For example:

```
Enter pathname of project <quit>:
```

You press `Return` to accept the default answer in angled brackets, or you type in your own response at the prompt.

Entering Information to MEDCONFIG

As you use MEDCONFIG, you find that if the program needs information, you are prompted for it. For example, if you mistype a pathname in the product list during configuration, MEDCONFIG tells you your information is not correct:

```
Enter pathname of product> c:\medusa\med2
mxe: product c:\medusa\med2 does not exist or is not accessible
Enter pathname of product>
```

If the information you supply is not suitable, or you have made an error, perhaps a typing mistake in a pathname, MEDCONFIG tells you there was an error and asks for the information again.

The Interactive Level of MEDCONFIG

You use the set of commands available in MEDCONFIG to build menus and configure workstations, and to compile customized files if you add them after the initial configuration.

When you quit from an interactive MEDCONFIG session, either save your configuration file, or abort if you need to do an initial configuration again.

To see the commands that are available, enter `help` in response to the `Enter command>` prompt in MEDCONFIG.

For further information on these commands, refer to the *MEDUSA Customization Guide*.

Aborting MEDCONFIG

If you need to stop MEDCONFIG without completing a configuration, you can use `control c`. After you entered `control c`, you can choose to abort the configuration or continue.

Overview of Configuring a Project

The following subsections give an overview of the procedure to follow to configure a MEDUSA project.

If you are familiar with this topic, you may find the information of this overview enough for configuring a project. If you doing project configuration the first time, you should work through this chapter in succession given below.

Configuring a New Project

When you configure a project, these are the general steps you follow:

1. You create a project directory to hold the configuration files and start `<medusa>\medsys\login` (see “[Accessing MEDCONFIG](#)” on page 24).
2. You start an initial configuration in MEDCONFIG and give the pathname of the project directory.
MEDCONFIG carries out an initial configuration, and starts by prompting you for, or reading in, the list of MEDUSA products that you want to include in the project. This list must include any plotting products you want to use with the project.
3. MEDCONFIG then generates the binary files for the project, and prompts you for any information it needs. MEDCONFIG informs you if it had any problems with part of the configuration. If there were any problems, you use the MEDCONFIG interactive commands to try parts of the configuration again.
4. MEDCONFIG asks you whether you want to link the 2D drafting image in order to create a new draft executable or copy a pre-configured one for the project.
5. You configure the workstation(s) for the project.

Please note: A TTY workstation (non-grafic) and a `WS_DEFAULT` workstation (grafic) is automatically set up in your project directory in the directory `<path to the customer project>\med\ws\`.

6. If you did not create a draft executable for the project earlier on in the configuration you can create one now.

Please note: Never copy a project to a new location, because the defined path inside the batch files and the binaries do not work.

Reconfigure a Project

You can repeat any stage of the project creation and configuration procedure by reconfiguring the project. You may want to do this if you need to add customizations or products to your project, or if you want to move products within your project. There are two ways of reconfiguring:

- You specify the `-reconfigure` option when you start up MEDCONFIG. This takes the existing file called `mxddef.bin` in the project and uses its values as the defaults for the new settings, but MEDCONFIG configures the project as if it were new.
- You give the `reconfigure` command at the interactive level of MEDCONFIG, as shown below:

```
Enter command> reconfigure
```

This restarts MEDCONFIG and continues as if you had started MEDCONFIG with the `-reconfigure` option, as above.

Default workstations `TTY` and `WS_DEFAULT` are not modified when you reconfigure a project.

Please note: If you reconfigure an existing project, you may need to relink the 2D draft executable.

Contents of the Sample Project

The examples in the following sections use a project called `proj` that contains every MEDUSA product directory. This project, therefore, shows all the MEDCONFIG commands available, but the command set you see at your site depends on which MEDUSA products you include in the project.

The order in which MEDCONFIG configures products and prompts you for information depends on the order of the products in the product list. The sample project shows products being configured in the order in the sample product list; your own project may be ordered differently, so the order of configuration actions may not be the same.

The order of some of the products is important. See [“The Product List” on page 25](#) for rules on the order of products in the product list.

Configuration Display

Most products display messages to the screen when MEDCONFIG configures them. Some products in the product list are configured as part of other products, so do not display any configuration actions on the screen. This is the case, for example, for the product `med3dprop`.

Before You Start Configuration

Before you start to configure a project, you need to set up certain directories, and find out the exact pathnames of the products you are going to use in the project. In this section, you learn how to create these directories, where the products are stored, and how the names of the product are.

Please note: In this section paths and commands are given for a Windows system.

Finding Out the Product Names

When you run MEDCONFIG to configure a project, you need to enter the exact pathnames of the products to configure. To find out these names and pathnames look at the file `product_list.dat` created during the MEDUSA Installation inside the sub-directory `MASTER_PROJECT`.

Alternatively, if the products have all been loaded into one MEDUSA product directory, called `medusa` throughout this manual, you can list the contents of the directory to find out product names and pathnames.

The example below assumes the products have all been installed in the `c:\medusa` directory. This list varies depending on the MEDUSA products you have bought (lines beginning with dashes (--) are comment lines indicating products which are not used in the project).

```
c:\medusa\medsys
c:\medusa\med2d
c:\medusa\med3d
c:\medusa\med3dprop
c:\medusa\med3dshrink
c:\medusa\med3dtran
c:\medusa\med3dui
c:\medusa\medavw
--c:\medusa\akcat
--c:\medusa\am
c:\medusa\cadconvert
--c:\medusa\cvcat_ansi
--c:\medusa\ducat
--c:\medusa\ducting
--c:\medusa\ham
--c:\medusa\isogen
c:\medusa\mdbaccess
c:\medusa\mdbcustool
c:\medusa\mdbfortran
```

```
c:\medusa\mdbhetnet

--c:\medusa\medchangeeger
--c:\medusa\medchgedmger
--c:\medusa\medchinfofer
--c:\medusa\medchinfo_demoger
--c:\medusa\medchmechger
--c:\medusa\medchmedproger
--c:\medusa\medchrasterger
--c:\medusa\medchrefger
c:\medusa\meddars
c:\medusa\meddoc
c:\medusa\meddtm
c:\medusa\mededa
--c:\medusa\medgedm
--c:\medusa\medgedm_medinfor
--
--c:\medusa\medilink
--
--c:\medusa\medinfo
--c:\medusa\medinfo_demo
--c:\medusa\medinfo_web
c:\medusa\medmech
c:\medusa\medmechgedm
c:\medusa\medpara
c:\medusa\medparts
c:\medusa\medplot
c:\medusa\medplot_canon
c:\medusa\medplot_hp
c:\medusa\medplot_raster
c:\medusa\medplot_winplot
c:\medusa\medplot_winpost
c:\medusa\medpro
c:\medusa\medraster
c:\medusa\medref
c:\medusa\medshade
c:\medusa\medstl
c:\medusa\medtt
c:\medusa\medvda
c:\medusa\medvrml
--c:\medusa\piping
--c:\medusa\spcat
--c:\medusa\sscat
--c:\medusa\steel
--c:\medusa\support
```

Create Directories

Before you configure a project, check the following details:

1. Check that the installation procedure is complete.
2. Create a new directory for the project giving the directory a suitable name; for example:

```
mkdir c:\proj
```

3. Create a new workspace directory for the project, giving the directory a suitable name:

```
mkdir c:\medtemp
```

The MEDUSA product directories are static, that is, after installation they are not subsequently changed. Therefore they can be located on a disk that is not regularly backed up. The project directory may have data written to the disk periodically and should be backed up. It is suggested that the logical name `PROJ` should be created to reference the disk where projects are created.

MEDUSA uses a scratch disk space. Use of this space will vary. It is suggested that `medtemp` is used to reference this space. The `medtemp` directory should not be backed up and it is worth clearing `medtemp` on machine startup.

You may now start to configure the project. The following sections give an overview of the configuration utility, MEDCONFIG.

Accessing MEDCONFIG

If this is the first time you have used the MEDCONFIG utility, you need to enter the following command to make MEDCONFIG available to you:

```
c:\medusa\medsys\login
```

On Unix systems you can add this command to your `.cshrc` or `.login` file, so that MEDCONFIG is always available to you.

The Product List

MEDCONFIG uses a file in the top level of the project directory to store the names of the products to include in that project. This file is called `product_list.dat`. There are some rules on the order in which you enter the products in the product list, so that the Code Table and Data Definition Language (DDL) files are then available for use by other products.

- `medsys` must be the first product in the product list.
- `med2d` should be the first application product after `medsys` or after the kanji font if one is required.
- `med3d` must be the first 3D product.
- `medplot` should be the first plotting product. Any further plotting products can be entered in any order relative to other plotting products. It is suggested that plotting products follow `med2d` products.
- `medraster_base` is needed for `medraster`, `medplot_raster` and `medplot_hpgl2rtl`.
- `product_list.dat` should only contain the items that are available on disk. Unused products can also be commented, the line then starts with two dashes (`--`).

Entering the Product List

There are two ways of entering the product list for MEDCONFIG:

- Using an editor before you start MEDCONFIG
- Inside MEDCONFIG, where you are prompted for the product list. The file `product_list.dat` is created for you in the project directory.

Both these options are described in [“Step 1 - Creating a Product List” on page 26](#).

MEDCONFIG and Text Editors

If you edit the product list while you are running MEDCONFIG, you will use the default text editor, such as `vi` (Unix).

If the default text editor is not suitable, you may customize your startup files to select another editor. Details of how to change your default text editor are given in the *MEDUSA Customization Guide*.

Step 1 - Creating a Product List

Please note: In this section paths and commands are given for a UNIX system.

The following example shows how you create a project called `/proj`. This project has only some products of the MEDUSA products palette. For your project you include only those products you need.

Change Product List Before Starting Configuration with MEDCONFIG

1. Attach to your project directory, for example:

```
os% cd /proj
```

2. Create a file called `product_list.dat` that follows this format:

```
/medusa/medsys  
/medusa/med2d  
/medusa/med3d  
/medusa/med3dprop  
/medusa/med3dshrink  
/medusa/med3dtran  
/medusa/med3dui  
/medusa/medshade  
/medusa/mdbaccess  
/medusa/mdbcustool  
/medusa/mdbfortran  
/medusa/meddoc
```

You may add comments to this file by starting a comment line with two dashes:

```
-- This is a comment.
```

MEDCONFIG ignores lines starting with two dashes.

3. Finish the last line using `RETURN`.
4. Save `product_list.dat` and quit from the editor.

You now start MEDCONFIG. You may also enter the product list after you have started up MEDCONFIG, as shown below.

Change Product List with MEDCONFIG

Starting up MEDCONFIG

1. Enter the following command line to add the MEDUSA directories and products to the search path that you are using, so that you can access MEDCONFIG:

```
os% /medusa/medsys/login
```

2. Start up MEDCONFIG in Administrator mode, and enter the project name when prompted:

```
os% medconfig -admin
      MEDUSA - Configuration Utility
      ~~~~~
Enter pathname of project <quit>: /proj
Project /proj is not configured
Do you want to configure it? <yes>:
```

Entering the Product List interactively

If you did not create a product list before starting MEDCONFIG, you can enter it now:

1. After you have started MEDCONFIG and having defined the project directory, you are prompted for project configuration:

```
Do you want to configure it? <yes>: Return
```

2. Press the Return key on your keyboard.

the following message comes up:

```
Input or Edit (using vi) the product list, or Abort ? (I/E/A)
```

This command line has three options:

- I
Allows you to input a product list by typing each product name at the prompt. For example:

```
pathname of product> /medusa/medsys
pathname of product> /medusa/med2d
```

You end the list by pressing the Return key on your keyboard:

```
pathname of product> Return
```

If there is a product list available already, you have the option to overwrite it and start again. If you answer `yes` an existing product list is overwritten immediately. If you answer `no`, then the question above appears again.
- E
This option starts up a text editor with the existing product list. If there is no product list, a new one is created and stored.
- A
stops configuration and quits MEDCONFIG. You are back at the operating system level and the following message is given:

```
Configuration aborted: Product list not confirmed
```

Errors in the product list

If there is a mistake in the product list, MEDCONFIG tells you where it is, and then prompts you to edit the product list. You must correct the error before MEDCONFIG can continue with the configuration:

```
Reading the product list definition file
Product /medusa/meds does not exist or is not accessible
Error(s) found in the product list definition file
Input or Edit the product list, or Abort ? (I/E/A)
```

Confirming the Product List

MEDCONFIG asks you to confirm the product list, and lists the products you entered:

The product list is defined as follows:

```
/medusa/medsys
/medusa/med2d
/medusa/med3d
/medusa/med3dprop
/medusa/med3dshrink
/medusa/med3dtran
/medusa/med3dui
/medusa/medshade
/medusa/mdbaccess
/medusa/mdbcustool
/medusa/mdbfortran
/medusa/meddoc
```

Is this product list satisfactory? <yes>: return

If you answer **no**, the question for entering the product list is displayed again:

Input or Edit (using vi) the product list, or Abort ? (I/E/A)

If you answer **yes** by pressing the **Return** key, MEDCONFIG starts with the configuration.

Step 2 - Configuring the MEDUSA Products

After defining the products for your project, MEDCONFIG configures the MEDUSA products in the order you entered them in the product list, and builds the necessary binary files. MEDCONFIG compiles binary files for the following data:

- Messages
- Code Table
- Data Definition Language (DDL) file
- Decortext and standard fonts
- Prim table
- Symbol table
- Boldness map

The configuration process runs automatically, unless you need to give additional information. The configuration process and additional information that you may need to enter is described in the following sections.

Configuring the System Product

When you have confirmed the product list, MEDCONFIG configures the MEDUSA system product, `medsys`:

```
Configuring product /medusa/medsys
```

Configuring the Workspace Directory

MEDCONFIG then prompts you to enter the temporary workspace directory for the project:

```
Enter name of temporary workspace directory </medtemp>:
```

You press `return` to accept the default name in angled brackets, or you type in the full path-name of the directory you want to use. If you reconfigure the project, MEDCONFIG uses the most recent entries for this directory as the default.

If the temporary workspace directory specified does not exist you will be prompted:

```
Directory does not exist or is not accessible - Continue? <yes>
```

If you press `return` to continue the configuration you must remember to create the temporary workspace directory before you try to use the project.

Compiling the Message Files

MEDCONFIG next checks the binary versions (.bmf) of all source message files (.nem) for the products in the product list (except for the user interface message files), and displays information for each message file. For example, a message such as the following is output:

```
Compiling MESSAGE 'raster_messages' ...
```

Configure Workstations

At this stage the default workstations TTY and WS_DEFAULT (graphics) are configured. You have no possibility to interfere for customizing workstations to your needs. You can do this later. For details see the *MEDUSA Customization Guide*.

Compiling the User Interface Message Files

There are two user interface message files. These are:

- **HELP_CONTEXT_MESSAGES**
This set of messages is used to tie the `help_context` attribute values (the message id) to the document locator strings (the message string) used by the online documentation system.
- **MED2D_MESSAGES**
This set of messages is used to define user interface menu strings (for example, pulldown menu titles) and browse text.

As already described above for the other message files, MEDCONFIG checks the binary versions of the user interface source files, and displays information for each message file. For example, if the binary files predate the source files, the following messages are output:

```
Compiling UISTRING 'MED2D/HELP_CONTEXT_MESSAGES' ...  
Compiling UISTRING 'MED2D/MED2D_MESSAGES' ...
```

When the binary files have later date stamps than the source files, MEDCONFIG displays these messages:

```
UISTRING binary 'MED2D/HELP_CONTEXT_MESSAGES' is up-to-date  
UISTRING binary 'MED2D/MED2D_MESSAGES' is up-to-date
```

Definition of the Data Exchange

You are asked whether you want to exchange sheets with STHENO or MEDUSA or not.

If you want to exchange STHENO sheets, answer `Yes` and specify the location of your STHENO installation, when the appropriate question appears.

If you want to exchange MEDUSA sheets, answer `Yes` and specify the location of your MEDUSA project, when the appropriate question appears.

Please note: For details see the chapter “Sheet Converter“ of the “Migration“ part.
“Migration“ is available as pdf file within the product
<medusa>\meddoc\doc\english\administration\migration.pdf.

Query for Changing Administration Password

A query for changing the Administration password appears. Pressing Return keeps the old one.

Compiling the 2D Binary Files

Most binary files are compiled as part of the 2D product, med2d. This is shown below:

```
Compiling DDL ...

Reading /medusa/med2d/med/src/ddl.dat ...
DDL compiled into /proj/med/bin/ddl.bin

Compiling CODE ...

Reading /medusa/med2d/m2d/src/code.dat ...
CODE compiled into /proj/m2d/bin/code.bin

Generating boldness map ...

Reading /medusa/med2d/m2d/src/boldness.map ...
Boldness map generated in /proj/m2d/bin/boldness.map

Compiling PRIMS ...

Reading /medusa/med2d/m2d/src/prims.she ...
PRIMS compiled into /proj/m2d/bin/prims.bin
Number of prims generated: 93
Storage space used for Table: 2769 words

Compiling PATTS ...

Reading /medusa/med2d/m2d/src/patts.she ...
PATTS compiled into /proj/m2d/bin/patts.bin
Number of patterns generated: 22
Storage space used for Table: 2391 words

Compiling PSYMS ...

Reading /medusa/med2d/m2d/src/psyms.she ...
PSYMS compiled into /proj/m2d/bin/psyms.bin
```

```
Number of point symbols generated: 10
Storage space used for Table: 968 words
```

Compiling the Fonts

MEDCONFIG then compiles the decortext and standard fonts by reading in the MEDUSA sheets where the fonts are defined:

```
Compiling decortext fonts ...

Loading /medusa/med2d/m2d/dfont/arial_default.bin ...
Loading /medusa/med2d/m2d/dfont/arial_fill_default.bin ...
Reading /medusa/med2d/m2d/dfont/express.she ...
Reading /medusa/med2d/m2d/dfont/gothic.she ...
Reading /medusa/med2d/m2d/dfont/hand.she ...
Reading /medusa/med2d/m2d/dfont/leroy.she ...
Reading /medusa/med2d/m2d/dfont/pica.she ...
Reading /medusa/med2d/m2d/dfont/pical0.she ...
Reading /medusa/med2d/m2d/dfont/sean.she ...
Decortext fonts compiled into /proj/m2d/bin/dfont.bin

Compiling font ...

Reading /medusa/medsys/med/src/font.dat ...
Font compiled into /proj/med/bin/font.bin
```

Further Compilations

MEDCONFIG runs further compilation, for example for defaults, fits and styles.

Linking the 2D Drafting Image

MEDCONFIG then asks you whether you want to link the 2D drafting image in order to create a draft executable for the project:

```
Link 2D drafting image? <yes>:
```

If you enter `n` or `no` at this prompt, you can create a draft executable for the project at a later stage by running the link command. This command can be found in the project directory `<project-dir>/m2d/build/draft/`. The executable is `draft_link.bat` (Windows) respectively `link` (UNIX).

Configuring the 3D Design Products

When MEDCONFIG configures the MEDUSA 3D related products it will report items such as:

```
Configuring product /medusa/med3d
Configuring product /medusa/medshade
```


Products like `med3dprop` and `med3dshrink` do not display any configuration actions. For this see also the section [“Configuration Display” on page 21](#).

Configuring the Superprint Driver

The last task of MEDCONFIG is the configuration of the superprint driver. If the configuration is successful, the following message is given:

```
Superprint driver for winplot creation succeeded
```

If superprint is already configured and you reconfigure your project for some reason, the following message is given:

```
Superprint driver for winplot exists
```

What you do next

As the case may be MEDCONFIG prompts you to enter further commands appropriate to your project configuration. The full list of commands is given in [Table 2, “MEDCONFIG Commands”, on page 17](#).

It is advisable to quit from MEDCONFIG and save your configuration environment at this point. You can then re-enter MEDCONFIG and continue the configuration process if required. If you do not do this, any hardware or operating system crash may cause you to lose all of the current configuration.

ADMINISTRATION

This chapter describes the administration of styles and standards:

- Administrator Mode 36
- Creating and Editing MEDUSA Styles 37
- Setting up Standard Colors 43
- Setting up Standard Boldness (Thickness) Values 46
- Setting up Boldness for Plotting Dimensions 47
- Setting up Standard Sheet Filename 48
- Setting up the Page Label..... 51
- Setting up the Sheet Header Editor 52
- Setting up Symbol Manager..... 62
- Layer Set Management..... 66
- Customize Layers 70
- Customize Layer Sets 71
- User-Defined Dashboards 74
- Dashboard Element Selection Limits 83
- Create Balloon Text Symbols..... 85
- Define Used Codepage..... 91

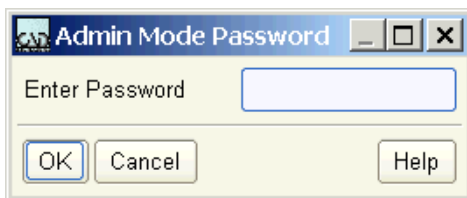
Administrator Mode

Only users with System Administrator privilege can create or modify styles, change standard settings for color, boldness and sheet filename, or adjust settings for the sheet header editor, the symbol manager and the layer management.

To enter Administrator mode:

1. Select Admin Mode -> Admin Mode from the Options menu.
If this option is already enabled, it shows a tick next to Admin Mode. In this case do not select it because this would switch off the Administrator modus.
If there is no tick, select Admin Mode to open the following dialog.

Figure 2 Admin Mode Password Dialog



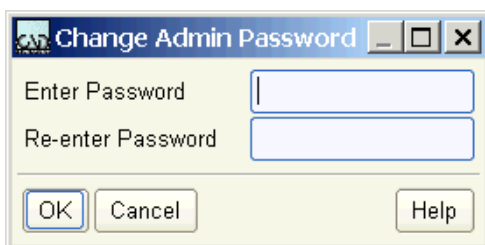
2. Type the correct password and click OK.

Please note: MEDUSA is provided with the default password Admin. If the password is requested, enter Admin.
Once you changed the password, only this string can be used for the password from then on.
It is recommended that your System Administrator changes the password after program installation.

To change the Administration password:

1. Select Admin Mode -> Change Admin Password from the Options menu.
This entry is only accessible if you are in administrator mode. After clicking this item the following dialog comes up:

Figure 3 Admin Password Change



2. Type the new password two times and click OK.
If the entered and re-entered passwords do not match, an error message comes up and the password is not changed.

Creating and Editing MEDUSA Styles

What are MEDUSA Styles?

Each set of creation tools (for example, Line, Text, Prim, and Dimension) has a list of properties that define how the geometry will be presented on the sheet. **Styles** are the mechanism by which MEDUSA describes geometry. For example, there are many line creation tools available, but it is the style that fully describes how a line will be presented on the drawing, for example, solid thin, dashed thin and so on.

Each style has a unique name and a particular grouping of properties, which reflect how the geometry will be created.

Creating a Style

Please note: Administrator mode must be enabled before you can create or modify styles within MEDUSA since style changes affect all MEDUSA users.

You access style creation via the `Properties` dialogs associated with each type or class of geometry, all of which are available from the MEDUSA Dashboard.

The following example shows you how to create a new line style called `My Style`, which will have the following attributes:

Line Type:	Dotted Line
Line Thickness:	1.0
Color:	Blue


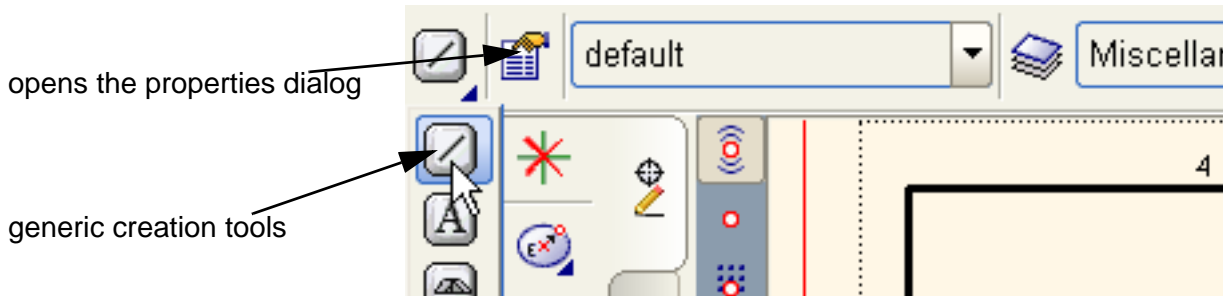
1. First set the geometry type to line in order to create the new style from the displayed one. You can do this by either
 - selecting the generic creation tool `Creates lines of specified type and properties` , which is located at the far left of the Dashboard, as shown below, or
 - selecting a line inside the drawing area.

Figure 4 Selecting a Line Creation Tool




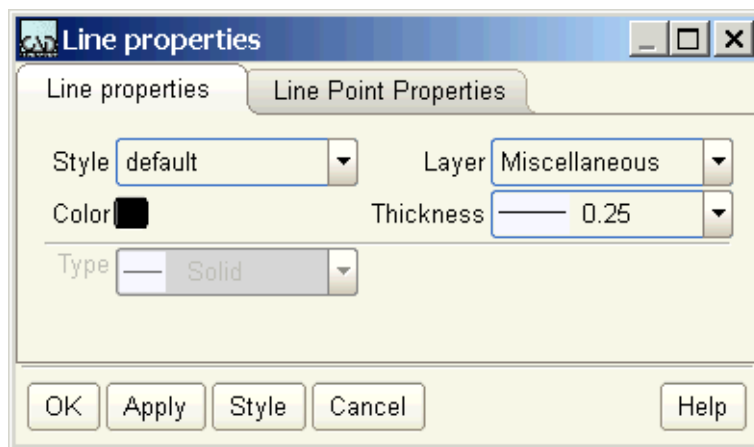
2. Now select the Properties tool  from the dashboard. The Line Properties dialog appears.

Figure 5 Line Properties Dialog

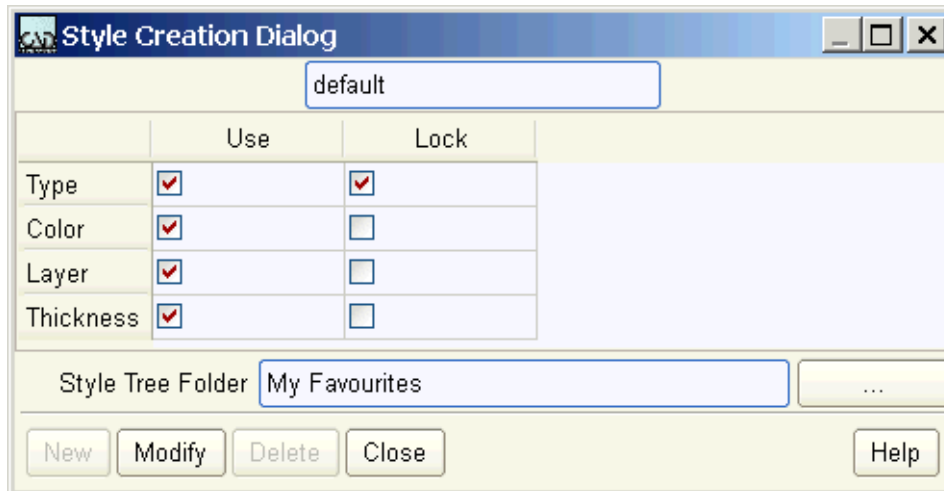


The Properties dialog is context sensitive. If text, dimensions or one of the other types was selected when you selected the Properties tool then the relevant Properties dialog is presented.

The properties described in Figure 5 show the current style settings for the default style. Note that the Type setting is disabled. This is because the default style has defined the Type to be locked, which means that no user, other than the System Administrator, can change the setting. This allows the enforcement of company-wide drawing style standards.

3. To create a new style, select the Style button from the Line Properties dialog. The Style Creation dialog appears.

Figure 6 Style Creation Dialog



The Style Creation dialog lists all the attributes associated with a particular geometry type. Each attribute displays the options Use and Lock, which can be set accordingly.

The Use field

The Use field indicates what attributes will be stored in the style.

4. You need to define Type, Thickness and Color attributes, so tick those choice boxes.

The Lock field

The Lock field is only enabled if you have set the corresponding Use choice box, and instructs the system not to allow any user to change the setting of that attribute.

5. For this example, lock the Color attribute.

Changing the Attribute Values

Now that both the Properties and Style Creation dialogs are visible, you will see that before you can modify the Type attribute on the Properties dialog, it is necessary to unlock it within the Style Creation dialog. By toggling the Lock field you will see the properties type switches from disabled to enabled, allowing you to change the attribute.

6. Now change the line attributes. Change:
 - Color to blue,
 - Thickness to one of the thicknesses of the pulldown list,
 - Type to Dotted Line.

Please note: You will need to unlock the color attribute before you can change the color.

7. Before you save the style, reset the **Lock** choice box for the **Color** attribute.

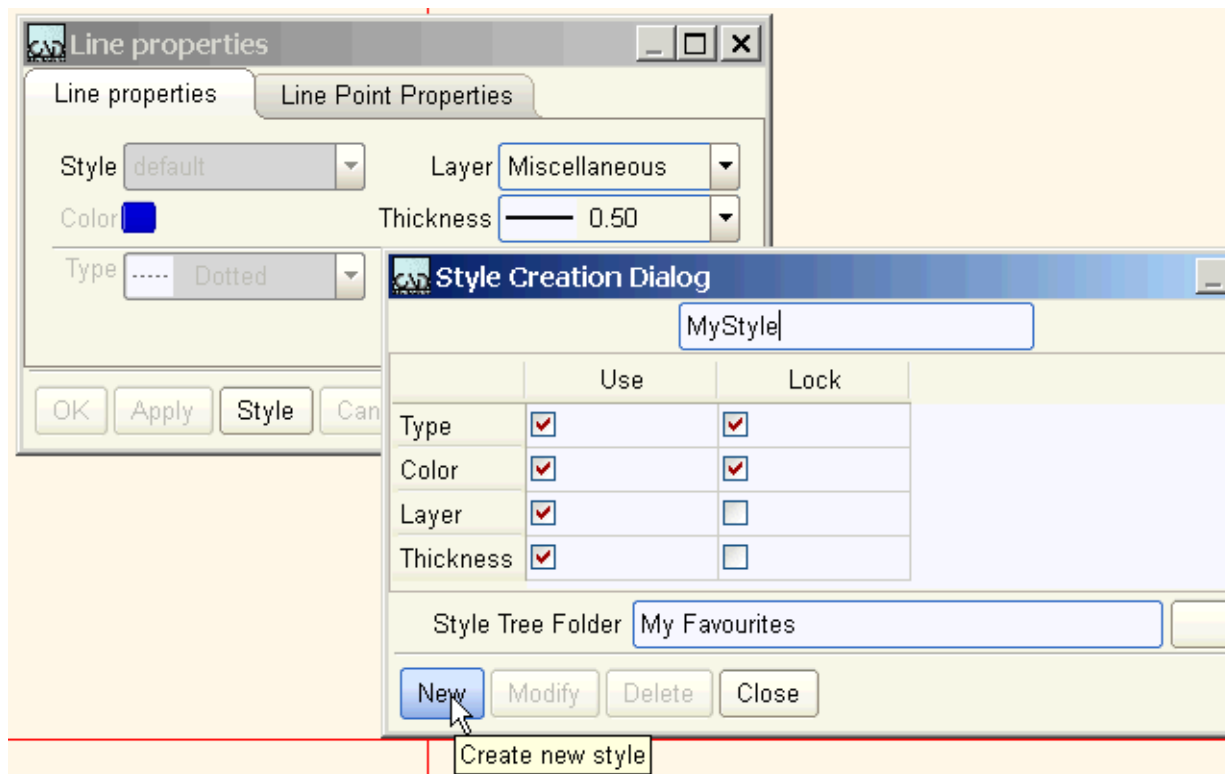
Saving Your Style

8. Type the style name **MyStyle** into the text entry area at the top of the **Style Creation** dialog. The context sensitive buttons at the bottom of the dialog change as you type the new name. The **New** button becomes enabled and the **Modify** and **Delete** buttons are disabled.

Please note: As you type the new style name into the field, the system tries to match the name with an existing style. If the **New** button remains disabled after you have typed a new style name, then a style with that name already exists in the system.

The two dialogs look like this:

Figure 7 Example: Create Line Style



9. Choose the **Style Tree Folder** to define the place of your style inside the style tree.

10. Click **New** on the **Style Creation** Dialog.

MyStyle is saved. The new style name appears in the style field of the **Properties** dialog. If you decide not to save the style, select **Close** to cancel the operation and reset the **Properties** dialog to its original settings.

Please note: At this stage the new style is only saved temporary for the current session. For saving permanently go on with the steps below.

Saving Your Style Permanently

11. Open the style tree by clicking the button *Sheet structure tree* inside the status area, and then choose the tab *Style Tree* inside the tree viewer area.
12. Move the cursor upon the displayed style tree and *click right*.
The popup menu for the style tree opens.
13. Choose the entry *Save Style Tree*.
The whole style tree is stored (including your new style *MyStyle*) and when you start MEDUSA the next time, all the current styles are loaded and can be used for drawing.

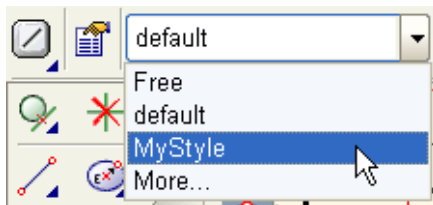
Please note: If you are the System Administrator and if you want all users to have the styles you created, then you have to edit the file *styles.xml*.


Using Your Style

To select the style for use, either:

- Select *Apply* from the *Properties* dialog to switch the *Dashboard* style to *MyStyle*, or
- Select *MyStyle* from the *Dashboard* style drop down list as shown below:

Figure 8 Selecting a Style from the Dashboard




Use the generic line creation tool  from the *Dashboard* to test out your new style. Alternatively, you can associate the *MyStyle* style directly with a custom tool. (See the *Drafting User Guide* for details).

Modifying Styles

You can also modify existing styles. The next example shows how to modify *My Style* so that *My Style* lines will be red, and users cannot change either the thickness or the line type attributes.


To modify the style:

1. Select *My Style* from the *Style* pulldown menu located on the *Dashboard*.
2. Select the *Properties* tool .
3. Select the *Style* button from the *Line properties* dialog.

The Style Creation dialog appears.

4. Unlock the Color attribute.
 5. Set the color to red in the Properties dialog.
 6. Lock Color, Thickness and Type.
 7. Select the Modify button, which will update the style.
 8. Close the Style Creation dialog and Apply as before.
- The Dashboard attributes for Color, Thickness and Type are now locked.

Deleting Styles

1. Select My style either
 - a. from the style pull down menu, located on the Dashboard, or
 - b. from the style tree popup menu, entry Use Style.
2. Select the Properties tool .
3. Select the Style button from the Line properties dialog.
The Style Creation dialog appears.
4. Select Delete.
My style is deleted from the system and a backup style called my_style is created in order to keep the settings in case that you wish to recreate the deleted style. The backup style is only available inside the dashboard and if you close MEDUSA it is lost.

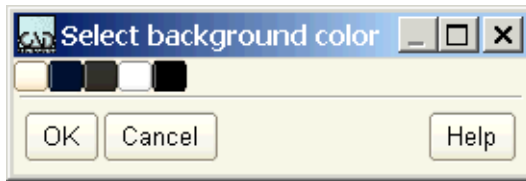
Please note: Inside the style tree you can use Remove Style from Favourites from the popup menu, which removes the selected style from the favorites folder of the tree only. Inside the dashboard it is still available.

Setting up Standard Colors

Color Display on User settable Background Color

The Select background color dialog which you can call up by Options > Configuration > Select Background Colors allows you to select a background color other than the default ivory background.

Figure 9 Select background color Dialog



Changing the background color requires the alteration of some displayed colors to be visible on the new background. For example a black line is invisible on a black background, but is still required to be black when printed on paper. To achieve this effect MEDUSA has an algorithm built in to ensure all colors are visible on the screen. This effect can be seen in Figure 10, “The Select Color Dialog”. Compare both color scales.

Figure 10 The Select Color Dialog



The upper color scale (Current Colors) shows the colors displayed on the screen. It changes when changing the background color from ivory to dark blue. Only colors that would not be visible are altered.

The lower palette displays the plot colors. They are still unchanged.

Any new or imported colors are automatically handled by the MEDUSA algorithm, but if you do not like the color matches created by MEDUSA you can create your own color mappings. Any color is exactly defined by a triplet of numbers, the RGB value. Define exact RGB values for each color for every background.

The default set of colors is defined in the file `<medusa>\med2d\m2d\src\colours.map` as shown in the following table.

Figure 11 The Color Map Editor

RGB values define dark blue background color

```

-- default
--
BG      background / 255 247 230 / 0 18 50 / 51 50 46 / 255 255
HL      red       / 255  0  0 / 255  0  0 / 255  0  0 / 255  0
COLOUR black     /  0  0  0 / 255 255 255 / 255 255 255 /  0  0
COLOUR green    /  0 204  0 /  0 204  0 /  0 204  0 /  0 204
COLOUR blue     /  0  0 204 / 133 133 204 /  0  0 204 /  0  0
COLOUR skyblue  /  0 179 179 /  43 179 176 /  0 187 183 /  0 179
COLOUR brown   / 143 125 84 / 143 124 84 / 255 221 150 / 143 125
COLOUR ochre   / 209 117 94 / 209 115 94 / 209 63 30 / 209 117
COLOUR yellow  / 209 204 54 / 209 203 54 / 209 203 42 / 209 204
--
-- additional colours|
COLOUR royal blue /  0  0 255 / 129 129 255 /  0  0 255 /  0  0
COLOUR grey      / 127 127 127 / 128 128 128 /  96  96  96 / 127 127
COLOUR magenta   / 255  0 255 / 255  2 255 / 255  0 255 / 255  0
COLOUR bluegreen /  32 178 170 /  39 178 168 /  32 210 198 /  32 178
COLOUR olive     /  0 128  0 /  0 128  0 /  0 128  0 /  0 128
COLOUR red       / 255  0  0 / 255  0  0 / 255  0  0 / 255  0
COLOUR purple    / 128  0 128 / 128  35 128 / 185  0 185 / 128  0

```

RGB values define displayed white color

- The first line defines the background colors (BG)
- The second line defines the highlight colors (HL)
- The following lines define the colors visible on the screen. (COLOUR)

You find a name field in the second column and 5 alternative RGB triplets for each color in the following columns. These correspond to the color values to be used for alternative for non-ivory background. You can define up to 8 alternative RGB triplets for each color. Additional background colors will be displayed in the `Select background color` dialog (Figure 9, “`Select background color Dialog`”)

Sample:

You might select the dark blue background, which would be the second button on the `Select background` dialog (see Figure 9, “`Select background color Dialog`”). This background is defined by RGB values `<0 18 50>` which you can find in the first line and second column of RGB values. If you now display the `Select Color` dialog again (see Figure 10, “`The Select Color Dialog`”, which can be done by selecting any piece of geometry and clicking on the color button on the dash-

board) the colors in the lower area will reflect the colors below the RGB values <0 18 50> and with COLOUR at the beginning of the line.

The highlight color is still red, defined by RGB values <255 0 0>.

A black line is now displayed as white, defined by RGB values <255 255 255>.

A green line is still displayed green defined by RGB value <0 204 0>.

a default blue line is displayed as a light blue, RGB <133 133 204>, to be visible on the dark blue background and so on. (Figure 10, "The Select Color Dialog")

Please note: Plotting is not affected, as any changes on the screen are graphical only and not written to the sheet. At any point you may switch the background color to the default sheet you see the sheet exactly as plotable.

Setting up Standard Boldness (Thickness) Values

The boldness (thickness) used for drawing lines and text are defined in the following file in your MEDUSA product: `<medusa>\med2d\m2d\src\boldness.map`.

Please note: Any adjustments which you make should not be made to the original *boldness.map* but in the *boldness.map* file of a product area (`<path to customer product>\m2d\src\boldness.map`). It will be merged with the contents of any *boldness.map* files in any of your listed products to generate the `<path to project>\m2d\bin\boldness.map` file. This task is performed by the BOLD option within MEDCONFIG, and is done automatically when a project is configured, or reconfigured.

Default thicknesses in the file *boldness.map*:

BOLDNESS	1	0.18
BOLDNESS	2	0.25
BOLDNESS	3	0.35
BOLDNESS	4	0.5
BOLDNESS	5	0.7
BOLDNESS	6	1.0
BOLDNESS	7	1.4
BOLDNESS	8	2.0

You can have a maximum of 16 boldnesses. You can modify this file to setup your system boldness values as required. Each BOLDNESS definition starts with the keyword BOLDNESS, followed by the boldness index (1 through 16) and the boldness value. The values do not have to be unique so you could setup all your boldness values to be the same. Any text including and following `--` is treated as a comment and is ignored.

Setting up Boldness for Plotting Dimensions

You can set up the boldness values for plotting dimension lines and texts for your project.

Use the following environment variables for this:

Windows:

```
set BOLD_DIM_TXT=<value>
set BOLD_DIM_LINE=<value>
```

Unix:

```
setenv BOLD_DIM_TXT <value>
setenv BOLD_DIM_LINE <value>
```

The environment variables can be set e.g. in the *login.bat* of your MEDUSA user project.

For the default values for thickness (<value>) see the the following table:.

<value>	Boldness
1	0.18
2	0.25
3	0.35
4	0.5
5	0.7
6	1.0
7	1.4
8	2.0

If the variables are not set, the default value for boldness is set to 2 (0.25).

Please note: Consider that these settings are used only, if the option for Dimension Text inside the Dimension dialog (available with Options > Defaults > Dimension) is set to Bold.

Setting up Standard Sheet Filename

When saving a new sheet the default name is automatically generated from the sheet header. This is normally defined by the style `drawing_number`, and therefore the file name would be, for example, *DrgNo.she*.

If you wish to use more, or other items, from the sheet header you can modify the default.

1. Open the file `<medusa>\med2d\m2d\src\defaults.dat` to display the text entries for the sheet saving.

The relevant section looks like following, for example:

```
-- SHEET_SAVE "<drawing_number>_<sheet_issue_number>_<sheet_number>"  
-- SHEET_SAVE "<drawing_number>_<sheet_number>"  
SHEET_SAVE @tTSH
```

The two lines which begin with `--` are comments only, and will be ignored when the *defaults.dat* file is compiled during the project configuration process.

The `SHEET_SAVE` expression tokens can be recorded in one of two formats.

- a. The active entry shown here has the form (`@tTSH`) which uses the tradition MEDUSA Text Type method to identifying a text item on the sheet. The three characters following the `@t` are a type code (or CAN-code).
- b. Alternatively the style form (`<drawing_number>`) can be used, as shown in the two alternative examples. Each item inside the brackets `<>` is taken to be an existing style name. Characters outside the brackets `<>` are fixed.

Please note: The style names used here are the internal ones, e.g. `drawing_number`, and not those ones used in the graphical user interface, e.g. Drawing Number.

As the standard `drawing_number` style is defined as having to be of text type TSH, the two forms `@tTSH` and `<drawing_number>` are identical in effect.

Similarly the alternate form

```
"<drawing_number>_<sheet_issue_number>_<sheet_number>"
```

is synonymous with

```
"@tTSH_@tTIS_@tTSN".
```

Please note: Each token should identify a unique text item on the sheet. In practice the first text element found that matches the token is used to provide the relevant string.

2. Create your own *defaults.dat* file.

Use the med2d *defaults.dat* file as a template to create your own *defaults.dat* that will be merged with the med2d one when your project is configured. However all that you will require is the line containing the `defaults` keyword that separates the alias definitions at the beginning of the file from the actual default definitions in the remainder of the file and a line defining the `SHEET_SAVE` expression.

Delete the characters `--` in one of the two lines described above to set up the data as the new standard sheet filename, and comment out the previous active entry by putting `--` at the start of that line.

You can also create your own standard by entering your own styles into the definition for the standard sheet name.

Example:

```
defaults
sheet_save string "<drawing_number>_<sheet_issue_number>_<sheet_number>"
```

alternatively, with the same effect:

```
defaults
sheet_save string "@tTSH_@tTIS_@tTSN"
```

3. Save the *defaults.dat* file in the project area or merge your additions into the one that is already there.

The process that compiles the *defaults.dat* files searches every product in your project's product list and loads all of the *defaults.dat* that it finds. The contents of them all are merged into the final *defaults.bin* file that is saved within the project.

The standard settings are stored in:

```
<MEDUSA installation path>\med2d\m2d\src\defaults.dat
```

but these can merge with, for example:

```
<MEDUSA installation path>\medmech\m2d\src\defaults.dat
```

A customized version would normally be inserted into:

```
<path to customer project>\m2d\src\defaults.dat
```

Only after recompiling of the project the entries are available. A binary file *defaults.bin* is generated in:

```
<path to the project>\m2d\bin
```

Please note: Using some characters has different effects on different operating systems. Care should be given to special characters, for example: `.` `/` `\` and so on. Avoid these characters.

For more details about *defaults.dat* see the *Customization Guide*.

Please note: When using the **Design Objects** product (`medmech`) within your project, and while working within a working set, the defined sheet saving expression is ignored. It is essential to the Design Objects processes that each sheet involved within the design of a component is unique, and so it always uses the form "`@tTSH_@tTIS_@tTSN`". The effect of this can be seen when using the sheet Save as... tool, if you have a different `SHEET_SAVE` expression.

If the sheet is a member of your working set the suggested sheet file will conform with the `<drawing_number>_<sheet_issue_number>_<sheet_number>` form.

However when the sheet is not a member of the working set your normal sheet saving expression will be used to generate the suggested file identifier.

Setting up the Page Label

When you have opened a MEDUSA sheet, the tab below the drawing area usually displays the filename. This tab label can be customized, for example to give a hint on the content, especially when a drawing consists of several sheets.

You can configure the displayed label in the file *defaults.dat*.

```

-----
-- Page Label Setting --
-- true / false
set_page_label boolean false

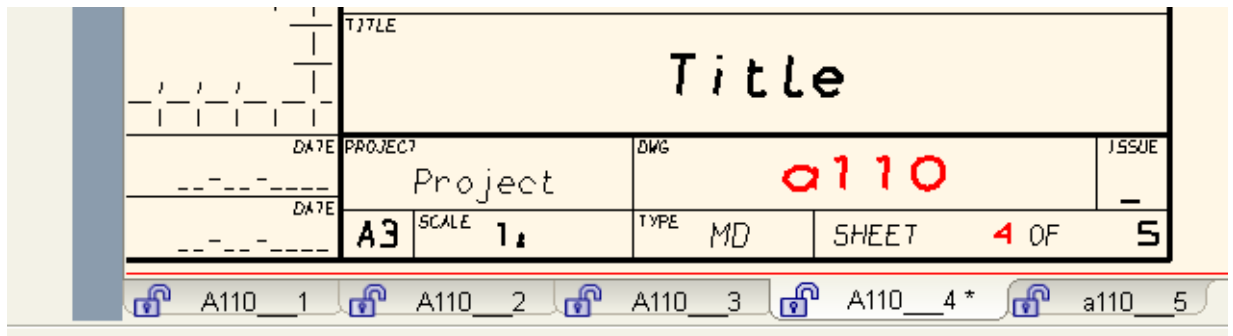
-- inside '$$....$$!' can be either:
--             style name           ($$!drawing_number$$)
--             localized style label ($$drawing number$$)
--             text type             ($$tsh$$)
page_label_style string "$$!drawing_number$$_$$!sheet_number$$"
-----

```

Please note: Only if `set_page_label` has the value `true` the configured label `page_label_style` will be used.

Following figure shows an example. The appropriate settings in the file *defaults.dat* are:
`set_page_label boolean true`
`page_label_style string "$$!drawing_number$$_$$!sheet_number$$"`

Figure 12 Example Sheet Header and Tab Label



Setting up the Sheet Header Editor

The operation and appearance of the **Sheet Header Editor** can be configured according to the user's needs, via a configuration file.

Name and Location of the Configuration file

It is possible to use several configuration files for the sheet header editor, either the *sfeld.cfg* file or a *sfeld_XXX.cfg* file.

sfeld.cfg

The configuration file *sfeld.cfg*, is located in the directory `<LW>:\<medusa>\med2d\m2d\src\`. `<LW>` is the letter of the hard disk and `<medusa>` is the installation directory of MEDUSA.

All **adjustments**, which you make, **should not be made in the original file** *sfeld.cfg* in `<medusa>\med2d\m2d\src\`, since these adjustments will be lost with a version change. Specify any adjustments in `<path to customer product>\m2d\src\sfeld.cfg`.

It is searched in all `...\m2d\src` directories over all MEDUSA products for a *sfeld.cfg* file. The last found file is used.

sfeld_XXX.cfg

In the *defaults.dat* file is defined, if the sheet should be scanned for a text type or attribute (on sheet level).

Example of the accordant section in the *defaults.dat*:

```
-----  
-  
-- Sheet header configuration  
-- sfeld_conf_typ: 0 -> disabled  
--                 1 -> text type  
--                 2 -> attribute on sheetlevel  
-- sfeld_conf: can code      (type -> 1)  
--                 attributename (type -> 2)  
-----  
--sfeld_conf_typ      integer      1  
--sfeld_conf          string       "TB1"  
  
sfeld_conf_typ      integer      2  
sfeld_conf          string       "sfconf"
```

In the example shown above the two last lines specify, to scan the sheet for the attribute. `sfconf` is the name of the attribute (6 characters are maximum).

In our example, the two lines which cause the search for the texttype are uncommented. `TB1` is the texttype.

The used name convention for the configuration file is: `sfeld_xxx.cfg`.

The `xxx` characters are the value of the attribute and thus they define the name convention of the configuration file.

It is searched in all `...m2d\src` directories over all MEDUSA products for a `sfeld_xxx.cfg` file. The last found file is used.

The product may contain several configuration files. A change of the attributes value defines which configuration file is used.

If more than one attribute/texttype is found on the sheet a warning is displayed.

Structure of the Configuration File

The structure of the configuration file orients itself on the INI-files of the operating system Windows and supports the following elements:

- **Chapter heading**

All configuration parameters that belong to a certain task area are combined to logical units (chapters). A chapter is initiated with the chapter heading and terminated with the next chapter or the file end. The chapter header is followed by data lines, comment lines or blank lines in optional sequence.

Syntax: [`<Chapter name>`]

Only alpha-numeric characters as well as underscores “`_`” are allowed in the chapter header. It won't be differentiated between upper- and lower case.

Sample: [`sfeld_cfg`]

- **Data lines**

The individual configuration parameter is placed in a data file, following the chapter header. The line will be ignored, if the data line does not have a chapter header defined at the time of read in.

Syntax: `<Keyword> = <value>`

Sample: `Coordinate_1 = 1.23`

- **Comment lines**

All lines starting with the character string: `--`, are recognized as comment lines. The content of the line is ignored.

Sample: `-- this is a comment`

- **Blank line**
The configuration file may contain as many blank lines as desired. They are ignored.

Chapters of the Configuration File

In the following sections you find information on the chapters of a configuration file. Finally, an example file is given.

Basic Settings: `sfield_cfg`

The chapter `sfield_cfg` describes the basic settings. The keywords of this chapter are optional, this means that they overwrite constant program default settings but they need not to be set explicitly. You find these program default settings inside the tables below. In detail the following keywords are defined:

Name : `zoom_hor`

Meaning : Defines the width of the zoom-area that the text field is zoomed to.

Type of Data : Real number > 0.0

Default value : 190.0

Name : `zoom_ver`

Meaning : Defines the height of the zoom-area that the text field is zoomed to.

Type of Data : Real number > 0.0

Default value : 85.0

Name : `dlg_maxrow`

Meaning : Defines the maximum amount of the lines of the dialog box.

Type of Data : Integer > 0

Default value : 15

Name : **dlg_maxchar**
Meaning : Defines globally the maximum amount of characters for an input field in the dialog box. Can possibly be overwritten with specifications from the definition of a dialog box. The entry will be shortened if it exceeds the length.
Type of Data : Integer > 0
Default value : 128

Name : **emptytextchar**
Meaning : A fill character can be entered since MEDUSA removes text with empty content ("") when saving a sheet.
Type of Data : Character
Default value : "."

Name : **layer_protect**
Meaning : Defines the layers that are to be protected from alterations during the termtime of MEDUSA.
Type of Data : Character string
Default value : "", i.e. no protected layers

Name : **consider_ng**
Meaning : If a style on a sheet cannot be found in MEDUSA, the CAN code of the missing style is extracted and this is used to search for the sheet header text. If this switch is set to `off`, the sheet is not searched for the CAN code and the message "...style not found" is written into the sheet header dialog.
Type of Data : Boolean
Default value : on

A final sample for the chapter `sfeld_cfg`:

```
[sfeld_cfg]
zoom_hor      = 250.0
zoom_ver      = 100.0
dlg_maxrow    = 20
dlg_maxchar   = 3
emptytextchar = #
layer_protect = 1020-1022
```

Dialog Areas: `sfeld_dlgarea_`

All chapters starting with `sfeld_dlgarea_`, contain the definition for each individual input line in a dialog box and also summarize those input lines into a group making up a dialog area. Possible groups could be, for example, the standard input, or the area of the update index, or simply all data in all.

Please note: The configuration file must contain at least one dialog area chapter.

Syntax:

```
[sfeld_dlgarea_<number>]
```

`<number>` represents a whole number and therefore defines the chapter name explicitly.

Sample:

```
[sfeld_dlgarea_1]  
[sfeld_dlgarea_12]  
[sfeld_dlgarea_4]
```

The sequence of the chapter names in the configuration file is optional. Each chapter name can only be given once.

Keywords are available for each chapter. To define a line in a dialog box, the keywords `label` and `texttyp` must be stated.

You have the possibility to split an existing chapter into several chapters at any time. To do so, simply enter a new chapter header at the desired spot and the keywords later mentioned under general keywords (see below).

If you would like to combine several chapters into one chapter, you may do so by deleting the chapter name as well as the belonging keywords.

Keywords inside Dialog Area Chapters

The following keywords are defined in detail:

General Keywords

Name	: <code>area_dlglabel</code>
Meaning	: Defines the heading, which is displayed for the respective group on the dialog box pages. If the name should contain dollar characters, it has to be written in single quotation marks (e.g. <code>a\$b</code>).
Type of Data	: character string

Keywords for the definition of a **dialog box input line**

In the following tables you find the entries `<xxx>`, which have to be replaced by a number,

which makes the keyword in the whole configuration file unique. The same number is to be used for all fields, which describe a dialog box input line, for example:

label1, texttyp1, maxchar1, ...

Name : **label<xxx>**

Meaning : Determines the text before the input field in a dialog box line. If the name should contain dollar characters, it has to be written in single quotation marks (e.g. a\$b).

Type of Data : character string

Comment : Needed for the definition of an input line!

Name : **texttyp<xxx>**

Meaning : Determines the style of text in the MEDUSA sheet, which is to be connected with the dialog box input line. If several texts of the same style are available in the sheet, the one found first is used.
Sample: Sheet title or sheet_title (internal style name)

Type of Data : character string

Comment : Is necessary for the definition of an input line!

Name : **maxchar<xxx>**

Meaning : Defines the max. amount of characters in the input field. Overwrites the [sfeld_cfg]dlg_maxchar defined value, if existing.

Type of Data : whole number > 0

Comment : optional Keyword

Name : **mandatory<xxx>**

Meaning : Defines, whether or not an input field must contain data or if 'empty' data is accepted as well.

Type of Data : character string on or off

Comment : optional keyword:
Active only if the keyword enabled<xxx> has the value on or if the keyword enabled<xxx> is missing and the program defaults (on) come to the course.
If the value of the keyword is unequal to on, the setting off applies.
If the keyword is not used, likewise the setting off applies.

Name : **enabled<xxx>**
Meaning : Determines whether inputs are possible in an input field or not.
Type of Data : character string `on` or `off`
Comment : optional keyword
`enabled<xxx> = off` has the consequence that the setting for `mandatory<xxx>` is not considered.
If the value of the keyword is unequal to `on`, the setting `off` applies.
If the keyword is not used, the `on` setting applies.

Name : **multitext<xxx>**
Meaning : Describes whether all texts of given type are synchronized.
Type of Data : character string `on` or `off`
Comment : optional keyword
`multitext<xxx> = off` has the consequence, that only the first found `texttyp` is modified, `on` modifies all texts of the found type.
If the value of the keyword is unequal to `on`, the setting `off` applies.
If the keyword is not used, the `off` setting applies.

Name : **suppress<xxx>**
Meaning : If this attribute is set to `on`, the attribute `mandatory` is ignored and the check for the `texttyp` is suppressed.
Due to it no error message is displayed, if a text of this type/style set for this input field does not exist on the sheet. In this case the Sheet Header Tool dialog opens and the input field to which the attribute `suppress =ON` has been assigned is disabled.
Type of Data : character string `on` or `off`
Comment : optional keyword
If the value of the keyword is unequal to `on`, the setting `off` applies.
If the keyword is not used, the `off` setting applies.

Please note: In addition to the keywords listed above the configuration file contains the two other keywords, `ilink` and `itype`. Both are irrelevant for MEDUSA.

A Sample Configuration File

```
-----  
--  
-- MEDUSA Sheet Header Tool --  
--  
--   - Configuration -  
--  
-----  
  
-- Common Description  
[sfeld_cfg]  
dlg_maxrow      = 7  
dlg_maxchar     = 20  
emptytextchar  = " "  
date_format     = %d-%m-%Y  
  
[sfeld_dlgarea_1]  
area_dlglabel  = sub_sheethead$label_compulsory_input  
  
label1         = sub_sheethead$label_sheet_title  
texttyp1       = Sheet title  
maxchar1       = 25  
mandatory1     = On  
enabled1       = On  
multitext1     = On  
ilink1         = Sheet_Title  
itype1         = string  
  
label2         = sub_sheethead$label_drawing_label  
texttyp2       = Drawing number  
maxchar2       = 15  
mandatory2     = On  
enabled2       = On  
multitext2     = Off  
ilink2         = Drawing_Number  
itype2         = string  
  
label3         = sub_sheethead$label_drawn_by  
texttyp3       = Drawn by  
maxchar3       = 10  
mandatory3     = On  
enabled3       = On  
multitext3     = Off  
ilink3         = Drawn_By  
itype3         = string
```

label4 = sub_sheethead\$label_date
texttyp4 = Date
maxchar4 = 20
mandatory4 = On
enabled4 = On
multitext4 = Off
ilink4 = Creation_Date
itype4 = date

label5 = sub_sheethead\$label_checked_by
texttyp5 = Checked by
maxchar5 = 10
mandatory5 = On
enabled5 = On
multitext5 = Off

label6 = sub_sheethead\$label_checked_date
texttyp6 = Checked date
maxchar6 = 10
mandatory6 = On
enabled6 = On
multitext6 = Off

[sfeld_dlgarea_2]

area_dlglabel = sub_sheethead\$label_additional_input

label7 = sub_sheethead\$label_sheet_number
texttyp7 = Sheet number
maxchar7 = 3
mandatory7 = Off
enabled7 = On
multitext7 = Off

label8 = sub_sheethead\$label_number_of_sheets
texttyp8 = Number of sheets
maxchar8 = 3
mandatory8 = Off
enabled8 = On
multitext8 = Off

label9 = sub_sheethead\$label_sheet_issue_number
texttyp9 = Sheet issue number
maxchar9 = 3
mandatory9 = On
enabled9 = On
multitext9 = Off

```
ilink9      = Sys_Revision

label10     = sub_sheethead$label_format
texttyp10   = Format
maxchar10   = 3
mandatory10 = ON
enabled10   = Off
multitext10 = Off

label11     = sub_sheethead$label_principle_scale
texttyp11   = Scale
mandatory11 = Off
enabled11   = On
multitext11 = Off

[sfeld_dlgarea_3]
area_dlglabel = sub_sheethead$label_additional_input

label12     = sub_sheethead$label_document_type
texttyp12   = Document type
maxchar12   = 5
mandatory12 = On
enabled12   = On
multitext12 = Off
ilink12     = Cad_System
itype12     = string

label13     = sub_sheethead$label_design_project
texttyp13   = Design Project
maxchar13   = 10
mandatory13 = Off
enabled13   = On
multitext13 = Off
```

Setting up Symbol Manager

The environment variable `MEDUSA_SYMBOL_PATH` in the file
`<medusa>\MASTER_PROJECT\login.bat`

contains paths to XML-files (*.xml) separated by semicolons. These XML-files are used for configuring the symbol libraries inside the symbol manager.

Structure of the XML-Files

```
<symbol_manager>
  <symbol_library>
    <symbols name=str path=str mask=str icon=str>
    </symbols>
    . . . .
  </symbol_library>
  . . . .
</symbol_manager>
```

An XML file consists of the following elements:

`<symbol_manager>`

tag which defines the start of the definitions for the symbol manager.

`</symbol_manager>`

tag which defines the end of the definitions for the symbol manager.

`<symbol_library>`

tag marking the start of defining a symbol library.

`</symbol_library>`

tag marking the end of defining a symbol library.

`<symbols name=str path=str mask=str icon=str>`

tag marking the start of defining a symbol, which can define a single symbol or a whole symbol family. The symbol tag can have up to four attributes:

`name` - describing name which is displayed in the appropriate tree node of the symbol manager.

`path` - is the directory to the symbols. A dot defines the directory with the XML file.

`mask` - filter for the recognized symbol file names which shall appear (optional).

`icon` - is displayed in front of the tree node inside the symbol manager (optional).

`str` - is a text string which determines the parameter name, path, mask and icon.

`str` has to be written inside quotation marks (e.g. `icon="symbols.gif"`)

`</symbols>`

tag marking the end of defining a symbol.

Default Path

The default path displayed in the Symbol Manager can be set either to an existing path name or to the symbol tree names as defined in the file *example_symtree.xml*. The definition is done by Bacis2 commands which can be applied in the console or in a customized UID file.

The lines which have to be entered in the console are:

```
ui!sym_load_named_dialog!pathname:-"symtree:Symbole 1\Schrauben"  
ui!sym_load_named_dialog!filename:-" "
```

The lines for the UID file are:

```
sym_load_named_dialog.pathname:"symtree:Symbole 1\Schrauben"  
sym_load_named_dialog.filename:" "
```

If you want a standard folder to be default path, enter for example:

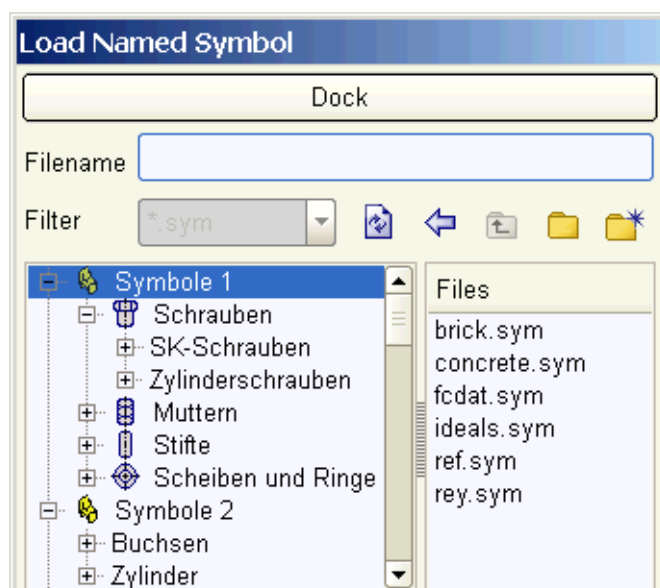
```
ui!sym_load_named_dialog!pathname:-"c:\prod\symbols\screws"
```

Please note: The defined path is temporary. If you change the path and load a symbol out of it, this last used path will become default directory and will be displayed when opening the Symbol Manager the next time.

Example

The following picture shows the symbol manager for the file *example_symtree.xml*, whose syntax is given on the following pages.

Figure 13 Example Symbol Manager



Contents of the file *example_symtree.xml*:

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE medusa>
<!-- Level;Name;Pfad;Dateimaske;Iconname -->
<symbol_manager>
  <!-- Symbolbibliothek 1 -->
  <symbol_library>
    <symbols name="Symbols 1" path="." mask="*.sym" icon="symbols.gif">
      <symbols name="Screws" path="./MASCHBAUBIB" mask="*Schraube*.sym"
        icon="csg_schraube.gif">
      <symbols name="Hexagon Screws" path="./MASCHBAUBIB"
        mask="SK_Schraube*.sym">
    </symbols>
    <symbols name="Cylinder Head Screw" path="./MASCHBAUBIB"
      mask="Z_Schraube*.sym">
    </symbols>
  </symbols>
  <symbols name="Nuts" path="./MASCHBAUBIB" mask="*utter*.sym"
    icon="csg_mutter.gif">
    <symbols name="Hexagon Nuts" path="./MASCHBAUBIB" mask="SK_Mutter*.sym">
  </symbols>
  <symbols name="Cap Nut" path="./MASCHBAUBIB" mask="Hutmutter*.sym">
  </symbols>
</symbols>
<symbols name="Pins" path="./MASCHBAUBIB" mask="*stift*.sym"
  icon="csg_stifte.gif">
  <symbols name="Set Pin" path="./MASCHBAUBIB" mask="G*stift*.sym">
  </symbols>
  <symbols name="Spring Pin" path="./MASCHBAUBIB" mask="Sp*stift*.sym">
  </symbols>
</symbols>
<symbols name="Washers and Rings" path="./MASCHBAUBIB" mask="Scheibe*.sym
  Fe*.sym" icon="csg_scheibe.gif">
  <symbols name="Washers" path="./MASCHBAUBIB" mask="Scheibe*.sym">
  </symbols>
  <symbols name="Spring Rings" path="./MASCHBAUBIB" mask="Feder*.sym">
  </symbols>
</symbols>
</symbol_library>
  <!-- Symbolbibliothek 2 -->
  <symbol_library>
    <symbols name="Symbols 2" path="." mask="*.sym" icon="symbols.gif">
      <symbols name="Sockets" path="./BUCHSEN">
        <symbols name="Sockets" path="./BUCHSEN/BUCHSEN"/>
        <symbols name="slitted Sockets" path="./BUCHSEN/GESCHLITZTE_BUNDBUCHSEN"
          mask="*.sym">
      </symbols>
```



```
</symbols>
<symbols name="Cylinders" path="./ZYLINDER">
  <symbols name="Hydraulics" path="./ZYLINDER/hydraulik">
    <symbols name="Guide Slide Bearing" path="./ZYLINDER/hydraulik
      /gleitfuehrung" mask="*.sym">
      </symbols>
    <symbols name="Slide Mounting" path="./ZYLINDER/hydraulik
      /schienenfuehrung" mask="*.sym">
      </symbols>
    </symbols>
  </symbols>
  <symbols name="Pneumatics" path="./ZYLINDER/pneumatik" mask="*.sym">
  </symbols>
</symbols>
</symbol_library>

<!-- Symbolbibliothek 3 -->
<symbol_library>
  <symbols name="Diagram Symbols" path="./diagrams" mask="*.sym" icon="symbols.gif">
  </symbols>
</symbol_library>
</symbol_manager>
```

Layer Set Management

Please note: For the layer set management you have to be in Administrator mode. To enter Administrator mode choose the entry Admin Mode from the Options menu.
If you are not in Administrator mode all entries of the tab below are disabled.


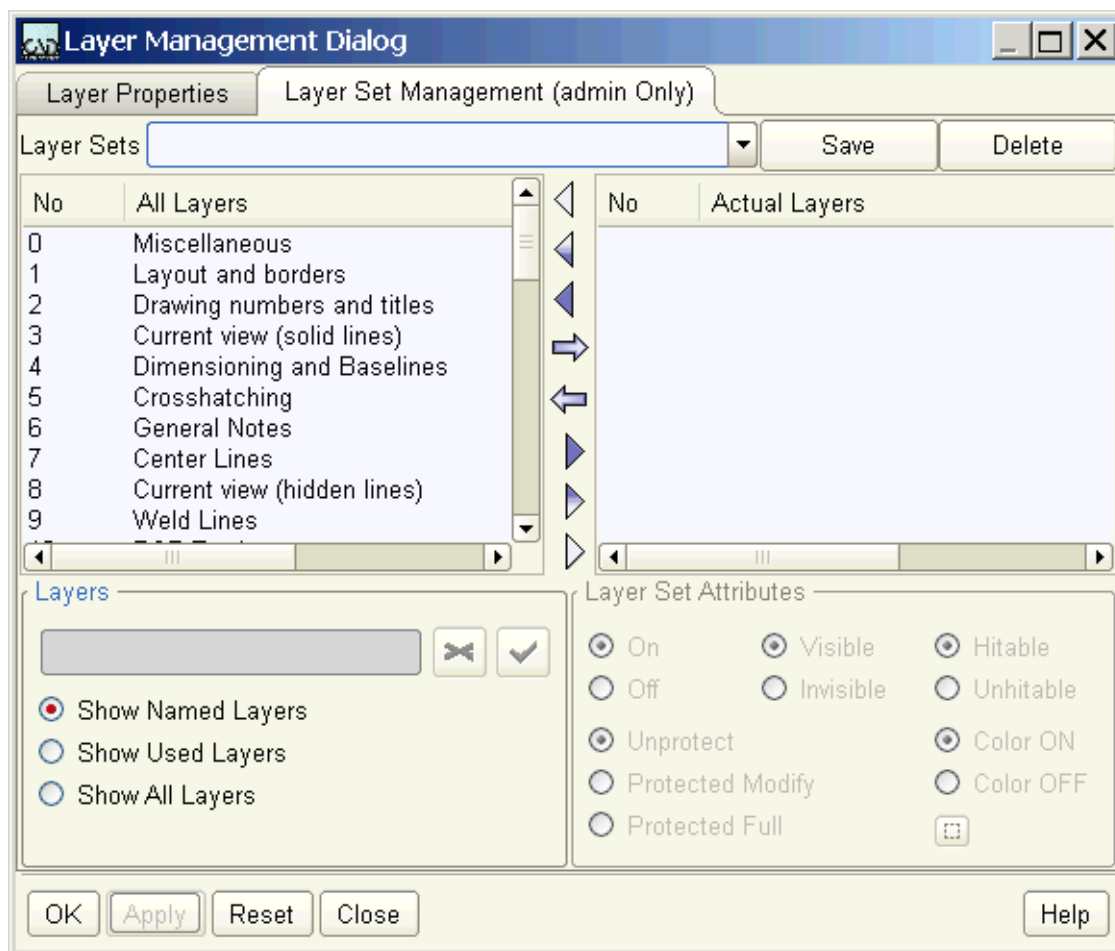
1. Choose either the Layer Manager option from the entry Layers in the menu bar, or, if you are just creating or editing an element, *click left* on the button View or change the layer properties  in the dashboard for opening the Layer Management Dialog.
Two tabs are available, for Layer Properties and for Layer Set Management (admin only).
2. Click on Layer Set Management (admin only) for displaying its contents.

Figure 14 Layer Manager: Tab Layer Set Management (admin Only)



The first entry on the tab Layer Set Management (admin only) is:

Layer Sets

This text field displays the currently chosen layer set. It is empty by default, indicating that no layer set is chosen. If you select a layer set from the pulldown list available by

the arrow on the right of the text field, the Layer Set Attributes of the chosen set are displayed. How to add, edit and delete a layer set is given in “Customize Layer Sets” on page 71.

Save

This button allows you to save the current layer set.

The appropriate file, *layerset.xml*, is in the user project of the current MEDUSA project, e.g. *master_project\user\username\layerset.xml*.

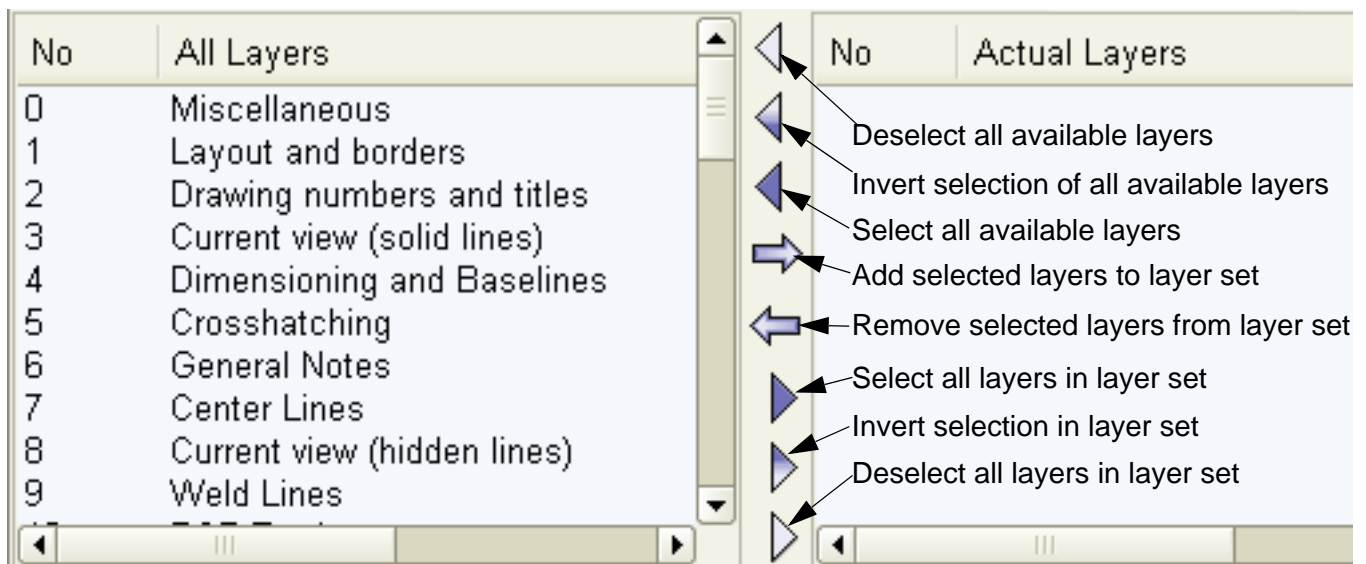
Please note: If you create new the user project, the layer set file will be deleted. Therefore we recommend to copy this file in the user product directory, e.g. *custom\m2d\src\layerset.xml*, and then reconfigure your MEDUSA project. Details on this can be found in the *Customization Guide*, chapter *Running MEDCONFIG*, section *Adding a User-Product Directory*.

Delete

deletes the current layer set.

Below these entries you find **two lists**.

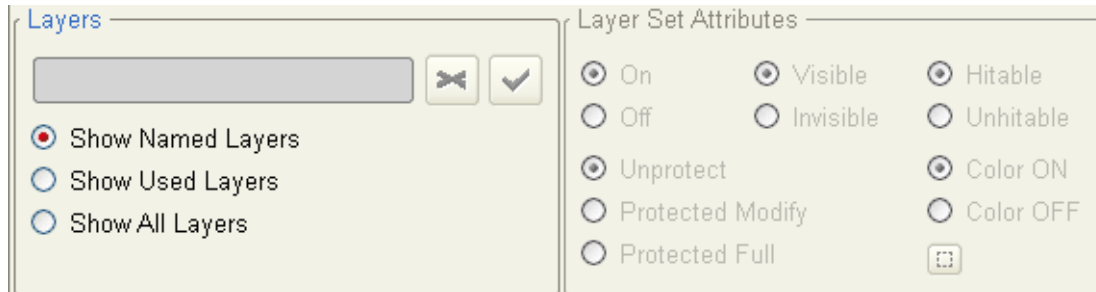
Figure 15 Layer Manager: Tab Layer Set Management, Layer Lists



- On the left you see the list of All Layers (depending on the show options in the Layers area; by default all layers are listed which have a name)
- On the right you see the list of layers which belong to the current layer set (Actual Layers).
- Between the lists you find some buttons used for selecting and deselecting layers either in the left or right list, and for moving layers from one list to the other.

Below the lists of the Layer Set Management tab you find the section **Layers** on the left hand side and **Layer Set Attributes** on the right.

Figure 16 Layer Manager: Tab Layer Set Management, Options



The **Layers** area:

Text field (below the list of all layers on the left)

The field is disabled by default. If you click double on a layer number inside the list above, the field becomes enabled and the name is displayed in it. Now you can delete or change the layer name. Details are given in [“Customize Layers” on page 70](#).

Show Named Layers, Show Used Layers, Show All Layers

Options for displaying either all named layers, all layers used in the current sheet, or all layers available in MEDUSA (which are the layers numbered from 0 to 1023) in the left list.

On the right of the Layer Set Management (admin only) tab you find the section **Layer Set Attributes**.

Please note: All entries in the area Layer Set Attributes are enabled not before the current layer set was saved.

On, Off

switches on or off the visibility and selectivity at the same time.

Visible, Invisible

switches on or off the visibility for the selected layer set.

Hitable, Unhitable

switches on or off the selectivity for the selected layer set.

Protected Modify, Protected Full, Unprotected

- Protected Modify behaves like full protection except that if you load or save symbols with elements which are on layers of the layer set, these elements are not loaded or saved.
- Protected Full prevents elements on layers of the layer set from being created, deleted or modified.
- Unprotected allows to create, change or delete elements on layers of the layer set.

Color ON, Color OFF

switches on or off the color chosen by the color selection button.



Color selection button

This is the color used for drawing elements on the layers of the layer set when the color switch `Color` is on. By default no color is defined. How to select a color or define a new one is explained in the *Drafting Guide*, chapter *Grids*, section *Grid Properties*, sub-section *Specifying the Grid Line Style, Color*.

The **buttons at the bottom** of the Layer Set Management (admin only) tab are:

Reset

sets the layer settings in the dialog back to the values of the sheet.

OK, Apply,

uses the current settings. In case of `OK` the dialog is closed. After `Apply` the button is disabled until you changed something in the dialog.

Cancel, Help

work as usual.

Customize Layers

Please note: You can customize layers when you are in Administrator mode only.

In MEDUSA you always have 1024 layers numbered from 0 to 1023, so you cannot add new layers which have a higher number.

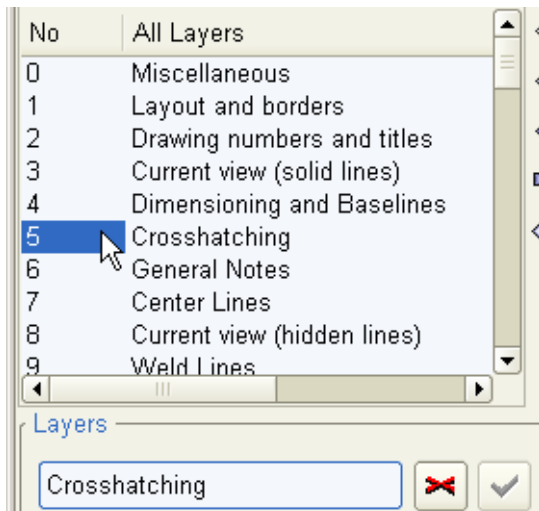
This section shows you how to rename layers and how to delete layer names.

Rename a Layer

For renaming a layer do the following:

1. Open the dialog Layer Management Dialog by choosing Layers > Layer Manager in the menu bar and then the tab Layer Set Management (admin only) (see [Figure 14, "Layer Manager: Tab Layer Set Management \(admin Only\),"](#) on page 66).
2. *Double click left* either on a layer number or the layer name.
Below the list the text field becomes enabled and the layer name is displayed.

Figure 17 Example: Rename Layer Name




3. *Click left* in the text field.
 4. Type a new name or change the displayed name.
As you do so the button Delete layer name becomes inactive and the button OK to set layer name is activated.
 5. *Click left* on the button OK to set layer name .
- The layer name is changed and the list is updated to the new name.

The text field and both buttons on the right from it are disabled.
If the currently displayed layer set contains the changed layer, the name is also updated in the list of layers defined in the layer set.

Delete a Layer Name

For deleting a layer name do the following steps:

1. Open the dialog Layer Management Dialog by choosing Layers > Layer Manager in the menu bar and then the tab Layer Set Management (admin only) (see Figure 14, “Layer Manager: Tab Layer Set Management (admin Only),” on page 66).
2. *Double click left* either on a layer number or the layer name.
Below the list the text field becomes enabled and the layer name is displayed. Additionally the button for deleting the layer name is activated.
3. *Click left* on the button Delete layer name .
The appropriate name in the list changes to Layer_<No> (No is the current layer number) to mark it as undefined. If the show option Show Named Layers is chosen, the layer is removed from the list.

Customize Layer Sets

Please note: You can customize layer sets when in Administrator mode.

There are two possibilities to define your own layer sets:

- a. using the Layer Manager or
- b. creating a *layerset.xml* file in the customer product.

Defining Layer Sets using the Layer Manager

To create, modify or delete layer sets open the dialog Layer Management Dialog by choosing Layers > Layer Manager in the menu bar and then the tab Layer Set Management (admin only) (see Figure 14, “Layer Manager: Tab Layer Set Management (admin Only),” on page 66).

To add a new layer set:


1. Click left in the text field Layer Sets.
2. Type a new name for the set into the entry box.

3. Click the *Save* button to create the new layer set.
The new layer set is available in the pulldown list now.


To **change the name** of a layer set:

1. Select a layer set from the list available by clicking on the arrow on the right of the layer sets text field.
2. Enter a new unique name in the *Layer Sets* field.
3. Click the *Save* button.
If the name is unique the new layer set is saved immediately.
If the layer set already exists you are asked for overwriting it.

To **add layers** to a layer set:

1. Select a layer set.
The list on the right *Actual Layers* displays the layers which are defined in the current set.
2. *Click left* on layers in the left list which shall be added to the layer set.
If a layer is highlighted, it is selected and ready for moving to the right list *Actual Layers*.
If you click on an already highlighted layer, it is deselected.
3. *Click left* on the button *Add layer to set* .
4. Click the *Save* button for storing the changes on the layer set.

To **remove layers** from a layer set:

1. Select a layer set.
The list on the right *Actual Layers* displays the layers which are defined in the current set.
2. *Click left* on layers which shall be removed from the layer set in the list *Actual Layers*.
If a layer is highlighted it is selected and ready for moving to the other list.
If you click on an already highlighted layer, it is deselected.
3. *Click left* on the button *Remove layer from set* .
4. Click the *Save* button for applying the changes on the layer set.

To **delete a layer set**, select the set and click the *Delete* button.

Defining Layer Sets in Product Directories

It is possible to have more than one *layerset.xml* files. They have to be placed into the product directories into *m2d\src*. MEDUSA looks through all *m2d\src* directories that are in the product list, starting from top, and merges the founded *layerset.xml* files. An example for a *layerset.xml* file is shown below:


```
<layersets>
  <layerset>
    <name>SET1</name>
    <color_on>>true</color_on>
    <color_pen>0</color_pen>
    <hitable>>true</hitable>
    <protection>3</protection>
    <visible>>true</visible>
    <layers>1-3 5</layers>
  </layerset>
  <layerset>
    <name>SET2</name>
    <color_on>>true</color_on>
    <color_pen>0</color_pen>
    <hitable>>true</hitable>
    <protection>2</protection>
    <visible>>true</visible>
    <layers>10-23 36-38 99</layers>
  </layerset>
</layersets>
```

protection:

- 1 - protected modify
- 2 - proteted full
- 3 - unprotected

Please note: Changes of the layer sets should be always done in the customer product not in the product directories of the original product. For information about adding user product directories see the *Customization Guide* please.

User-Defined Dashboards

Dashboards are an important part of the user interface of MEDUSA and many dashboards are available for giving fast information on selected elements like lines, texts and prims.

In addition to the usual dashboards it is possible to create your own dashboards for groups, for which you want to specify certain information which shall be displayed in both the dashboard itself and in the associated properties dialog. The information like properties and attributes displayed in the dashboard and properties dialog can be defined differently in order to give the main information in the dashboard only and all information on a group in the properties dialog, for example.

Please note: In order to define a new dashboard you have to be in **Admin Mode**.

This section contains the following topics:

- [Creating a User-Defined Dashboard](#)
- [“The Create Dialog” on page 77](#)
- [“Using User-Defined Dashboard” on page 79](#)
- [“Modify User-Defined Dashboard” on page 82](#)
- [“Deleting User-Defined Dashboard” on page 82](#)

Creating a User-Defined Dashboard

For creating a new dashboard, do the following steps:

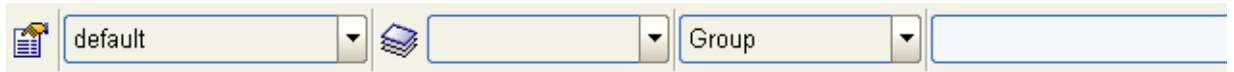
1. You firstly need to select the group for which the dashboard is required.
You can do this by:
 - a. selecting a group in the structure tree, or by
 - b. selecting an element in the group and press the `Up one level` button in the menu bar,

Figure 18 “Up One Level” button in the Menu Bar



When a group is selected, a group dashboard will be displayed. For a diagram symbol it looks like this:

Figure 19 Group Dashboard




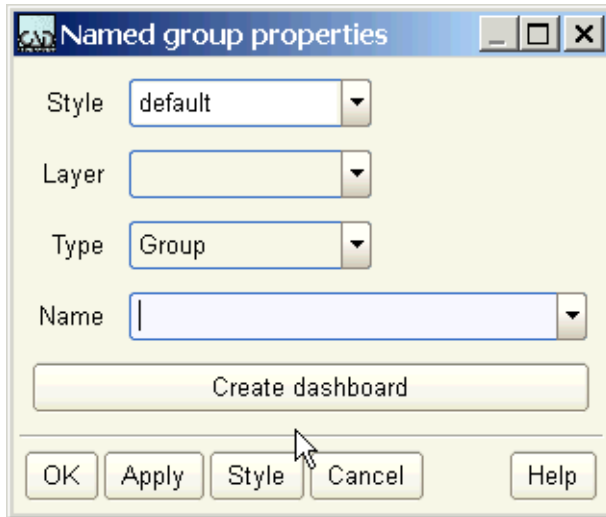
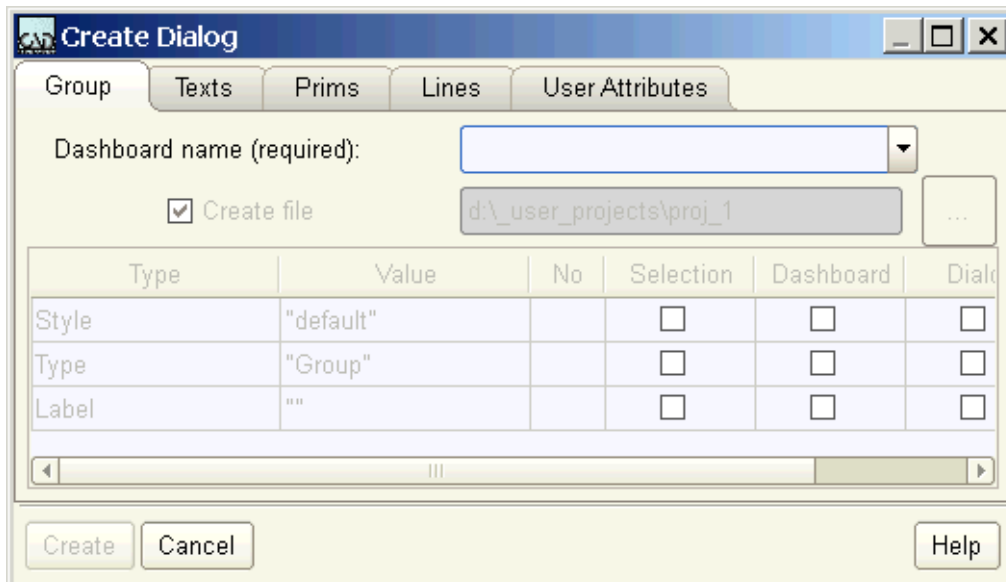
2. Call up the associated properties dialog by:
 - a. selecting the Properties ... option on the *right mouse popup* menu, or
 - b. using the Properties ... button  on the group dashboard.

Figure 20 The Group Properties Dialog



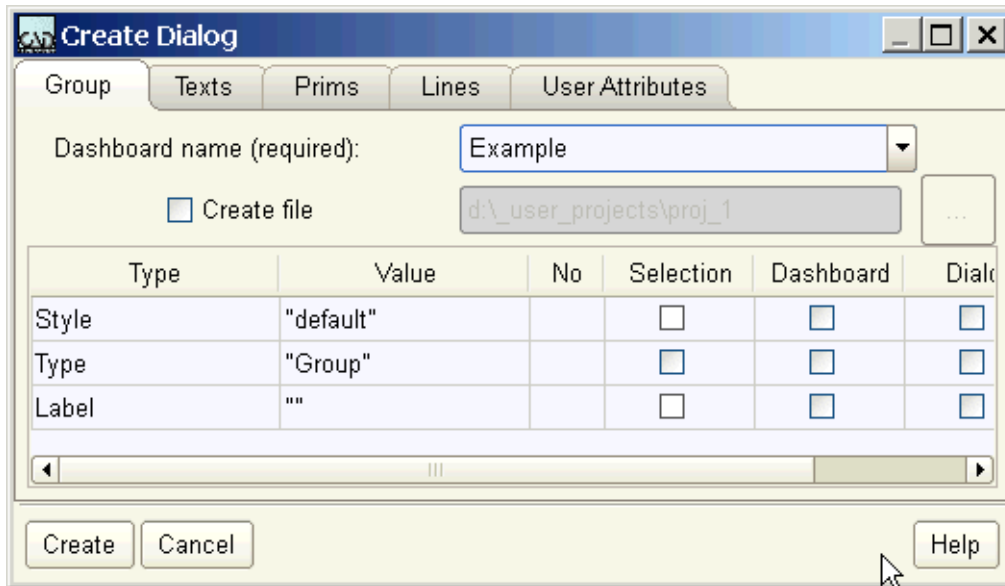
3. Click on the Create dashboard button.
The Create Dialog is displayed.

Figure 21 The Create Dialog



4. Enter a Dashboard name to enable the dialog.

Figure 22 Activated Create Dialog



5. Define the selection criteria and the information you want to be displayed in the Dashboard and in the properties Dialog.
For details on the entries of the dialog see [“The Create Dialog” on page 77](#).
6. If you want to use the dashboard also in subsequent sessions of MEDUSA, activate the option `Create File`.

Please note: If you do not activate `Create File`, the user defined dashboard is only created temporary and it will be lost after quitting the current MEDUSA session.

7. When you are satisfied with your settings, press the `Create` button.
The default group dashboard and properties dialog will be replaced by that one you have just defined.
The new dashboard and dialog will appear each time that you select a group that corresponds to your selection criteria during the current session.
If you want to make the new dashboard a permanent part of your MEDUSA environment read [“Adding a User-Defined Dashboard to Customer Product” on page 80](#)
The following figures show the user-defined dashboard and the accordant properties dialog for the **example** of a Diagram Symbol where in the `Texts` tab of the `Create Dialog` `Type LB2` was set to be displayed in the dashboard. All other texts were set to be displayed in the properties dialog.

Figure 23 User-Defined Dashboard

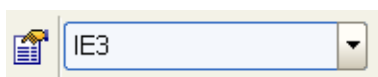
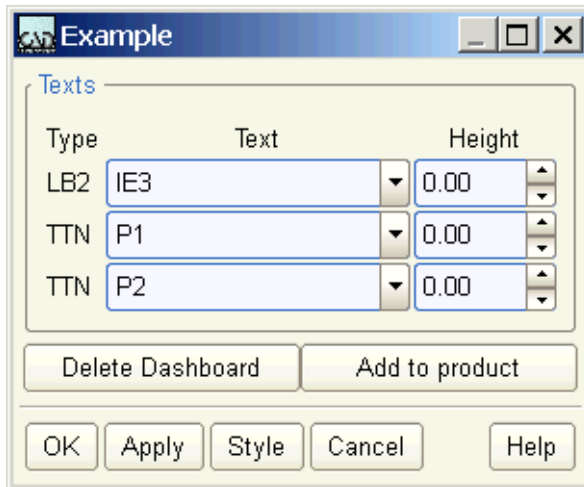


Figure 24 User Defined Properties Dialog



The Create Dialog

The Create Dialog can be opened inside the Named group properties dialog after selecting a group.

Please note: If a user defined dashboard already exists for the selected group, the Named group properties dialog is not available but the dialog you defined with the user defined dashboard.

Figure 25 The Group Properties Dialog

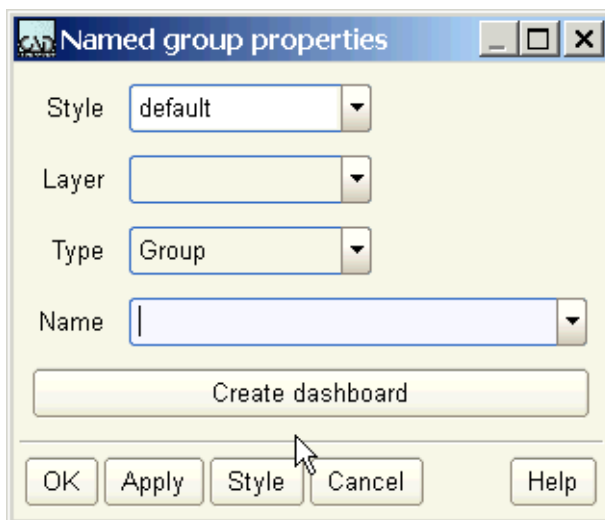
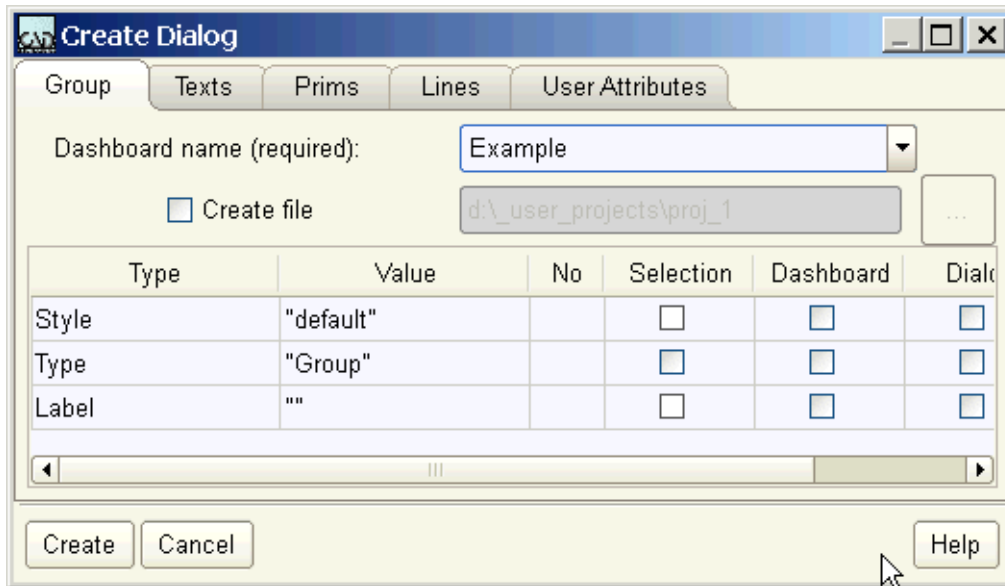


Figure 26 The Create Dialog



The tab pages of the dialog contain all the elements of the group and enable you to select those you wish to be displayed in the dashboard and the properties dialog, also those that identify this particular group.

The Create Dialog provides the following **tabs**:

Group

contains data associated with the group itself.

Texts

lists the texts contained in the group.

Prims

lists the prims contained in the group.

Lines

lists the lines contained in the group.

User Attributes

lists any user attribute of the group according to type and value.

Please note: If any of these particular elements or attributes is not present in the group, the accordant tab page is disabled.

Each of the tab pages contains a **table** which has the following **columns**:

Type identifies the types of elements for the particular tab. For example `Solid` might appear in the `Lines` tab. In the `User Attributes` tab the `Type` is the name of the attribute, for example `attr1`.

Value Where applicable, this shows the current value of the element. This might be the current text string or attribute value.

No	shows the number of elements of this type that are in the group. This applies only to texts, prims and lines.
Selection	<p>The option enables you to specify the characteristics of the group that this dashboard/dialog is defined for. The entries for which <code>Selection</code> is marked are used as selection criteria when selecting a group.</p> <p>It is usual to tick items that are specific to the group you intend.</p> <p>For a Diagram Symbol you might tick <code>LB2</code> in the <code>Texts</code>, <code>Superprim</code> in the <code>Prims</code> and some or all of the <code>User Attributes</code>. You might not tick <code>TTN</code> texts because the selection would be restricted to the number <code>TTN</code> texts in the particular diagram symbol you selected.</p> <p>It makes no sense to tick common group properties such as <code>Free for Style</code>, <code>Group for Type</code> and “ ” for <code>Label</code>, since these are used for many groups.</p>
Dashboard	<p>These tick boxes indicate the fields that are to appear on the dashboard itself.</p> <p>For the Diagram Symbol you might tick just <code>LB2</code> text.</p>
Dialog	<p>These tick boxes indicate the fields that are to appear in the properties dialog.</p> <p>For the Title Block you might repeat those on the <code>Dashboard</code> and add the other user-definable fields. For the Diagram Symbol you might select all texts.</p>

Using User-Defined Dashboard

There are the following cases in which you can use user-defined dashboards.

- Use in the session in which you create a user-defined dashboard.
- Use after adding a user-defined dashboard, created in a previous MEDUSA session, manually to the current session.
- Use by adding a dashboard permanently to a product.

Please note: To be able to use a dashboard created in a previous MEDUSA session, it has to be saved into a file by activating the `Create file` option in the `Create Dialog` as described in [“Creating a User-Defined Dashboard” on page 74](#).

For an example of a user defined dashboard and the appropriate properties dialog see the end of the section [“Creating a User-Defined Dashboard” on page 74](#).

Using a Dashboard in the Session in which you create it

As long as you run the session in which you created a user-defined dashboard, this dashboard is automatically displayed whenever you select a group which matches the selection criteria of the created dashboard.

Adding Dashboard to the Current Session Manually

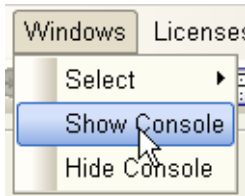
If you saved a user-defined dashboard to a file, you can re-use the file by executing it in the current session. The file is executed by typing the according command into a console window (DOS Box).

Please note: Adding a dashboard manually during a session can be executed only in API mode. Start with the command:

```
call <medusa>project\m2d\command\medusa.bat -api
```

The console is displayed by selecting **Show Console** from the **Windows** option of the menu bar.

Figure 27 Showing the Console



When the console is displayed, please type, for example:

```
exec diagram_symbol.bad
```

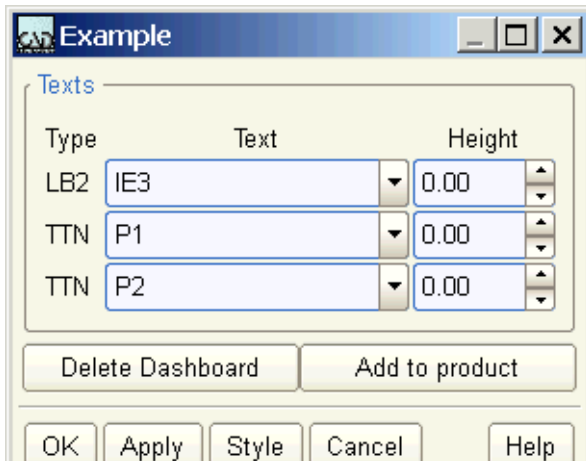
If you changed the directory, you have to insert the complete path to the file, for example:

```
exec D:\projects\medusa_proj_1\diagram_symbol.bad
```

Adding a User-Defined Dashboard to Customer Product

Having created a user-defined dashboard, you may wish to make it more permanent part of your MEDUSA environment. This is done by adding the dashboard to the product. The properties dialog provides the **Add to product** button.

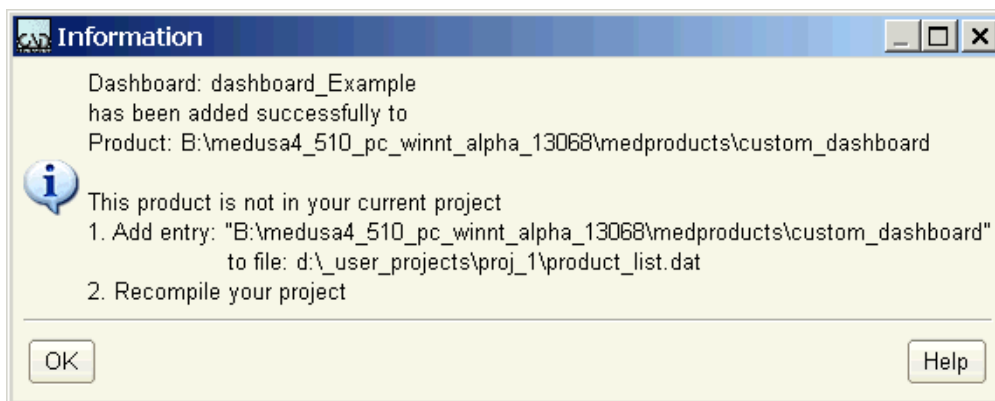
Figure 28 User-Defined Properties Dialog



Please note: Adding a user-defined dashboard to a customer product has to be done while creating the dashboard, see “[Creating a User-Defined Dashboard](#)” on page 74.

1. When you press the **Add to product** button, the **Product Directory Selector** opens. Here you can choose the directory, where you wish to save the dashboard. It is recommended to use the default directory. This is *custom_dashboard* in the product.
2. When you have selected the directory, click **OK**.
If this is the first time you have added a dashboard to this directory, you will see an **Information window**.

Figure 29 Information Window

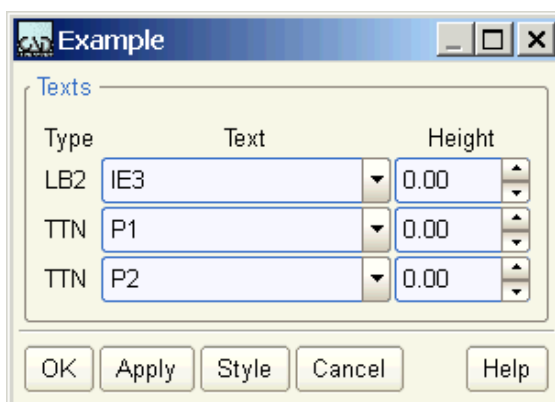


It provides information about how to add the directory to your product.

3. Add the chosen directory to your product list according to the indication in the dialog.
4. Quit MEDUSA.
5. Recompile your project.

In future MEDUSA sessions the new dashboard and the properties dialog are displayed, when you select a group matching the dashboard selection criteria. The properties dialog provides no longer both the **Delete dashboard** button and the **Add to product** button.

Figure 30 User-Defined Properties Dialog



Remove User-Defined Dashboard from Customer Product

Once a dashboard has been added to your product it can be removed in two ways:

- Deactivate the product in the product list.
Comment out or delete the line you added to your product list and recompile the project. This removes all user-defined dashboards.
- Remove a particular dashboard
To remove just a specific dashboard, find the directory where the definition file is stored, for example *custom_dashboard\ui\med2d*
Delete or comment out the dashboard in the *library-init.bar* file.

Example for a *library-init.bar* file:

```
proc(library)
  -- Callback routines for initialisation of custom dashboards
  -- To remove a dashboard, comment out or delete the relevant entry
  -- and recompile your project
  m2d_callback_set(!dashboard_startup,dashboard_diagram_symbol)
endproc
```

In this example the line which has to be deleted or commented out is:

```
m2d_callback_set(!dashboard_startup,dashboard_diagram_symbol)
```

Type into the MEDUSA console (visible via Window > Show console):

```
dir:-"<path to the source files>\custom_dashboard\ui\med2d"
compile_all_source_files(dir)
library_create_index(dir)
```

The dashboard will no longer be available in subsequent MEDUSA sessions.

Modify User-Defined Dashboard

You cannot modify a user defined dashboard. If you want to change the definitions for a dashboard, you need to delete the dashboard and then create it new.

Deleting User-Defined Dashboard

If the dashboard you have created is not correct or you no longer wish to keep it, press the Delete dashboard button (see [Figure 24, "User Defined Properties Dialog," on page 77](#)). Now the default group dashboard and properties dialog will be seen again.

Please note: If you delete a user-defined dashboard created also as a file, it does not delete the file! So, you can execute the file in following sessions in order to use the dashboard again (see “Adding Dashboard to the Current Session Manually” on page 80 and “Adding a User-Defined Dashboard to Customer Product” on page 80).

Dashboard Element Selection Limits

Some dashboard actions depend on the count of selected elements. If the number of selected elements exceeds the limit count, the dashboard is not displayed or updated. The following table shows the default and maximum values for limits:

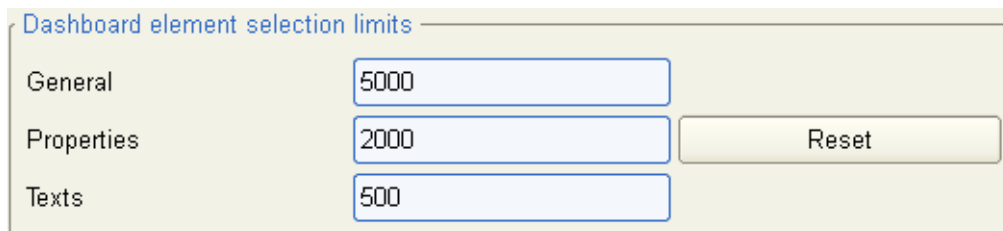
Element	Default Value	Maximum Value
General	5000	1000000
Properties	2000	4000
Texts	500	1000

You can define the limits either in the Defaults dialog (Options > Defaults > Common) or in the *defaults.dat* file of the product.

Setting Limits in the Defaults Dialog

Figure 31 shows the part of the Common tab, where you can insert the desired values.

Figure 31 Defaults Dialog: Tab Common, Dashboard Element Selection Limits



Dashboard element selection limits

General	<input type="text" value="5000"/>	<input type="button" value="Reset"/>
Properties	<input type="text" value="2000"/>	
Texts	<input type="text" value="500"/>	

Please note: The higher the values are, the longer MEDUSA needs for checking. If you exceed the maximum value, a message dialog is given and the value is reset to its maximum.

In order to return to the default values, press the *Reset* button.

When you quit MEDUSA, you are asked if you want to save the changes of the defaults or not. If you confirm the changes, the new values will be taken over, and are valid for following MEDUSA sessions.

Setting Limits in the defaults.dat File

The *defaults.dat* file is located in the directory *med2d\m2d\src* of the product. You find the keywords for the limits setting as follows:

```
-----  
-- Dashboard element selection limits for show / not show a dashboard  
-----  
-- countlimit_dashboard  integer  5000  
-- countlimit_props      integer  2000  
-- countlimit_text       integer  500
```

The rows are commented out by default. To use the defined limits delete the dashes and modify the values as desired. Consider the maximum values given above in the table. Recompile your project by running `medconfig -admin`.

Please note: Changes in the *defaults.dat* should be always done in the customer product not in the original product.

Create Balloon Text Symbols

This section shows how to create Parts Balloon Symbols and the Definition of Parts Balloon Tables.

Entries in the defaults.dat

To define the default symbol for loading Parts Balloons an entry inside the *defaults.dat* exists.

The name for the symbol (*.sym) and the description file (*.bac) must be the same. How this *bac* file is organized is described in [“Creating user defined symbols” on page 85](#).

```
parts_balloon_default string "balloon_def"
```

To define which texts of the Parts Balloon are written to the Parts Balloon table, you need to define the filename of the *bac* file:

```
parts_ballon_table string "balloontable"
```

A search mechanism is looking for the defined files in the products defined in *products_list.dat*. Inside the product the following structure is recommended:

```
...\M2D\SYMBOL\PARTS_BALLOON
```

The entries found inside the last product in the list will be used. The structure of the *bac* file is described in [“Definition of Parts Balloon Tables” on page 89](#).

```
parts_balloon_heights boolean false
```

With this setting you can define whether the height property in the properties dialog is displayed. Default setting is *false*.

Creating user defined symbols

To create a new parts balloon symbol you need to be in Administration mode.

1. First draw the symbol.

It needs a least an ITN clump with the texttypes LB1, PN1, SQT and SGR. You can get these by:

- a. Place a standard balloon text on the sheet (see *Drafting Guide*, chapter *Texts*, section *Balloon Text*).
- b. Edit it either by *double click* or choosing *Edit* from the popup menu.
- c. Modify it as requested.

Any other texts and lines can be added.

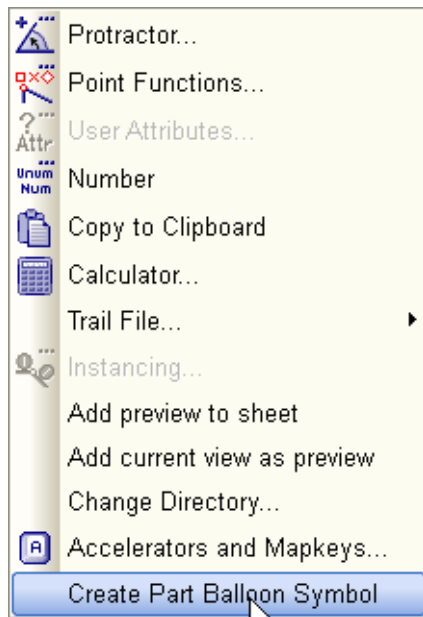
For texts make sure that the text values are always like this: \$1, \$2, \$3, ...

Figure 32 Example for a user defined parts balloon symbol



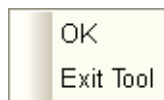
2. Choose the tool Create Part Balloon Symbol from the Utilities menu.

Figure 33 Utilities Menu



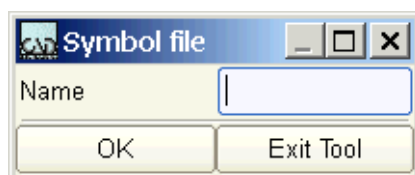
3. Select all the geometry you want to use for the Parts Balloon Symbol.
4. Once you have selected, use the *right mouse button* popup button “ok”.

Figure 34 Popup Menu while creating balloon symbol



A dialog opens.

Figure 35 Create balloon symbol dialog



5. You need to type the name of the symbol here (exactly 3 characters).

6. Confirm with the OK.
7. Finally set a probe for the loading point of the Part Balloon Symbol.
The symbol file will be created at once and the *bac* file will be written into the installation path *med2d\m2d\symbol\parts_balloon*.

For this example the names of the files are:

balloon_abc.sym

balloon_abc.bac

The *bac* file is opened in an editor.

You need to change the attributes of the \$ texts, which are at the end of the file:

```
-- default description for parts balloon abc

-- list of balloon symbol texts
-- (1 - 4 are predefined with standard text, do not change!)
balloon_def :- []

-- Item Number
balloon_descr :- dict_new()
balloon_descr!datatype :- !item_number
balloon_descr!cancode :- "LB1"
balloon_descr!visible :- true
balloon_def :- balloon_def >< [balloon_descr]

-- Part Number
balloon_descr :- dict_new()
balloon_descr!datatype :- !part_number
balloon_descr!cancode :- "PN1"
balloon_descr!visible :- true
balloon_def :- balloon_def >< [balloon_descr]

-- Quantity
balloon_descr :- dict_new()
balloon_descr!datatype :- !quantity
balloon_descr!cancode :- "SQT"
balloon_descr!visible :- true
balloon_def :- balloon_def >< [balloon_descr]

-- Issue
balloon_descr :- dict_new()
balloon_descr!datatype :- !issue
balloon_descr!cancode :- "SGR"
balloon_descr!visible :- true
balloon_def :- balloon_def >< [balloon_descr]

-- additional texts, user defined--

balloon_descr :- dict_new()
```

```
balloon_descr!datatyp :- !datatyp1
balloon_descr!textid  :- "$1"
balloon_descr!type    :- !string
balloon_descr!label   :- "Label1"
balloon_descr!browse  :- "Label Description1"
balloon_descr!default :- "DEFAULT"
balloon_descr!precb   :- unset
balloon_descr!postcb  :- unset
balloon_descr!visible :- true
balloon_def :- balloon_def >< [balloon_descr]

balloon_descr :- dict_new()
balloon_descr!datatyp :- !datatyp2
balloon_descr!textid  :- "$2"
balloon_descr!type    :- !string
balloon_descr!label   :- "Label2"
balloon_descr!browse  :- "Label Description2"
balloon_descr!default :- "DEFAULT"
balloon_descr!precb   :- unset
balloon_descr!postcb  :- unset
balloon_descr!visible :- true
balloon_def :- balloon_def >< [balloon_descr]
```

You need to give sensible data to the following issues:

```
balloon_descr!datatyp :- !datatyp1
    !datatyp needs to correspond to the !datatyp entry in the table description (see
    "Definition of Parts Balloon Tables" on page 89).
```

```
balloon_descr!textid  :- "$1"
    !textid needs not to be changed, this points to the text of the symbol.
```

```
balloon_descr!type    :- !string
    This describes the value type, !string, !integer or !real.
```

```
balloon_descr!label   :- "Label1"
    !label is the label displayed inside the Parts Balloon load dialog, you can either
    insert a string ("Label") or entry of the message system
    (mui_message_get(mui2d_messages, !SUB_BALL2TAB, !position)
```

```
balloon_descr!browse  :- "Label Description1"
    This is the browse text for the dialog field. Here you can also insert a string or message
    system.
```

```
balloon_descr!default :- "DEFAULT"
    This is the default value.
```

```
balloon_descr!precb   :- unset
    Here you can insert a callback to a function that is called before the symbol is loaded,
    e.g. values for a combo box. For an example see:
    M2D_PROD\MED2D\M2D\SYMBOL\PARTS_BALLOON\ balloon_usr.bac
```



```
balloon_descr!postcb :- unset
```

Here a callback to a function that is called after loading the symbol can be defined.

```
balloon_descr!visible :- true
```

This defines if the description will be visible, if it is set to `false` the value for this property cannot be modified in the dialog, dashboard or properties dialog. This property may also be changed for Part Number, Quantity or Issue.

WARNING: Do not change this rows!

```
balloon_def :- []
```

```
balloon_descr :- dict_new()
```

```
balloon_def :- balloon_def >< [balloon_descr]
```

Do not change `datatype` and `cancode` of Item Number, Part Number, Quantity and Issue!

For having a new created symbol to be available in the dialog Create Item Number Balloon called by the tool Create New Item Number Balloon in the tooltray Text + Dimension, you have to restart MEDUSA.

Definition of Parts Balloon Tables

It is possible to administrate the layout of Parts Balloon tables. The file

```
... \M2D\SYMBOL\PARTS_BALLOON\balloontable.bac
```

describes the table. If you want to use another file you need to change the setting in the `defaults.dat` (see [“Entries in the defaults.dat” on page 85](#)).

```
-- Configuration for Partsballoon table tool
```

```
balloontable_props :- dict_new()
```

```
-- table header position (!top or !bottom)
```

```
balloontable_props!header_pos :- !top
```

```
-- description of columns
```

```
balloontable_props!cols :- []
```

```
col :- dict_new()
```

```
col!datatype :- !item_number
```

```
col!label :- mui_message_get(mui2d_messages, !SUB BALL2TAB, !position)
```

```
col!width :- 5
```

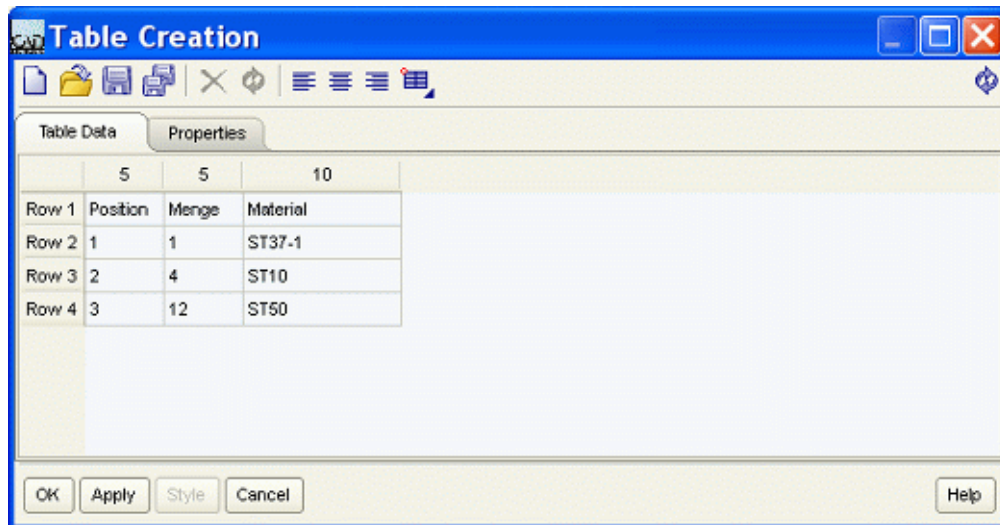
```
balloontable_props!cols :- balloontable_props!cols >< [col]
```

```
col          :- dict_new()
col!datatype :- !quantity
col!label    :- "Menge"
col!width    :- 5
balloontable_props!cols :- balloontable_props!cols >< [col]

col          :- dict_new()
col!datatype :- !material
col!label    :- "Material"
col!width    :- 10
balloontable_props!cols :- balloontable_props!cols >< [col]
```

Only the items that are described here will be added to the table, for the example above the following table will be created:

Figure 36 Create balloon table dialog



Description of the columns:

`balloontable_props!header_pos :- !top`

This describes the position of the header line. The values `!top` or `!bottom` are valid. If you choose `!top` the header will be the first row in the table, the next row will be the lowest balloon symbol position. If you choose `!bottom`, the first row will be the highest position and the last row will be the header line.

`col!datatype :- !material`

This is `datatype` of the symbol text, it must correspond to the entry of the file `balloon_xxx.bac`, that you use to define the symbol.

`col!label :- "Material"`

Here you can insert the name of the table header of each symbol text. You can either insert text or an entry of the message system.

```
col!width :- 10
```

This is the predefined width of a column. A table will always open with this column width.

WARNING: Do not change this rows!

```
balloontable_props :- dict_new()  
col :- dict_new()  
balloontable_props!cols :- balloontable_props!cols >< [col]
```

Define Used Codepage

For MEDUSA you can define which character set (codepage) is used by the operating system. For identifying characters of the used codepage following variables have to be defined inside the file `login.bat`:

```
Windows:  
set MED_APP_ENCODING=utf8  
set MED_OS_ENCODING=<Character Set>  
UNIX:  
setenv MED_APP_ENCODING utf8  
setenv MED_OS_ENCODING <Character Set>
```

<Character Set> can have following values:

<Character Set>	Description
cp1251	MS Windows Cyrillic
cp1253	MS Windows Greek
cp1254	MS Windows Turkish

These are only some examples. There are further character set encodings for other languages. Further information on codepages can be found on this website:

<http://www.microsoft.com/globaldev/reference/WinCP.msp>.

INDEX

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Symbols

? 18

Numeric

2D Binary Files 31
2D Drafting Image
link 32
3D Design Products 32

A

Accessing MEDCONFIG 24
Adding layer sets 71
Adding layers to layer set 72
Admin Mode Password dialog 36
Administration 35
Administrator Mode 36
Attributes
changing values 39
locking 39
use in style 39

B

basis 18
Balloon Text Symbols 85
bold 17
Boldness setting 46
button
View or change the layer properties 66

C

Changing layer set name 72
character set 91
code 17
Codepage 91
Colors
create color map 43
setting 43
configuration
3D Design Products 32
configuration link 2D Drafting Image 32
Configuring a New Project 20

Configuring the Products 29
Create Dialog, User-Defined Dashboard 77
create User-Defined Dashboard 74
Creating a Product List 26
Creating styles 37
Customizing
layer sets 71
layers 70

D

ddl 17
Decortext fonts 17
defaults 17
delete User-Defined Dashboard 82
Deleting
layer sets 72
styles 42
deleting a layer name 71
dfont 17
Dialogs
Admin Mode Password 36
Line Properties 38
Select background color 43
Style Creation 39
Documentation 14

E

Editing styles 37

F

Finding Out the Product Names 22
fits 17
font 17
Fonts
building
dfont command 17
compiling 32

H

Help 14
help 18

I

Installation 7
Installing MEDUSA 8

L

Layer
 customize 70, 71
Layer Manager
 List Buttons 67
layer properties, button 66
Layer Set Attributes 68
Layer Set Management 66
Layer sets
 adding 71
 adding layers 72
 changing name 72
 deleting 72
 remove layers 72
Layers
 adding to layer sets 72
 delete name 71
 remove from layer sets 72
 rename 70
Line Properties dialog 38
Lines
 setting thickness 46
Locking
 attributes 39

M

mdba_application_control 17
mdbc_rebuild_control 17
MED_COMMA_IS_DEZIMAL 14
MED_LANG 14
MED_UNICODE_FILENAMEES 15
MEDCONFIG
 description 13
 start first time 24
 using - 16
message 17
modify User-Defined Dashboard 82
Modifying styles 41

N

Name
 changing - of layer sets 72

O

Online Help 14
options of MEDCONFIG 16

P

Page Labels 51
patts 17
permanent saving of styles 41
plot_config 17
plotting_dev_kit 17
prims 17

Printing documentation from Portable Document Format (PDF) files 6
Product List 25
project reconfiguration 18
Prompts
 Enter command> 19
properties
 layer, button 66
psyms 17

Q

quit 18

R

raster_path 17
reconfigure 18
Reconfiguring a project 21
Removing layers from layer set 72
renaming a layer 70

S

Save Style Tree 41
Saving
 styles 40
 styles permanently 41
Selection criteria, user-defined dashboard 79
Separators of Decimal Values 14
Setting
 standard boldness 46
 standard colors 43
 standard thickness 46
Setting up Page Labels 51
Sheet Header
 area_dlglabel 56
 Blank line 54
 Chapter heading 53
 Chapter of the configuration file 54
 Comment lines 53
 Configuration file 52
 consider_ng 55
 Data lines 53
 dlg_maxchar 55
 dlg_maxrow 54
 emptytextchar 55
 enabled 58
 General Keywords 56
 Keywords 54, 56
 label 57
 layer_protect 55
 mandatory 57
 maxchar 57
 multitext 58
 sfeld_dlgarea_ 56
 Structure of the configuration file 53
 suppress 58
 texttyp 57
 zoom_hor 54
 zoom_ver 54
Sheet structure tree 41
Style Creation dialogs 39

style tree 41

Styles

attributes to use 39

creating 37

creating and editing 37

deleting 42

modifying 41

preventing alteration by locking attributes 39

saving 40

saving permanently 41

using 41

styles 17

T

Text

setting thickness 46

Thickness setting 46

ttable 17

U

UIE_COMMA_IS_DECIMAL 14

uistrings 17

Unicode Filename Support 15

Uninstalling 10

user interface

compile messages 17

User Project 12

User-Defined Dashboard

create 74

Create Dialog 77

delete 82

modify 82

selection criteria 79

use 79

Using styles 41

using User-Defined Dashboard 79

V

Values of attributes

changing 39

View or change the layer properties 66

W

wsconfig 18

X

xresources 18



A B C D E F G H I J K L M N O P Q R S T U V W X Y Z