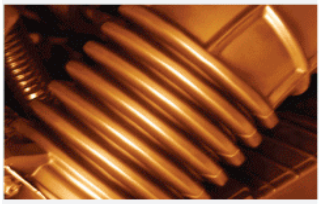
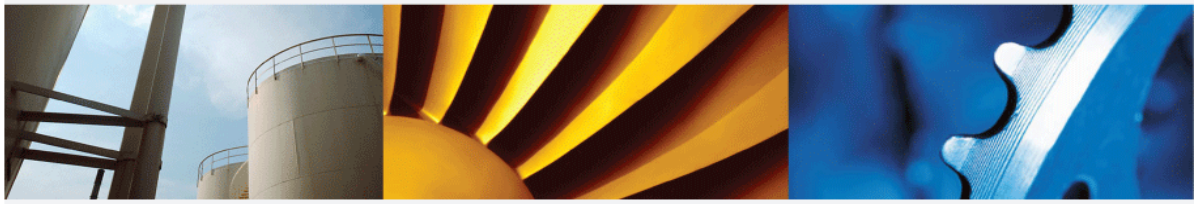
MEDUSA4

VERSION 6.3

Administration

GUIDE





CAD Schroer

www.cad-schroer.com

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, MPDS4

Third-Party Products and Trademarks:
Creo, Pro/ENGINEER, Pro/DETAIL, Pro/TOOLKIT and Windchill are registered trademarks of PTC, Incorporated.

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

September 2018

Copyright © CAD Schroer GmbH

Germany

CAD Schroer GmbH
Fritz-Peters-Str. 11
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

Installation 5
Installing MEDUSA4 . . . 6
Uninstalling . . . 10
Configuring MEDUSA4 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 MEDUSA4 Products . . . 29
Administration 35
Administrator Mode . . . 36
Creating and Editing MEDUSA4 Styles . . . 38
Setting up Standard Colors . . . 46
Setting up Standard Boldness (Thickness) Values . . . 49
Setting up Boldness for Plotting Dimensions . . . 50
Setting up Standard Sheet Filename . . . 51
Setting up the Page Label . . . 54
Setting up the Sheet Header Editor . . . 55
Setting up the Symbol Manager . . . 66
Layer Set Management . . . 71
Customizing Layers . . . 76

Customizing Layer Sets 77
Dashboard - Element Selection Limits 80
Creating Item Number Balloons 82
Generating Styles from DDL Elements 89
Defining the Used Codepage 92
Unicode Text 93

Index 95

© CAD Schroer GmbH

INSTALLATION

This chapter describes how to install MEDUSA4.

- Installing MEDUSA4 6
- Uninstalling 10

Installing MEDUSA4

This section gives some hints which are helpful for the installation of MEDUSA4 with and without the MEDUSA4 Information System (MEDInfo).

Before you start the installation, 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.

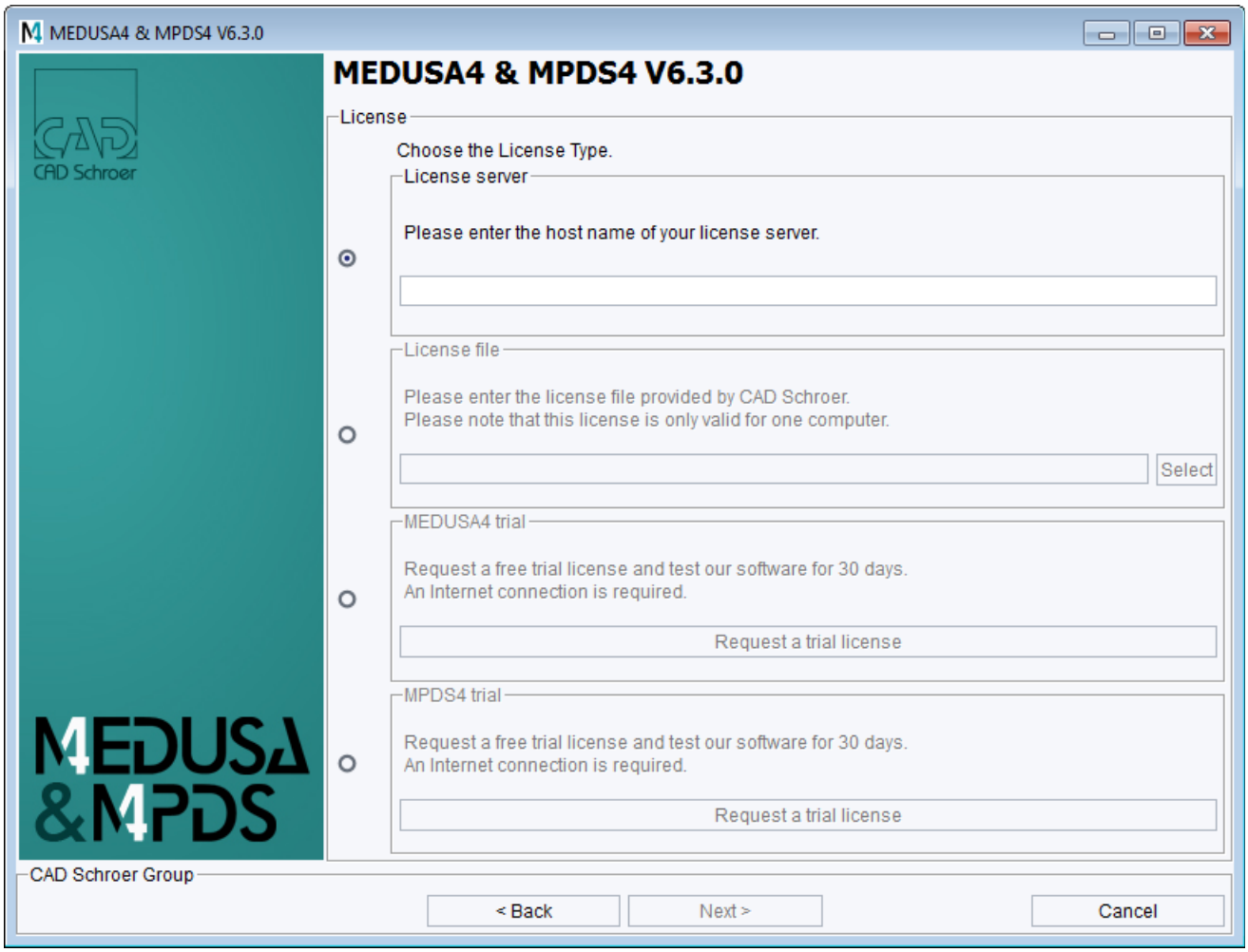
Installation without MEDInfo

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 cannot contain spaces, which is why the default installation path cannot be *C:\Program Files*.
- 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 MEDUSA4 project).

© CAD Schroer GmbH

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*.

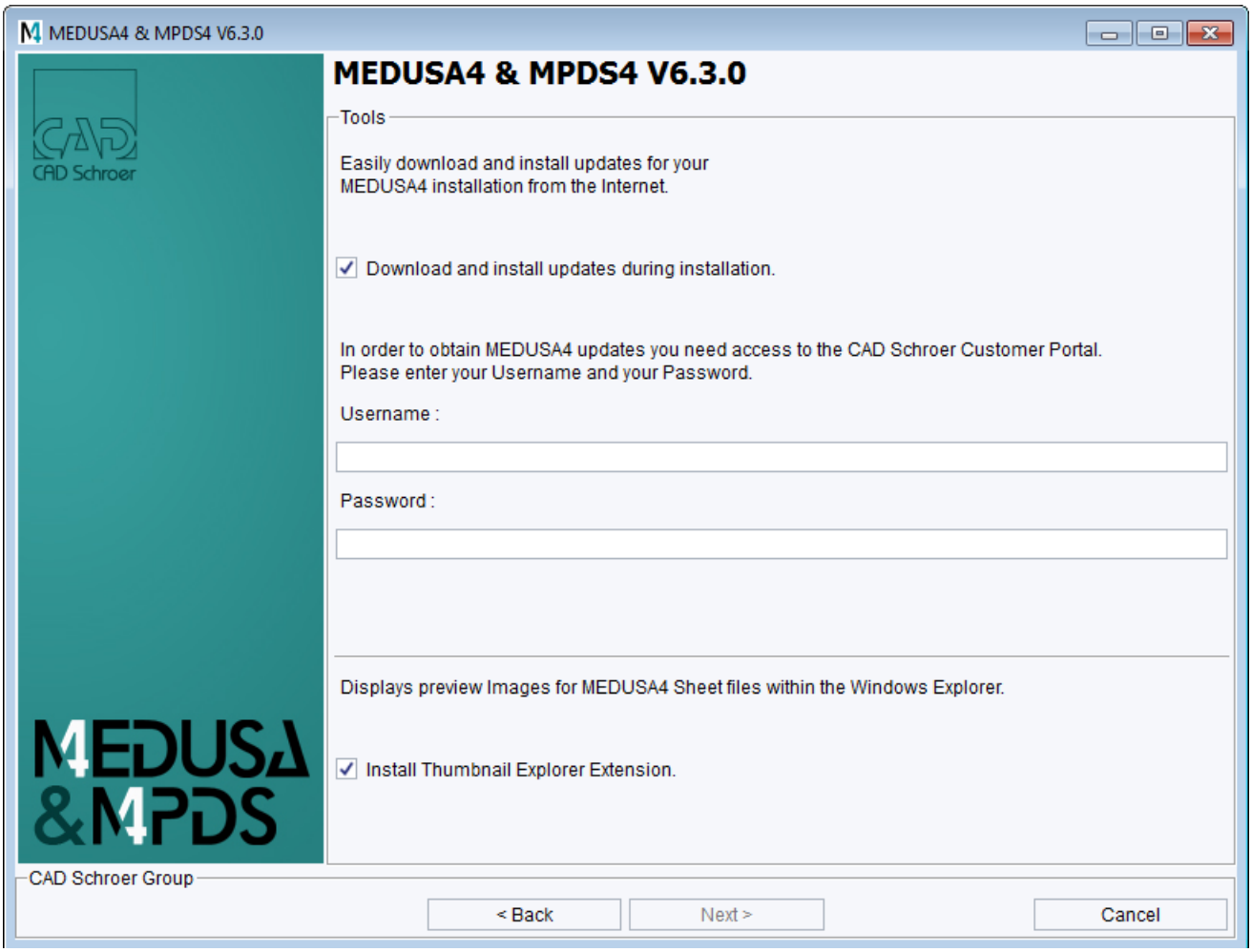
Please note: MEDUSA4 and MEDInfo have to installed on local hard disks, because MEDInfo uses services, which are available locally only. This is valid for the installation on a client as well as on a server.
MEDInfo can be installed any time after MEDUSA4 has been installed. To do this you only have to launch the installation executable from the installation disc and to choose the MEDInfo products.

MEDUSA4 Updates

The CSG Update Client is part of the supported software. In MEDUSA4 6.3 it can be found in the folder *C:\MEDUSA4_V6_3\CSGUpdateClient*.

The CSG Update Client allows you to download and install updates online.

Figure 2 Automatic Updates during MEDUSA4 Installation



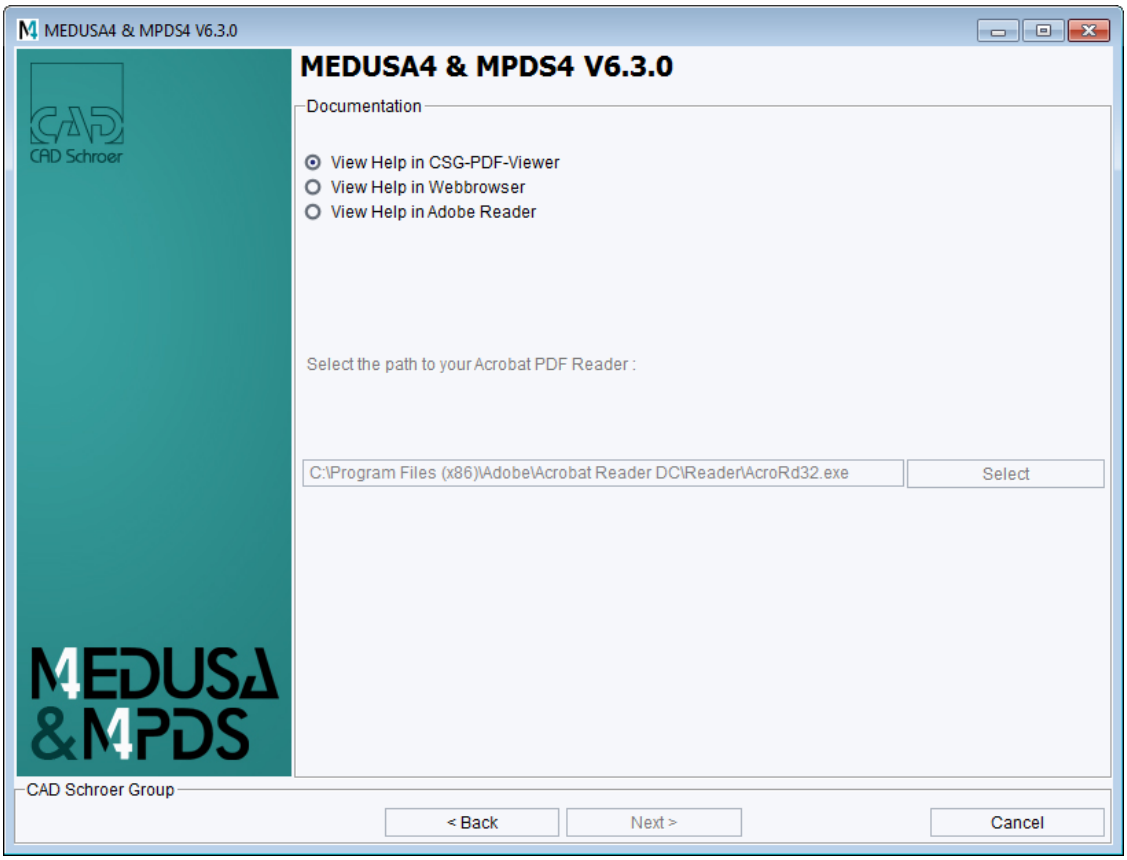
Details to the CSG Update Client are described in the „CSG Update Client“ Guide, which is available with the software.

© CAD Schroer GmbH

CSG PDF Viewer

MEDUSA4 provides its own CSG PDF Viewer for viewing the help (documentation). During the MEDUSA4 installation you are asked for the desired Viewer. Default setting is CSG PDF Viewer.

Figure 3 Installation of the CSG PDF Viewer



Uninstalling

To uninstall the software open in your start menu the options

Programs -> MEDUSA4 ->uninstall MEDUSA4

The deinstallation assistant is started which leads you through the deinstallation process. All components of the application and the whole MEDUSA4 product will be removed.

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

© CAD Schroer GmbH

CONFIGURING MEDUSA4

This chapter describes how you configure a MEDUSA4 project using the MEDUSA4 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 MEDUSA4 Products 29

Description of a Project

When you have finished installing MEDUSA4 (see "Installation" on page 5), you configure a project. A project provides the link between the installed MEDUSA4 products, and is one definition of how you want to run MEDUSA4. This definition includes, for example, the choice of products you want to use, and which the description of workstations you want MEDUSA4 to run on.

As a reference you can use the project *<MEDUSA4 Installation path>\master_project*. Do not change this project.

This following table shows the products of a MEDUSA4 installation in different projects.

Table 1 MEDUSA4 User Project Examples

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

Please note: In case of a localized version an additional product is required. For a German program version it is `medchangeger`. The language is defined by the last characters of the products, `ger` for German, `fra` for French (Französisch), `ja` for Japanese and `it` for Italian.

Each project shown in the previous table, `proj1` through `proj3`, has a different combination of MEDUSA4 products. Each project is independent, but the one thing they have in common is that they use the same source files within MEDUSA4. For example, all three projects use the same `med2d` product.

You can configure workstations to suit the hardware and MEDUSA4 products you intend to use with each project.

© CAD Schroer GmbH

Description of MEDCONFIG

MEDCONFIG is a MEDUSA4 utility you use to configure a project with the MEDUSA4 products you have installed, as well as any customization you have made to MEDUSA4.

How MEDCONFIG Works

MEDCONFIG runs under the MEDUSA4 Executive and uses information provided within the MEDUSA4 products to configure a project. When you start up MEDCONFIG, you give it the pathname 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 MEDUSA4 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 MEDUSA4 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 the project as already configured and 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

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 configu-

ration 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 `mxedef.bin`, `mxedef.fasl` and `login.bat`, which MEDCONFIG writes to your project directory. After this, you are ready to use MEDUSA4.

After Configuration

After configuration you can make same settings.

Online Help - Documentation

You can change the language display of the documentation. The documentation is currently available in English and German language. The language in which the documentation is displayed depends on the value of the environment variable `MED_DOC_LANG` which is located in the *login.bat* of your project. When you configure a project, the environment variable is automatically set in the *login.bat* by the following line:

```
set MED_DOC_LANG=<lang>
```

`<lang>` defines the language and is replaced by following values:
`en` for English, `ger` for German, `fra` for French, `it` for Italian

When you configure a project which contains a medchange-product in the product list, `<lang>` is set according to the relevant language. For a French or Italian medchange-product, `<lang>` is set to `fra` or `it`. However, since no documentation is available in these languages, the English documentation is displayed instead.

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

© CAD Schroer GmbH

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 (*<medusa4>\<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 the environment variable MED_UNICODE_FILENAMES=1 must be set.

Please note: Reconfiguring a project (see "Reconfiguring 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 details are explained 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
`project`	Starts MEDCONFIG in PROJECT mode, where `project` is the full pathname of the project.
`-admin`	Starts MEDCONFIG in ADMIN mode. This gives you the complete range of MEDCONFIG commands you use to compile fonts and menus, for example.
`-printdef filename`	Writes the definition of the MEDUSA4 environment into the file you specify in `filename`. This information includes the products configured in the project, and the variables used for configuration.
`-batch`	Starts MEDCONFIG as a batch job. Use this option with the `reconfigure` option. This option enables MEDCONFIG to run without user intervention as it assumes defaults for all configurable actions.
`-reconfigure`	Starts MEDCONFIG and allows you to reconfigure the project. This option uses existing configuration settings as the defaults and leaves you in ADMIN mode.
`-force`	Forces all configurations to recreate their associated binary files, whether or not the binary files are up to date. This option is used with `reconfigure` after you have added new products to an existing product list.
`-help`	Displays the available MEDCONFIG command line options.

© CAD Schroer GmbH

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 MEDUSA4 DB, you do not see the `mdba_application_control` command.

Table 2 MEDCONFIG Commands

Command	Description	ADMIN	PROJ
`bold`	Builds the merged boldness.map	Y	
`code`	Builds the code binary	Y	
`ddl`	Builds the ddl binary	Y	
`defaults`	Builds the user interface defaults binary	Y	
`dfont`	Builds a decortext character font	Y	
`fits`	Builds the fits binary	Y	
`font`	Builds alternative character font	Y	
`mdba_application_control`	Accesses the MEDUSA4 DB Application Control Menu	Y	
`mdbc_rebuild_control`	Accesses the MEDUSA4 DB Customization Control Menu	Y	
`message`	Builds the message binary files	Y	
`patts`	Builds the patts binary	Y	
`plot_config`	Configure this project's use of plotting	Y	
`plotting_dev_kit`	Define new custom plotting products	-	-
`prims`	Builds the prims binary	Y	
`psyms`	Builds the psyms binary	Y	
`raster_path`	Resets or checks Raster Backdrop search path	Y	
`styles`	Create XML Style file	Y	
`ttable`	Builds the translation table binary	Y	Y
`uistrings`	Builds the user interface string binaries	Y	

Table 2 MEDCONFIG Commands

Command	Description	ADMIN	PROJ
`wsconfig`	Configures a workstation	Y	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:\medusa4\med2
mxe: product c:\medusa4\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.

© CAD Schroer GmbH

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.

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 MEDUSA4 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 described 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 *<medusa4>\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 MEDUSA4 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-graphic) and a `WS_DEFAULT` workstation (graphic) is automatically set up in your project 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.

© CAD Schroer GmbH

Reconfiguring 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 *mxedef.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 MEDUSA4 product directory. This project, therefore, shows all the MEDCONFIG commands available, but the command set you see at your site depends on which MEDUSA4 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; the products in 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 within the product list 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 `med3dshrink`.

Before You Start Configuration

Before you start to configure a project, you need to set up certain directories and know 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.

Finding Out the Product Names

When you run MEDCONFIG to configure a project, you have to enter the exact pathnames of the products to configure. You can find the product- and pathnames in the *product_list.dat* which is created during the MEDUSA4 installation inside the *master_project* sub-directory.

Assumed that all products are loaded into one MEDUSA4 product directory, you also can find out the product names and path names by viewing the contents of this directory.

The example below assumes the products have all been installed in the *c:\medusa4* directory. This list varies depending on the MEDUSA4 products you have bought. Lines beginning with dashes (`--`) are comment lines indicating products which are not used in the project.

```
c:\medusa4\medsys
c:\medusa4\med2d
c:\medusa4\meddoc
c:\medusa4\cadconvert
-- c:\medusa4\cadconvertpro
c:\medusa4\medref
c:\medusa4\mdbaccess
-- c:\medusa4\mdbhetnet
c:\medusa4\medraster
-- c:\medusa4\medinfo
-- c:\medusa4\medinfo_web
-- c:\medusa4\medinfo_demo
c:\medusa4\medmech
c:\medusa4\medmech_demo
-- c:\medusa4\medgedm_medinfo
-- c:\medusa4\medgedm
c:\medusa4\mdbcustool
-- c:\medusa4\mdbfortran
-- c:\medusa4\meddars
c:\medusa4\medpara
-- c:\medusa4\mededa
c:\medusa4\medparts
c:\medusa4\medtt
-- c:\medusa4\medapi_demo
-- c:\medusa4\med2sap
```


© CAD Schroer GmbH

```
-- c:\medusa4\med2profile
-- c:\medusa4\med2tc
-- c:\medusa4\med2dgnc
-- c:\medusa4\medpid
-- c:\medusa4\medpdcat_csg
-- c:\medusa4\medads
c:\medusa4\med3d
c:\medusa4\med3dui
c:\medusa4\med3dshrink
c:\medusa4\med3dtran
c:\medusa4\medshade
c:\medusa4\meddtm
-- c:\medusa4\medsmd
c:\medusa4\medstl
c:\medusa4\medstepiges
c:\medusa4\med3dpdf
c:\medusa4\medvrml
c:\medusa4\medvda
c:\medusa4\medplot
-- c:\medusa4\medplot_hp
c:\medusa4\medplot_qtplot
c:\medusa4\medplot_qtraster
c:\medusa4\medpro
-- c:\medusa4\meduwgm
-- c:\medusa4\am
-- c:\medusa4\ampm
-- c:\medusa4\autosel
-- c:\medusa4\hoops
-- c:\medusa4\piping
-- c:\medusa4\cvcat_cs
-- c:\medusa4\steel
-- c:\medusa4\sscat_cs
-- c:\medusa4\support
-- c:\medusa4\spcat_cs
-- c:\medusa4\ducting
-- c:\medusa4\ducat_cs
-- c:\medusa4\electrical
-- c:\medusa4\elcat_cs
-- c:\medusa4\mech_handling
-- c:\medusa4\mhcat_cs
-- c:\medusa4\med_factory
-- c:\medusa4\cdcat
-- c:\medusa4\m4iso
-- c:\medusa4\rohr2
-- c:\medusa4\isogen
```

Creating 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:
   `c:\proj`
3. Create a new workspace directory for the project, giving the directory a suitable name:
   `c:\medtemp`
   This step is optional (see “Configuring the Workspace Directory” on page 29).

The MEDUSA4 product directories are static, i.e. after installation they are not subsequently changed.

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:\medusa4\medsys\login
```


© CAD Schroer GmbH

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.
- `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 *notepad*.

If the default text editor is not suitable, you may customize your startup files to select another editor.

Step 1 - Creating a Product List

The following example shows how you create a project called `c:\proj`. This project has only some products of the MEDUSA4 products palette. For your project you include only those products you need.

Changing the Product List Before Starting Configuration with MEDCONFIG

1. Attach to your project directory, for example:
   cd C:\proj
2. Create a file called *product_list.dat* that follows this format:
   ```
   C:\medusa4\medsys
   C:\medusa4\med2d
   C:\medusa4\med3d
   C:\medusa4\med3dshrink
   C:\medusa4\med3dtran
   C:\medusa4\med3dui
   C:\medusa4\mdbaccess
   C:\medusa4\mdbcustool
   C:\medusa4\mdbfortran
   C:\medusa4\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.

Changing the Product List with MEDCONFIG

Starting up MEDCONFIG

1. Enter the following command line to add the MEDUSA4 directories and products to the search path that you are using, so that you can access MEDCONFIG:
   `c:\medusa4\medsys\login`

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

© CAD Schroer GmbH

```
medconfig -admin
          MEDUSA4 - Configuration Utility
          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Enter pathname of project <quit>: C:\proj
Project c:\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 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>` **`C:\medusa4\medsys`**
     `pathname of product>` **`C:\medusa4\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 displayed:
     `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 C:\medusa4\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:

C:\medusa4\medsys
C:\medusa4\med2d
C:\medusa4\med3d
C:\medusa4\med3dshrink
C:\medusa4\med3dtran
C:\medusa4\med3dui
C:\medusa4\medshade
C:\medusa4\mdbaccess
C:\medusa4\mdbcustool
C:\medusa4\mdbfortran
C:\medusa4\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 the product list, or Abort ? (I/E/A)
```

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

© CAD Schroer GmbH

Step 2 - Configuring the MEDUSA4 Products

After defining the products for your project, MEDCONFIG configures the MEDUSA4 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 MEDUSA4 system product, `medsys`:

```
Configuring product C:\medusa4\medsys
```

Configuring the Workspace Directory

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

```
Enter name of temporary workspace directory <C:\temp>:
```

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' ...
```

Configuring 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.

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
```

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 ...
```


© CAD Schroer GmbH

```
Reading C:\MEDUSA4_V6_0_x86\med2d\med\src\ddl.dat ...
DDL compiled into c:\proj\med\bin\ddl.bin

Compiling CODE ...

Reading C:\MEDUSA4_V6_0_x86\med2d\m2d\src\code.dat ...
CODE compiled into c:\proj\m2d\bin\code.bin


Generating boldness map ...

Reading C:\MEDUSA4_V6_0_x86\med2d\m2d\src\boldness.map ...
Boldness map generated in /proj/m2d/bin/boldness.map

Compiling PRIMS ...

Reading C:\MEDUSA4_V6_0_x86\med2d\m2d\src\prims.she ...
PRIMS compiled into c:\proj\m2d\bin\prims.bin
Number of prims generated: 93
Storage space used for Table: 2769 words

Compiling PATTS ...

Reading C:\MEDUSA4_V6_0_x86\med2d\m2d\src\patts.she ...
PATTS compiled into c:\proj\m2d\bin\patts.bin
Number of patterns generated: 22
Storage space used for Table: 2391 words

Compiling PSYMS ...

Reading C:\MEDUSA4_V6_0_x86\med2d\m2d\src\psyms.she ...
PSYMS compiled into c:\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 MEDUSA4 sheets where the fonts are defined:

```
Compiling decortext fonts ...

Loading C:\MEDUSA4_V6_0_x86\med2d\m2d\dfont\arial_default.bin ...
Loading C:\MEDUSA4_V6_0_x86\med2d\m2d\dfont\arial_fill_default.bin ...
Reading C:\MEDUSA4_V6_0_x86\med2d\m2d\dfont\express.she ...
Reading C:\MEDUSA4_V6_0_x86\med2d\m2d\dfont\gothic.she ...
Reading C:\MEDUSA4_V6_0_x86\med2d\m2d\dfont\hand.she ...
Reading C:\MEDUSA4_V6_0_x86\med2d\m2d\dfont\leroy.she ...
```

```
Reading C:\MEDUSA4_V6_0_x86\med2d\m2d\dfont\pica.she ...
Reading C:\MEDUSA4_V6_0_x86\med2d\m2d\dfont\pica10.she ...
Reading C:\MEDUSA4_V6_0_x86\med2d\m2d\dfont\sean.she ...
Decortext fonts compiled into c:\proj\m2d\bin\dfont.bin

Compiling font ...

Reading C:\MEDUSA4_V6_0_x86\medsys\med\src\font.dat ...
Font compiled into c:\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`.

Configuring the 3D Design Products

When MEDCONFIG configures the MEDUSA4 3D related products it will report items such as:

```
Configuring product C:\MEDUSA4_V6_0_x86\med3d
```

Products like `med3dshrink`, for example, 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 displayed:

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

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

```
Superprint driver for winplot exists
```


© CAD Schroer GmbH

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 shown 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.

© CAD Schroer GmbH

ADMINISTRATION

This chapter describes the administration of styles and standards:

- Administrator Mode 36
- Creating and Editing MEDUSA4 Styles 38
- Setting up Standard Colors 46
- Setting up Standard Boldness (Thickness) Values 49
- Setting up Boldness for Plotting Dimensions 50
- Setting up Standard Sheet Filename 51
- Setting up the Page Label 54
- Setting up the Sheet Header Editor 55
- Setting up the Symbol Manager 66
- Layer Set Management 71
- Customizing Layers 76
- Customizing Layer Sets 77
- Dashboard - Element Selection Limits 80
- Creating Item Number Balloons 82
- Generating Styles from DDL Elements 89
- Defining the Used Codepage 92
- Unicode Text 93

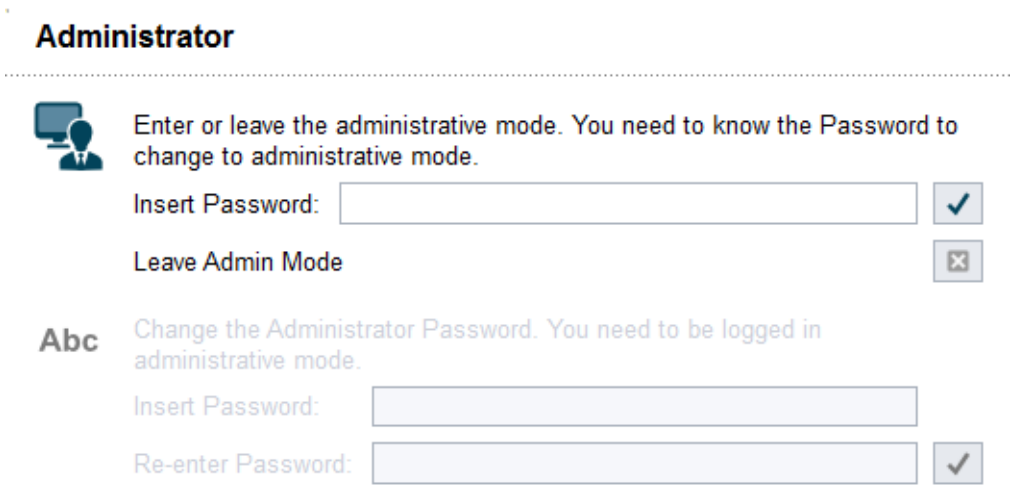
Administrator Mode

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.

Thus you enter the Administrator mode:

1. Select File > Options > Administrator on the ribbon.

Figure 4 Administrator Login



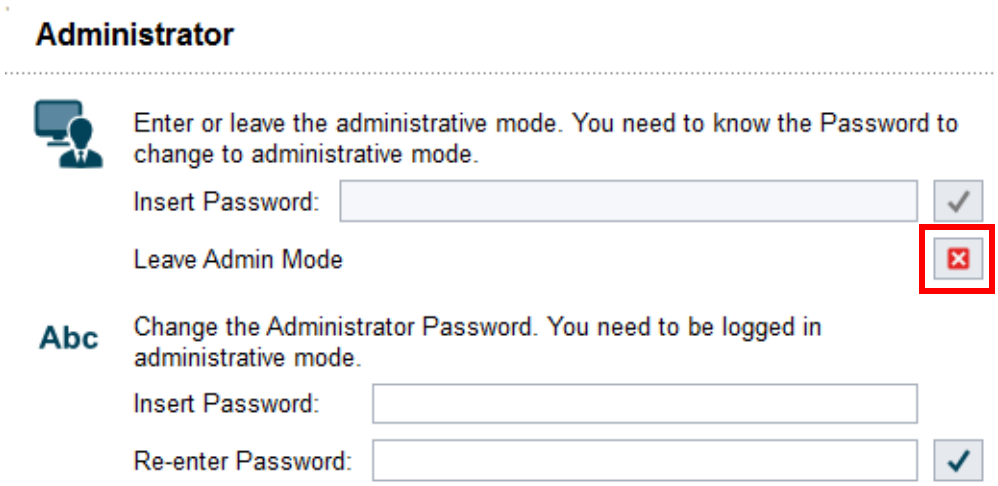
2. Enter the password and click either the `Enter` key or the check mark next to the password entry box.

Please note: MEDUSA4 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.

© CAD Schroer GmbH

If the login was successfully, the button to leave the Admin Mode is displayed red and the input fields for changing the password are activated.

Figure 5 After successful Log in as Administrator



Thus you change the Administrator password:

1. Enter a new password in the relevant input field and re-type the password in the field below.
2. Confirm the change by pressing the `Enter` key or click on the check mark.
   If the two passwords do not match, an error message will be displayed and the password will not be changed.

Please note: Lower and upper case are considered!

You leave the Administrator mode by clicking on the red cross button ☒.

Creating and Editing MEDUSA4 Styles

What are MEDUSA4 Styles?

Most of the tools for creating elements (for example, lines, texts, prims or dimensions) have properties that define how the element will be presented on the sheet. These properties are hold in a kind of container, the so-called **Style**. 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, e.g. solid thin, dashed thin and so on.

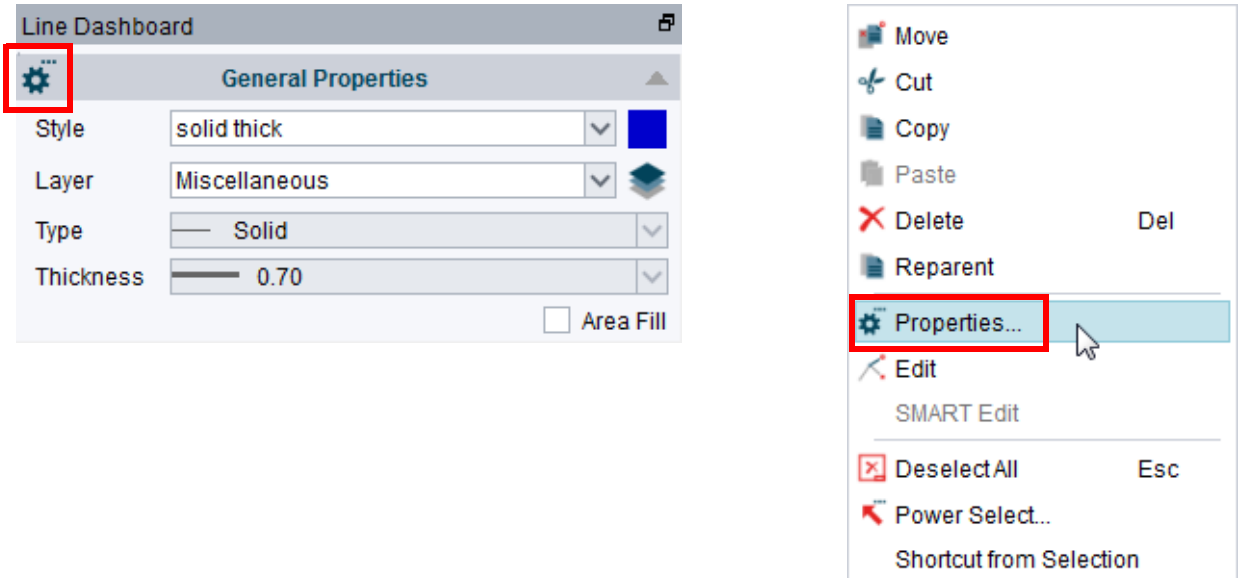
Each style has a unique name and a particular grouping of properties, which reflect how the geometry will be created.

Creating a Style

Please note: **Administrator mode** must be enabled before you can create or modify styles within MEDUSA4. Style changes affect all MEDUSA4 users.

The style creation function can be accessed via the properties dialogs associated with each type or class of geometry. Properties dialogs can be opened either in the Dashboard or by the Properties option of the RMB popup menu, when you have selected an element in your drawing.

Figure 6 **Example: Opening the Line Properties Dialog**



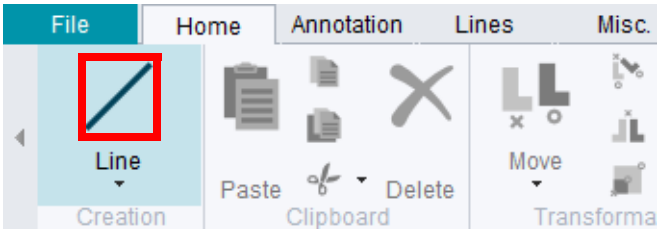
© CAD Schroer GmbH

The following example shows how you create a new line style called `My Style`, which has the following attribute properties:

Type:	Dotted Line
Thickness:	1.0
Color:	Blue

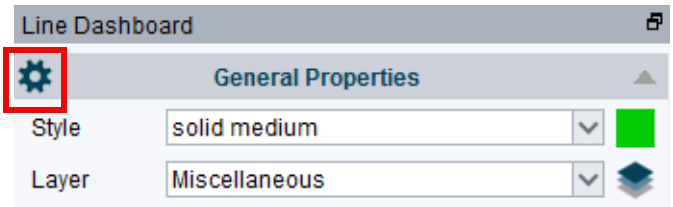
1. First set the geometry type to line. You have the following possibilities:
   a. Choose the Creates lines of the specified type and properties tool which is located on the Ribbon in the Creation tool group of the Home tab
   or
   b. select a line inside the drawing area.

Figure 7 Selecting Line Creation Tool



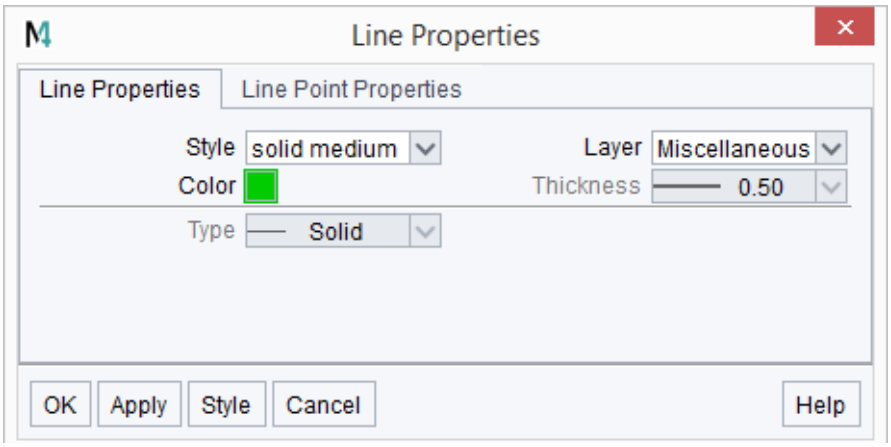
The Line Dashboard is displayed.

Figure 8 Line Dashboard with Button to open the Line Properties Dialog



2. Click on the button to open the Line Properties dialog.

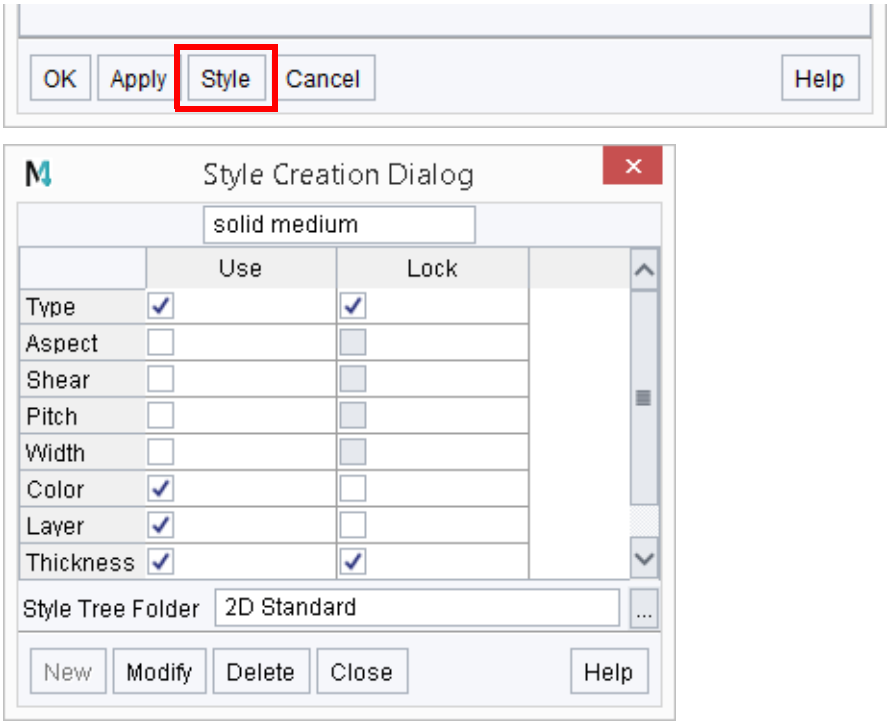
Figure 9 Line Properties Dialog



The Properties dialog is context sensitive. If text, dimensions or any other element type was selected when you click on the button which opens the properties dialog, then the relevant properties dialog is presented.
The properties displayed in Figure 9 show the current settings for the `solid medium` style. The Type and Thickness settings are disabled. This is because the `solid medium` style has defined the Type and Thickness 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. Click on the Style button of the Line Properties dialog to display the Style Creation Dialog.

Figure 10 Open the Style Creation Dialog



Style Creation Dialog

The Style Creation dialog lists all the attributes associated with a particular geometry type. For each attribute the Use and Lock options are available which can be activated as required.

Use
: The Use column indicates which attributes are stored under the style name.

Lock
: If an attribute is locked, the relevant fields are disabled in both the properties dialog and the Dashboard, i.e. they cannot be changed.
The Lock field is only enabled if you have set the corresponding Use choice box.

Style Tree Folder
: defines the location where a new style is placed in the style tree.

© CAD Schroer GmbH

When you choose the My Favourites folder, the new style is directly available in the Style pulldown menu of the Dashboard.

Changing the Attribute Values

To create the new style named `My Style` with the properties shown in the table on page 39, you have to define the Color, Thickness and Type attributes.

1. Make sure that the check boxes for all attributes which define the new style are activated in the Use column.
2. Remove the check marks from the Type and Thickness box of the Lock column to unlock.
3. Change the attributes properties in the Line Properties dialog as specified in the table:
   - Color:
   - Thickness: `1.0`
   - Type: `Dotted Line`
4. Before you save the style, lock the Color attribute.
5. Enter the style name `My Style` into the text field 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.

6. Choose the `Favorites` entry in the Style Tree Folder pulldown menu to place the new style under `Favorites` in the style tree.
7. Save the new style as described in the following section.

Saving a Style

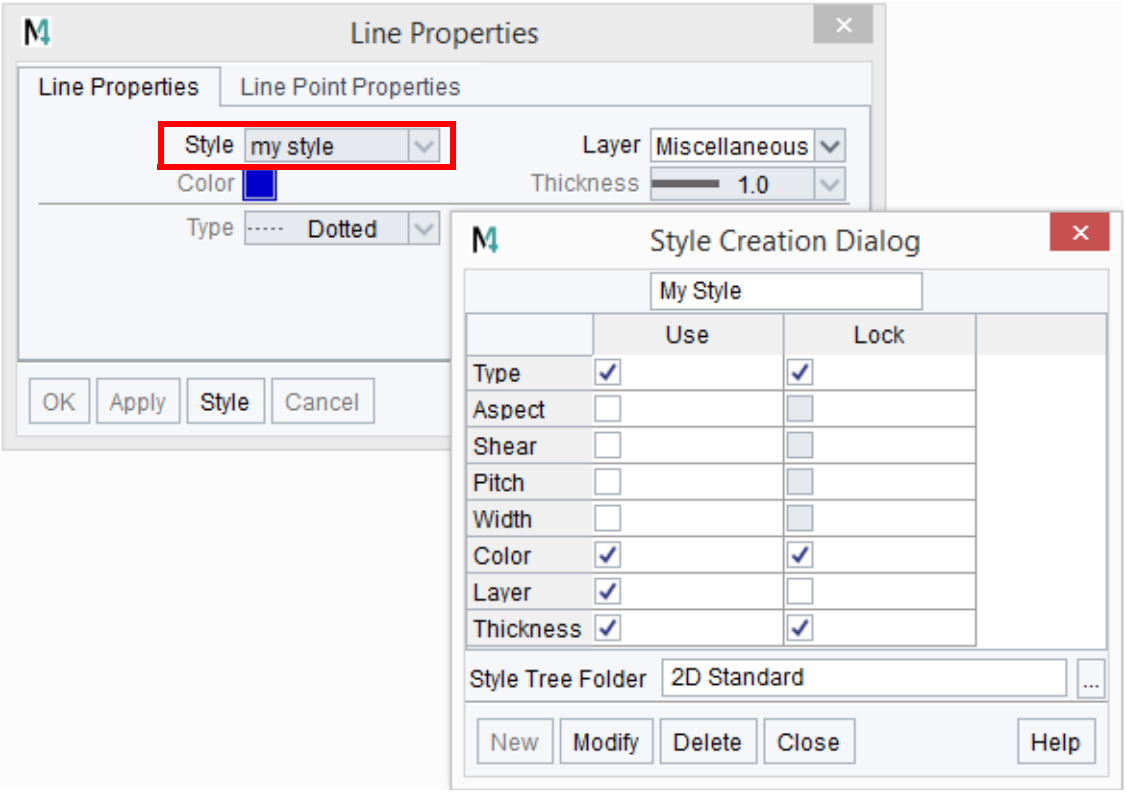
You can save a style temporarily, i.e. for the duration of your current session only, or permanently, then the style is also available for subsequent sessions.

To save a style permanently go on with the following steps after you have saved the style temporarily.

Saving a Style Temporarily

Click on the New button in the Style Creation Dialog.
The new style is saved temporarily.

Figure 11 Dialogs during Creation of a new Line Style after temporary Saving



The `My Style` style is displayed in the Style field of the Line Properties dialog. Type, Color and Thickness cannot be changed according to the settings in the Lock column in the Style Creation Dialog. When you open the style tree, the new style is displayed in the `Favorites`.

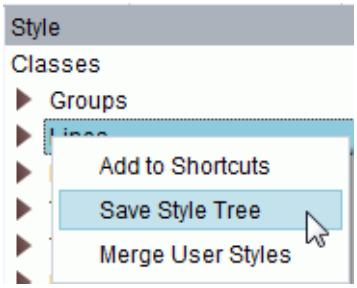
Saving a Style Permanently

To save a style permanently go on with the following steps after you have saved the style temporarily.

1. Open the style tree by clicking the button Opens Catalog, Tree and Browser Area at the bottom right of the status area.
2. Choose the Style tab.
3. Move the cursor over the displayed style tree and click the RMB.
   A popup menu opens.
4. Choose the Save Style Tree entry.

© CAD Schroer GmbH

Figure 12 Style Tree Popup



The whole style tree is stored. When you start MEDUSA4 the next time, all the current styles, including your new style `My Style`, 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*.

Canceling Style Creation

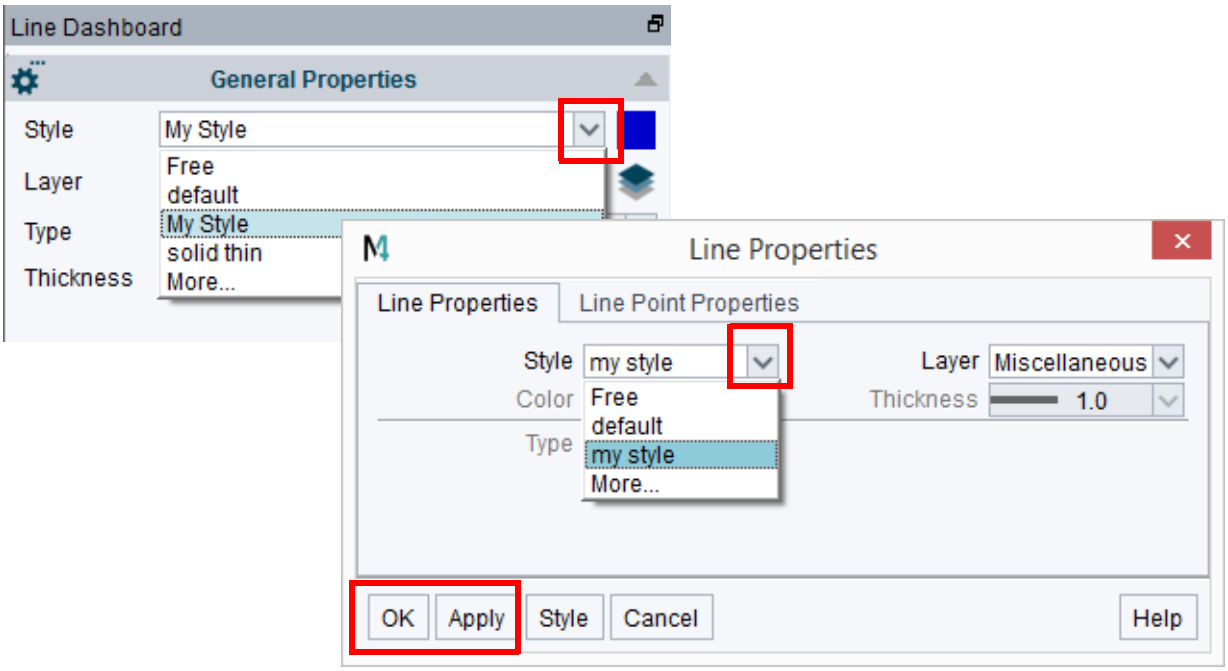
If you decide not to save the style, you can cancel the operation by using the Close button on the Style Creation Dialog. The dialog disappears and the Properties dialog is reset to its original settings.

Using Your Style

There are two ways to assign a style to an selected element:

- Select the desired style on the Style pulldown menu of the Line Properties dialog. Click either Apply or OK to assign the style to the element.
- Select `My Style` from the Style Pulldown menu on the Dashboard.
- Click the RMB on the `My Style` entry in the style tree and choose the Use Style option.

Figure 13 Assigning a Style via Properties Dialog and via Dashboard



Use the generic line creation tool on the Ribbon > Home tab > Creation tool group to test out your new style. Alternatively, you can associate the `My Style` style directly with a tool in the Shortcut area of the Dashboard (see the “Drafting User Guide, Dashboard, Element Specific Dashboard, Tools“ for details).

Modifying Styles

You can modify existing styles. The next example shows how to modify `My Style` so that lines of this style will be red.

1. Select `My Style` from the Style pulldown menu located on the Dashboard.
2. Open the Line Properties dialog .
3. Click on the Style button of 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 the Color attribute in the Style Creation dialog.
7. Select the Modify button, which will update the style.
8. Close the Style Creation dialog.
9. Click on Apply on the Line Properties dialog or on OK to close the dialog.
   The color of the line style `My Style` is displayed red also in the Dashboard.

© CAD Schroer GmbH

Deleting Styles

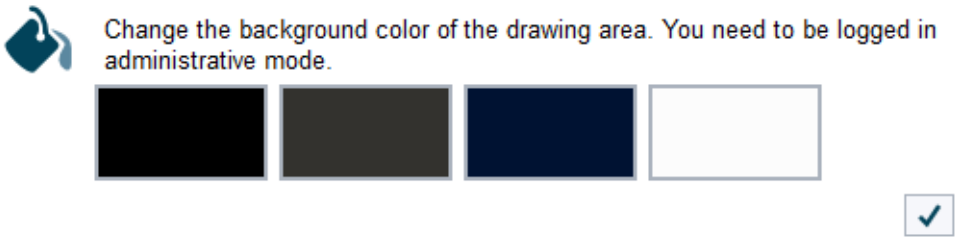
1. Select `My Style` on the Dashboard.
2. Open the Line Properties dialog .
3. Click on the Style button at the bottom of the dialog.
   The Style Creation dialog appears.
4. Select Delete.
   `My Style` is deleted from the system and a backup style called `my_style` is created in order to keep the settings in case that you wish to recreate the deleted style. The backup style is only available inside the dashboard and if you close MEDUSA4 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 My Favourites folder of the tree only. Inside the dashboard it is still available.

Setting up Standard Colors

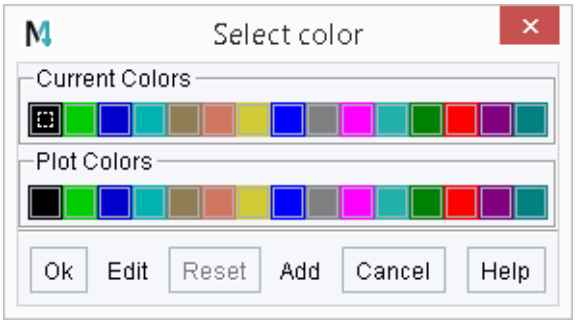
You can change the default white (`light`) background color of the Graphics area via File > Options > Administrator.

Figure 14 Range of available Background Colors

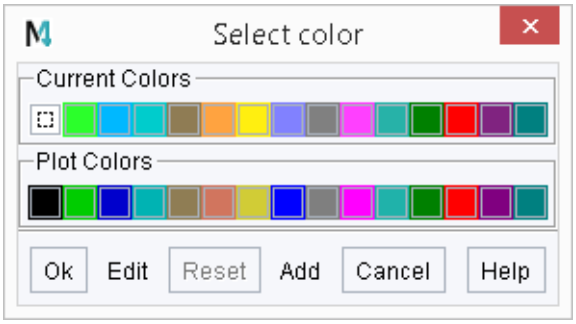


Changing the background color requires the alteration of some displayed colors to be visible on the new background. For example a black line is invisible on a black background, but is still required to be black when printed on paper. To achieve this MEDUSA4 contains an algorithm to ensure all colors are visible on the screen. The following figure shows an example.

Figure 15 The Select Color Dialog



default background color:
light (default setting)



background color:
black

The upper color scale - Current Colors - shows the colors displayed on the screen. It changes when changing the background color from white (`light`) to `black`. Only colors that would not be visible are altered.

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

© CAD Schroer GmbH

Any new or imported colors are automatically handled by the MEDUSA4 algorithm, but if you do not like the color matches created by MEDUSA4 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 <*medusa4*>*\med2d\m2d\src\colours.map* shown in the following figure.

Figure 16 The Color Table

RGB values define black background color

```
-- default
--
                          light        dark blue      dark grey        black
BG      background / 252 252 252 /   0  18  50 /  51  50  46  /   0   0   0 /
--
HL      red        / 255   0   0 / 255   0   0 / 255   0   0 / 255   0   0 /
COLOUR  black      /   0   0   0 / 255 255 255 / 255 255 255 / 255 255 255 /
COLOUR  green      /   0 204   0 /   0 204   0 /   0 204   0 /  43 255  43 /
COLOUR  blue       /   0   0 204 / 133 133 204 /   0   0 204 /   0 184 255 /
COLOUR  skyblue    /   0 179 179 /  43 179 176 /   0 187 183 /   0 204 204 /
COLOUR  brown      / 143 125  84 / 143 124  84 / 255 221 150 / 143 124  84 /
COLOUR  ochre      / 209 117  94 / 209 115  94 / 209  63  30 / 255 163  64 /
COLOUR  yellow     / 209 204  54 / 209 203  54 / 209 203  42 / 255 239  16 /
--
-- additional colours
--
COLOUR  royal blue /   0   0 255 / 129 129 255 /   0   0 255 / 129 129 255 /
COLOUR  grey       / 127 127 127 / 128 128 128 /  96  96  96 / 128 128 128 /
COLOUR  magenta    / 255   0 255 / 255   2 255 / 255   0 255 / 255  64 255 /
COLOUR  bluegreen  /  32 178 170 /  39 178 168 /  32 210 198 /  39 178 168 /
COLOUR  olive      /   0 128   0 /   0 128   0 /   0 128   0 /   0 128   0 /
COLOUR  red        / 255   0   0 / 255   0   0 / 255   0   0 / 255   0   0 /
COLOUR  purple     / 128   0 128 / 128  35 128 / 185   0 185 / 128  35 128 /
```

RGB values define white displayed line 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`).
- In the second column you see a text field showing the name of the color for the `light` background. This is not true for the first line because it defines the different background colors.
- The following columns contain RGB triplets for each color. These color values are used for the relevant background. For example, the `black` color (RGB=0 0 0) becomes white (RGB=255 255 255) for the `black` background.
  You can define up to eight alternative RGB triplets for each color. Additional background colors are displayed beside the already available colors (see Figure 14, “Range of available Background Colors,” on page 46).

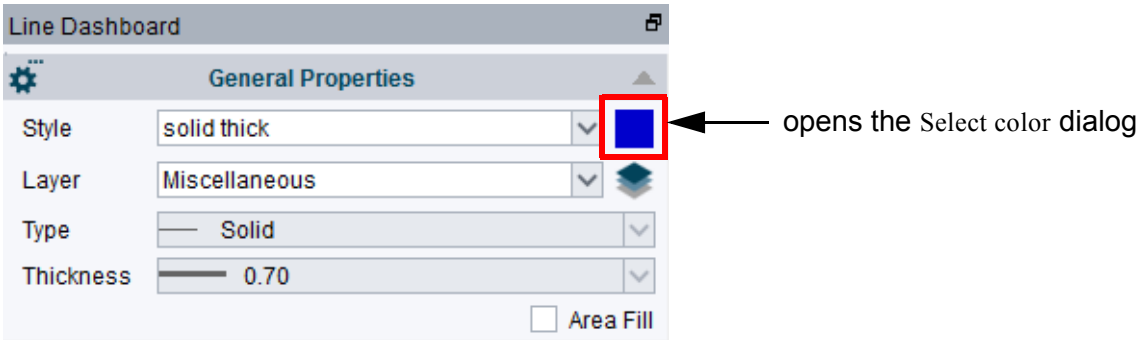
Example

Select the black background (see Figure 14, “Range of available Background Colors,” on page 46). This background is defined by the RGB values `<0 0 0>` which you can find in the first line and right column of RGB values in the Figure 16, “The Color Table”.

If you have set the background color to black, the Select color dialog (see Figure 15, “The Select Color Dialog,” on page 46) displays the colors below the RGB value `<0 0 0>` and with `COLOUR` at the beginning of the line.

Please note: You can display the Select color dialog by selecting an element on the sheet and clicking on the color button on the Dashboard.

Figure 17 Button on the Dashboard to open the Select color Dialog



The highlight color (`HL`) 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 displayed as a light green defined by RGB value `<43 255 43>`.
A blue line is displayed as a light blue, RGB `<0 184 255>`, to be visible on the black background and so on.

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.

© CAD Schroer GmbH

Setting up Standard Boldness (Thickness) Values

The boldness (thickness) used for drawing lines and text are defined in the following file in your MEDUSA4 product: *<medusa4>\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:

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

The environment variables can be set e.g. in the *login.bat* of your MEDUSA4 user project.

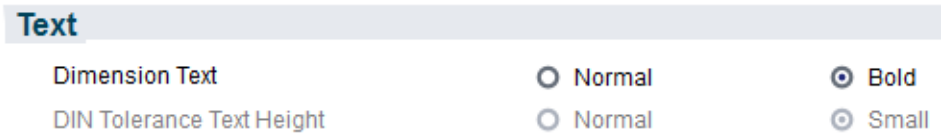
For the default values for thickness (<value>) see the following table:.

<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 when the Bold option on the Default Settings dialog, Dimensions > Text is activated.
The dialog is displayed via File tab > Default Settings.

Figure 18 Dimension Text Settings



© CAD Schroer GmbH

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 *<medusa4>\med2d\m2d\src\defaults.dat* to display the text entries for the sheet saving.
   The relevant section looks like following, for example:
   ```
   --sheet_save  string  "<drawing_number>_<sheet_issue_number>_<sheet_number>"
   --                  == "@tTSH_@tTIS_@tTSN"
   --sheet_save  string  "<drawing_number>_<sheet_number>"
   --                  == "@tTSH"
   sheet_save  string  "@TTSH_@TTIS_@TTSN"
   ```
   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_@TTIS_@TTSN`) which uses the old MEDUSA4 Text Type method for 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>`"
equates 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:
   <MEDUSA4 installation path>\med2d\m2d\src\defaults.dat

   but these can merge with, for example:
   <MEDUSA4 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: Take care with special characters, for example: `.` `/` `\` and so on.
Avoid these characters.

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

© CAD Schroer GmbH

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 MEDUSA4 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*.

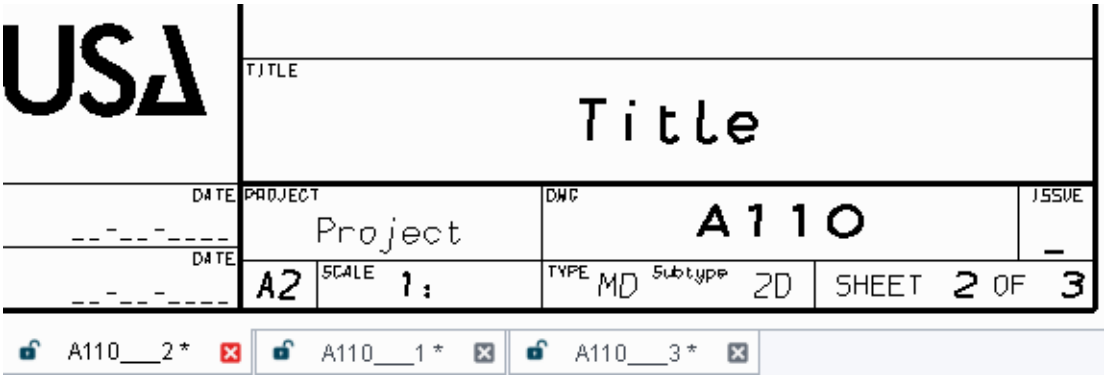
```
--------------------------------------------------
-- Page Label Setting --
-- true / false
set_page_label boolean false

-- 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 19 Example Sheet Header and Tab Label



© CAD Schroer GmbH

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>:\<medusa4>\med2d\m2d\src*. *<LW>* is the letter of the hard disk and *<medusa4>* is the installation directory of MEDUSA4.

All **adjustments**, which you make, **should not be made in the original file** *sfeld.cfg* in *<medusa4>\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 MEDUSA4 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 relevant 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 MEDUSA4 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`

© CAD Schroer GmbH

- **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 shown.

Basic Settings: `sfeld_cfg`

The chapter `sfeld_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 MEDUSA4 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 MEDUSA4.
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 MEDUSA4, 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
```


© CAD Schroer GmbH

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	**:**	**area_dlglabel**
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. `a$b`).
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 MEDUSA4 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	**:**	**attnam<xxx>**
Meaning	:	This is used in conjunction with the `attval` entry described below. Together they provide an additional filter for selecting the relevant text element from the MEDUSA4 sheet. The text items can have a User Attribute attached to themselves, and each one can be given a specific value. For example if the sheet header was to have several lines of text that could be used to describe the sheet content, then these would all have the same style and each would have a user attribute with different values set, for example LINE_1, LINE_2 etc. `attnam` defines the name of the user attribute, for example ZBEDEU (see the *MEDInfo Configuration Guide's* section about *MEDUSA4 Link*).
Type of Data	:	case independent character string, limited to 6 characters
Comment	:	optional Keyword

Name	**:**	**attval<xxx>**
Meaning	:	This is used in conjunction with the `attnam` entry described above. `attval` defines the value of the User Attribute, for example Line_1 or BENENN.

© CAD Schroer GmbH

Type of Data : case independent character string

Comment : optional Keyword

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 MEDUSA4.

© CAD Schroer GmbH

A Sample Configuration File

```
--------------------------------
--                            --
--  MEDUSA4 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
```


© CAD Schroer GmbH

```
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 the Symbol Manager

The environment variable `MEDUSA_SYMBOL_PATH` in the file
<medusa4>\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 contain the following attributes:

  `name` – describing name which is displayed in the appropriate tree node of the symbol manager. `name_translation` contains the subsystem and the message keyword in the *mui_2d_messages.nem* file. Example:
  `name_translation="symbol_library$symlib1_screws`

  `path` – is the directory to the symbols. A dot defines the directory with the XML file. Note that paths starting with `/` or `\` or a volume letter (e.g. `C:`) will be processed as an absolute path, not a path relative to the location of the XML-file. Example:

  ```
  <symbols name="Stifte" path="C:\SYMBOL_LIB\MASCHBAUBIB" mask="*stift*.sym"
           icon="csg_stifte.gif">
  ```

  `mask` – filter for the symbol file names which shall appear in the preview (optional).

© CAD Schroer GmbH

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

`str` – is a text string which determines the parameter explained above. *`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:

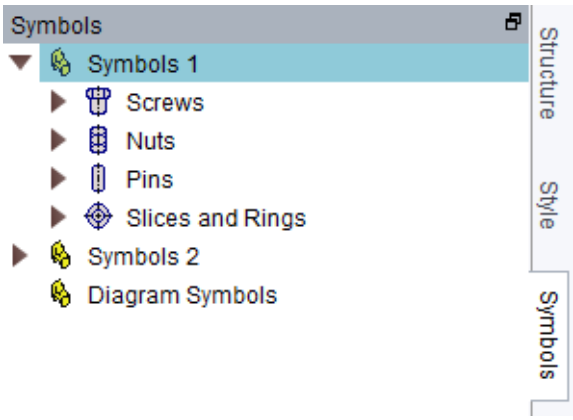
```
ui!sym_load_named_dialog!pathname:-"c:\prod\symbols\screws"
```

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* (located in *med2d\m2d\symbols*), whose syntax is shown on the following pages. The Diagram Symbols entry is defined in the file *\example_symtree_diagram.xml*.

Figure 20 Example Symbol Manager



Contents of the *example_symtree.xml* file:

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE medusa4>
<!-- Level;Name;Pfad;Dateimaske;Iconname -->
<symbol_manager>
  <!-- Symbolbibliothek 1 -->
  <symbol_library>
        <symbols name="Screws" name_translation="symbol_library$symlib1_screws"
                path="./MASCHBAUBIB" mask="*Schraube*.sym" icon="csg_schraube.gif">
           <symbols name="Hexagon Bolts"
                    name_translation="symbol_library$symlib1_hexagon_bolts"
                    path="./MASCHBAUBIB" mask="SK_Schraube*.sym">
           </symbols>
           <symbols name="Cylinder Screw"
                    name_translation="symbol_library$symlib1_cylinder_screws"
                    path="./MASCHBAUBIB" mask="Z_Schraube*.sym">
           </symbols>
        </symbols>
        <symbols name="Nuts" name_translation="symbol_library$symlib1_nuts"
                path="./MASCHBAUBIB" mask="*utter*.sym" icon="csg_mutter.gif">
           <symbols name="Hexagon Nuts"
                    name_translation="symbol_library$symlib1_hexagon_nuts"
                    path="./MASCHBAUBIB" mask="SK_Mutter*.sym">
           </symbols>
           <symbols name="Cap Nuts"
                    name_translation="symbol_library$symlib1_cap_nuts"
                    path="./MASCHBAUBIB" mask="Hutmutter*.sym">
           </symbols>
        </symbols>
        <symbols name="Pins" name_translation="symbol_library$symlib1_pins"
                 path="./MASCHBAUBIB" mask="*stift*.sym" icon="csg_stifte.gif">
           <symbols name="Threaded Pins"
                    name_translation="symbol_library$symlib1_threaded_pins"
                    path="./MASCHBAUBIB" mask="G*stift*.sym">
```


© CAD Schroer GmbH

```
            </symbols>
            <symbols name="Dowel Pins"
                     name_translation="symbol_library$symlib1_dowel_pins"
                     path="./MASCHBAUBIB" mask="Sp*stift*.sym">
            </symbols>
         </symbols>
         <symbols name="Slices and Rings"
                  name_translation="symbol_library$symlib1_slices_rings"
               path="./MASCHBAUBIB" mask="Scheibe*.sym Fe*.sym" icon="csg_scheibe.gif">
            <symbols name="Slices"
                     name_translation="symbol_library$symlib1_slices"
                     path="./MASCHBAUBIB" mask="Scheibe*.sym">
            </symbols>
            <symbols name="Spring Washer"
                     name_translation="symbol_library$symlib1_spring_washer"
                     path="./MASCHBAUBIB" mask="Feder*.sym">
            </symbols>
         </symbols>
      </symbols>
   </symbol_library>

   <!-- Symbolbibliothek 2 -->
   <symbol_library>
      <symbols name="Symbols 2" name_translation="symbol_library$symlib2_title"
               path="."  mask="*.sym" icon="symbols.gif">
         <symbols name="Sockets" name_translation="symbol_library$symlib2_sockets"
                  path="./BUCHSEN">
            <symbols name="Sockets" name_translation="symbol_library$symlib2_sockets"
                     path="./BUCHSEN/BUCHSEN"/>
            <symbols name="Slotted Sockets"
                     name_translation="symbol_library$symlib2_slotted_sockets"
                     path="./BUCHSEN/GESCHLITZTE_BUNDBUCHSEN" mask="*.sym">
            </symbols>
         </symbols>
         <symbols name="Cyinder" name_translation="symbol_library$symlib2_cylinder"
                  path="./ZYLINDER">
            <symbols name="Hydraulics"
                     name_translation="symbol_library$symlib2_hydraulics"
                     path="./ZYLINDER/hydraulik">
               <symbols name="Sliding Guide"
                        name_translation="symbol_library$symlib2_sliding_guide"
                        path="./ZYLINDER/hydraulik/gleitfuehrung" mask="*.sym">
               </symbols>
               <symbols name="Rail Guide"
                        name_translation="symbol_library$symlib2_rail_guide"
                        path="./ZYLINDER/hydraulik/schienenfuehrung" mask="*.sym">
               </symbols>
```

```
            </symbols>
            <symbols name="Pneumatics"
                     name_translation="symbol_library$symlib2_pneumatics"
                     path="./ZYLINDER/pneumatik" mask="*.sym">
            </symbols>
          </symbols>
        </symbols>
      </symbol_library>

    </symbol_manager>
```

Adding Symbol Directory for several Project Users

Users can add directories to the Symbol Manager by using the Add Directory button. When a directory was added, a *symbols.xml* is created in the *<project>\user\<username>* directory. Only user *<username>* can use this directory. To give all project users access to an added directory, the Administrator has to create a *common* directory in the *<project>\user*directory which contains the *symbols.xml* file.

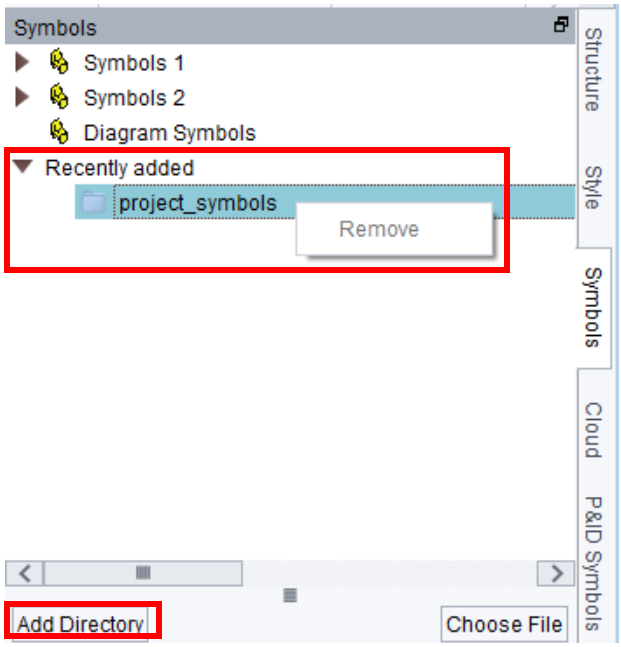
If the *common* directory exists, the symbol browser reads the *symbol.xml* file from the directory *user\common,* before reading the *symbols.xml* from the directory *user\<username>*. Directories specified in *user\common* cannot be removed from the symbol browser.

Procedure:

1. Click on the Add Directory button in the Symbol Manager to add the directory (e.g. *project_symbols)* that contains the symbols, which are to be accessed by all project users.
   The *symbols.xml* file is created in *<project>\user\<username>*.
2. Make the new *symbols.xml* file accessible to all users of the project:
   a. If a *common* directory does not exist, just rename the *<username>* directory into *<common>*.
   b. If a *common* directory already exists, move the *symbols.xml* file from the *<username>* to the *common* directory.
3. Restart MEDUSA4.
   When you click the RMB on the *project_symbols* folder in the Symbols Manager, the Remove option on the popup is disabled so that the directory cannot be removed.

© CAD Schroer GmbH

Figure 21 Example of a Directory added for all Users

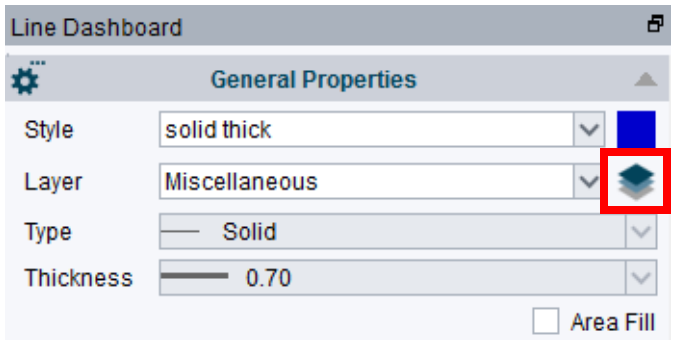


Layer Set Management

Please note: For the layer set management you have to be in Administrator Mode. To enter Administrator Mode choose File > Options > Administrator.
If you are not in Administrator Mode all entries of the tab described in the following (see Figure 23) are disabled.

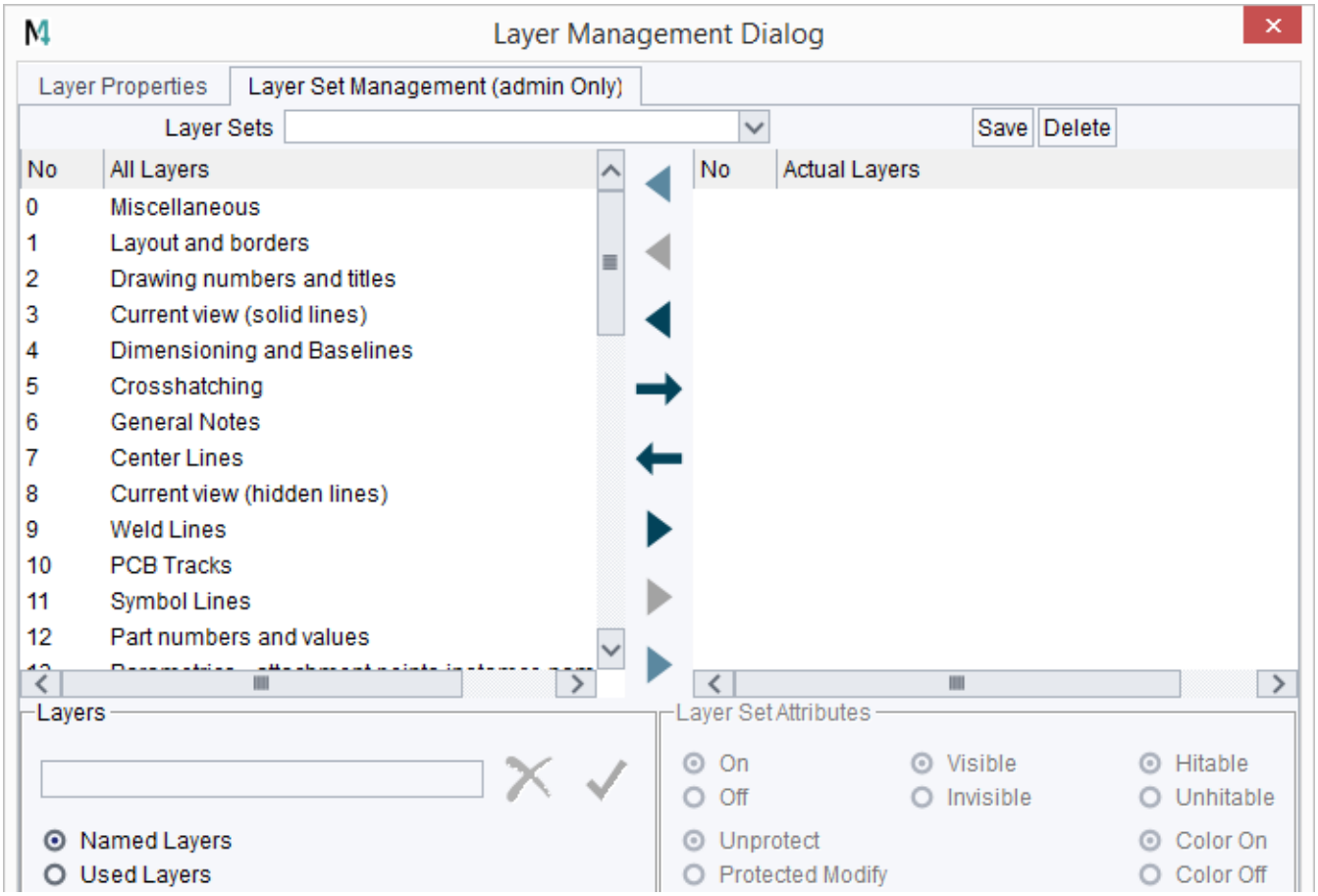
1. Display the Layer Management Dialog.
   Use either te tab File > Options > Layer Management or, if you are just creating or editing an element, use the button in the General Properties area of the Dashboard.

Figure 22 Button to open Layer Management Dialog on the Dashboard



2. Click on Layer Set Management (admin only) for displaying its contents.

Figure 23 Layer Management Dialog: Tab Layer Set Management (admin only)



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

Layer Sets
: 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 described in "Customizing Layer Sets" on page 77.

Save
: saves the current layer set in the file *layerset.xml* located in the user directory of the current MEDUSA4 project, e.g. *master_project\user\username\layerset.xml*.

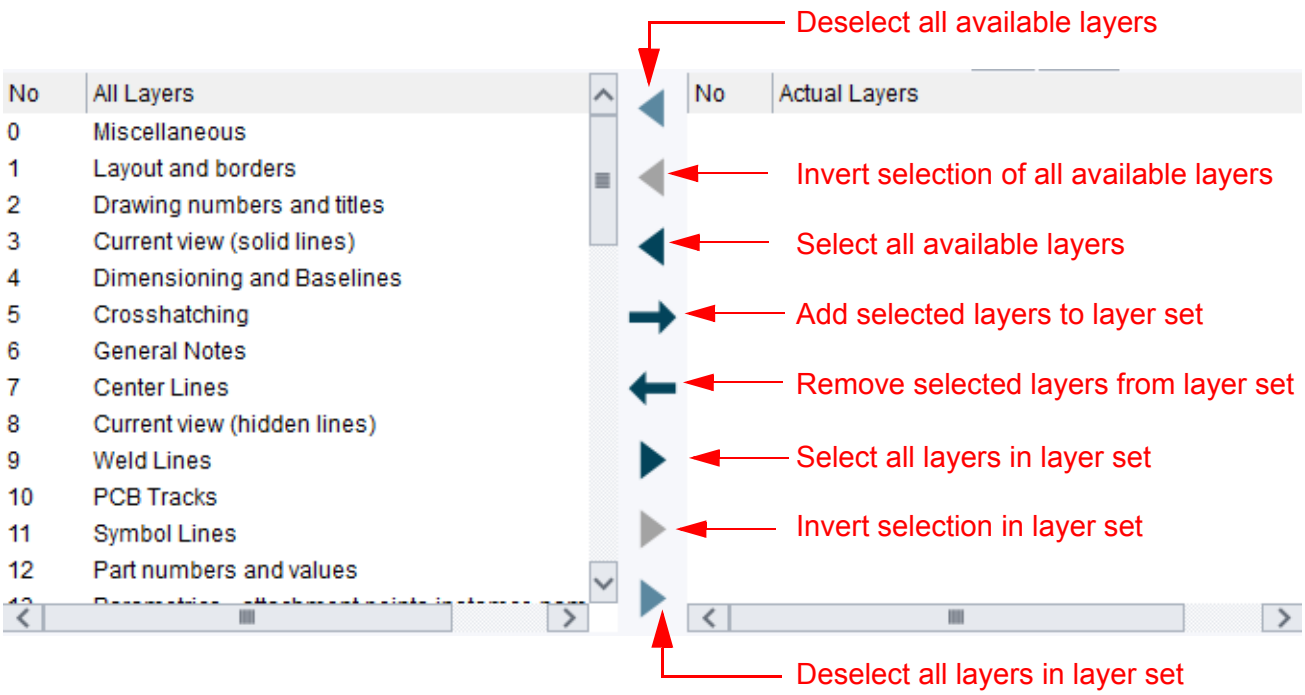
Please note: If you re-create the user project, the layer set file will be deleted. Therefore we recommend to copy this file in the user product directory, e.g. *custom\m2d\src\layerset.xml*, and then reconfigure your MEDUSA4 project. Please read the "Customization Guide, Running MEDCONFIG, Adding a User-Product Directory" for details.

Delete
: deletes the current layer set.

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

© CAD Schroer GmbH

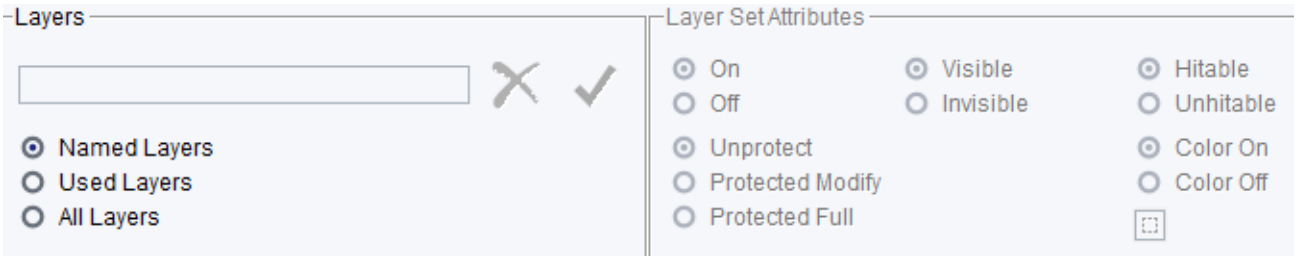
Figure 24 **Layer Manager Dialog: Tab Layer Set Management, Layer Lists**



- The All Layers list on the left hand side lists by default all layers which have a name. The content depends on the show options in the Layers area.
- The Actual Layers list on the right hand side lists all layers which belong to the current layer set.
- Between the lists you find 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 two lists you find the Layers and Layer Set Attributes areas.

Figure 25 **Layer Management Dialog: Tab Layer Set Management, Options**



Layers Area

Text field (below the list of All Layers)
: The field is disabled by default. If you click double on a layer number inside the list of All Layers, the field becomes enabled and the name is displayed in it. Now you can delete or change the layer name. Details are described in “Customizing Layers” on page 76.

Named Layers, Used Layers, All Layers
: Options for displaying either all named layers, all layers used in the current sheet or all layers available in MEDUSA4 (which are the layers numbered from 0 to 1023) in the left list.

Layer Set Attributes Area

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

Please note: All entries in the Layer Set Attributes area 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 you 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, Grid Properties, 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.

© CAD Schroer GmbH

OK, Apply,
: use 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.

Customizing Layers

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

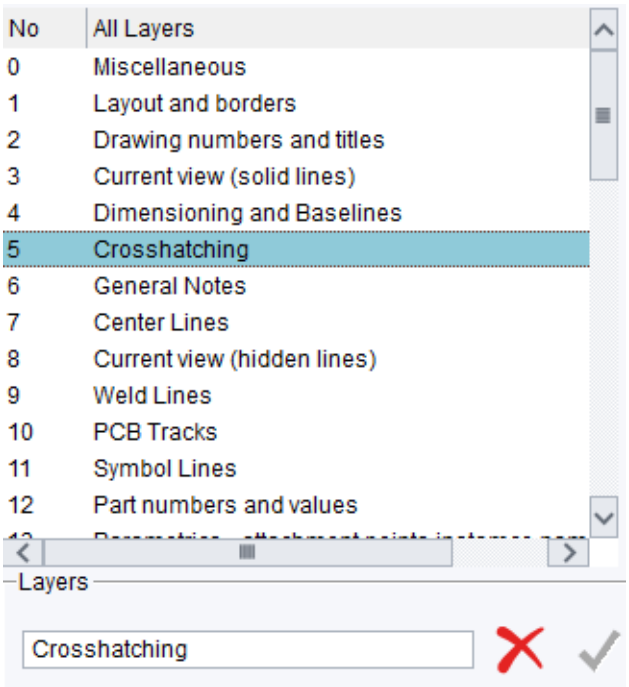
In MEDUSA4 you always have 1024 layers numbered from 0 to 1023, so you cannot add new layers which have a higher number.

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

Renaming a Layer

1. Open the dialog Layer Management Dialog either by using File > Options > Layer Management or the button in the General Properties area of the Dashboard.
2. Double click the LMB either on a layer number or the layer name.
   Below the list the text field becomes enabled and the layer name is displayed.

Figure 26 Example: Rename Layer Name



3. Click the LMB 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.

© CAD Schroer GmbH

5. Click the LMB 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 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.

Deleting a Layer Name

1. Open the dialog Layer Management Dialog either by using File > Options > Layer Management or the button in the General Properties area of the Dashboard.
2. Double click on the LMB 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 the LMB 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.

Customizing Layer Sets

Please note: You can customize layer sets when you are 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 via the File tab > Options > Layer Management or the button in the General Properties area of the Dashboard.

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 Actual Layers list on the right displays the layers which are defined in the current set.
2. Click on the layers in the left list which are to 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 on the Add layer to set button.
   The selected layers are shifted from the left list to the right one.
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 Actual Layers list on the right displays the layers which are defined in the current set.
2. Click on layers which shall be removed from the layer set in the Actual Layers list.
   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 on the Remove layer from set button.
   The selected layers are shifted from the right list to the left one.
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* file. These files have to be placed in the product directories into *m2d\src*. MEDUSA4 looks through all *m2d\src* directories that are in the product list, starting from top, and merges the found *layerset.xml* files. Example for a *layerset.xml* file is shown below:

```
<layersets>
```


© CAD Schroer GmbH

```
  <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>
```

The meaning of the `protection` settings is as follows:

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

Details are described in the “Drafting Guide, Layers, Layer Properties“.

Please note: **Changes** of the layer sets should be **always** done **in the customer product** not in the product directories of the original product. Please read the “Customization Guide, Running MEDCONFIG, Adding a User-Product Directory“ for information about adding user product directories.

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 Sheet Dashboard is displayed. The following table shows the default and maximum values for limits:

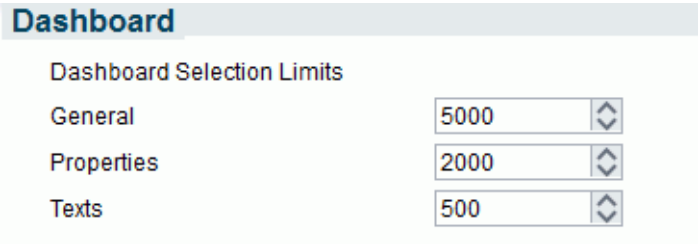
Element	Default Value	Maximum Value
General	5000	1000000
Properties	2000	10000
Texts	500	100000

You can define the limits either on the Defaults Settings dialog (File > Default Settings> General) or in the *defaults.dat* file of the *med2d* product.

Setting Limits in the Defaults Dialog

Figure 27 shows the part of the General page, where you can insert the desired values.

Figure 27 Defaults Dialog: Tab Common, Dashboard Element Selection Limits



Please note: The higher the values are, the longer MEDUSA4 needs for checking.
If you exceed the maximum value, a message dialog is displayed and the value is reset to its maximum.

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

When you quit MEDUSA4, 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 MEDUSA4 sessions.

© CAD Schroer GmbH

Setting Limits in the *defaults.dat* File

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

```
-----------------------------------------------------------------------
-- Dashboard element selection limits for show / not show a dashboard
-----------------------------------------------------------------------
--                            Internal
--                        Default   Maximum
-- countlimit_dashboard     5000   1000000
-- countlimit_props         2000     10000
-- countlimit_text           500    100000
```

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 shown in the table above. 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.

Creating Item Number Balloons

This section shows how you create item number balloons and define parts lists tables which contain the data of item number balloons.

Before you start creating your item number balloon symbol:

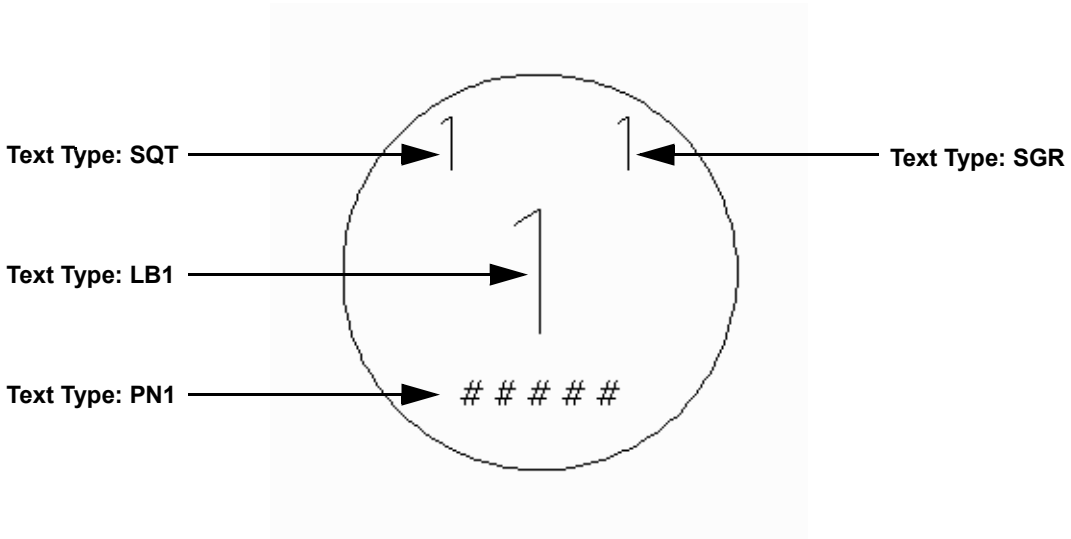
make the following settings:

- Enter the Administrator Modus (File > Options > Administrator).
- Activate the Admin Tab, which is not displayed in the Ribbon by default, because you will need the tools on this tab later. To do this, press the RMB on the File tab label and check Admin in the displayed menu.

... and you should know the following:

- You create a customer specific item number balloon based on an existing symbol, i. e., you load a standard symbol and adjust it to your requirements.
- An item number symbol must contain the following mandatory text types: LB1, PN1, SQT and SGR. The figure below shows these text types using the `balloon_def` symbol file, which you can use as the base symbol. If you select the individual text elements of the symbol, the text dashboard displays the text type in the General Attributes area.

Figure 28 Standard Item Number Balloon Symbol with Indication of the Mandatory Text Types



By using a standard symbol for your symbol, you ensure that all mandatory text types are definitely included.

© CAD Schroer GmbH

Entries in the *defaults.dat* File

Which symbol of an item number balloon is loaded by default when creating a new item number balloon is defined in the *defaults.dat* (*<MEDUSA4_installation>\med2d\m2d\src*).

```
parts_balloon_default  string  "balloon_def"
```

The name for the symbol (**.sym*) and the description file (**.bac*) must be the same. How this *bac* file is organized is described below in Creating User Defined Item Number Balloon.

To define which data of the item number balloon are written to the parts list, you need to define the filename of the *bac* file:

```
parts_ballon_table  string  "balloontable"
```

You can define whether the height property in the properties dialog is displayed. Default setting is `false`.

```
parts_balloon_heights  boolean  false
```

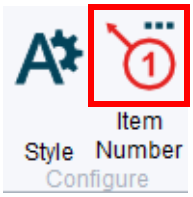
Creating User Defined Item Number Balloon

1. Choose the Create a new Item Number Balloon tool on the Home tab > Text + Prims.
   The Create Item Number Balloon dialog is displayed.
2. Select the symbol file `balloon_def` as the basis for your symbol.
   The symbol is attached to the crosshairs.
3. Place the symbol on the sheet and exit the tool.
4. Now edit the symbol as desired:
   a. either by double click on an element of the symbol or choosing Edit from the popup menu.
   b. 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 29 Example for a User defined Item Number Symbol

5. Choose the tool Creates user defined symbol for Item Number Balloons on the Admin tab > Configure tool group.

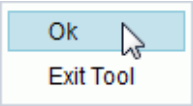
Figure 30 **Admin > Configure > Creates user defined symbol for Item Number Balloons**



Please note: The Admin tab is not displayed by default, it first must be activated, see page 82.

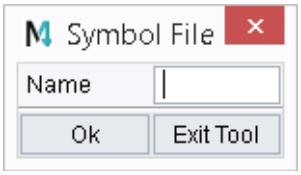
6. Select all the geometry you want to use for the item number balloon.
7. Open the RMB popup menu and click on Ok.

Figure 31 **Popup Menu while creating Item Number Symbol**



A dialog opens.

Figure 32 **Symbol File Dialog while creating Item Number Balloon**



8. Enter a name for the symbol **consisting exactly of three characters.**
9. Confirm with OK.
10. Finally set a probe for the loading point of the Item Number Balloon.
    The symbol file is created promptly and the *bac* file is 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; you can find it below `-- additional texts, user defined--`, highlighted bold.

```
-- default description for parts balloon abc

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


© CAD Schroer GmbH

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

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

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

-- Issue
balloon_descr :- dict_new()
balloon_descr!datatyp :- !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 List" on page 87).

`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 Item Number 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 whether 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 the following rows!

```
balloon_def :- []
balloon_descr :- dict_new()
balloon_def :- balloon_def >< [balloon_descr]
```

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

© CAD Schroer GmbH

For having a new created symbol to be available in the Create Item Number Balloon dialog called by the Creates a New Item Number Balloon tool in the Text + Prims tool group of the Home tab, you have to restart MEDUSA4.

Definition of Parts List

It is possible to administrate the layout of the parts list table. The file

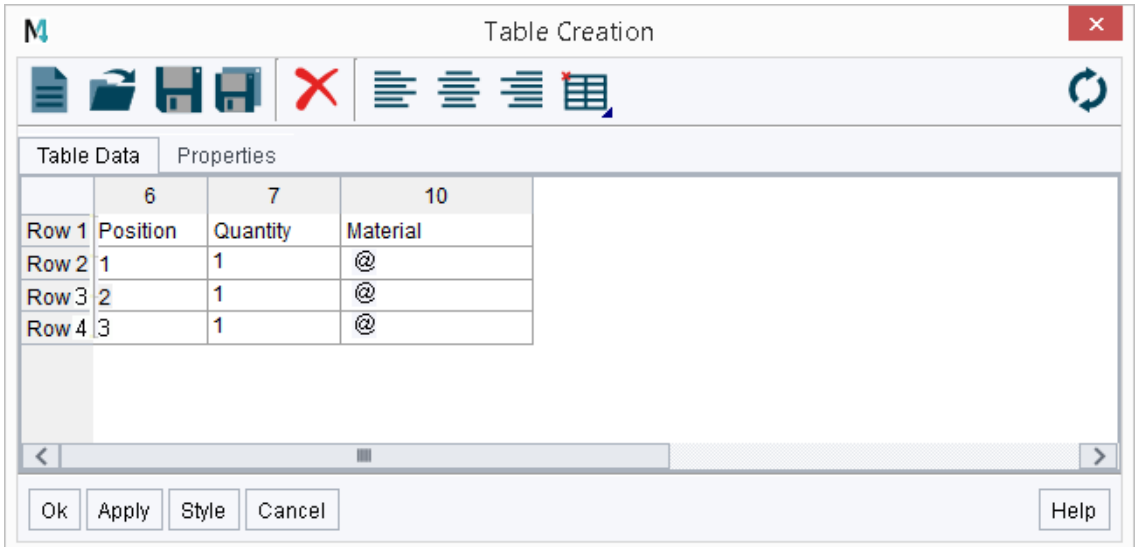
...\M2D\SYMBOL\PARTS_BALLOON\balloontable.bac

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 File" on page 83).

```
-- 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!datatyp :- !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!datatyp :- !quantity
col!label   :- "Menge"
col!width   :- 5
balloontable_props!cols :- balloontable_props!cols  >< [col]

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

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

Figure 33 Create Parts List - 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 item number. If you choose `!bottom`, the first row will be the highest item number and the last row will be the header line.

`col!datatyp :- !material`

This is `datatyp` of the symbol text, it must correspondent 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** these rows!

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

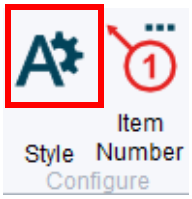

© CAD Schroer GmbH

Generating Styles from DDL Elements

MEDUSA4 provides a tool which can be used to assign style properties to DDL elements which do not yet have a specific style. The tool is located on the Admin tab in the Configure tool group. By default the tool is disabled. To be able to use it, you first have to set the `autogenstyles` variable in the *med2d\m2d\src\defaults.dat* file on `true` and reconfigure your user project.

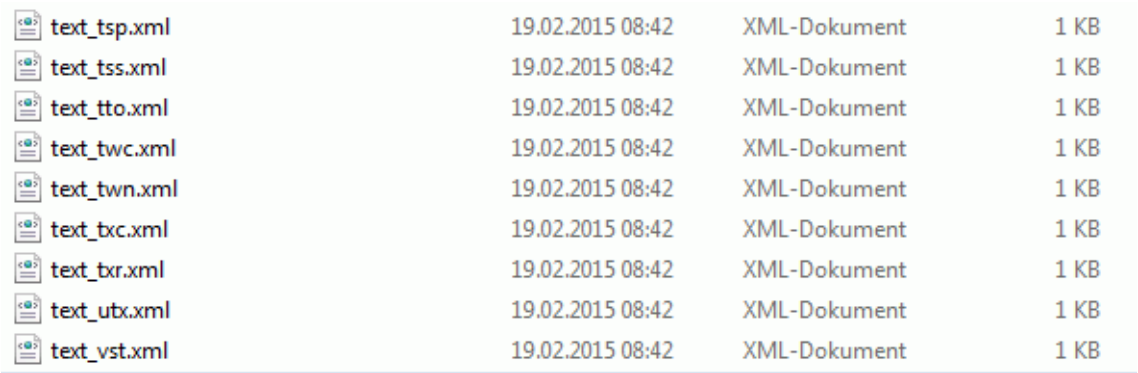
Please note: You must be logged in as Administrator.

Figure 34 Admin > Configuration > Style



When you click once on the button , an XML file for each element type is generated in the directory *\userproj\user\<username>*. Below you can see a selection of XML files.

Figure 35 Example of generated XML Files



text_tsp.xml	19.02.2015 08:42	XML-Dokument	1 KB
text_tss.xml	19.02.2015 08:42	XML-Dokument	1 KB
text_tto.xml	19.02.2015 08:42	XML-Dokument	1 KB
text_twc.xml	19.02.2015 08:42	XML-Dokument	1 KB
text_twn.xml	19.02.2015 08:42	XML-Dokument	1 KB
text_txc.xml	19.02.2015 08:42	XML-Dokument	1 KB
text_txr.xml	19.02.2015 08:42	XML-Dokument	1 KB
text_utx.xml	19.02.2015 08:42	XML-Dokument	1 KB
text_vst.xml	19.02.2015 08:42	XML-Dokument	1 KB

The XML file names are built of the MEDUSA4 element (the element class) and the element type. Examples are *text_vst.xml, text_utx.xml* and *text_txr.xml*, where

- *text* is the element class and
- *vst, utx, txr* are element types.

The file *text.vst.xml* contains the following definition:

Figure 36 Example: text_vst.xml

```
<?xml version="1.0"?>
- <style_classes>
  - <style_class>
      <name>text</name>
    - <styles>
      - <style>
            <!-- text_vst -->
          <name>vst</name>
          <default_label>vst</default_label>
        - <items merge-action="delete">
          - <item>
              <name>type</name>
              <value>vst</value>
              <lock>true</lock>
            </item>
          </items>
        </style>
      </styles>
    </style_class>
  </style_classes>
```

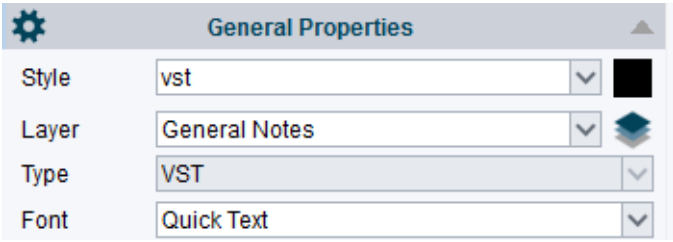
Only the class (`text`) and the type (`vst`) are defined. To assign additional properties to this style proceed as follows:

1. Click on the Opens catalog, tree and browser area button at the bottom right of the status area and choose the Style tab.
   Now all element classes are displayed in the style tree.
2. Click on the triangle in front of text class to display all text styles.
   When you scroll down, any style generated by using the button is displayed by its name, e.g. vst.

Now you can add specific style properties to the vst style and create a new style.

1. Create a text of any style on the sheet.
2. In the style tree click the RMB on the vst style and choose the Use Style option of the popup menu.
   The Text Dashboard displays the General Properties of the vst style.

Figure 37 Text Dashboard displaying the General Properties of vst Style



© CAD Schroer GmbH

3. Open the Text Properties dialog and change the style as described in “Creating and Editing MEDUSA4 Styles” on page 38.
4. Create a new style under a different name (for example `vst_new`) with the properties of the changed `vst` style.
   A new XML file is generated in *\userproj\user\<username>*, which contains the changed style properties.

Figure 38 Definition of the new Style - vst_new.xml File


```xml
<?xml version="1.0"?>
- <style_classes>
  - <style_class>
      <name>text</name>
    - <folders>
      - <folder>
          <name>sub_styles$folder_favourites</name>
        - <stylerefs>
          - <styleref>
              <name>vst_new</name>
            </styleref>
          </stylerefs>
        </folder>
      </folders>
    - <styles>
      - <style>
            <!-- text_vst_new -->
          <name>vst_new</name>
          <default_label>vst_new</default_label>
        - <items merge-action="delete">
          - <item>
              <name>type</name>
              <value>VST</value>
              <lock>true</lock>
            </item>
          - <item>
              <name>height</name>
              <fvalue>8.00000</fvalue>
              <lock>true</lock>
            </item>
          - <item>
              <name>font</name>
              <ivalue>1000</ivalue>
              <lock>true</lock>
            </item>
          - <item>
              <name>color</name>
              <rgbvalue>0 128 0</rgbvalue>
              <lock>true</lock>
            </item>
          - <item>
              <name>thickness</name>
              <fvalue>0.350000</fvalue>
              <lock>true</lock>
            </item>
          </items>
        </style>
      </styles>
    </style_class>
  </style_classes>
```

Defining the Used Codepage

For MEDUSA4 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`:

```
set MED_APP_ENCODING=utf8
set 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.

© CAD Schroer GmbH

Unicode Text

General

MEDUSA4 can use system true type fonts for the display of text. A Unicode (true type font) has font number of 1000 or greater.

The Unicode fonts are defined by two files:

- *unicodefont.dat*:
  This file defines the font number and the full system name of a font. For example:
  ```
  1000 Arial
  1001 Time New Roman
  1002 MingLiU
  ```
  This file is found in *m2d\src* and the `MEDconfig ufont` command compiles it to *m2d\bin* in the project. The file is loaded into 2D using the `font` command.
- *unicodefonts.map*:
  This file contains the font numbers and the display name for the font in the user interface. For example:
  ```
  -- This is the unicode fonts file.
  -- Format : FONT_LABEL <number> <name>
  -- The Unicode font numbers start at a 1000

  FONT_LABEL  1000 "Arial"                    --> Arial
  FONT_LABEL  1001 "Time New Roman"           --> Times New Roman
  FONT_LABEL  1002 "MingLiU"                  --> MingLiU
  FONT_LABEL DEFAULT "Arial" unicodefont
  ```

Unicode text can be used for any piece of MEDUSA4 text.

Please note: It is possible that the underlying OS does not support all the Unicode characters in file and directory names or text in a file so care must be used in using Unicode in MEDUSA4 text that may be used for this purpose.

You can set the default MEDUSA4 font to the defined Unicode in the Default Settings dialog. Choose File > Default Settings > Drawing Elements and select the desired font in the Default Font pull-down menu (see the “Drafting User Guide“, chapter “Default Settings“, section “Text“).

The font used by part balloons or feature control frames can be set by editing the appropriate symbol. The font for a table can be set using the properties dialog or the dashboard.

Limitations

- Only the MEDPLOT_QTPLOT and MEDPLOT_QTRASTER drivers support Unicode text.
- You can convert a text created from Unicode to normal MEDUSA4 and vice versa but the font may not have the character to display.

Usage

To enter Unicode character it may be necessary to use the operating system input method (IME).

© CAD Schroer GmbH

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 30
2D Drafting Image
  link 32
3D Design Products 32

A
Accessing MEDCONFIG 24
Adding layer sets 77
Adding layers to layer set 78
Administration 35
Administrator Mode 36
Attributes
  changing values 41
  lock 40
  use in style 40

B
bacis 18
Background Colors 46
bold 17
Boldness setting 49

C
Changing layer set name 78
character set 92
code 17
Codepage 92
Colors
  setting 46
configuration
  3D Design Products 32
configuration link 2D Drafting Image 32
Configuring a New Project 20
Configuring the Products 29
Creating a Product List 26
Creating styles 38
Customizing
  layer sets 77
  layers 76

D
ddl 17
Decortext fonts 17
defaults 17
Deleting
  layer sets 78
  styles 45
deleting a layer name 77
dfont 17
Dialogs
  Line Properties 39
  Style Creation 40
Documentation 14

E
Editing styles 38

F
Finding Out the Product Names 22
fits 17
font 17
Fonts
  building
    dfont command 17
  compiling 31

H
Help 14
help 18

I
Installation 5
Installing MEDUSA 6
Item Number Symbols, creation 82

L
Layer
  customize 76, 77
Layer Manager

List Buttons 73
Layer Set Attributes 74
Layer Set Management 71
Layer sets
adding 77
adding layers 78
changing name 78
deleting 78
remove layers 78
Layers
adding to layer sets 78
delete name 77
remove from layer sets 78
rename 76
Line Properties dialog 39
Lines
setting thickness 49
Lock
Attributes 40

M

mdba_application_control 17
mdbc_rebuild_control 17
MED_COMMA_IS_DEZIMAL 15
MED_DOC_LANG 14
MED_UNICODE_FILENAMES 15
MEDCONFIG
description 13
start first time 24
using - 16
message 17
Modifying styles 44

N

Name
changing - of layer sets 78

O

Online Help 14
options of MEDCONFIG 16

P

Page Labels 54
patts 17
permanent saving of styles 42
plot_config 17
plotting_dev_kit 17
prims 17
Product List 25
project reconfiguration 18
Prompts
Enter command> 19
psyms 17

Q

quit 18

R

raster_path 17
reconfigure 18
Reconfiguring a project 21
Removing layers from layer set 78
renaming a layer 76

S

Save Style Tree 42
Saving
styles 41
styles permanently 42
styles temporarily 41
Separators of Decimal Values 15
Setting
standard boldness 49
standard colors 46
standard thickness 49
Setting up Page Labels 54
Sheet Header
area_dlglabel 59
attnam 60
attval 60
Blank line 57
Chapter heading 56
Chapter of the configuration file 57
Comment lines 56
Configuration file 55
consider_ng 58
Data lines 56
dlg_maxchar 58
dlg_maxrow 57
emptytextchar 58
enabled 61
General Keywords 59
Keywords 57, 59
label 60
layer_protect 58
mandatory 61
maxchar 61
multitext 61
sfeld_dlgarea_ 59
Structure of the configuration file 56
suppress 62
texttyp 60
zoom_hor 57
zoom_ver 57
Sheet structure tree 42
Style Creation dialog 40
style tree 42
Styles
attributes to use 40
creating 38
creating and editing 38
deleting 45
modifying 44
prevent changing attributes by using lock 40
saving 41
saving permanently 42
saving temporarily 41
using 43

© CAD Schroer GmbH

styles 17
symbols.xml 70

T

temporare saving of styles 41
Text
  setting thickness 49
  UNICODE 93
Thickness setting 49
ttable 17

U

UIE_COMMA_IS_DECIMAL 15
uistrings 17
Unicode Filename Support 15
UNICODE Text 93
Uninstalling 10
user interface
  compile messages 17
User Project 12
Using styles 43

V

Values of attributes
  changing 41

W

wsconfig 18

© CAD Schroer GmbH