



MSet application Manual

MSet 8.40 – support drawing tool for Bentley CAD products

CONTENTS

CONTENTS	1
INTRODUCTION	2
About MSet application.....	2
Main features of MSet	3
License.....	3
Operation system requirements	4
Starting of MSet	4
GRAPHIC INTERFACE	5
Main window of MSet.....	5
Main menu item “File“	6
Main menu item “Info“	7
Context menu.....	8
Dialog “Scale“	8
Dialog “Options“	9
MSET IN USE	11
Definition of Instruction directory	11
Instruction – DGN file Relation.....	12
INSTRUCTION	12
About Instruction	12
Instruction structure	12
How to modify Instruction	14
COMMANDS IN INSTRUCTION	14
Standard commands	14
Special syntax of dedicated commands	14
Macros.....	15
Examples of often used MicroStation Key-In commands	17
INSTALLATION AND CONFIGURATION	18
Installation.....	18
Configuration.....	19
Uninstall.....	19

MSet variables.....	19
Demonstration example Mset_demo.dgn	20
Notations and Abbreviations.....	20

INTRODUCTION

About MSet application

Program MSet is a support tool for CAD Bentley products MicroStation, PowerDraft, PowerMap, PowerCivil and Bentley Map PowerView (hereinafter MicroStation) which increases the efficiency and comfort of drawing according to given rules – Instruction.

MSet uses the fact that almost all common tasks in MicroStation can be activated via text commands or so-called *Key-Ins*. After selecting a desired item from MSet tree menu, one or more commands are sent to MicroStation. These commands are immediately executed in the same way, as it would be entered in the command prompt of MicroStation. So for example, user can set the layer, color, thickness, line style, select the active cell, select the drawing tool etc. In more advanced cases the user can control the visibility of layers - both individually and in groups, connect/disconnect reference files, including management visibility of the layers and the like. Drawing rules and their graphical representation in the tree menu are defined in the so called *INSTRUCTION* – saved in the XML file. The user can edit that file according to his requirements, or derive new instructions from it.

The structure of the Instruction is designed so that it does not have to repeat commands valid for the entire group of lower orders (tree), thus greatly improving the transparency of its registration.

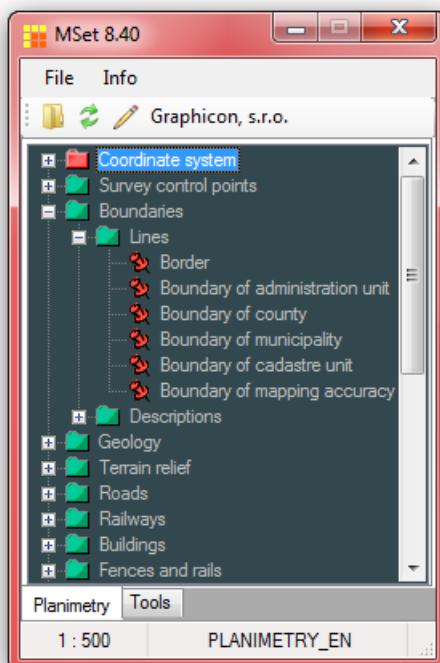


Fig. 1 Main MSet window

```
<Group Cpt="Boundaries">
lv=Boundaries;co=0
<Group Cpt="Lines">
wt=1;place lstring
<Feature Cpt="Border" >lc=2.012</Feature>
<Feature Cpt="Boundary of administration unit" >lc=2.032</Feature>
<Feature Cpt="Boundary of county" >lc=2.042</Feature>
<Feature Cpt="Boundary of municipality" >lc=2.052</Feature>
<Feature Cpt="Boundary of cadastre unit" >lc=2.062</Feature>
<Feature Cpt="Boundary of mapping accuracy" >lc=2.082</Feature>
</Group>
```

Fig. 2 Passage from Instruction

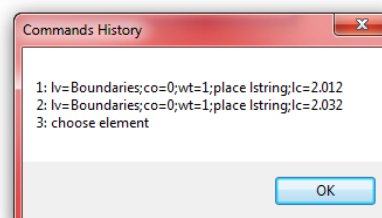


Fig. 3 Example of sent commands

The MSET application consists of two cooperating programs:

- *mset.ma* MDL application runs as the first in the MicroStation environment,

- *mset.exe*, the Windows application. Its start is triggered by the MDL application *mset.ma* directly or indirectly (in case of PowerDraft, using *MStart.exe*).

Main features of MSet

- ✓ It works with Bentley CAD products of version V8 MicroStation, PowerDraft, PowerMap, PowerCivil, Bentley Map Standalone and Bentley Map PowerView
- ✓ It has a user-friendly interface, clear menus
- ✓ When working with cells, texts and user lines it uses a scale
- ✓ It has a structured tree menu to select commands
- ✓ It has a possibility to define more tree menus on separate pages
- ✓ Commands and also their graphic representation are stored in the Instruction - XML file
- ✓ Instruction can use all *Key-Ins* commands recognized by MicroStation
- ✓ Commands defined in the upper node are inherited by all lower (child) nodes
- ✓ Lower nodes can modify its inherited command or only its part, or cancel them
- ✓ MSet saves the information about used scale in the DGN file
- ✓ MSet saves the name of the used Instruction in the DGN file
- ✓ When opening a DGN file, MSet automatically adjusts the scale and activates the relevant Instruction
- ✓ Allows quick access to the editing of Instruction
- ✓ All important settings are handled through the MicroStation configuration variables

License

For unlimited use MSet requires a license. The license is bound to the computer name and its owner. A single computer license is valid for all appropriate Bentley products. The number of MSet instances running simultaneously on one computer is not limited. All required data are saved in the license file *mset.lic*, which is usually sent via email after purchasing a license.

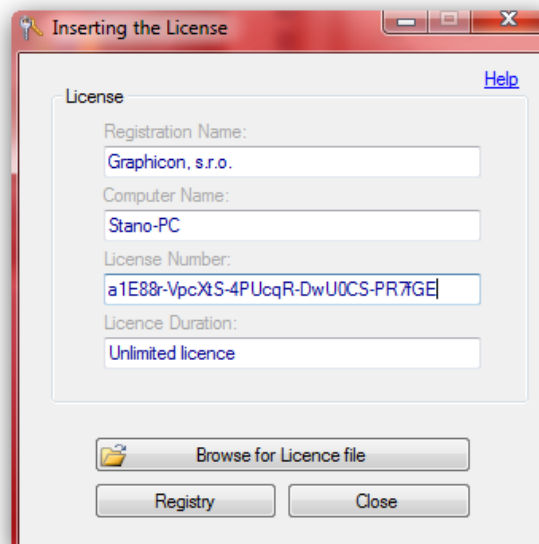


Fig. 4 Dialog for a License

The license file is saved in the same directory as the *mset.exe* program. We recommend the insertion of the license using the *MLic.exe* application, which can be activated from the MSet menu *File > Tools > License* or from Windows *Start > All Programs > MSet .. > License*.

After entering the license restarting MSet is necessary, but restarting MicroStation itself is not needed. The license is usually time unlimited, but in certain cases it may be time limited, for example for testing. More information about the license is on the [MSet web site](#).

If there is no valid license, MSet will run in Trial mode. There are three restrictions in Trial mode:

- Start delay of about 20 seconds,
- Limited time of running - 15 minutes,
- An Information dialog window *Trial mode* is active.

All other functions of MSet during Trial mode are fully available.

Note: Licenses for MSet 8.1x, 8.2x and 8.3x are valid for MSet 8.4x too, i.e. the upgrade is free.

Operation system requirements

Operating system:

- Windows XP, Vista, Windows 7 with current updates
- Microsoft .NET Framework 4 Client Profile, or higher
- PDF viewer for viewing this document

Target Bentley products:

- MicroStation, PowerDraft a PowerMap in versions 8.05, 8.09 a 8.11
- PowerCivil, Bentley Map PowerView in version 8.11

Hardware requirements:

- - are fulfilled if one of the CAD Bentley products is installed on the computer and working

Starting of MSet

MSet is a MDL application therefore its starting is standard:

- By entering the command "mdl load mset" into the Key-in prompt of MicroStation
- By clicking on its name in the dialog for starting MDL applications ("Utilities > MDL Applications")
- Recommended is to define the starting command as a certain function key
- A menu item for MSet starting is available in Bentley products from version 8.09.



Fig. 5 Item for MSet in menu of MicroStation from version 8.09

By using MSet with one of the products the PowerDraft family (PowerDraft, PowerMap a PowerCivil) it is necessary to have the MStart application running in the background of the operating system. This application is responsible for a successful start of MSet for PowerDraft. All required settings are made by the install program. Readiness is indicated by an icon in the notify area - in the right bottom part of monitor screen:



Fig. 6 Icon of program MStart in the notification area

If MStart is not running, its start is available via *Start > All Programs > MSet 8.40.xx > MStart*. Unlike the products in the PowerDraft family, MicroStation and Bentley Map don't need MStart, they can start mset.exe directly.

GRAPHIC INTERFACE

Main window of MSet

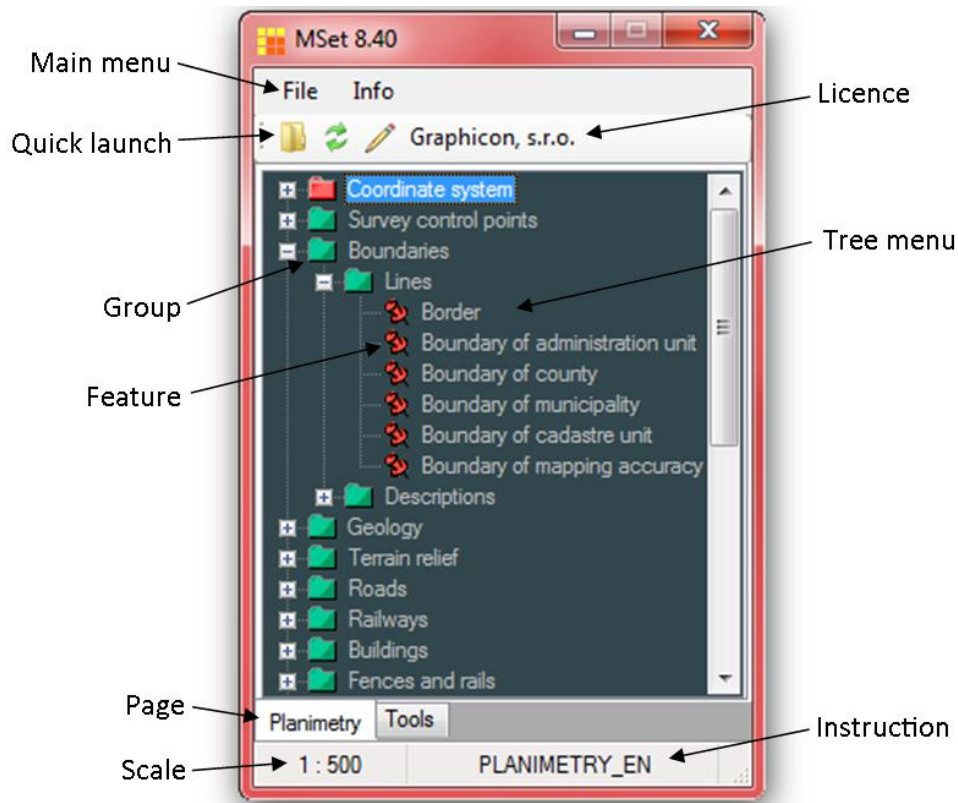


Fig. 7 Main window of MSet and its parts

As shown in Figure 7 the Main window of MSet has one or more Pages in addition to common controls. On each Page is always one tree menu. The tree menu consists of two types of nodes: Group and Feature nodes. A Group node is represented by a green folder icon if it is selected - by a red folder icon. A Feature node is represented by a pin icon. On the upper band near the quick buttons the name of license owner is shown. The bottom state panel is split in two areas. On the left side is the current scale and on the right side is the name of the active Instruction.

Main menu item “File”

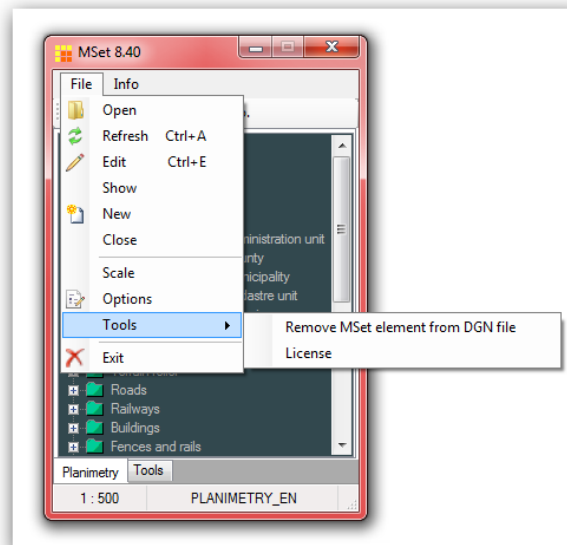


Fig. 8 Menu item „File“

Open

This item brings up a dialog to open the Instruction file. The initial directory in which dialog is opened is first selected in the dialog "Select Directory", which precedes the dialog “Select Instruction”.

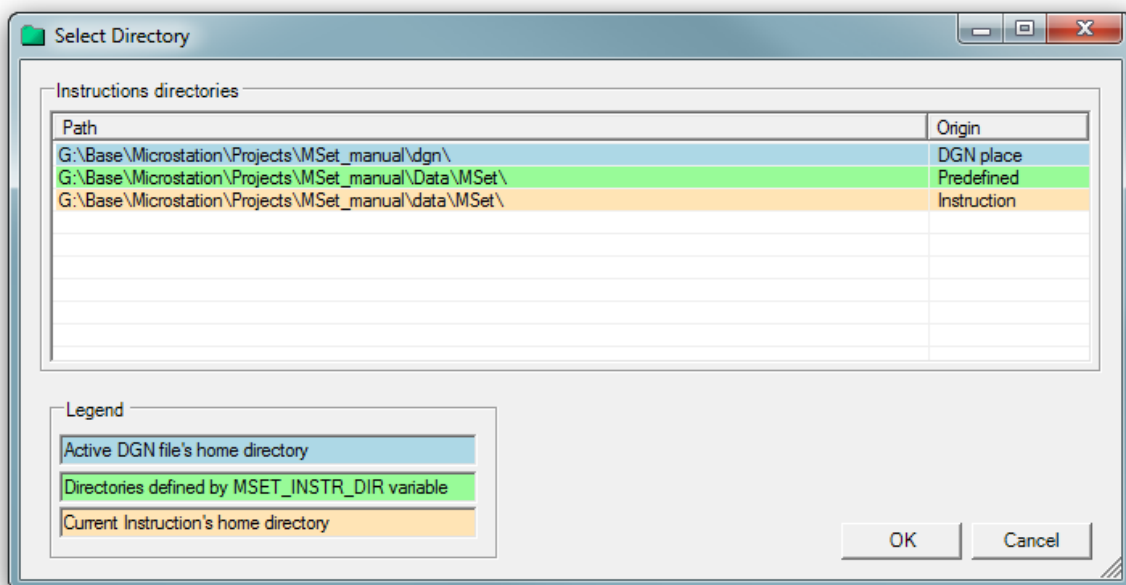


Fig. 9 Selecting initial directory for dialog Open Instruction

Refresh

Reopen the currently opened Instruction, usually for the purpose of applying changes that were made in the Instruction file.

Edit

Open current Instruction in the default editor for editing.

Show

Opens the current Instruction in the program that is associated with the XML extension on the OS level. It is usually Internet Explorer. This tool is especially useful for finding errors in compiling the Instruction.

New

This item is dedicated to creating a new Instruction file, which consists of one page, one group and two features. Instruction is displayed in Figure 16. The user can modify and develop the Instruction according to his requirements.

Close

Closes currently open Instruction.

Scale

The "Scale" dialog opens, [see standalone article](#).

Options

The "Options" dialog opens, [see standalone article](#).

Tools >

- **Remove MSet element from DGN file** - removes MSet application data (name of Instruction and scale factor) from the DGN file,
- **License** – the MLic application starts for insertion or modification the License.

Exit

Terminates the MSet application.

Main menu item "Info"

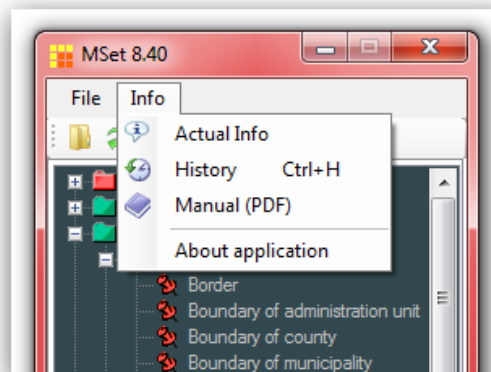


Fig. 10 Menu item „Info“

Actual Info

Opens dialog window with actual information:

- DGN file – full path to the active DGN file
- MSet data saved in the DGN file:
 - Name of the instruction
 - Scale factor
- Instruction file – displays full path to actual opened Instruction file
- Instructions directories – displays all the directories that are searched at the opening of the Instruction

- Communication channel – displays info about the channel under which the current MSet communication is running (used for connection with other applications).

History of commands

The dialog window is opened with information about commands sent to MicroStation. Displays a maximum of the last 6 commands sent. See Figure 3.

Manual

Open this document.

About application

The dialog window opens with information about the MSet application. At the same time a full path and version of mset.ma is displayed on the status panel of MicroStation – see Figure 11.



Fig. 11 Information about MDL application mset.ma on the MicroStation status bar

Context menu

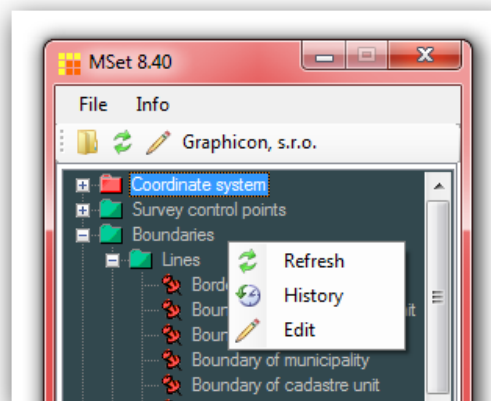


Fig. 12 Context menu is accessible via right mouse click

By right click on the window a context menu appears. In this menu are often used commands, all were discussed in previous chapters.

Dialog “Scale”

Dialog window “Scale” is accessible via menu *File > Scale*, or by double clicking on the status bar of the MSet application. Dialog is opened in the short form, via control [More...](#) expands to its full form, see Figure 13.

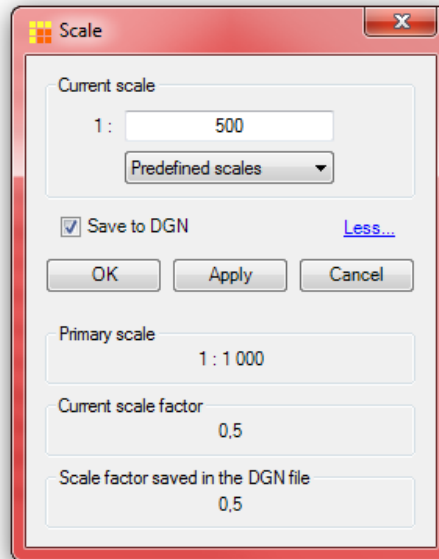


Fig. 13 Dialog “Scale”

In the upper part of the dialog is text box, with current scale. In this text box user can set desired scale. Scale can be only integer, greater than zero. Below this text box is combo box with range of predefined scales. After selection one of them its value is written into upper text box. Every change of the current scale must be confirmed by button *Apply*, or *OK*. If option *Save to DGN* is checked, the information about scale (scale factor) will be saved to the DGN file.

MSet application has defined so-called Primary scale. Primary scale is scale in which libraries, user lines etc. are made. If the primary scale is the same as an active scale, so cells, user lines and other elements are used unchanged in size – scale factor is equal 1. Within MSet application Scale factor (SF) is ratio by which is size of cells/terminators, texts, user lines etc. multiplied in the drawing.

$$\text{Actual scale factor} = \frac{\text{Aktual scale}}{\text{Primary scale}}$$

Relation between the scale factor and scales

Primary scale used to be defined in the Instruction in the *Scales* element, see the article about Instructions. If doesn't, the Primary scale is set to the default value 1000. In the lower part of the dialog window is Primary scale displayed. Below it is the actual scale factor, which is calculated using the above formula. The last item in the dialog window is the scale factor saved in the active DGN file. It is used to calculate and set the active scale after opening of the instruction ($AS = SF \times PS$). The active scale factor can be set by macro $\$F(x)$, see article about macros.

Dialog “Options”

Dialog window “Scale” is accessible via menu *File > Options*. Most of the settings in dialog are saved in the user's personal profile. After the next start of MSet, the saved settings are applied.

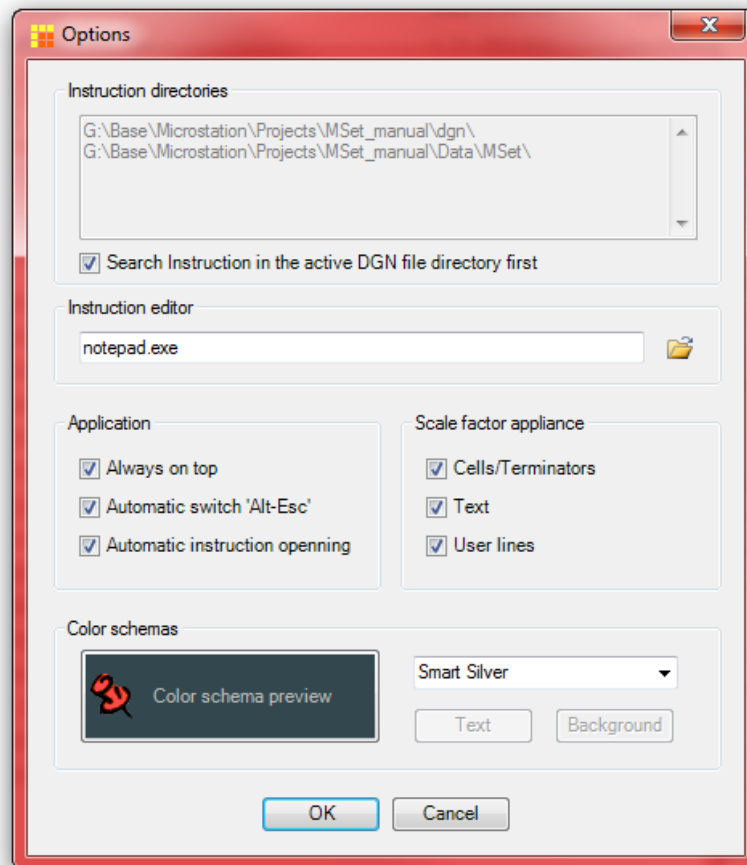


Fig. 14 Dialog window „Options“

In the following articles individual options are described.

Instruction directory

Contains paths to directories in which the MSet required by the Instruction seeks by open DGN file. Directories are set by the MicroStation variable MSET_INSTR_DIR. Its proper setting is essential for smooth functioning of the automatic opening of the associated Instructions. The list includes the home directory of the active DGN file, if the option *Search Instruction in the current DGN Directory first* is checked.

Search Instruction in the active DGN Directory first

If checked, application searches in the active DGN file directory first. Instructions saved in the same directories as active DGN files have higher priority to than Instructions saved in directories defined by MSET_INSTR_DIR variable (in the case of the same name).

Instruction Editor

The preferred editor for editing of Instructions is in the text box. The default editor is Notepad.

Application

The panel has three options:

- **Always on top** – if checked, the application window is always above the other running applications. The advantage is that user doesn't need to manually switch to MicroStation by pressing keys Alt-Tab, Alt-Esc or select it from the Task bar to make it visible.

- **Automatic switching 'Alt-Esc'** - if checked, the application immediately after sending commands to MicroStation emulates pressing of keys Alt-Esc. In this way next application in the list of running Windows applications become active – it is mostly desired MicroStation and the user doesn't need to manually switch to MicroStation. This immediately enables use the function keys in MicroStation, or text can be written into text dialog which is focused etc.
- **Automatic instruction opening** – this switch determines whether after opening a new file in MicroStation, which contains the application element, the application MSet attempts to find and open the associated Instruction.

Scale factor appliance

The scale factor is applied to command parameters so that a particular command parameter is multiplied by the scale factor. Eventually, an additional command, which is transparent to the user is generated to achieve the desired effect. Commands which should be affected by the scale factor must have a prescribed syntax; otherwise the scale factor does not apply. More information is in the article [Special syntax of dedicated commands](#).

The panel has three options:

- **Cells/Terminators** – if checked, the scale factor is applied to cells and terminators
- **Text** – if checked, the scale factor is applied to texts, eventually to text nodes
- **User lines** – if checked, the scale factor is applied to user defined lines

Color schemas

In this section the user can choose the text and background color for the tree menu. User can choose from several preset color schemes or select their own color combination.

MSET IN USE

Definition of Instruction directory

Instructions directories are dedicated to storing Instruction files and they are defined by MicroStation variable MSET_INSTR_DIR on the application level. After the MSet install process, variable MSET_INSTR_DIR is set to directory *MSet 8.40.xx\Data* – the area dedicated to MSet applications data. The full path depends on the operation system:

OPERATING SYSTEM	DIRECTORY FOR MSET APPLICATION DATA
Windows XP	C:\Documents and Settings\All Users\Documents\MSet 8.40.05\Data
Windows Vista	C:\Users\Public\Documents\MSet 8.40.05\Data
Windows 7	C:\Users\Public\Documents\MSet 8.40.05\Data

Table 1 Default value of MSET_INSTR_DIR for different OS

The user can use this directory for their own Instructions, but more recommended is to define their own directory (directories) by setting variable MSET_INSTR_DIR on some level (site, project or user). The directory can also be a network drive, shared by several users.

After opening the DGN file and loading the associated instruction name, MSet tries to find the relative Instruction just in the Instructions directories. In addition if the option *Search Instruction in the active DGN file directory* (Dialog "Options") is checked, then the first searched directory is the directory, where the active DGN file is saved. Variable MSET_INSTR_DIR can hold more than one Instructions directory. In this case searching proceeds according to the order in which the directories are written in the variable.

Instruction – DGN file Relation

Any DGN file can be associated with just one Instruction, which is activated by opening the DGN file.

Figure 15 clarifies the mechanism of finding and opening Instruction associated with the DGN file.

The necessary condition for automatic searching is to set option *Automatic instruction opening* in the options dialog. The mechanism is triggered in two cases:

- By starting MSet application
- By opening the DGN file, if MSet is running

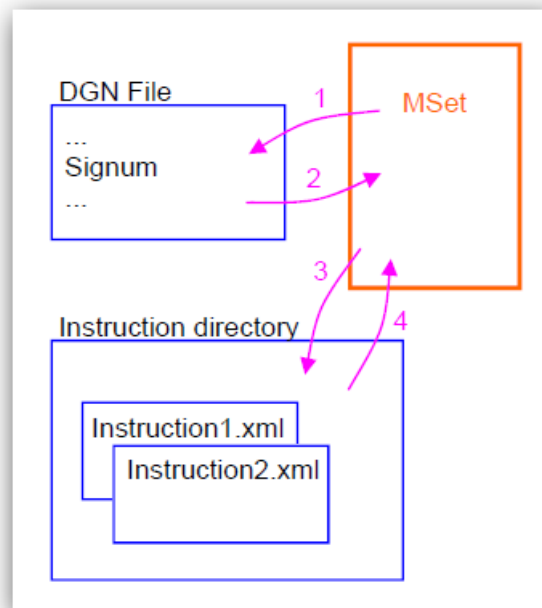


Fig. 15 Instruction activating mechanism

The sequence of events to search and open Instruction by MSet is as follows:

- Step 1: The active DGN file is searched for the MSet application element
- Step 2: From MSet application element is extracted the scale factor and name of the Instruction
- Step 3: MSet searches in Instructions directories for a suitable Instruction
- Step 4: The Instruction is opened

INSTRUCTION

About Instruction

Instruction for MSet is an XML file (text file, ANSI Encoding) with a special structure. In Instruction are defined commands and groups of commands and at the same time their graphical interpretation in the MSet window. In the header of the file is DTD - Document Type Definition which determines the formal rules of the Instruction structure. DTD is helpful for the user to avoid mistakes in composing or changing the Instruction. The instruction name without an extension can be saved in the DGN file into so-called application element. When the application element is saved, an association between the DNG file and Instruction is created.

Instruction structure

Because the Instruction file is an XML document, an XML declaration is the first line of the text. The next lines are the DTD - Document Type Definitions. It is displayed in red in Figure 16.

```
<?xml version="1.0" encoding="windows-1250"?>
<!-- *** MSet, Graphicon, s.r.o. *** -->
<!DOCTYPE Instruction [
<!ELEMENT Instruction (#PCDATA | Page | Scales)* >
<!ELEMENT Scales (#PCDATA) >
<!ATTLIST Scales PrimaryScale CDATA #REQUIRED >
<!ELEMENT Page (#PCDATA | Group | Feature)* >
<!ATTLIST Page Cpt CDATA #REQUIRED >
<!ELEMENT Group (#PCDATA | Group | Feature)* >
<!ATTLIST Group Cpt CDATA #REQUIRED >
<!ELEMENT Feature (#PCDATA) >
<!ATTLIST Feature Cpt CDATA #REQUIRED >] >
<!-- _____ -->

<Instruction>
<Scales PrimaryScale="1000">50,100,200,500,1000,2000,5000</Scales>
  <Page Cpt="Page example">
    <Group Cpt="Group example" >wt=0;lc=0
      <Feature Cpt="Line" >co=3;pla 1st</Feature>
      <Feature Cpt="Circle" >co=4;pla cir</Feature>
    </Group>
  </Page>
</Instruction>
```

Fig. 16 Example of trivial Instruction

After the DTD is the body of the Instruction (element *Instruction*). In coordination with DTD, the Instruction can contain only 5 XML elements:

- Instruction
- Scales
- Page
- Group
- Feature

With the exception of the *Scales* element, it is possible to define the commands for every element in MicroStation. Commands are defined as an element text. If there are more commands, they are separated by a semicolon. After the last command a semicolon is not required.

Instruction element is a root element, a grandparent of the other elements. At the level of this element it is possible to define the command/commands for MicroStation. These commands are executed only once - when the Instruction is opening or refreshing. Element *Instruction* consists of one or more child elements *Page* and can contain *Scales* element.

Scales element – defines range of common used scales which are offered in the dialog “Scales” and defines primary scale.

- Attributes : *PrimaryScale* – defines primary scale (integer number > 0)
- Child elements: : none
- Example:
< Scales PrimaryScale="200">50,100,200,400,500,1000,2000</Scales>

Page element - at this level are defined named pages with their own tree menus. MSet displays as many Pages elements with named tabs as are defined. Each *Page* element usually represents a certain integrated part of Instruction. Commands defined on this level are executed ones, on entering the Page

- Attributes : *Cpt* - (Caption) is obligatory - it carries text which is displayed on the tab of the page
- Child elements: minimum one child *Group* element or *Feature* element.
- Graphical interpretation: Standalone page with a tree menu
- Example: <Page Cpt="Planimetry"> ... </Page>

Group element is dedicated to grouping relative commands - *Feature* elements or *Group* elements.

- Attributes: *Cpt* - (Caption) is obligatory – carries text which is displayed on the node of the tree menu. At *Group* element level is possible to define commands for MicroStation.
- Child elements: minimum one child element *Group* or *Feature*.
- Graphical interpretation: Green folder, at focus red folder
- Example: `<Group Cpt="Borders">lv=2;co=0; ... </Group>`

Feature element – end element (leaf). Its specialty is that after it is chosen from the tree menu, a comprehensive command is sent to MicroStation. This command contains all particular commands defined from the first *Group* element to this end *Feature* element in this order.

- Attributes: *Cpt* - (Caption) is obligatory – carries text which is displayed on the node of the tree menu. At the *Feature* element level is expected to define some commands for MicroStation.
- Child elements: - no child elements
- Graphical interpretation: Pin icon
- Example: `<Feature Cpt="Cadastre border">pla lstring;lc=2.062</Feature>`

Note: The first element on the Page should be an element *Group* instead of *Feature*, because if *Feature* is first, then the application has no chance after clicking to delegate focus to another (upper) node. In this case the next click on *Feature* will result in no action. However, if only *Feature* elements are on the Page, it is recommended the first *Feature* be a dummy (without commands) for focus parking.

Example: `<Feature Cpt="--- MSet ---"></Feature>`

How to modify Instruction

The simplest way to modify Instruction is by using *Notepad* text editor. The main advantage is that editing is straightforward – everything is under control of the user. This editor is well known and comes as a part of Windows installation. Disadvantage is that it doesn't recognize syntax and so it doesn't warn about potential mistakes in the XML text.

Another way is use editors which recognize XML syntax. They are helpful for writing Instruction correctly. This is, for example, *XML Notepad* or *Frontpage*, both from Microsoft or *XML Pad* etc.

Tip: Jumping to the desired line in Notepad can be done only with Word Wrap turned off via Ctrl-G.

COMMANDS IN INSTRUCTION

Standard commands

The Instruction mostly uses standard MicroStation commands. They are described in the MicroStation manual. Moreover commands for running MDL applications could be used. These commands are called *Key-In* commands. All commands used in the Instruction must work as if they are entered directly into MicroStation Key-in prompt and vice-versa. This is one way to debug Instruction at composing time. How it was described in the previous article – commands are defined as text of XML element. If there are more commands, they are separated by a semicolon. After the last command, a semicolon is not required. See Figure 16, where commands are displayed in black color.

Special syntax of dedicated commands

All key-in commands can be used in the Instruction in all their variants except commands which are modified by the scale factor. For illustration the same command for circle placement could be written in any manner because scale factor is not applied: `place circle`, `place cir`, `pla circle`, `pla cir`, `PLA CIR` etc. Opposite this is for example the command for setting text

parameters which must be written only in specified form: `th=text_high`, or `TH= text_high`. MSet must recognize this command and modify its argument – numeric value according to scale factor.

Vice-versa, if syntax is not kept the MSet application overlooks the command and the scale factor is not applied. For example if a *place cel* (instead of *pla cel*) command is used it will only be understood by MicroStation, but not MSet, so MSet will not send the correct command to set the cell scale. The list of standard commands with special syntax is in the next table:

Calls and terminators	
<code>pla cel</code>	select place active cell tool with relative off
<code>pla cel rel</code>	select place active cell tool with relative on
<code>ac=</code>	set active cell and select place active cell tool with relative off
<code>ar=</code>	set active cell and select place active cell tool with relative on
<code>cm=</code>	place active cell matrix tool
<code>pla ter</code>	select place terminator tool
<code>lt=</code>	active terminator
Texts	
<code>th= or TH=</code>	active high
<code>tw= or TW=</code>	active width
<code>tx= or TX=</code>	active high and width
<code>ls= or LS=</code>	active line spacing
Line styles	
<code>lc=</code>	active line style

Table 2 List of commands with special syntax

Note:

- Other cases when it is necessary to take into account the scale factor can be addressed using a macro `$$S(x)` whose use is explained in Article *Macros*,
- Because of backward compatibility with products older than version 8.9, user lines scaling runs in the background and is transparent for the user. Applying Scale factor in other cases is made by modifying the sending commands. The user can monitor this mechanism in the history of sent commands.

Macros

In the Instruction can be used also special commands so-called macros to achieve another way of non-accessible functionality. A macro begins with character '\$' following its name and if it is necessary, round brackets with parameters.

MSet contains next macros:

- `$$S` – Scale Factor macro
- `$$F` – Scale Factor set macro
- `$$X` – Ignore macro for ignoring commands defined on higher levels
- `$$C`– Continue macro

Scale Factor macro `$$S(x)`

This macro can be used if you need to adapt the numerical value depending on the scale factor. The scale factor macro `$$S(x)` has one parameter `x` –which is number entered by the user that will be multiplied by actual scale factor. The result will replace an entire string of macro. The macro can be applied mostly while constructing the graphical elements which the user wants on the reprographic

output to be always the same size (independent of the scale factor). For example, macro can be used to construct a circle for a part number, see Fig. 17.

Example 1: There is a requirement to draw the circle, which will be on reprographic output in the same size, regardless of the used scale factor. In other words, its radius in the drawing will be dependent on the scale factor. For example if the scale factor = 1, the radius will be 2, or if the scale factor = 2 the radius will be 4 etc. If the command for circle drawing with a given radius is: `pla cir radius $S(2)` and the scale factor is 2.88, the final command send to MicroStation will be:
`pla cir radius 5.76`

The following command was used for construction part number on Figure 17:

```
<Feature Cpt="Part number">pla cir radius $S(2)</Feature>
```

Example 2: The macro is often used by parallel copy/moving of elements, to achieve parallel lines with a constant distance on the reprographic output independent of the scale factor. The following command was used to move the dashed line in Figure 17:

```
<Feature Cpt="Dash line shunt">move parallel keyin $S(0.3)</Feature>
```

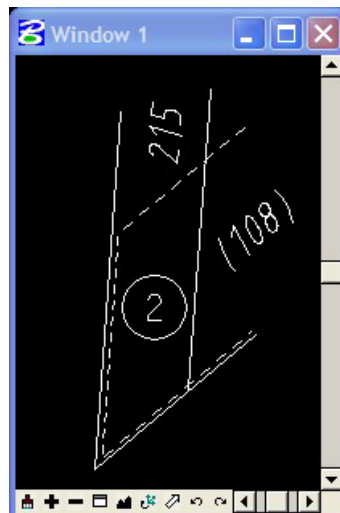


Fig. 17 Example of drawing - the circle and shifting the line by using macro `$S(x)`

Scale Factor set macro `$F`

This macro can be used to set the actual MSet Scale Factor without writing it to the DGN file. Macro `$F(x)` has one parameter `x` –which is a number entered by the user – required for the scale factor. By executing the `$F` macro MicroStation will send the command `RESET` to force the user to reactivate the current tool and this way apply the new scale factor. If macro `$F` is part of several commands, it is executed first, regardless of its position within the package.

Example 1: Setting different scales (if primary scale is 1:1000)

```
<Feature Cpt="Scale 200"      >$F(0.2)      </Feature>
<Feature Cpt="Scale 500"     >$F(0.5)      </Feature>
<Feature Cpt="Scale 1000"    >$F(1)         </Feature>
```

Example 2: Setting different scales (if primary scale is 1:200)

```
<Feature Cpt="Scale 200"      >$F(1)         </Feature>
<Feature Cpt="Scale 500"     >$F(2.5)       </Feature>
```



```
<Feature Cpt="Scale 1000"      >$F(5)      </Feature>
```

Ignore macro \$X

This macro can be used if you need to ignore all the commands that preceded it, i.e. commands defined in the higher nodes, which would otherwise be inherited and executed. \$X macro has no parameter.

Continue macro \$C

It serves to split a single MicroStation command into two MSet lines. The macro is on the end of commands defined on the Group level. The macro has no parameter. Its presence ensures that a semicolon is not automatically inserted at the junction of two groups of commands. In this way it is possible to start the MicroStation command at a certain group level and complete it at a lower level in many different ways.

Example 1: Level setting. Suppose that in the DGN file are levels with names 50 CKN parcels boundaries, 50 CKN inner drawing a 50 CKN parcels numbers. A common beginning of the command 'lv=50 CKN ' for the level setting is on the Group level and depending on the selected Feature will be completed in different ways:

```
<Group Cpt="C-KN"                >co=7;lv=50 CKN $C
    <Feature Cpt="Parcels boundaries" >parcels boundaries</Feature>
    <Feature Cpt="Inner drawing"      >inner drawing;co=215</Feature>
    <Feature Cpt="Parcels numbers"    >parcels numbers</Feature>
</Group>
```

The resulting commands are:

```
co=7;lv=50 CKN parcels boundaries
co=7;lv=50 CKN inner drawing;co=215
co=7;lv=50 CKN parcels numbers
```

Examples of often used MicroStation Key-In commands

Attributes settings

COMMAND	MEANING
lv=level_name	Set active level to <i>level_name</i>
co=x	Set active color to number x
wt=x	Set width to number x
lc=x	Set line style to number x
lc=style_name	Set line style to <i>style_name</i>
aa=x	Set active angle to x
as=x	Set active scale factor to x (meaning in MicroStation)

Elements construction

COMMAND	MEANING
place line	Select tool for line drawing
place smartline	Select tool for smartline drawing
place circle	Select tool for circle drawing
place circle radius 50	Select tool for circle drawing with radius 50
place circle radius \$\$ (50)	Select tool for circle drawing with radius 50 multiplied by scale factor
place cell	Select tool for cell placement
place cell relative	Select tool for cell placement – relative to levels

ac=cel_name	Select active cell - <i>cel_name</i>
place shape	Select tool for shape drawing

Working with levels

COMMAND	MEANING
set level off all	All levels are switched off
set level on all	All levels are switched on
set level off level_name	Level <i>level_name</i> is switched off
set level on level_name	Level <i>level_name</i> is switched on
update all	All MicroStation windows are redrawn (refreshed)

Working with files

COMMAND	MEANING
rd=<file_name>	Open file <i>file_name</i> as active
rf= <file_name>	Attach file <i>file_name</i> as reference
raster attach fixed <file_n>	Open raster file <i>file_n</i>
raster detach <file_name>	Close raster file <i>file_name</i>
reference Exchange <file_n>	Exchange active file with reference file <i>file_name</i>
attach library <cell_library>	Attach cell library

Table 3 Examples of often used Key-In commands

A complete list of Key-In commands is present in the MicroStation documentation, or in the documentation of the particular MDL applications.

INSTALLATION AND CONFIGURATION

Installation

For MSet installation it is necessary to download the installation program MSet_8.40.xx_setup.exe (where xx stands for the minor version and it could change over time) from [MSet web page](#). Before upgrading an existing MSet installation to a new version it's recommended to uninstall the old one. The installation process is standard, except the end of installation, when the configuration program MSetup is launched. In this step it's possible to choose from the list of Bentley products installed on the computer, which will be bound with the MSet application. All of them are selected by default. The products which are already bound with another MSet installation are in pink.

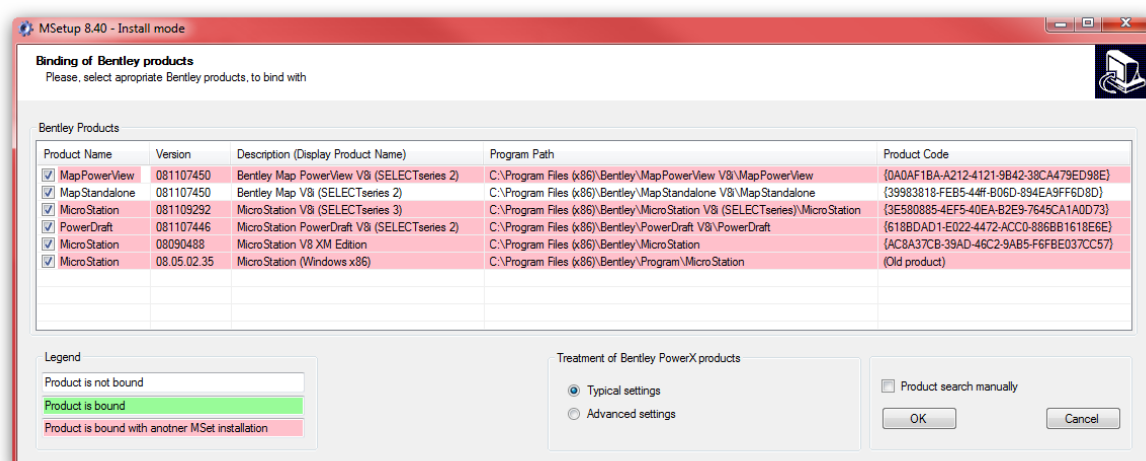


Fig. 18 Dialog of MSetup program in install mode

The user can manually browse for a Bentley product by checking the *Find product manual* in the lower right corner of the dialog. The button for browsing appears and the user can browse and select the desired product. Product should be Bentley application, for example *ustation.exe*, *draft.exe* etc. Notice: Before installation the user should end all running MSet programs, including MStart program. Note: MStart is not on the Task bar, but its icon is displayed in the notification area.

Configuration

For modification the MSet linkage to Bentley products can use the MSetup configuration program, which is accessible via *Start > All Programs > MSet 8.40.xx > MSetup*. The program starts in configuration mode, check boxes and highlighted rows show actual state of binding to Bentley products. The MSetup program can create or remove a connection to the desired Bentley product. Note: The link between the MSet and Bentley product is created by making a copy of the *mset_app.cfg* file in ...\\config\\appl directory of the given product.

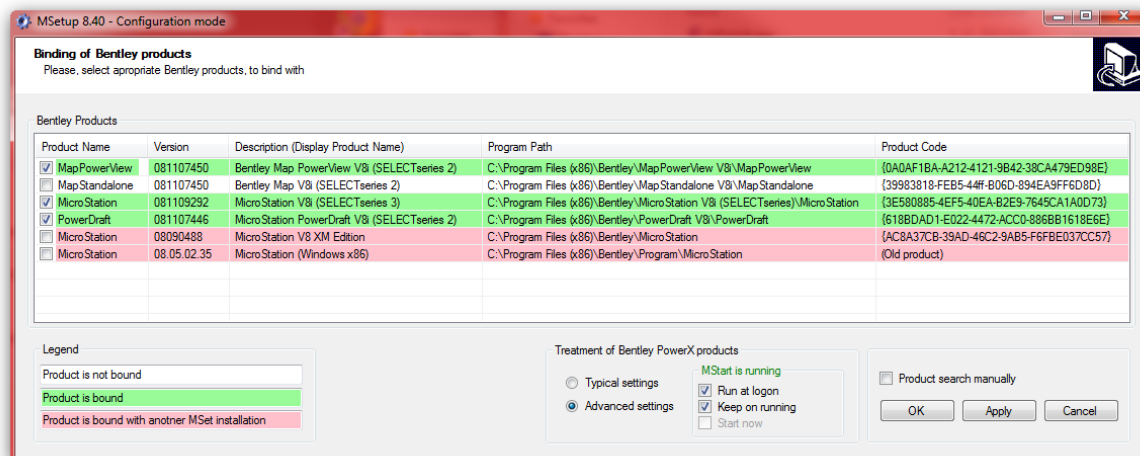


Fig. 19 Dialog of MSetup program in configuration mode

Uninstall

For uninstalling MSet use the standard uninstall program accessible via *Start > All Programs > MSet 8.40.xx > Uninstall MSet*. Before uninstalling it is recommended to save the license file. At the end of uninstall MStart configuration program starts in uninstall mode (no interaction with user) and removes files *mset_app.cfg* from the linked Bentley products directories ...\\config\\appl.

MSet variables

The MSet application introduces a few MicroStation configuration variables on the application level. In the case of necessity, is possible to modify them. Normally, it is recommended to change only the MSET_INSTR_DIR variable.

VARIABLE	MEANING
MSET_DGNLIB	Path to <i>menu_mset.dgnlib</i> file – add item MSet into MicroStation menu (V8.9+)
MSET_DIR	Main MSet directory
MSET_FORUM	Directory for MSet communication
MSET_INSTR_DIR	Instruction Directory/Directories
MSET_LANGUAGE	Language (en, sk-SK)
MSET_START_WAITING*	Maximum waiting time, which application mset.ma waits to start the mset.exe. Default value is not defined, but there is a possibility to increase the waiting time to a higher value than the standard 20 seconds.

Table 4 MSet configuration variables

Demonstration example Mset_demo.dgn

At the end of the install process it is possible to open MSet demonstration example DGN file. We recommend the first time to open and study it. Later, an example is accessible from the *Start > All Programs > MSet 8.40.xx > MSet_demo*.

Notations and Abbreviations

- **AS** – active scale
- **Application element** – Signum and Scale factor written in the DGN file
- **DTD** - Document Type Definition specified formal rules of the Instruction structure
- **Primary scale** – scale in which libraries, user lines etc. are made
- **PS** – primary scale
- **Scale factor** – actual scale divided by primary scale
- **SF** – Scale factor
- **Signum** – Name of a file without extension, transferred into capital letters
- **UAC** – is a technology and security infrastructure introduced in Microsoft's Windows Vista and Windows Server 2008 operating systems