



Conversion of chemical files
into spreadsheet compatible
file formats

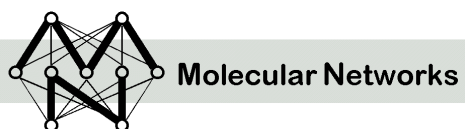
User Manual

Version 1.0

for program version 1.0 (or higher)

November 2003

Molecular Networks GmbH – Computerchemie
Nägelsbachstraße 25, 91052 Erlangen, Germany



Molecular Networks

Molecular Networks GmbH

Computerchemie

Nägelsbachstr. 25

91052 Erlangen

Germany

Phone: +49-(0)9131-815668

Fax: +49-(0)9131-815669

Email: info@mol-net.de

WWW: www.mol-net.de

This document is copyright © 2002 by Molecular Networks GmbH Computerchemie. All rights reserved. Except as permitted under the terms of the Software Licensing Agreement of Molecular Networks GmbH Computerchemie, no part of this publication may be reproduced or distributed in any form or by any means or stored in a database retrieval system without the prior written permission of Molecular Networks GmbH Computerchemie.

The software described in this document was originally designed and implemented for the CACTVS system by W. D. Ihlenfeldt. It is not part of the standard CACTVS toolkit distribution; it is furnished under a license and may be used and copied only in accordance with the terms of such license. For licensing this software please contact Molecular Networks GmbH, Nägelsbachstr. 25, 91052 Erlangen, Germany.

Product names and company names may be trademarks or registered trademarks of their respective owners, in the Federal Republic of Germany and other countries. All rights reserved.

Table of Contents

1.	General Information about MN.TABLE.....	5
2.	Installation	6
2.1.	Requirements.....	6
2.2.	Installation steps for UNIX operating systems (IRIX, Solaris, Linux)	6
2.3.	Installation steps for Windows operating systems (NT4/2000/XP).....	6
3.	Uninstallation.....	6
3.1.	Uninstallation steps for UNIX operating systems (IRIX, Solaris, Linux).....	6
3.2.	Uninstallation steps for Windows operating systems (NT4/2000/XP)	6
4.	Problems and Help!.....	7
5.	Release Notes	8
5.1.	Version 1.0.....	8
6.	Getting Started.....	9
6.1.	UNIX operating systems.....	9
6.2.	Windows operating systems.....	9
7.	Program Use	10
7.1.	Synopsis	10
7.2.	General program features.....	11
7.3.	Supported file formats for input files.....	13
7.4.	Program features for file handling.....	13
-	format cactvs/comma/dbase3/excel/html/htmlpage/semicolon/space/sql/tab	13
-	outfile <filename.extension>	13
-	count <n>	14
-	eoltype mac/pc/unix	14
-	imagedirectory dirname	14
-	imageurl urlprefix.....	15
-	feedback n.....	15
7.5.	Program features for setting the layout of the table	15
-	assign propertypairlist	15
-	colnames 0/1	15
-	flatten 0/1	16
-	imageborder n	16
-	footer text.....	16
-	header text.....	17
-	mapping propertypairlist.....	17
-	sort sortspec	17
7.6.	Program features for setting the layout of the embedded images	18
-	imgheight <pixels>.....	18
-	imgwidth <pixels>	19
-	imgbgcolor <colorname>/transparent	19
-	imgborder <pixels>	19
-	imgheader <text>	20

-imgheadercolor <colorname>	20
-imgheaderproperty <propertyname>	21
-imgfooter <text>	22
-imgfootercolor <colorname>	22
-imgfooterproperty <propertyname>	22
-imglogfile <filename>	23
-imglogoscale <factor>	24
7.7. Program features for setting the layout of molecules in the embedded images	25
-imgasymbol symbol/xsymbol/label/index/box/compact	25
-imghsymbol none/special/all	25
-imgatomcolor <colorname>/type	26
-imghcolor <colorname>	27
-imgbondcolor <colorname>/split	27
-imgsymbolfontsize <pts>	28
-imgannotationfontsize <pts>	28
-imgbonds <n>	29
-imgshowcharge <0/1>	29
-imgshowisotope <0/1>	30
-imgshowradical <0/1>	30
-imgshowstereo <0/1>	31
-imgshowstereoh <0/1>	31
-imgwedges 0/1	32
-imgdashes 0/1	32
-imgusehatch 0/1	33
7.8. Program features of general usage	33
-version	33
8. Extended Features only available for the UNIX operating systems	33
9. Application examples	35
9.1. Table of 20 natural amino acids	35
10. Frequently Asked Questions (FAQ)	36
11. Error messages	36
12. Known problems and limitations	36
13. Technical Support	37
The MN.TABLE Web Site	37
Reporting Problems	37
Updates	37
Contact information	37
14. Report Form	38
15. Index	39

1. General Information about MN.TABLE

MN.TABLE is a powerful tool for building tables in various formats from structure data. Besides data, images may be generated and embedded into the tables if the output format supports it.

The program MN.TABLE

- supports several options for setting the table layout
- processes datasets with 99.9% conversion rate
- handles datasets of hundreds of thousands of chemical structures
- supports over 5 output file formats (Excel, dBASEIII, HTML, etc.)

2. Installation

2.1. Requirements

MN.TABLE is available for common UNIX platforms (x86 Linux, Sun Solaris, SGI IRIX). It is also available for Microsoft Windows NT4/2000/XP.

The program is running in a batch mode.

2.2. Installation steps for UNIX operating systems (IRIX, Solaris, Linux)

- 1.) Create a subdirectory, e.g., `mn_table`
(for system administrators when installing software locally, e.g. `/usr/local/bin/mn_table`).
- 2.) Copy the file `mn_table_<version>.<os>.gz` to the subdirectory `mn_table`
- 3.) Unpack the distribution by executing the `gunzip` command:
`gunzip mn_table_<version>.<os>.gz`
- 4.) Rename the file `mn_table_<version>.<os>` to `mn_table`.
Please note: `mn_table_<version>.<os>` is a binary file.
- 5.) Add the `mn_table` subdirectory name to the environment variable `PATH` in your `.login` or `.cshrc` files (`.profile` or `.bashrc`).

Launch MN.TABLE with the command

```
mn_table -version or  
/usr/local/bin/mn_table/mn_table -version
```

2.3. Installation steps for Windows operating systems (NT4/2000/XP)

Although administrator privileges are not necessary, we recommend logging in as administrator. Double-click on the executable setup program and follow the instructions on the screen.

After successful installation there is no need to reboot your PC.

3. Uninstallation

3.1. Uninstallation steps for UNIX operating systems (IRIX, Solaris, Linux)

Log in as root and delete the file `mn_table` in your installation directory carefully (default path during installation was `/usr/local/bin/mn_table/`).

3.2. Uninstallation steps for Windows operating systems (NT4/2000/XP)

Log in as administrator, launch the uninstaller and follow the on-screen instructions.

4. Problems and Help!

If you have any difficulties with the installation of MN.TABLE or if any problems occur while running MN.TABLE, please send all your inquiries to the following address:

Molecular Networks GmbH Computerchemie
Nägelsbachstr. 25
91052 Erlangen
Germany,

or contact us by email
or by fax

support@mol-net.de
+49-(0)9131 - 81 56 69.

Please mention the program version of MN.TABLE (`mn_table -version`), include your input file, and the output file on an MS/DOS diskette (3½") or send it to us by email. These files will help us to analyze the problem; if your system displays any error messages, please add them to your report.

You can also use the report form at the end of this manual.

5. Release Notes

5.1. Version 1.0

First release of MN.TABLE

6. Getting Started

6.1. UNIX operating systems

The example file `six_examples.sdf` submitted with the distribution contains the structure information of six molecules in SDF format. Copy this example file into your working directory and type the following command:

```
mn_table -outfile six_examples.xls -col1 Image -col2 "E_NAME  
Name" six_examples.sdf
```

MN.TABLE now creates the output file named `six_examples.xls`. Figure 1 shows the output file.

6.2. Windows operating systems

The example file `six_examples.sdf` submitted with the distribution contains the structure information of six molecules in SDF format. Copy this example file into your working directory and open a DOS shell. Change the working directory to the directory where you installed `mn_table` by using the `cd` command then type the following command:

```
mn_table -outfile six_examples.xls -col1 Image -col2 "E_NAME  
Name" six_examples.sdf
```

MN.TABLE now creates the output file named `six_examples.xls`. Figure 1 shows the output file.

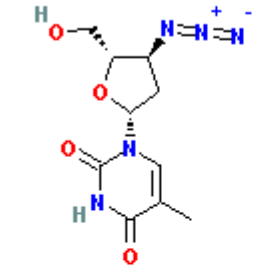
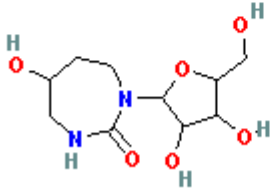
	A	B	C	D
1	Image	Name		
2		AZT		
3		NSC323230 Coaracine (racemic)		

Figure 1: First two records of the input data set converted into a table format with embedded 2D images.

7. Program Use

7.1. Synopsis

The general synopsis for using MN.TABLE is:

```
mn_table [ -option(s) ] [ infile ]
```

An overview of the various options is given in Table 1 and in more detail in the following chapter. Infile is the input file name. If no file names are given, the program reads from standard input.

[-assign propertypairlist]	[-imgfooterproperty propertyname]
[-colnames 0/1]	[-imgheader text]
[-count nrecs]	[-imgheadercolor colorname]
[-eoltype mac/pc/unix]	[-imgheaderproperty propertyname]
[-feedback n]	[-imghcolor colorname]
[-flatten 0/1]	[-imgheight pixels]
[-footer text]	[-imghsymbol none/special/all]
[-format cactvs/comma/dbase3/excel/ html/htmlpage/semicolon/space/sqp/tab]	[-imglogfile filename]
[-h] or [-help]	[-imglogoscale factor]
[-header text]	[-imgshowcharge 0/1]
[-imageborder n]	[-imgshowisotope 0/1]
[-imagedirectory dirname]	[-imgshowradical 0/1]
[-imageurl urlprefix]	[-imgshowstereo 0/1]
[-imgannotationfontsize points]	[-imgshowstereooh 0/1]
[-imgasymbol symbol/xsymbol/label/index/ box/compact]	[-imgsymbolfontsize points]
[-imgatomcolor colorname/type]	[-imgusehatch 0/1]
[-imgbgcolor colorname/transparent]	[-imgwidth pixels]
[-imgbondcolor colorname/split]	[-imgwedges 0/1]
[-imgbonds n]	[-mapping propertypairlist]
[-imgborder pixels]	[-outfile filename]
[-imgdashes 0/1]	[-sort sortspec]
[-imgfooter text]	[-version]
[-imgfootercolor colorname]	

Table 1: Overview of all options supported by MN.TABLE.

7.2. General program features

The program MN.TABLE reads structures from one file specified on the command line, build a table with data in memory, and writes it out in the selected format to the file name specified by the **-outfile** option.

The program detects and understands two standard exchange formats, MDL MOL and SDF. The desired columns in the table are specified by one or more **-coln** options. *n* is a number in the range between 1 and 255. The numerically ordered sequence of the actually used numbers determines the order of the columns. So, for example, if **-col3**, **-col5**, and **-col1** are specified in that order, a table with three columns is generated, with the column defined by **-col1** inserted as first table column, **-col3** as second, and **-col5** as third. No empty filler columns are generated. The order of the **-col** options does not matter, only their number components. A **-coln** option has one to three parameter fields. If more than one field is used, it must be specified as a proper Tcl list, for example as **'-col1 "E_WEIGHT {Molecular Weight} {fgcolor red format {bold right}}"**'. This construct defines a column which stores the property E_WEIGHT, is titled "Molecular Weight", and is written as red, bold, right-aligned text.

The **first** and mandatory **field** of a **-coln** option is either a property name, or one of the three magic words Image, Blank, or Record. A property name can be either specified in the standard CACTVS syntax (such as E_WEIGHT for the ensemble molecular weight), or in its original name as it is stored in a data file (for example, the name used in the data section of an SD-file). The name lookup is case-sensitive. Note that all property names which use identifiers gathered from the file must be already defined in the first record of the first input file. The three magic column property names Image, Blank, and Record insert a structure drawing in a suitable format, a blank column, or the file record number, respectively. These magic columns can be named and assigned display attributes just like normal property data columns.

The **second field** is the table column name. If it is not specified, or empty, the name of the column is the same as the name of the property or magic name.

The **third** optional parameter **field** allows the setting of display attributes for the column. This is a list of attribute-value pairs. The following attributes are currently officially supported:

- fgcolor - foreground color, specified as a color name in the X11 color database, or in the #rrggbb format
- bgcolor - background color, specified as a color name in the X11 color database, or in the #rrggbb format
- font - a font family name in short notation, i.e. Arial, Symbol, Courier
- fontsize - the font size in points (may be a floating point number)
- comment - a free-text comment
- description - a free-text column description
- precision - the number of digits after the decimal point for floating-point values
- width - the column width in characters
- format - a list of keywords from the set of:
 - none - no options, default

- left - explicitly left-aligned data
- center - data is centered
- right - data is right-aligned
- bold - use bold font for display
- highlight - use highlight format for display
- histogram - display as histogram bars instead of numeric values
- border - surround by border
- padding - use extra cell padding
- expand - stretch to 100% width
- top - data is vertically aligned on top
- middle - data is vertically centered in the middle
- bottom - data is vertically aligned on bottom
- italic - use italic font
- underline - text is underlined

While all these attributes can be freely set regardless of the selected table output format, their effect and whether they have an effect at all depends on the table output format. A simple comma-separated file simply cannot encode any formatting informations, and thus these attributes are ignored. The formats which support the highest percentage of the various formatting flags are the native CACTVS binary table format (this is the only format which fully preserves all information), the Excel BIFF (.xls) format and HTML.

One or more of the table columns may hold a structure depiction. In case of Excel output, the image is a resizable WMF (Windows Metafile) vector drawing. When the table is written as HTML, HTML page, SQL command file or CACTVS native table file, the structure is embedded as a GIF image. The display style of images is modified by the various `img*` options of this program.

As an alternative to displaying numerical data as simple numbers, the Excel output format supports also the generation of histograms. Selected data columns with numerical values may be overlaid by a resizable histogram (using the `fgcolor`, `bgcolor`, `width` attributes) which is scaled between 0% and 100% using the minimum and maximum values of all rows in that data column.

The theoretical maximum size of tables which can be handled by the software is $2^{32}-1$ rows and columns. Since the structures read from the file are not kept in memory after their data content has been copied to the table cells, the memory requirements even for assembling large tables are not excessive. Note however that some table file formats have rather low limitations on the maximum number of rows and columns - for example, MS Excel will not allow more than 65534 rows and 256 columns.

This software supports NULL (missing) values in property cells and will try to preserve this information when writing output files.

7.3. Supported file formats for input files

The program will automatically detect the file format of the input file. Two standard exchange formats are supported. Thus, there is no need for a parameter specifying the input format. The supported file formats are listed in the table below.

Full Format Name	Default Input-Extension	Read	Comment
MDL Molfile	mol	Yes	
MDL SDF	sdf	Yes	

Table 2: Overview of the supported input file formats

7.4. Program features for file handling

-format cactvs/comma/dbase3/excel/html/htmlpage/semicolon/space/sql/tab

Select a format for the output table file. If this option is not specified, an attempt is made to identify the format from the suffix of the output file name, such as .csv or .xls.

The number of formatting attributes supported by the various formats and also the acceptable data types of columns are dependent on the format. Only the excel (embedded EMF image), sql (BLOB column with GIF data) and cactvs/html/htmlpage (GIF image) formats support embedded images in a form which goes beyond a simple file reference. Excel is the only format which currently supports histograms. The difference between the html and htmlpage formats is that the former format is just an HTML <TABLE> section, which is intended to be integrated as a building block into an HTML page for display, while the latter is a simple, but complete HTML page. The semicolon, comma, space and tab formats are variants of simple text dumps. If a data value contains the separator character, it will be protected by putting the value into quotes. In these formats, NULL values are represented by empty entries.

Default value:

This parameter has no default value.

Example:

Generating a Microsoft Excel file of the input structure stored in six_examples.sdf:

```
mn_table -format excel -col1 Image -col2 "E_NAME Name"  
./examples/six_examples.sdf
```

-outfile <filename.extension>

The parameter **-outfile** defines the name of the output file. MN.TABLE automatically recognizes the desired output format, thus in most cases it is not necessary to specify the output format.

Default value:

Parameter without a default value

Example:

Generating a Microsoft Excel file named 6examples.xls:

```
mn_table -outfile 6examples.xls -col1 Image -col2 "E_NAME Name"
./examples/six_examples.sdf
```

-count <n>

The parameter **-count** is able to limit the amount of processed structures. A maximum of <n> records from the input file(s) is processed. Especially while handling large input files with hundreds of thousands of structures this parameter is useful to convert only the first hundred structures, if you would like to test other parameters and would not like to wait until the entire file was processed.

Default value:

No limitation (value= ∞)

Example:

Generating a Microsoft Excel file of the first four alkanes (from methan to butane):

```
mn_table -outfile alkanes1_4.xls -col1 "E_NAME Name" -count 4
./examples/alkanes1_12.sdf
```

Remark:

The **-offset** parameter can be used to position the file before the count begins and thus convert only a region of a large file.

-eoltype mac/pc/unix

This parameter is used for choosing between different end-of-line characters.

NL (Unix)

CR (Macs)

CR/NL (PC)

Default value:

Example:

Generating a table file with end-of-line characters for Microsoft Windows operating systems:

```
mn_table -outfile alkanes1_12.txt -col1 "E_NAME Name" -eoltype
pc ./examples/alkanes1_12.sdf
```

Remarks:

This option is ignored if the output file format is binary such as the excel format.

-imagedirectory dirname

The name of a directory where external images are stored for output formats (such as *html*, *htmlpage*, etc.) which cannot store images internally. The directory is created if necessary. If no directory is specified, external images are put into the current directory.

Example:

Generating a HTML table with embedded structures saved in the directory pictures:

```
mn_table -outfile alkanes1_12.html -imagedirectory pictures
-col1 Image -col2 "E_NAME Name" ./examples/alkanes1_12.sdf
```

-imageurl urlprefix

If this option is specified, the html-based output formats (*html* and *htmlpage*) will use its value as prefix to references to embedded structure GIF images. Note that a possible relative image directory (**-imagedirectory**) is already automatically added to the reference path, so it is not necessary to repeat it with this option.

Example:

```
Generating a HTML table with redirected URL address for the generated pictures:  
mn_table -outfile alkanes1_12.html -imageurl http://www.mol-  
net.de/test -col1 Image -col2 "E_NAME Name"  
./examples/alkanes1_12.sdf
```

-feedback n

If the parameter **-feedback** is set, a dot is printed on the standard error channel for every n^{th} completed entry.

Default value:

It is not active by default.

Example:

```
Generating a Microsoft Excel file with feedback information during processing:  
mn_table -outfile maybridge.xls -col1 Image -col2 "E_NAME Name"  
-feedback 100 ./examples/maybridge.sdf
```

7.5. Program features for setting the layout of the table

-assign propertypairlist

Assign one or more properties read from the input files to other properties, which are, for example, used in the table columns. Only properties of the same object class can be assigned. The system will attempt to convert the data type, if the property data types of source and destination property are not the same. Property names can be given either in CACTVS syntax or with the name they appeared in the original file. Case is important.

Default value:

It is not active by default.

Example:

```
Generating a Microsoft Excel file with assigned property information:  
mn_table -outfile alkanes.xls -col1 Image -col2 "E_FORMULA  
Formula" -assign "E_NAME E_FORMULA" ./examples/alkanes1_12.sdf
```

-colnames 0/1

Choose whether the column names should be included in the output table or not. This flag is ignored in formats which differentiate between cell content and column names, such as the native CACTVS table format or SQL table definitions.

Default value:

By default the flag is set.

Example:

Generating a Microsoft Excel file without the column names:

```
mn_table -outfile alkanes.xls -col1 Image -col2 "E_NAME Formula"
-colnames 0 ./examples/alkanes1_12.sdf
```

-flatten 0/1

The **-flatten** option, which is set by default except when the output format is the native CACTVS table format, temporarily expands multi-element columns into multiple single-value columns. The columns usually have data types which can be represented in export formats. The column names of the temporary columns are set to *colname(0)*, *colname(1)*, etc. If multi-value cells are not flattened, they are output as a string in Tcl list notation.

Default value:

This option is set by default.

Example:

Generating a Microsoft Excel file with flattened multi-field data cells:

```
mn_table -outfile alkanes.xls -col1 Image -col2 "A_XY 2D"
-flatten 1 ./examples/alkanes1_12.sdf
```

Remark:

By using this option the table file now contains two columns "2D(x)" and "2D(y)".

-imageborder n

The option **-imageborder** specifies the size of a border around an embedded image. This option is only used for HTML-style and Excel output. For HTML, it is the thickness of the border in pixels around the GIF image on the HTML page (the BORDER=n field in the tag). For Excel, it is the distance between embedded WMF structure drawings or histogram bars and the cell border. In this case, the unit is 1/255 of the cell height and width. Using a non-zero border in Excel files may prevent problems when sorting image cells. They may not move in sync with other sort columns if their size is very close to the cell size. On the other hand, values above 2-3, depending on the image size, prevent resizing of the drawings when the size of the cell is changed. Depending on the application, this can be a feature or a problem.

Default value:

The default value is 0.

Example:

Generating a Microsoft Excel file with borders around the embedded images:

```
mn_table -outfile alkanes.xls -col1 Image -imageborder 2
./examples/alkanes1_12.sdf
```

-footer text

Define a free-form text for the table footer shown in the lowest part of the table page.

Default value:

It is not active by default.

Example:

Generating a Microsoft Excel file with a footer line:

```
mn_table -outfile alkanes.xls -col1 Image -col2 "E_NAME Formula"
-footer "Molecular Networks GmbH" ./examples/alkanes1_12.sdf
```

-header text

Define a free-form text for the table header shown in the uppermost part of the table page.

Default value:

It is not active by default.

Example:

Generating a Microsoft Excel file with a header line:

```
mn_table -outfile alkanes.xls -col1 Image -col2 "E_NAME Formula"
-header "Molecular Networks GmbH" ./examples/alkanes1_12.sdf
```

-mapping propertypairlist

This option is a method for associating data fields in input files with predefined CACTVS properties. For example, the specification „*E_NAME Catalogname E_WEIGHT Molweight*“, will associate the data field *Catalogname* with the standard property *E_NAME* and the data field *Molweight* with *E_WEIGHT*. This kind of mapping is especially useful for correctly setting the data type of yet unspecified properties.

Default value:

It is not active by default.

Example:

Generating a Microsoft Excel file mapping the property *E_NAME* to "Formula":

```
mn_table -outfile alkanes.xls -col1 Image -col2 Formula -mapping
"E_NAME Formula" ./examples/alkanes1_12.sdf
```

-sort sortspec

By default, the order of the rows in the table is the same as the structures read from the input file.

With this option, the sequence of the rows may be changed by sorting the table according to the values in one or more columns. The sort specification is a list of column names or name/direction pairs. Column names to the left have higher priority, column names to the right are only used to break ties. The default sort order is *up*, meaning that rows with a lower value of the cells in the sort column show up first. By combining a column name with either *up* or *down*, the sort order of each column may be specified individually. So, a sort specification such as *-sort "{Weight down} {Record up}"* would arrange the rows with the structures of high molecular weight first, with a preference for earlier file records in case of ties. The sort does take the data type of the columns into account, so truly numerical values are sorted differently than the same data encoded as strings if the strings are longer than a single character.

Default value:

It is not active by default.

Example:

Generating a Microsoft Excel file sorted by the molecular weight in decreasing order:

```
mn_table -outfile alkanes.xls -col1 Image -col2 "E_WEIGHT  
Weight" -sort "{Weight down}" ./examples/alkanes1_12.sdf
```

Remark:

Note that it is not always possible to determine the value type of property data when reading from file, and thus numerical data may be unintentionally read as strings. The value types of properties can be unambiguously defined by providing proper property definition files.

7.6. Program features for setting the layout of the embedded images

The following figure shows an overview of all options used for specifying the layout of the images.

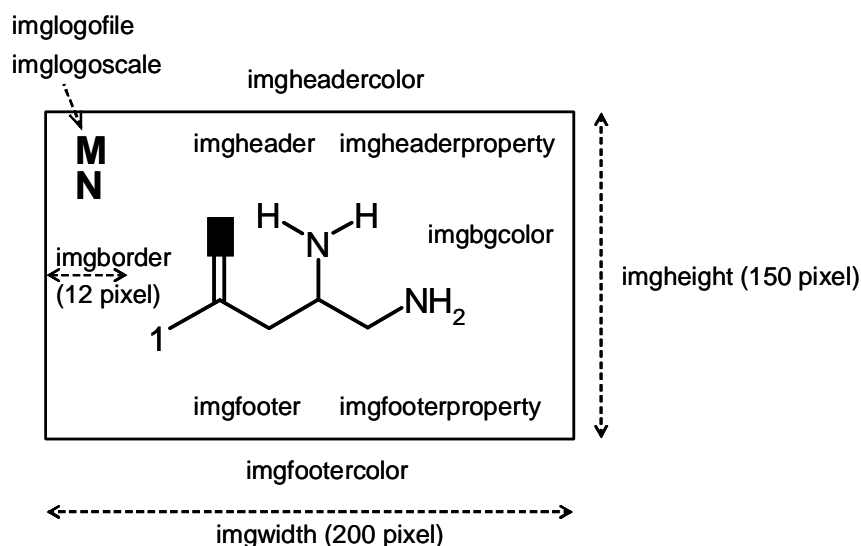


Figure 2: Overview of all options used for setting the layout of the embedded images; in parenthesis the default values are given.

-imgheight <pixels>

The parameter **-imgheight** sets the height of the embedded structure drawing in pixels.

Default value:

The default value is 150 pixels.

Example:

Generating a table file with embedded images having a height of 300 pixels:

```
mn_table -outfile alkanes.xls -col1 Image -imgheight 300  
./examples/alkanes1_12.sdf
```

Remarks:

Use the option **-imgwidth** to set the width of the image.

-imgwidth <pixels>

The parameter **-imgwidth** sets the width of the embedded structure drawing in pixels.

Default value:

The default value is 200 pixels.

Example:

Generating a table file with embedded images having a width of 300 pixels:

```
mn_table -outfile alkanes.xls -coll Image -imgwidth 300
./examples/alkanes1_12.sdf
```

Remarks:

Use the option **-imgheight** to set the height of the image.

-imgbgcolor <colorname>/transparent

The parameter **-imgbgcolor** determines the background color of the embedded images. If the special value transparent is chosen, the background is transparent.

Default value:

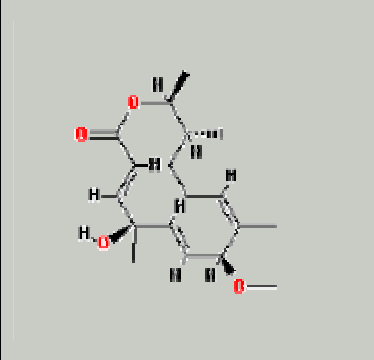
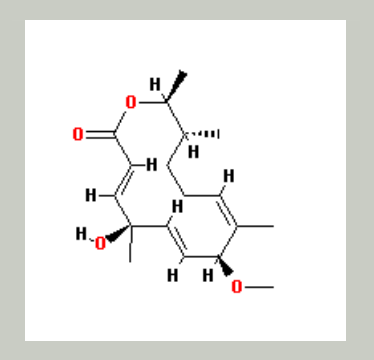
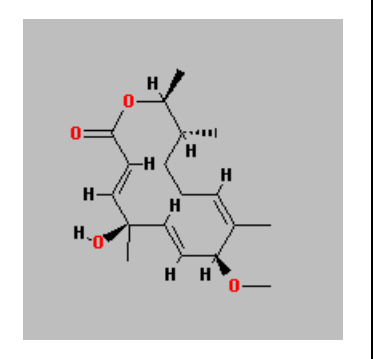
This option is deactivated by default.

Example:

Generating a table file with embedded images having a transparent background:

```
mn_table -outfile six_examples.xls -coll Image -imgbgcolor
transparent ./examples/six_examples.sdf
```

Pictures:

		
background transparent (the background color of the cell was set to grey)	background white (the background color of the cell was set to grey)	background grey (the background color of the cell was set to white)

-imgborder <pixels>

The parameter **-imgborder** determines the width of the border from the center of the outmost atoms of a structure which fits tightly into the display area to the outer border of the image. Note that atoms with plotted symbols require a few pixels in all directions around the atom center, so setting this parameter to very small values should be avoided.

Default value:

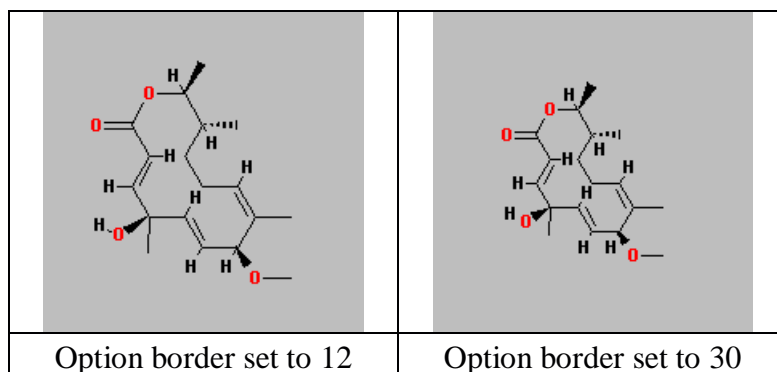
The default is 12 pixels.

Example:

Generating a table file with embedded images having a larger border:

```
mn_table -outfile six_examples.xls -col1 Image -imgborder 30
./examples/six_examples.sdf
```

Picture:



-imgheader <text>

This is a free text which is centered on the top of every image. Compound data can be automatically inserted into the header with the **-imgheaderproperty** option.

Default value:

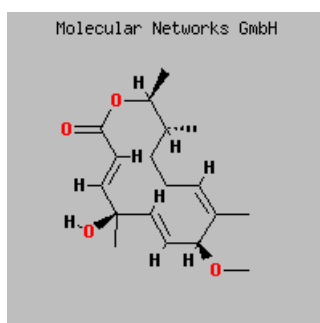
This option is deactivated by default.

Example:

Generating a table file with embedded images showing a header text:

```
mn_table -outfile six_examples.xls -col1 Image -imgheader
"Molecular Networks GmbH" ./examples/six_examples.sdf
```

Picture:



-imgheadercolor <colorname>

The option **-imgheadercolor** selects the color for the text written with the **-imgheader** option.

Default value:

The default color is black.

Example:

Generating a table file with embedded images showing the header text in red color:

```
mn_table -outfile six_examples.xls -col1 Image -imgheader
"Molecular Networks GmbH" -imgheadercolor red
./examples/six_examples.sdf
```

-imgheaderproperty <propertyname>

If no explicit header is set with the **-imgheader** option, this option can be used to transfer the data associated with a defined property into the header field. Useful properties are for example E_NAME, E_FORMULA, E_WEIGHT, or E_SMILES. If the data is not yet present, but a method is available to compute the data from available information, it is automatically invoked. Currently, the header property must be of the ensemble property attachment type, and subfield extraction is not yet supported.

Default value:

This option is deactivated by default.

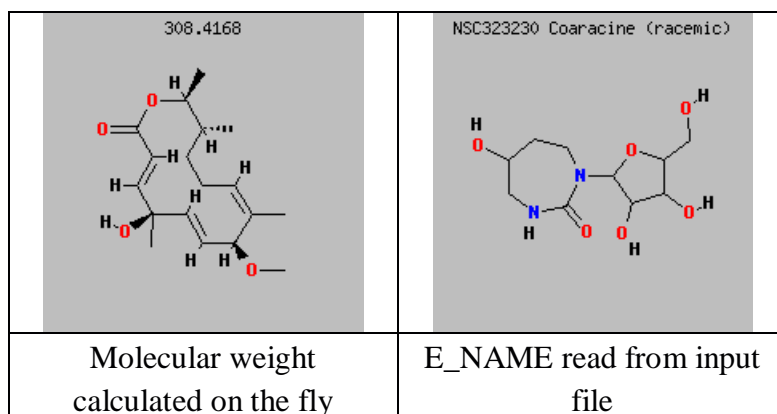
Example:

Generating a table file with embedded images showing the molecular weight or the name of the compound in the header line:

```
mn_table -outfile six_examples.xls -coll Image

```

Picture:



Remarks:

Note that only properties associated with a molecular ensemble can be used, but no atom, molecule, and bond properties. In case of externally defined properties (such as SD file data fields), the corresponding internally defined name should be used. If MN.TABLE knows about a computational method to derive the requested information from the data it read from the input file, it is not required that the data is already present. This means that you can use E_FORMULA with any file, but E_IDENT only with files which contain data which is mapped to this property, since it is not computable. Table 2 shows a list of properties which can be calculated with MN.TABLE:

Property name	Description	UNIX	Windows
E_WEIGHT	molecular weight	yes	yes
E_FORMULA	molecular formula	yes	yes
E_NHDONORS	amount of H-donors	yes	no
E_NHACCEPTORS	amount of H-acceptors	yes	no
E_NROTBONDS	amount of rotatable bonds	yes	no
E_INPUTDATE	date the ensemble was created or read	yes	no

E_EXACT_MASS	exact molecular weight based on the monoisotopic mass	yes	no
E_COMPLEXITY	complexity value	yes	no
E_SMILES	SMILES string of the compound	yes	yes
E_HASH	64 bit hash code	yes	yes
E_HASHSY	64 bit hash code with stereo information	yes	yes

Table 2: Some properties which can be calculated by MN.TABLE.

-imgfooter <text>

This is a free text which is centered on the bottom of every image. Compound data can be automatically inserted into the footer with the **-imgfooterproperty** option.

Default value:

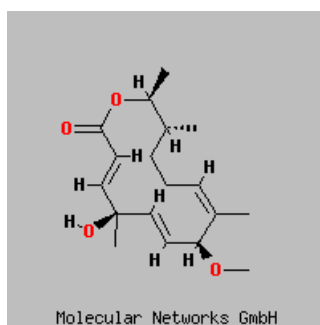
This option is deactivated by default.

Example:

Generating a table file with embedded images showing a footer text:

```
mn_table -outfile six_examples.xls -coll Image -imgfooter
"Molecular Networks GmbH" ./examples/six_examples.sdf
```

Picture:



-imgfootercolor <colorname>

The option **-imgfootercolor** selects the color for the text written with the **-imgfooter** option.

Default value:

The default color is black.

Example:

Generating a table file with embedded images showing the footer text in red color:

```
mn_table -outfile six_examples.xls -coll Image -imgfooter
"Molecular Networks GmbH" -imgfootercolor red
./examples/six_examples.sdf
```

-imgfooterproperty <propertyname>

If no explicit footer is set with the **-imgfooter** option, this option can be used to transfer the data associated with a defined property into the footer field. Useful properties are for example E_NAME, E_FORMULA, E_WEIGHT, or E_SMILES. If the data is not yet present, but a

method is available to compute the data from available information, it is automatically invoked. Currently, the footer property must be of the ensemble property attachment type, and subfield extraction is not yet supported.

Default value:

This option is deactivated by default.

Example:

Generating a table file with embedded images showing the molecular weight of the compound in the footer line:

```
mn_table -outfile six_examples.xls -coll Image
-imgfooterproperty E_WEIGHT ./examples/six_examples.sdf
```

Remarks:

Note that only properties associated with a molecular ensemble can be used, but no atom, molecule, and bond properties. In case of externally defined properties (such as SD file data fields), the corresponding internally defined name should be used. If MN.TABLE knows about a computational method to derive the requested information from the data it read from the input file, it is not required that the data is already present. This means that you can use E_FORMULA with any file, but E_IDENT only with files which contain data which is mapped to this property, since it is not computable. Table 3 shows a list of properties which can be calculated with MN.TABLE:

Property name	Description	UNIX	Windows
E_WEIGHT	molecular weight	yes	yes
E_FORMULA	molecular formula	yes	yes
E_NHDONORS	amount of H-donors	yes	no
E_NHACCEPTORS	amount of H-acceptors	yes	no
E_NROTBONDS	amount of rotatable bonds	yes	no
E_INPUTDATE	date the ensemble was created or read	yes	no
E_EXACT_MASS	exact molecular weight based on the monoisotopic mass	yes	no
E_COMPLEXITY	complexity value	yes	no
E_SMILES	SMILES string of the compound	yes	yes
E_HASH	64 bit hash code	yes	yes
E_HASHSY	64 bit hash code with stereo information	yes	yes

Table 3: Some properties which can be calculated by MN.TABLE.

-imglogfile <filename>

If this is not an empty parameter, an attempt is made to read the file as GIF or PNG logo. If the image could be read, it will be inserted into the upper left corner of all structure images as a logo picture. The logo image can be scaled by the **-imglogoscale** option.

Default value:

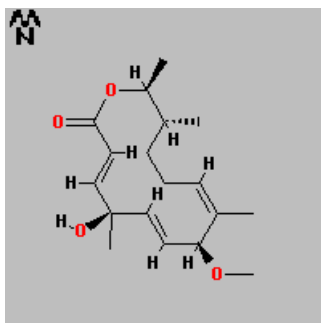
This option is deactivated by default.

Example:

Generating a table file with embedded images showing a logo picture:

```
mn_table -outfile six_examples.xls -coll Image -imglogfile
./examples/mn_logo.gif ./examples/six_examples.sdf
```

Picture:



-imglogoscale <factor>

This factor can be used to resize a logo file specified by the **-imglogfile** option. The scale factor is a floating point number.

Default value:

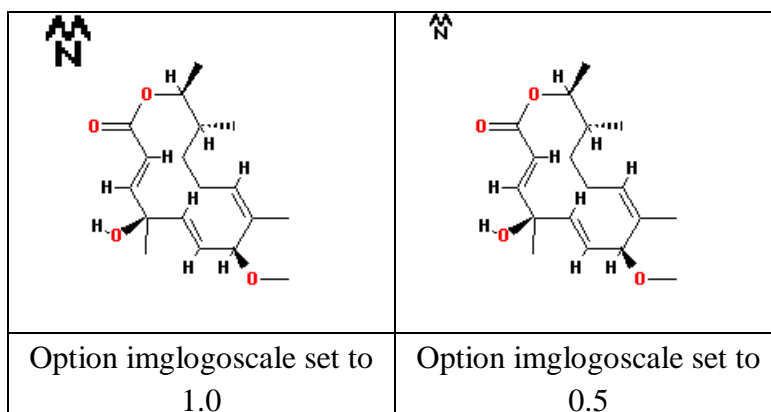
The default scaling factor is 1.0.

Example:

Generating a table file with embedded images showing a logo picture scaled to half of the size:

```
mn_table -outfile six_examples.xls -coll Image -imglogfile
./examples/mn_logo.gif -imglogoscale 0.5
./examples/six_examples.sdf
```

Picture:



Remark:

Note that this option should not be used on a regular base - a cleanly rendered logo image in the intended final resolution generally has a better graphical quality than a rescaled logo.

7.7. Program features for setting the layout of molecules in the embedded images

-imgasymbol symbol/xsymbol/label/index/box/compact

The option **-imgasymbol** sets the type of symbol to print. For normal atoms, the display types **symbol** and **xsymbol** are equivalent, but **xsymbol** will produce a more detailed text for certain types of query atoms, such as atom lists. With the **label** style, atom labels replace the atomic symbols. These are either taken from the input file (if the files store this information) or correspond otherwise to the internal atom ordering, which is always the same as in the file. **Index** displays the index of the atom in the atom list, starting with one. In the **box** style, hetero atoms are depicted as small rectangles. The **compact** mode displays hydrogen atoms on hetero atoms and other special hydrogen atoms as a common symbol, such as OH or NH₂. Subscripting of hydrogen count numbers only works with Unicode fonts.

Default value:

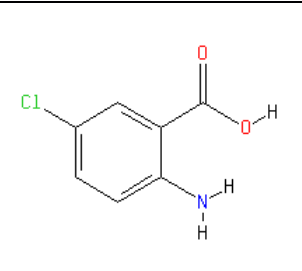
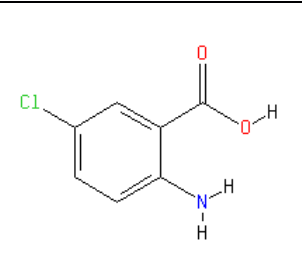
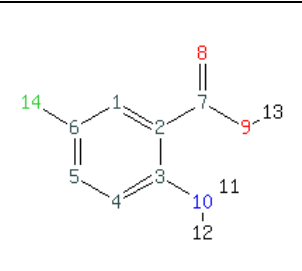
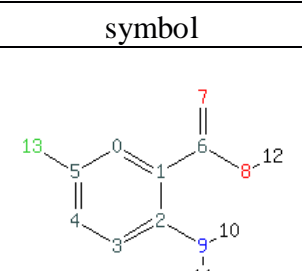
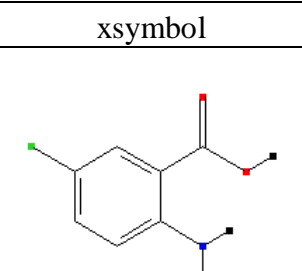
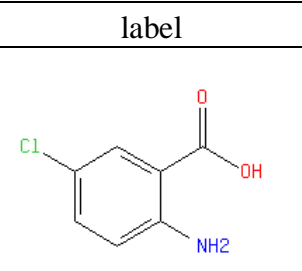
By default this option is set to symbol.

Example:

Generating a table file with embedded images showing the atom symbols of each structure in compact mode:

```
mn_table -outfile six_examples.xls -coll Image -imgasymbol
compact ./examples/six_examples.sdf
```

Picture:

		
symbol	xsymbol	label
		
index	box	compact

-imghsymbol none/special/all

The parameter **-imghsymbol** controls the rendering of the hydrogen atoms. By default (mode *special*) only hydrogen atoms which are traditionally plotted are printed, such as hydrogen atoms on hetero atoms and aldehydes. The mode *all* will display all hydrogen atoms, whereas mode *none* suppresses them all. In contrast to carbon atoms, the bonds to suppressed hydrogen atoms also vanish.

Default value:

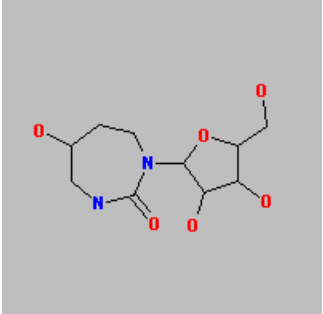
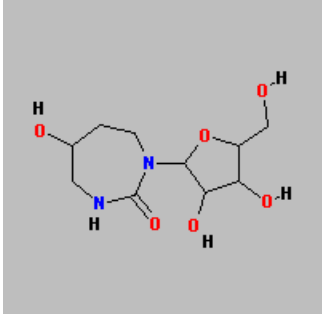
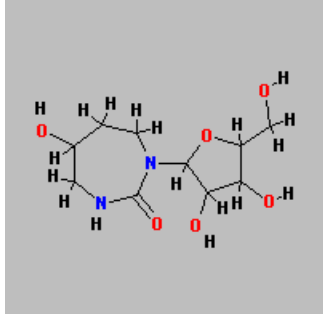
special (plot special hydrogen atoms)

Example:

Generating a table file with embedded images showing no hydrogen atoms:

```
mn_table -outfile six_examples.xls -coll Image -imghsymbol none
./examples/six_examples.sdf
```

Picture:

		
Option imghsymbol set to none	Option imghsymbol set to special	Option imghsymbol set to all

-imgatomcolor <colorname>/type

Select a global color for all atom symbols. If the default color **type** is chosen, the color of atom symbols is determined individually. If the input data contains color information, it is used. Otherwise, a standard element-specific coloring scheme is applied. For hydrogen atoms, the hydrogen color specified with the **-imghcolor** option will override both a global atom color and an individual atom color, if it is not an empty string.

Default value:

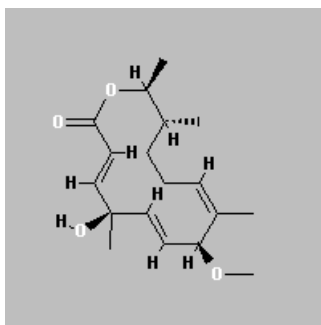
By default this option is set to “type”.

Example:

Generating a table file with embedded images showing white colored atom symbols for non-hydrogen atoms:

```
mn_table -outfile six_examples.xls -coll Image -imgatomcolor
white ./examples/six_examples.sdf
```

Picture:



-imghcolor <colorname>

The option **-imghcolor** specifies an override color for hydrogen atoms. If the color name is not an empty string, it overrides both the global atom color, and an individual hydrogen color taken from file or the standard element color table.

Default value:

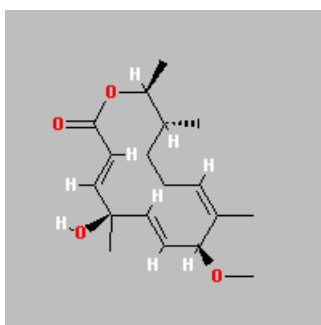
This flag is deactivated by default.

Example:

Generating a table file with embedded images showing white colored atom symbols for hydrogen atoms:

```
mn_table -outfile six_examples.xls -col1 Image -imghcolor white
./examples/six_examples.sdf
```

Picture:



-imgbondcolor <colorname>/split

The parameter **-imgbondcolor** allows the selection of a global color for bond lines. In contrast to atom colors, individual bond colors are currently not supported. The special value *split* splits the bond into two halves. Each half bond is colored in the same color as its associated atom symbol, with the exception of carbon atoms. In case of half bonds to carbon atoms, the split color is either black or white, automatically selected according to the image background.

Default value:

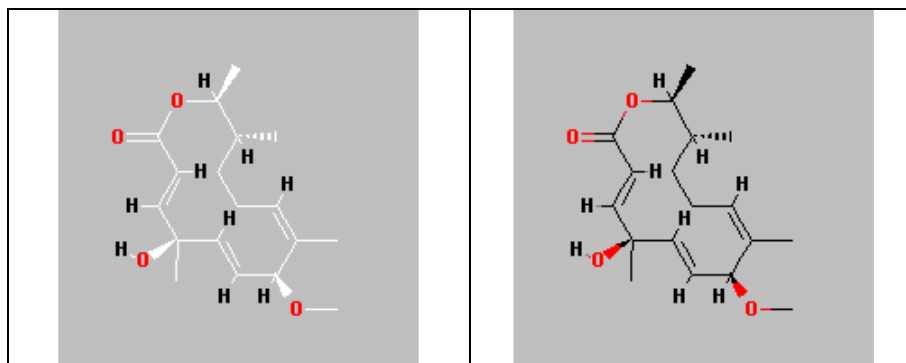
By default, the color of bonds is either set to black or white.

Example:

Generating a table file with embedded images showing white colored bonds:

```
mn_table -outfile six_examples.xls -col1 Image -imgbondcolor
white ./examples/six_examples.sdf
```

Picture:



Option imgbondcolor set to white	Option imgbondcolor set to split
----------------------------------	----------------------------------

-imgsymbolfontsize <pts>

The option **-imgsymbolfontsize** specifies the size of element symbols or labels in the structure images. By default (or if you set this number to less than 0) the program chooses suitable font sizes for atomic symbols automatically. It is possible to override this choice with this option. If this value is set to 0, no symbols will be printed. Hetero atoms are then marked by small (colored, if in color mode) squares. The desired point size can be a floating point number.

Default value:

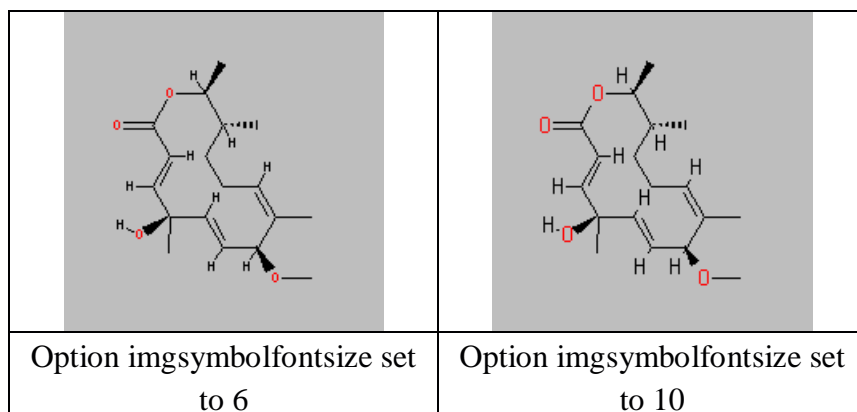
By default the program chooses suitable font sizes for atomic symbols automatically.

Example:

Generating a table file with embedded images showing atom symbols in font size 10 pts:

```
mn_table -outfile six_examples.xls -coll Image
        -imgsymbolfontsize 10 ./examples/six_examples.sdf
```

Picture:



Remark:

Note that the number of available font sizes is limited if you use the built-in fonts instead of external TrueType fonts. The program will choose the closest built-in font (from the list 6pt, 7pt, 8pt, 10pt) in this case.

-imgannotationfontsize <pts>

The option **-imgannotationfontsize** controls the font size for atom annotations, such as formal charges or isotope labels. If the value is negative (the default), the font size is automatically derived from the element symbol size. If it is set to zero, no atom annotations are printed.

Default value:

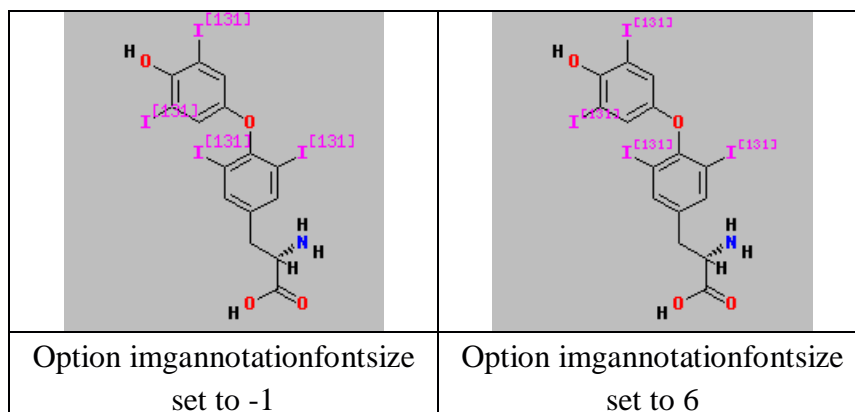
By default the font size is set to -1, meaning automatic adjustment.

Example:

Generating a table file with embedded images showing text for annotations in 6 pts:

```
mn_table -outfile thyroxin.xls -coll Image
        -imgannotationfontsize 6 ./examples/thyroxin_i131.sdf
```

Picture:



Remark:

Note that the number of available font sizes is limited if you use the built-in fonts instead of external TrueType fonts. The program will choose the closest built-in font (from the list 6pt, 7pt, 8pt, 10pt) in this case. The desired point size may be a floating point number.

-imgbonds <n>

The option **-imgbonds** determines how many standard-length bonds should fit on the image in x-direction. Structures which possess fewer bonds are centered on the image. Structures which have more bonds are shrunk until they fit tightly into the image. A larger value will let the average structure appear smaller, but more structures will fit in the image without re-scaling, maintaining their correct size relationships. Smaller values will make plots larger, but re-scaling is necessary for more compounds.

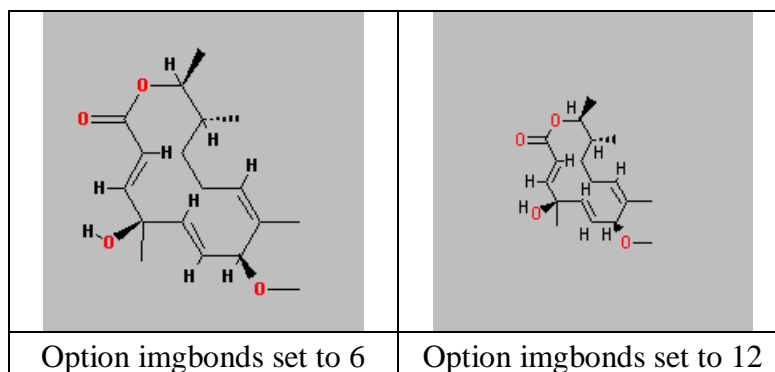
Default value:

By default the option bondscale is set to 10.

Example:

Generating a table file with embedded images having a bondscale number of 12:
`mn_table -outfile six_examples.xls -coll1 Image -imgbonds 12
./examples/six_examples.sdf`

Picture:



-imgshowcharge <0/1>

The parameter **-imgshowcharge** controls the plotting of charge symbols. If this flag is set to 0, atomic charge symbols are not plotted.

Default value:

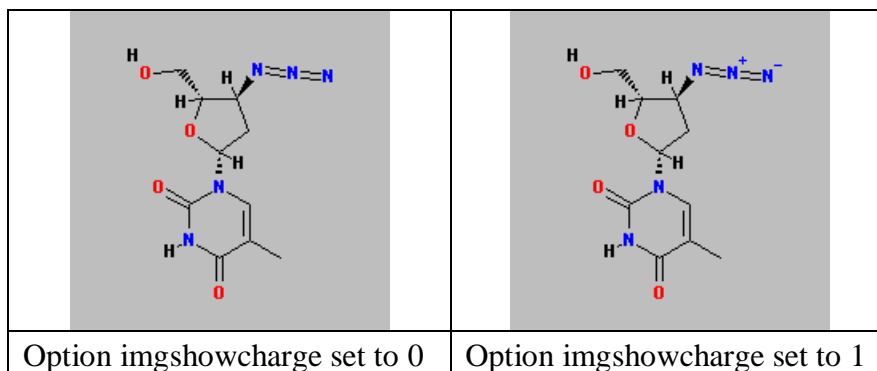
By default, charge symbols are plotted (value 1).

Example:

Generating a table file with embedded images showing no charge symbols:

```
mn_table -outfile six_examples.xls -coll Image -imgshowcharge 0
./examples/six_examples.sdf
```

Picture:



-imgshowisotope <0/1>

The parameter **-imgshowisotope** controls the plotting of the nucleonic number. If this flag is set, isotopically labeled atoms are annotated with their nucleonic number. Heavy hydrogen atoms are displayed as D or T. If the flag is unset, isotopic information is suppressed.

Default value:

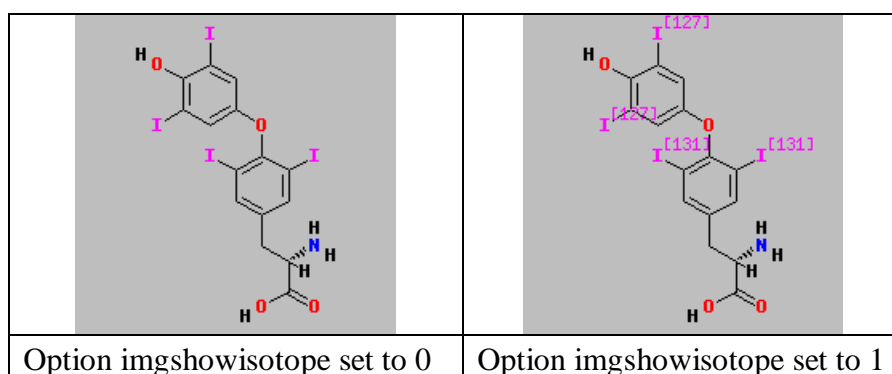
By default, nucleonic symbols are plotted (value 1).

Example:

Generating a table file with embedded images showing isotope information:

```
mn_table -outfile thyroxin.xls -coll Image -imgshowisotope 1
./examples/thyroxin_i131.sdf
```

Picture:



-imgshowradical <0/1>

The parameter **-imgshowradical** controls the plotting of radical centers.

Default value:

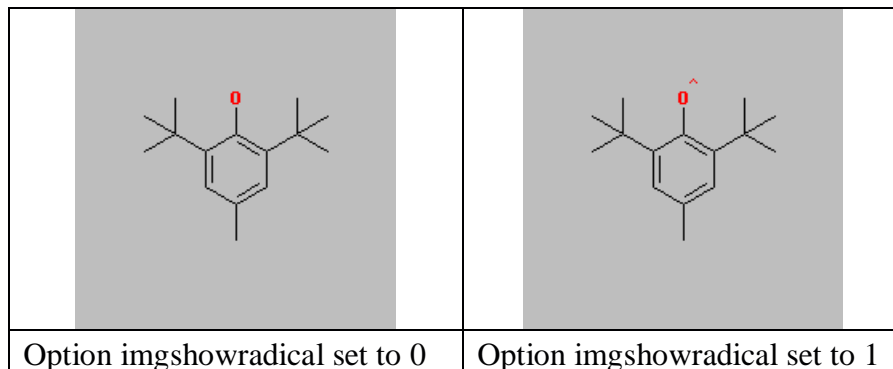
By default, radical centers are plotted (value 1).

Example:

Generating a table file with embedded images showing no radical centers:

```
mn_table -outfile bht_radical.xls -coll Image -imgshowradical 0
./examples/bht_radical.mol
```

Picture:



-imgshowstereo <0/1>

The parameter **-imgshowstereo** controls the plotting of stereo descriptors. If this flag is set to 0, no stereo descriptors are plotted. Note that this flag has no influence on the display of wedge bonds (see **-imgwedges** and **-imgdashes** options to control their appearance). It only applies to atomic stereo descriptors such as CIP R or S, which might be present in the input file. These descriptors are not computed if not explicitly present in the input data. They are only plotted if read from file.

Default value:

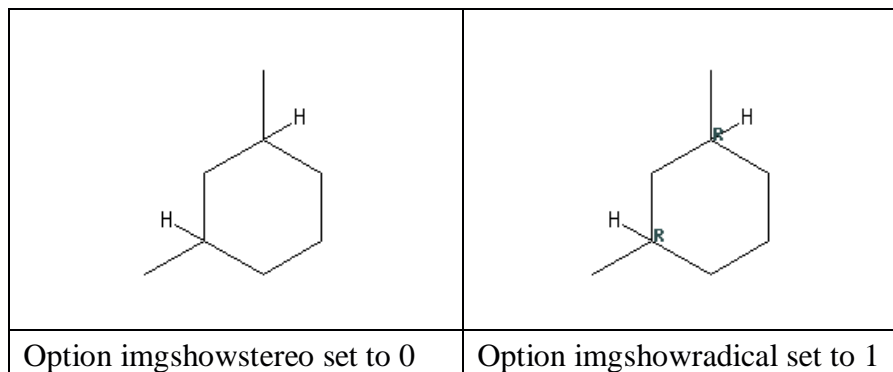
By default, no stereo descriptors are plotted (value 0).

Example:

Generating a table file with embedded images showing stereo descriptors:

```
mn_table -outfile stereo.xls -coll Image -imgshowstereo 1
./examples/stereo.sdf
```

Picture:



-imgshowstereoH <0/1>

The parameter **-imgshowstereoH** controls the plotting of hydrogen atoms at stereo centers. If this flag is set, hydrogen atoms at stereo centers which are not linked via a wedge bond are explicitly drawn in order to obtain an unambiguous stereocenter display. Hydrogen atoms linked via a wedge bond are always drawn regardless of the value of this flag.

Default value:

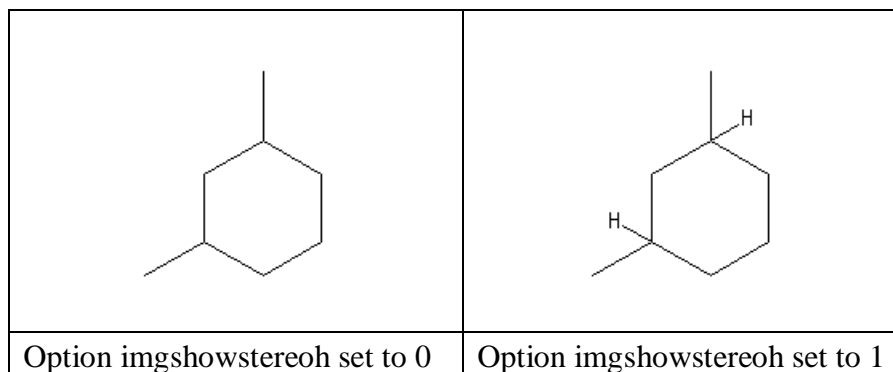
By default, hydrogen atoms at stereo centers are plotted (value 1).

Example:

Generating a table file with embedded images showing no stereo descriptors at hydrogen atoms:

```
mn_table -outfile stereo.xls -coll Image -imgshowstereoh 0
./examples/stereo.sdf
```

Picture:



-imgwedges 0/1

The option **-imgwedges** controls whether wedge bonds are displayed as such or not.

Default value:

This flag is activated by default (value=1).

Example:

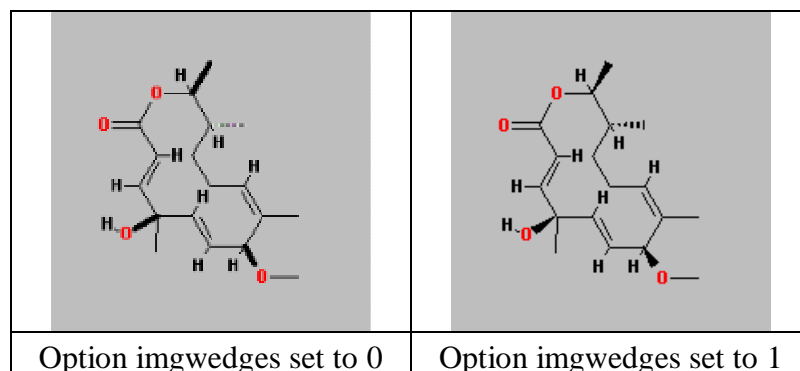
Generating a table file with embedded images showing no attributes on wedge bonds:

```
mn_table -outfile six_examples.xls -coll Image -imgwedges 0
./examples/six_examples.sdf
```

Remarks:

If this parameter is set to 0, but the **-imgdashes** parameter is still switched on, full wedges will be printed as a bold bond and dashed wedges as simple dashed bond. If the **-imgdashes** option is also inactive, the display of bond stereo attributes is completely suppressed.

Pictures:



-imgdashes 0/1

The option **-imgdashes** controls whether the dash attribute on wedges and simple bond lines are displayed as such or not.

Default value:

This flag is activated by default (value=1).

Example:

Generating a table file with embedded images showing no attributes on wedge and dash bonds:

```
mn_table -outfile six_examples.xls -coll Image -imgwedges 0  
-imgdashes 0./examples/six_examples.sdf
```

Remarks:

In order to avoid wrong display of stereochemistry, it should only set to 0 in combination with a **-imgwedges 0** option. Resetting this option alone, without also resetting **-imgshowstereo**, is not useful.

-imgusehatch 0/1

If the option **-imgusehatch** is set to 1, dashed wedges in WMF plots are drawn as a single triangle and filled with the built-in hatch patterns of the Windows GDI. By default, every sector of such wedges is drawn as an independent polygon. In case of very small wedges, the use of the build-in hatch pattern may yield visually more pleasing results than polygon drawings. Larger wedges or printer output in a resolution higher than the screen display generally look better when drawn as polygons.

Default value:

This flag is activated by default (value=0).

Example:

Generating a table file with embedded images showing dashed wedges in hatch patterns:

```
mn_table -outfile six_examples.xls -coll Image -imgusehatch 1  
./examples/six_examples.sdf
```

7.8. Program features of general usage

-version

If this flag is set, the version and licensing information is printed.

Default value:

This flag is deactivated by default.

Example:

Showing the program version:

```
mn_table -version
```

8. Extended Features only available for the UNIX operating systems

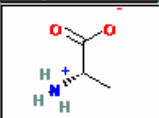


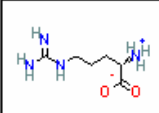


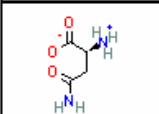


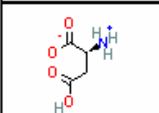


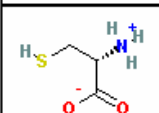


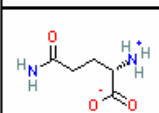


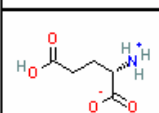


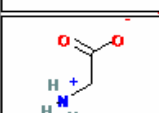


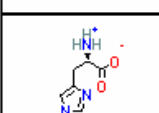


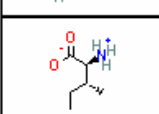
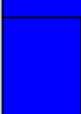

Input files can be processed in compressed or gzip-ed form without prior unpacking. The input file name arguments may each be a local file, an URL (http, ftp, gopher, file) or an email message file containing the structure data in the main body or as one or more attachments. URL retrieval and compression can be combined.

The output file name can also be an anonymous ftp URL.

9. Application examples

9.1. Table of 20 natural amino acids

A SD file containing the 20 natural amino acids (20as_prop.sdf) with some property information was converted into a spreadsheet format. The first eleven entries and the corresponding command line are shown below.

	A	B	C	D	E	F	G	H
1	Image	Name	Abbrev.	Code	IPoint	IPoint Histogram	pks1	pks1 Histogram
2		Alanine	Ala	A	6.01		2.34	
3		Arginine	Arg	R	10.76		2.18	
4		Asparagine	Asn	N	5.41		2.02	
5		Aspartic Acid	Asp	D	2.77		1.88	
6		Cysteine	Cys	C	5.07		1.71	
7		Glutamine	Gln	Q	5.65		2.17	
8		Glutamic Acid	Glu	E	3.22		2.19	
9		Glycine	Gly	G	5.97		2.34	
10		Histidine	His	H	7.59		1.82	
11		Isoleucine	Ile	I	6.02		2.32	

```
mntable -outfile 20as.xls -col1 Image -imgheight 75 -imgwidth 100 -imgbonds 4 -col2 "E_NAME
Name {width 17}" -col3 "Abbrev. Abbrev." -col4 "Code Code" -col5 "IPoint IPoint {fgcolor
blue}" -col6 "IPoint {IPoint Histogram} {fgcolor blue format histogram bgcolor grey width 20}"
-col7 "pks1(COOH) pks1 {fgcolor red}" -col8 "pks1(COOH) {pks1 Histogram} {fgcolor red format
histogram bgcolor grey width 20}" 20as_prop.sdf
```

10.Frequently Asked Questions (FAQ)

11.Error messages

12.Known problems and limitations

Embedding images into Microsoft Excel files in a way that they move along with their cells when columns are sorted is problematic. There seem to be some rounding problems within Excel when determining whether an image is still contained within a cell and thus should be moved or not. When image objects are formatted to exactly fill a spreadsheet cell, they frequently fail to move properly. This is strangely dependent of the image size – enlarging or shrinking it by one pixel frequently resolves the problem, or makes it appears. Therefore, such images (structure drawings and histograms) are currently encoded by this software to be slightly smaller than the cells they are embedded in. An attempt is made to minimize visible gaps, but they cannot be completely avoided in all cases. The fudge factor is already dependent on an estimate of the cell size, but these are heuristics. If you encounter parameter settings where either the images do not move, or the gaps are unacceptably large (or the images do not scale while changing row heights or column widths, which is a side effect of the presence of a notable gap), please let us know.

13. Technical Support

The MN.TABLE Web Site

If you have problems while running MN.TABLE please have a look at the Support- and FAQ web site of MN.TABLE. The pages are available at <http://www.mol-net.de>

Reporting Problems

If your problem is not listed in these web pages please report it to the MN.TABLE team at Molecular Networks. Please make sure to provide us with all important data for replicating your problem on our machines. Therefore please use the report form on the next page.

Updates

If you have licensed the program MN.TABLE with maintenance you will automatically receive updates every time a new release is launched.

Contact information

Distribution and Maintenance for MN.TABLE is handled by Molecular Networks Computerchemie, Erlangen, Germany.

Molecular Networks GmbH
Computerchemie
Nägelsbachstraße 25
91052 Erlangen
Germany

e-Mail: support@mol-net.de

Tel. +49 9131/815668

Fax +49 9131/815669

14. Report Form

In case of problems occurring during installation or running MN.TABLE, please complete the following form and send it or fax it to

Molecular Networks GmbH Computerchemie
Nägelsbachstraße 25
91052 Erlangen
Germany
FAX: +49-(0)9131-815669

User:

MN.TABLE program and version number (mn_table -version):

Command line to run MN.TABLE:

Error and warning messages by MN.TABLE:

System messages:

Short description:

Please include the input file and output file generated by MN.TABLE on a 3½" diskette written in MS/DOS format or send an e-mail to support@mol-net.de attaching these files. These files will help us to analyze your problems. All data will be treated confidentially.

15.Index

inputfile
 20as_prop.sdf 35
 alkanes1_12.sdf 14, 15, 16, 17, 18, 19
 bht_radical.mol 31
 maybridge.sdf 15
 mn_logo.gif 24
 six_examples.sdf 9, 13, 14, 19, 20, 21,
 22, 23, 24, 25, 26, 27, 28, 29, 30, 32,
 33
 stereo.sdf 31, 32
 thyroxin_i131.sdf 28, 30

option
 assign 15
 colnames 15
 count 14
 eoltype 14
 feedback 15
 flatten 16
 footer 16
 format 13
 header 17
 imageborder 16
 imagedirectory 14
 imageurl 15
 imgannotationfontsize 28
 imgasymbol 25
 imgatomcolor 26
 imgbgcolor 19
 imgbondcolor 27
 imgbonds 29
 imgborder 19
 imgdashes 32
 imgfooter 22
 imgfootercolor 22
 imgfooterproperty 22
 imghcolor 27
 imgheader 20
 imgheadercolor 20
 imgheaderproperty 21
 imgheight 18
 imghsymbol 25
 imglogofile 23
 imglogoscale 24
 imgshowcharge 29
 imgshowisotope 30
 imgshowradical 30
 imgshowstereo 31
 imgshowstereoh 31
 imgsymbolfontsize 28
 imgusehatch 33
 imgwedges 32
 imgwidth 19
 mapping 17
 outfile 13
 sort 17
 version 33