



Combining Structure Files and Data Files
into a Single File

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 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.MERGE.....	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 Microsoft 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 Microsoft 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.	Microsoft 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.....	11
7.4.	Program Features in More Detail	11
	-format <abbreviation of the output format>	11
	-outfile <filename.extension>	12
	-directory <dirname>	12
	-feedback 0/n	12
	-tablefile <filename>	13
	-tablekey <columnname/index>	13
	-tablekeytype <single/multi/strict>.....	14
	-structurekey <property>	14
	-stat	15
	-version	15
	-h or -help.....	15
8.	Extended Features Only Available for the UNIX Operating Systems.....	15
9.	Frequently Asked Questions (FAQ)	16
10.	Error Messages	16
11.	Known Problems and Limitations	16
12.	Technical Support	17
	The MN.MERGE Web Site.....	17
	Reporting Problems.....	17
	Updates.....	17
	Contact Information	17
13.	Report Form.....	18
14.	Index.....	19

1. General Information about MN.MERGE

MN.MERGE reads a structure file and a corresponding data file and builds a single SDFfile containing the structural and textual information.

The program MN.MERGE

- reads data files saved as dBASEIII, Sylk or Sybyl or separator delimited text
- reads structure files saved in SD file format
- supports various options to define the primary key
- processes datasets with 99.9% conversion rate
- handles datasets of hundreds of thousands of chemical structures
- supports the SD file format for saving the output files

2. Installation

2.1. Requirements

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

The program runs in a batch mode.

2.2. Installation Steps for UNIX Operating Systems (IRIX, Solaris, Linux)

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

Launch MN.MERGE with the command

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

2.3. Installation Steps for Microsoft 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_merge` in your installation directory carefully (default path during installation was `/usr/local/bin/mn_merge/`).

3.2. Uninstallation Steps for Microsoft 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.MERGE or if any problems occur while running MN.MERGE, 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.MERGE (`mn_merge -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.MERGE

6. Getting Started

6.1. UNIX operating systems

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

```
mn_merge -tablefile alkanes_prop.txt alkanes.sdf
```

MN.MERGE now creates the output file named `alkanes.mdl` written to the same directory where the file `alkanes.sdf` is located. Figure 1 shows the content of this file.

6.2. Microsoft Windows operating systems

The example file `alkanes.sdf` submitted with the distribution contains the structure information of twelve molecules in SD 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_merge` by using the `cd` command then type the following command:

```
mn_merge -tablefile alkanes_prop.txt alkanes.sdf
```

MN.MERGE now creates the output file named `alkanes.mdl` written to the same directory where the file `alkanes.sdf` is located. Figure 1 shows the content of this file.

If you have no permission writing to the directory in which the program was installed, set the **-directory** option for specifying another directory:

```
UNIX: mn_merge -directory /tmp -tablefile alkanes_prop.txt  
alkanes.sdf
```

```
Windows: mn_merge -directory C:/temp -tablefile alkanes_prop.txt  
alkanes.sdf
```




-182.5  Methane	-161.5	-183.3	-88.6 Ethane	-187.7 0.493 Propane	-42.1	-138.4 0.573 Butane	-0.5
-129.7  0.621 Pentane	36.1	-95.3	68.7 Hexane	-90.6 0.68 Heptane	98.4	-56.8 0.698 Octane	125.7
-53.5  0.714 Nonane	150.8	-29.7	174.1 Decane	-25.6 0.737 Undecane	195.9	-9.6 0.745 Dodecane	216.3

Figure 1: Content of the output file `alkanes.mdl`

7. Program Use

7.1. Synopsis

The general synopsis for using MN.MERGE is:

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

An overview of the various options is given in Table 1 and in a more detailed one in the following chapter. `Infile` is the input file name. If no file name is given, the program reads from standard input.

[-directory dirname]	specifies the output directory
[-feedback 0/n]	prints a control message after processing a block of n items
[-format fmt]	specifies the output format name
[-h] or [-help]	shows a brief help message about the usage of the program
[-outfile filename]	defines the name of the output file
[-properties propertyname/all]	determines additional data fields that are output
[-stat]	writes statistical information about the number of successfully processed records and processing failures
[-structurekey property]	set the name of the key in the structure file
[-tablefile filename]	specifies the name of a table file which contains additional data information
[-tablekey columnname/index]	set the name of the key column
[-tablekeytype single/multi/strict]	defines the processing of cases where a structure key matches more than one table row
[-version]	prints version and licensing information

Table 1: Overview of all options

7.2. General Program Features

The file type of the input file is automatically recognized. If no input file is specified, or the file name „-“ is used, the program reads from standard input.

If you are running MN.MERGE under a UNIX operating system, there are some more features reading input files (see chapter 8 “Extended Features Only Available for the UNIX Operating Systems” for more details).

The file name of the output file is either explicitly set with the **-outfile** option or automatically derived from the input file and the given output file format (**-format**). The special filename `stdout` can be used to direct output to the standard output channel.

7.3. Supported file formats for input files

The program will automatically detect the file format of the input files. 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 in More Detail

-format <abbreviation of the output format>

The parameter **format** is specified for selecting the output format.

The supported file formats are listed in the table below.

Full Format Name	Default Output-Extension	Write	Comment
MDL SDF	mdl	Yes	

Table 3: Overview of the supported output file formats

Please use the abbreviation of the format names for specifying your desired file format using the **-format** option. If no output file is specified, the output has the same name (but with an updated suffix) and is written in the same directory as the input file. The extension of the resulting output file is sometimes different to the given abbreviation (see the previous table). If the output file is specified explicitly with the **-outfile** parameter, this file name including the chosen suffix, will be used.

Default value:

Parameter without a default value

Example:

Generating a MDL SD-file:

```
mn_merge -format sdf -tablefile examples/alkanes_prop.txt
examples/alkanes.sdf
```

Remarks:

If this option is not used, an attempt is made to guess the output file format from its suffix by the given **-outfile** parameter.

-outfile <filename.extension>

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

If you are using MN.MERGE on a UNIX operating system the output file name can also be an anonymous ftp URL.

Default value:

Parameter without a default value

Example:

Generating a MDL SD-file:

```
mn_merge -outfile alkanes_prop.sdf -tablefile
examples/alkanes_prop.txt examples/alkanes.sdf
```

-directory <dirname>

This parameter sets the target directory. If the directory does not yet exist, it will be created.

Default value:

The directory of the output files is the same as of the corresponding input files, or the current directory, if the input file names do not contain directory information.

Example:

Generating a MDL SD-file saved in a given directory:

```
UNIX: mn_merge -outfile alkanes_prop.sdf -directory /tmp
-tablefile examples/alkanes_prop.txt examples/alkanes.sdf
Windows: mn_merge -outfile alkanes_prop.sdf -directory C:/Temp
-tablefile examples/alkanes_prop.txt ./examples/alkanes.sdf
```

-feedback 0/n

If the parameter **feedback** is set to a value larger than zero, a control message is printed after processing a block of n structures. The current record number and the object name are printed on the standard error channel. Only structures which are actually written out are counted.

Default value:

It is not active by default.

Example:

Generating a MDL SD-file printing dots for every fifth records:

```
mn_merge -outfile alkanes_prop.sdf -tablefile
examples/alkanes_prop.txt -feedback 5 examples/alkanes.sdf
```

-tablefile <filename>

This option specifies the name of a table file which contains additional structure information. Data from this table will be merged to the current structure.

By default, if the parameter **-tablekey** is not used, the data in the table is assumed to be in the same sequence as the structures in the input file(s). Data assignment takes place before any operations which could change the number of structures coming from the input sources, i.e. before any filtering, tautomer generation, etc. If the table contains explicit column names, the output data fields, will have the same name as the table column. In case the columns are not named, synthetic names in the form coln will be used, with n being the index of the column, starting with one. The format of the table file is automatically determined. Supported table formats include dBase3 files, Sylk, Sybyl tables, and simple text tables with fields separated by a separator character like tab, vertical tab, whitespace or semicolon. In case of text tables, data fields with embedded separator characters are allowed if the data is enclosed in quotes, and quotes in the quoted text are allowed if escaped with a backslash. An attempt is made to identify and optimize the data type of columns automatically. If this fails, data is assumed to be a string. In case any columns resolve to a non-string data type if the first row is ignored, it is assumed that the first row represents column names in formats which do not provide explicit column naming.

Default value:

By default this property list is empty.

Example:

```
mn_merge -outfile alkanes_prop.sdf -tablefile
examples/alkanes_prop.txt examples/alkanes.sdf
```

Remark:

This simple example assumes that there is one table row for every record in the input file, and both input sources follow the same sequence. Both the structure and table keys default to record. Data from one table row is merged to every structure record.

Please note that the input of Excel files, a commonly requested feature, is not yet supported.

-tablekey <columnname/index>

This option is used to set the name of the key column when a structure file is merged with a data table (see option **-tablefile**). If no merging takes place, the option is ignored.

The default value for this parameter is record, indicating that the sequence of structures in the input file(s) is the same as in the table. Alternatively, a name of a column in the table, or its numerical index (starting with 0) may be specified. The special value record will always work, even if the table does not contain an explicit record column, because it is added when the table is read and filled with the row number in case it is not present as an explicit column. The column name of the index column is by default used as structure key, but this may be overridden by the **-structurekey** option. The structure key is interpreted as the name of a

property (in toolkit nomenclature, or as used in the structure input file) of the current structure or reaction. It is possible to use structure keys which are computed on the fly from more elementary structure data.

Default value:

By default this property list is empty.

Example:

```
mn_merge -outfile alkanes_prop_regid.sdf -tablefile
examples/alkanes_prop_regid.txt -tablekey REGID
examples/alkanes1_12_regid.sdf
```

Remark:

This program invocation will merge data from the input table with the structures read from the SD file. The table must have a column named REGID, and the SD file likewise.

-tablekeytype <single/multi/strict>

This parameter defines the processing of cases where a structure key matches more than one table row, or no row. The default value is single. In this case, if there are multiple matching rows, only the first will be used, and it is no error to have structure keys without corresponding table rows. Mode strict is nearly the same as single, but it is an error to find no matching table row for a structure record. In mode multi, multiple table rows are assigned to distinct property instances and output as such in file formats which support this.

Default value:

By default this option is set to single.

-structurekey <property>

This option is used in combination with the **-tablefile** and **-tablekey** parameters. If this parameter is not specified, it defaults to the name of the table column identified by the **-tablekey** parameter. If both are unspecified, both default to record. If the **-tablekey** parameter is a table column index, the structure key is not identical to the table key – it is the name of the indexed column, or coln (n being the index plus 1) in case the table columns are not named. This structure key is interpreted as a name of a property, either in toolkit nomenclature or as used in the input file, which is used to find rows corresponding to the current structure. If it is the special value record, the current global record count (ignoring any offsets, and not reset in case multiple input files are processed) is used as row number. Otherwise, the property value is extracted from the record (it may be computed in case of computable properties) and used to find the corresponding table row with data to be added to the structure.

Default value:

By default this property list is empty.

-stat

If this flag is set, statistical information about the number of successfully processed records and conversion failures is written to the standard error channel.

Default value:

This flag is deactivated by default.

Example:

Generating a MDL SD-file showing the statistical information of the conversion:

```
mn_merge -outfile alkanes_prop.sdf -tablefile
examples/alkanes_prop.txt -stat examples/alkanes.sdf
```

Output:

```
Convert file alkanes.sdf
Successfully read 12 records, failed 0
Successfully wrote 12 records, failed 0.
```

-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_merge -version
```

-h or -help

If this flag is set, a brief help message about the usage of the program is shown.

Default value:

This flag is deactivated by default.

Example:

Show the help message:

```
mn_merge -h or mn_merge -help
```

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.

9. Frequently Asked Questions (FAQ)

10. Error Messages

11. Known Problems and Limitations

12. Technical Support

The MN.MERGE Web Site

If you have problems while running MN.MERGE please have a look at the Support- and FAQ web site of MN.MERGE. 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.MERGE 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.MERGE with maintenance you will automatically receive updates every time a new release is launched.

Contact Information

Distribution and Maintenance for MN.MERGE 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

13.Report Form

In case of problems occurring during installation or running MN.MERGE, 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.MERGE program and version number (`mn_merge -version`):

Command line to run MN.MERGE:

Error and warning messages by MN.MERGE:

System messages:

Short description:

Please include the input file and output file generated by MN.MERGE 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.

14.Index

inputfile

alkanes.sdf 9, 12, 13, 15
alkanes_prop.txt 9
alkanes_prop_regid.txt 14
alkanes1_12_regid.sdf 14

option

directory 12
feedback 12
format 11
h 15
help 15

outfile 12

stat 15

structurekey 14

tablefile 13

tablekey 13

tablekeytype 14

version 15

outputfile

alkanes_prop.sdf 12

alkanes_prop_regid.sdf 14