Mjølner System: Man Pages

Mjølner Informatics Report

March 2004

Table of Contents

1 Manpage for beta....1

1.1 NAME....1

1.2 SYNOPSIS....1

1.3 AVAILABILITY....1

1.4 DESCRIPTION....1

1.5 OPTIONS....1

1.6 ENVIRONMENT....4

1.7 FILES....5

1.8 SEE ALSO....6

1.9 BUGS....7

1.10 AUTHORS....7

2 Manpage for mjolner....8

2.1 NAME....8

2.2 SYNOPSIS....8

2.3 AVAILABILITY....8

2.4 DESCRIPTION....8

2.5 OPTIONS....8

2.6 ENVIRONMENT....8

2.7 SEE ALSO....9

2.8 AUTHORS....10

3 Manpage for betatar....11

3.1 NAME....11

3.2 SYNOPSIS....11

3.3 AVAILABILITY....11

3.4 DESCRIPTION....11

3.5 OPTIONS....12

3.6 ENVIRONMENT....13

3.7 SEE ALSO....13

3.8 BUGS....13

3.9 AUTHORS....13

4 Manpage for betafs....15

4.1 NAME....15

4.2 SYNOPSIS....15

4.3 AVAILABILITY....15

4.4 DESCRIPTION....15

4.5 OPTIONS....15

4.6 ENVIRONMENT....15

4.7 SEE ALSO....15

4.8 AUTHORS....16

5 Manpage for betawc....17

5.1 NAME....17

5.2 SYNOPSIS....17

5.3 AVAILABILITY....17

5.4 DESCRIPTION....17

5.5 OPTIONS....18

5.6 ENVIRONMENT....18

5.7 SEE ALSO....18

Table of Contents

5.8 AUTHORS....................18

6 Manpage for psbrowser....................20

6.1 NAME....................20

6.2 SYNOPSIS....................20

6.3 AVAILABILITY....................20

6.4 DESCRIPTION....................20

6.5 OPTIONS....................21

6.6 SEE ALSO....................21

6.7 AUTHORS....................21

1 Manpage for beta

1.1 NAME

beta − BETA Compiler (version 5.4)

1.2 SYNOPSIS

beta [−−help|−h] [−−repeat|−r] [−−noRepeat] [−−link] [−−noLink|−x] [−−static] [−−dynamic] [−−list] [−−noList|−l] [−−debug] [−−noDebug|−d] [−−code] [−−noCode|−c] [−−checkQua] [−−noCheckQua|−Q] [−−checkNone] [−−noCheckNone|−N] [−−checkIndex] [−−noCheckIndex|−I] [−−warn] [−−noWarn|−w] [−−warnQua] [−−noWarnQua|−q] [−−verbose] [−−quiet] [−−mute] [−−traceCheck] [−−noTraceCheck] [−−traceCode] [−−noTraceCode] [−−out *file* | −o *file*] [−−preserve|−p] [−−noPreserve] [−−job] [−−noJob|−j] [−−switch *sw−1*...*sw−n* [0] | −s *sw−1*...*sw−n* [0]] [−−linkOpts *string*] *files*...

1.3 AVAILABILITY

The Mjolner BETA Compiler is available as part of the Mjolner System from Mjolner Informatics.

1.4 DESCRIPTION

beta is an efficient compiler for the object oriented programming language BETA. The compiler is using native code generation, automatic garbage collection, and separate compilation. The compiler also allows for easy interfacing into code and data structures, originating from sources written in other languages such as C, Pascal and assembly language. The Mjolner System includes (besides this BETA compiler), a persistent store for BETA objects, a source−level debugger, a hyper structure editor and a wide variety of libraries and application frameworks (data structures, window system frameworks, metaprogramming system, etc.), an object−oriented database for BETA objects (prerelease), a distributed object system for BETA objects (experimental), etc.

1.5 OPTIONS

−h
−−help
Print this help info

−r
−−repeat

```
     Run compiler in repeating mode
--noRepeat
     Do not run compiler in repeating mode (default)

--link
     Link program (default)
 -x
--noLink
     Do not link program

--static
     Use static linking
--dynamic
     Use dynamic linking (default)

--list
     Generate .lst file, if semantic errors (default)
 -l
--noList
     Do not generate .lst file, if semantic errors

--debug
     Generate debug info to enable debugging (default)
 -d
--noDebug
     Do not generate debug info

--code
     Generate code (default)
 -c
--noCode
     Do not generate code

--checkQua
     Generate qualification runtime checks (default)
 -Q
--noCheckQua
     Do not generate qualification runtime checks

--checkNone
     Generate runtime checks for NONE references (default)
 -N
--noCheckNone
     Do not generate runtime checks for NONE references

--checkIndex
     Generate runtime checks for  repetition  index  out  of
     range (default)
 -I
--noCheckIndex
     Do not generate runtime checks for repetition index out
     of range

--warn
     Generate warnings (default)

 -w
--noWarn
     Do not generate warnings

--warnQua
     Generate warnings about runtime QUA checks (default)
 -q
--noWarnQua
     Do not generate warnings about runtime QUA checks
```

```
--verbose
     Verbose compiler info output
--quiet
     Only compiler info on parse, check, etc. (default)
--mute
     No compiler info output

--traceCheck
     Trace the compiler during semantic checking
--noTraceCheck
     Do not trace the compiler during semantic checking

--traceCode
     Trace the compiler during code generation
--noTraceCode
     Do not trace the compiler during code  generation  (de-
     fault)

 -o  file
--out file
     Specify name to use for executable

 -p
--preserve
     Preserve generated .job and assembly files
--noPreserve
     Do not preserve generated .job and assembly files  (de-
     fault)

--job
     Execute the .job file (default)
 -j
--noJob
     Do not execute the .job file

 -s  sw-1...sw-n [0]
--switch
     sw-1 ... sw-n [0]
     Set/unset one or more compiler switches. Please see the
     section Compiler switches below for details.

--linkOpts string
     Specify text string to be append to the link directive

Note that short options can be combined, e.g.  -q -c can  be
written  as  -qc  .  Long option names are case insensitive,
whereas single character options are case sensitive.

Compiler switches
     This section describes the  most  interesting  compiler
     switches with respect to parameterization.  Please con-
     sult the compiler reference manual for details.

      12:  Force code generation for all  fragments  in  the
           dependency  graph.   Since  this switch may force
           code generation of standard libraries, it  should
           only  be  used  by  the system administrators and
           only in the case of  serious  problems  with  the
           installation.

      21:  Continue translation after semantic errors.

      191: Print each descriptor just before it is checked.

      192: Print each declaration just before it is checked.
```

193: Print each imperative just before it is checked.

308: Print each declaration just before code is generated for it.

311: Print each imperative just before code is generated for it.

Note that switches 191, 192 and 193 are jointly set by --traceCheck and that switches 308 and 311 are jointly set by --traceCode .

1.6 ENVIRONMENT

The *beta* compiler recognizes the environment variables listed below. Please note that many of these variables are given default values in the Bourne Shell script $*BETALIB/configuration/env.sh* (see the *BETALIB* environment variable below). If the default values for some of these variables are to be changed for an entire site-installation, the easiest way to do it is by changing them directly in this file.

BETALIB

Specifies where ~*beta* is located. If not set, *beta* is assumed to be a username, and ~*beta* being the home directory of that user. Is used by many tools in the Mjolner System.

BETAOPTS

Specifies options that the *beta* compiler should be invoked with by default.

BETALINKOPTIONS

Specifies the linker options to be used by the BETA compiler when linking (using std. UNIX linker). If set, it totally overwrites the default link options, the compiler would have used otherwise.

LD_LIBRARY_PATH

This is a colon separated list of directories to search for external libraries during linking. Notice that not all standard UNIX linkers supports this variable directly, but the *..job* files generated by the beta-compiler will still use this variable.

TMPDIR

Normally, the link-directives in the *..job* files will use */tmp* for temporary files. If another directory is to be used (e.g. because */tmp* is full), setting *TMPDIR* to the name of a directory, prior to compilation, will cause the link-directives to place temporary files in this directory.

MACHINETYPE

Is set automatically by the compiler during the execution of the *..job* files and *make* files.

BETART

Is used to set various characteristics of the BETA run-time system. See documentation on the Web:

$*BETALIB/doc/betarun/BETART.html*:

Local copy of the document as released with release 5.0.

http://www.mjolner.com/mjolner-system/documentation/betarun/BETART.html:
Latest version of the document.

BETARS
Corresponds to BETART. Is used by valhalla (the source level debugger) for specifying the BETART entries for the program being debugged (and thus not affecting the behavior of the debugger itself). Setting BETART is in this case used to control the entries of the debugger itself.

1.7 FILES

file.bet
The file containing the BETA source-code

file.ast
On a big-endian architecture, this file will contain the abstract syntax tree representation of the compiled source code. This file is used by many tools in the Mjolner System. This file used to be called *file.group* in previous releases of the Mjolner System.

file.astL
On a little-endian architecture, this file will contain the abstract syntax tree representation of the compiled source code. This file is used by many tools in the Mjolner System.

file.lst
This file is generated in the case of errors during the compilation process. It will then contain error-messages etc. along with the source code if errors are discovered in the source code (syntax errors and semantic errors).

file.o
This file will contain the object-code for the compiled source code. *file.o* files are located in subdirectories named according to the machine type, to which the source code have been compiled. Currently, the directories: *sun4s*, *hpux9pa*, *linux*, *nti_ms*, *nti_gnu*, *sgi*, and *ppcmac* can be created. These directories are automatically created by the compiler, if not present already.

file..job
(or *file.job*) This file contains commands controlling the assembly and link process (asm and link instructions, etc.). This file is generated during the compilation process, and will normally be removed immediately before the compilation terminates. Like *file.o* the job-file is placed in a subdirectory corresponding to the machine type, please see *file.o* for a descriptrion.

file..s
This file (only generated on HP-UX machines) will contain the assembly code for the compiled source code

files. This file is also normally removed immediately before the compilation terminates. Like *file.o* the assembly−file is placed in a subdirectory corresponding to the machine type, please see *file.o* for a descrip− trion.

file.db
file with debug info used by valhalla. Like *file.o* the debug−file is placed in a subdirectory corresponding to the machine type, please see *file.o* for a descriptrion.

file.dump
If a run−time error occurs during execution of the executable, a dump of the current object and the call chain that activated that object will be written to this file. Please note, that in some situations, the program state may be so corrupted that the dump becomes unprecise, or it may fail completely to produce the dump. Try using valhalla. and reproduce the error if the dump is not enough to understand the error. Note that this file was previously named *beta.dump* for all applications.

$*BETALIB/configuration/env.sh*
Bourne Shell script used by the tools in the Mjolner System. Contains default set−up of environment variables for the architectures currently supported by the Mjolner System.

1.8 SEE ALSO

mjolner(1) − Mjolner Integrated Development Environment

betatar(1) − BETA **tar(1)** front−end

betawc(1) − BETA fragment analyser

betafs(1) − Mjolner BETA Fragment Structure Lister

The USENET newsgroup *comp.lang.beta* is intended for discussions about the BETA language and the programs and systems written in or supporting BETA. Discussions concerning object−oriented programming principles based on the concepts known from BETA will also take place in *comp.lang.beta*, possibly cross−posted to *comp.object*.

The *beta−language−faq* will be cross−posted to *comp.lang.beta*, and the most frequently asked questions from *comp.lang.beta* will be included in the subsequent versions of the FAQ. The faq is also available on the Web:

http://www.daimi.au.dk/~beta/faq/beta−language−faq.html

Two home pages are accessible on the World Wide Web, at the URLs

http://www.mjolner.com/

http://www.daimi.au.dk/~beta/

The Mjolner System − Online Documentation; available in two forms:

$BETALIB/doc/index.html:
Local copy of the documentation as released with release 5.0.

http://www.mjolner.com/mjolner-system/documentation/index.html:
Latest version of the documentation.

[MIA 90-02] The Mjolner System - Compiler Reference Manual; available in two forms:

$BETALIB/doc/compiler/index.html:
Local copy of the compiler documentation as released with release 5.0.

http://www.mjolner.com/mjolner-system/documentation/compiler/index.html:
Latest version of the compiler documentation.

O. Lehrmann Madsen, B. Moller-Pedersen, K. Nygaard: *Object-Oriented Programming in the BETA Programming Language*, Addison-Wesley, 1993, ISBN 0-201-62430-3.

J. L. Knudsen, M. Lofgren, O. L. Madsen, B. Magnusson (Eds.): *Object-Oriented Environments - The Mjolner Approach*, Prentice Hall, 1994, ISBN 0-13-009291-6.

1.9 BUGS

The BETA compiler does not currently implement the entire BETA language. A few constructs are not supported. For a precise description of the limitations (and additions), see the compiler reference manual.

1.10 AUTHORS

The BETA Compiler is developed by Mjolner Informatics as part of the Mjolner System.

Questions, bug-reports, etc. may be directed to support@mjolner.com if the local support organization cannot find solutions to the problems.

For more information of the Mjolner System, please contact Mjolner Informatics, Helsingforsgade 27, DK-8200 Aarhus N, Denmark, phone: +45 70 27 43 43, fax: +45 70 27 43 44, e-mail: info@mjolner.com, web: http://www.mjolner.com.

2 Manpage for mjolner

2.1 NAME

mjolner − Mjolner Integrated Development Tool

2.2 SYNOPSIS

mjolner *file*...

2.3 AVAILABILITY

The Mjolner Integrated Development Tool is available as part of the Mjolner System from Mjolner Informatics.

2.4 DESCRIPTION

Mjolner is a general structure editor, especially targeted for browsing and editing BETA programs. *Mjolner* is an integrated development tool that consists of the following components: A source browser, a general struc− ture editor, especially targeted for browsing and editing BETA programs, a source level debugger, a GUI editor (inter− face builder) a class diagram editor (CASE tool).

Mjolner is integrated with the BETA compiler *beta* (*1*). This integration gives a good support for locating and correcting semantic errors.

2.5 OPTIONS

(*none*)

2.6 ENVIRONMENT

mjolner recognizes the environment variable $BETALIB. Please note that this variable is given default value in the Bourne Shell script $*BETALIB/configuration/env.sh*

BETALIB

Specifies where ~*beta* is located. If not set, *beta* is assumed to be a username, and ~*beta* being the home directory of that user. Is used by many tools in the Mjolner System.

EDITOR

If specified, simple textediting (as opposed to structure editing) can be done using the text editor, specified by EDITOR. Simple textediting is activated using the 'External textedit' command of the Edit menu of *mjolner*. After textediting, the modified text is parsed according to the corresponding syntactic category. E.g *emacs* (*1*) can be used by setting EDITOR to */usr/local/bin/emacsclient* and starting *emacs* as a server by *emacs −f server−start*.

2.7 SEE ALSO

beta(1) − BETA Compiler

betatar(1) − BETA **tar(1)** front−end

betafs(1) − Mjolner BETA Fragment Structure Lister

betawc(1) − BETA fragment analyzer

The Mjolner System − Online Documentation; available in two forms:

$*BETALIB/doc/index.html*:
Local copy of the documentation as released with release 5.0.

http://www.mjolner.com/mjolner−system/documentation/index.html:
Latest version of the documentation.

[MIA 99−39] The Mjolner System: Mjolner Integrated Development Tool − Overview; available in two forms:

$*BETALIB/doc/mjolner−overview/index.html*:
Local copy of the Mjolner Integrated Development Tool overview as released with release 5.0.

http://www.mjolner.com/mjolner−system/documentation/mjolner−overview/index.html:
Latest version of the Mjolner Integrated Development Tool overview.

[MIA 99−40] The Mjolner System: Mjolner Integrated Development Tool − Tutorial; available in two forms:

$*BETALIB/doc/mjolner−tut/index.html*:
Local copy of the Mjolner Integrated Development Tool tutorial as released with release 5.0.

http://www.mjolner.com/mjolner−system/documentation/mjolner−tut/index.html:
Latest version of the Mjolner Integrated Development Tool tutorial.

[MIA 99−34] The Mjolner System: Mjolner Integrated Development Tool − Reference Manual; available in two forms:

$*BETALIB/doc/mjolner/index.html*:
Local copy of the Mjolner Integrated Development Tool reference manual as released with release

5.0.

http://www.mjolner.com/mjolner−system/documentation/mjolner/index.html:
Latest version of the Mjolner Integrated Development Tool reference manual.

2.8 AUTHORS

Mjolner is developed by Mjolner Informatics as part of the Mjolner System.

Questions, bug−reports, etc. may be directed to support@mjolner.com if the local support organization cannot find solutions to the problems.

For more information of the Mjolner System, please contact Mjolner Informatics, Helsingforsgade 27, DK−8200 Aarhus N, Aarhus C, Denmark, phone: +45 70 27 43 43, fax: +45 70 27 43 44, e−mail: info@mjolner.com, web: http://www.mjolner.com.

3 Manpage for betatar

3.1 NAME

betatar − BETA archiving program

3.2 SYNOPSIS

betatar [−−help|−h] [−−extent|−e] [−−domain|−d] [−−full|−f] [−−ast|−a] [−−asm|−s] [−−code|−c] [−−debug|−b] [−−job|−j] [−−dump|−u] [−−total|−t] [−−ignore|−x *rexps*] [−−include|−i *rexps*] [−−verbose|−v] [−−compress|−m] [−−gzip|−g] [−−zip|−z] [−−list|−l] *file*...

3.3 AVAILABILITY

The Mjolner System **betatar(1)** utility is available as part of the Mjolner System from Mjolner Informatics.

3.4 DESCRIPTION

betatar is an archiving program. **betatar(1)** makes use different external programs, such as **tar(1)** or **zip(1)** and for compressing the files, **betatar(1)** makes use of **compress(1)** or **gzip(1)** Which is actually used depends on the options (see later).

betatar is used to create an archive of all files related to a BETA fragment file (BETA source files, etc.) *betatar* is intended to be used for packaging the entire set of files, contributing to a given BETA program in order to move this program to another installation for further work (debugging or further development). Typical usages include moving a program between, say, a PC at home and the UNIX workstation at work, or between different development teams.

betatar offers many different options for controlling which files are packaged into the archive file. The most important aspect of *betatar* is that it ensures that all necessary files are collected into the archive file (except if certain options are used − see later), such that, when unpacked, all needed files will be in place for continued work.

betatar works by traversing the dependency graph of the fragment graph, following ORIGIN, INCLUDE, BODY and MDBODY fragment links, starting at the fragment specified in the file given as argument to *betatar*. During this traversal (identical to the dependency analysis conducted by the BETA compiler), *betatar* selects the fragments to be included in the archive file, depending on the different options specified to *betatar*.

3.5 OPTIONS

```
 -h
--help
     Print this help info
 -e
--extent
     Traverse the entire dependency  graph,  including  BODY
     and MDBODY fragments.
 -d
--domain
     Traverse the dependency graph, ignoring BODY and MDBODY
     fragments.
 -f
--full
     Choose all fragments found during the traversal of  the
     dependency  graph.   This  includes  standard libraries
     from the Mjolner BETA System.  If --full is not  speci-
     fied,  all  fragments  located in $BETALIB/ are ignored
     (i.e. standard files are not packed  into  the  archive
     file).
 -a
--ast
     Include .ast/.astL files in the archive  file  (if  the
     corresponding .bet file are selected).
 -s
--asm
     Include ..s assembler files (if present) in the archive
     file (if the corresponding .bet file are selected).
 -c
--code
     Include .o code files (if present) in the archive  file
     (if the corresponding .bet file are selected).
 -b
--debug
     Include ..db debug files (if present)  in  the  archive
     file (if the corresponding .bet file are selected).
 -j
--job
     Include the .job file (if present)
 -u
--dump
     Include the .dump file (if present)
 -t
--total
     Include all file types (equiv. to "--ast  --asm  --code
     --debup --job --dump")
 -x
--ignore string
     Ignore fragments found in the traversal if they contain
     rexps  in  their  filename (e.g. --ignore basiclib will
     result in all fragments containing 'basiclib'  will  be
     ignored.  Note: rexps may be any regexp pattern.
     This option may have more than one "rexp" in  the  com-
     mand line, with different regexps.  The effect will be,
     that all these fragments are ignored.  "--ignore  rexp"
     has precedence over all above options.
 -i
--include string
     Include fragments found in the traversal if  they  con-
     tain  rexps  in their filename (e.g. --include basiclib
     will result in all fragments containing 'basiclib' will
```

be included. Note: *rexps* may be any regexp pattern. This option may have more than one "rexp" in the command line, with different regexps. The effect will be, that all these fragments are included. "−−include rexp" has precedence over all above options.

−v
−−verbose

print what is saved onto the archive file

−m
−−compress

compress the archive file

−g
−−gzip

use **gzip(1)** instead of **compress(1)** to compress the archive file.

−z
−−zip

use **zip(1)** instead of **tar(1)** to pack the files.

−l
−−list

list the files to be packed. Do not actually pack the files.

3.6 ENVIRONMENT

betatar recognizes the environment variable $BETALIB. Please note that this variable is given default value in the Bourne Shell script $*BETALIB/configuration/env.sh*

BETALIB

Specifies where *~beta* is located. If not set, *beta* is assumed to be a username, and *~beta* being the home directory of that user. Is used by many tools in the Mjolner System.

3.7 SEE ALSO

beta(1) − BETA Compiler

mjolner(1) − Mjolner Integrated Development Environment

betawc(1) − BETA fragment analyser

betafs(1) − Mjolner BETA Fragment Structure Lister

3.8 BUGS

Currently, *betatar* packs the files with full file path specifications, making it difficult to unpack the files at another location.

3.9 AUTHORS

The **betatar(1)** utility is developed by Mjolner Informatics as part of the Mjolner System.

Questions, bug−reports, etc. may be directed to support@mjolner.com if the local support organization cannot find solutions to the problems.

For more information of the Mjolner System, please contact Mjolner Informatics, Helsingforsgade 27, DK−8200 Aarhus N, Denmark, phone: +45 70 27 43 43, fax: +45 70 27 43 44, e−mail: info@mjolner.com, web: http://www.mjolner.com.

4 Manpage for betafs

4.1 NAME

betafs − Mjolner BETA Fragment Structure Lister

4.2 SYNOPSIS

betafs *file*...

4.3 AVAILABILITY

The Mjolner BETA Fragment Structure Lister is available as part of the Mjolner System from Mjolner Informatics.

4.4 DESCRIPTION

Betafs is a small utility for printing out the entire dependency graph (in textual format) of a fragment file. Is usefull for documentation purposes, and for identifuing possible version problems in the dependency graph.

4.5 OPTIONS

(*none*)

4.6 ENVIRONMENT

betafs recognizes the environment variable $BETALIB. Please note that this variable is given default value in the Bourne Shell script $*BETALIB/configuration/env.sh*

BETALIB

> Specifies where ~*beta* is located. If not set, *beta* is assumed to be a username, and ~*beta* being the home directory of that user. Is used by many tools in the Mjolner System.

4.7 SEE ALSO

beta(1) − BETA Compiler

mjolner(1) − Mjolner Integrated Development Environment

betatar(1) − BETA **tar(1)** front−end

betawc(1) − BETA fragment analyzer

4.8 AUTHORS

Betafs is developed by Mjolner Informatics as part of the Mjolner System.

Questions, bug−reports, etc. may be directed to support@mjolner.com if the local support organization cannot find solutions to the problems.

For more information of the Mjolner System, please contact Mjolner Informatics, Helsingforsgade 27, DK−8200 Aarhus N, Aarhus C, Denmark, phone: +45 70 27 43 43, fax: +45 70 27 43 44, e−mail: info@mjolner.com, web: http://www.mjolner.com.

5 Manpage for betawc

5.1 NAME

betawc − BETA fragment analyser

5.2 SYNOPSIS

betawc [−−help|−h] [−−all|−a] [−−conflict|−c] [−−full|−f]
[−−ignore|−x] [−−include|−i] [−−list|−l] *file*...

5.3 AVAILABILITY

The Mjolner BETA fragment analyzer is available as part of the Mjolner System from Mjolner Informatics.

5.4 DESCRIPTION

betawc is a BETA fragment analyzer. *betawc* is used to find out how many fragment groups, how many fragment forms, how many lines of code, how many words, and how many characters are in the dependency graph of the specified fragment.

Furthermore, *betawc* tries to evaluate which libraries are used in the specified fragment (either directly or indirectly throughout the entire dependency graph), and gives a listing of all fragment groups found during the analysis of the dependency graph of the specified fragment.

During the analysis of the dependency, *betawc* will give proper warnings of suspicious library versions, in the sense that there seems to be two versions of the same library in use in the same dependency graph.

betawc works by traversing the dependency graph of the fragment graph, following ORIGIN, INCLUDE, BODY and MDBODY fragment links, starting at the fragment specified in the file given as argument to *betawc*. During this traversal (identical to the dependency analysis conducted by the BETA compiler), *betawc* selects the fragments to be included in the further analysis (count of fragment forms, etc.).

The different *betawc* options may be used to control the part of the dependency graph that are analysed, and the types of analysis conducted (see below).

5.5 OPTIONS

−a
−−all

same as −−conflict −−full −−list

−c
−−conflict

check for version conflicts

−f
−−full

include standard files (files located in $BETALIB)

−x
−−ignore *rexp*

ignore all files with names containing *rexp*. Note, that *rexp* may be any regexp pattern. "−−ignore rexp" has precedence over all above options.

−i
−−include *rexp*

include all files containing *rexp*. Note, that *rexp* may be any regexp pattern. "−−include rexp" has precedence over all above options.

−l
−−list

list the fragment groups in the dependency graph.

−h
−−help

Print this help info

5.6 ENVIRONMENT

betawc recognizes the environment variable $BETALIB. Please note that this variable is given default value in the Bourne Shell script $*BETALIB/configuration/env.sh*

BETALIB

Specifies where ~*beta* is located. If not set, *beta* is assumed to be a username, and ~*beta* being the home directory of that user. Is used by many tools in the Mjolner System.

5.7 SEE ALSO

beta(1) − BETA Compiler

mjolner(1) − Mjolner Integrated Development Environment

betatar(1) − BETA archiving program

betafs(1) − Mjolner BETA Fragment Structure Lister

5.8 AUTHORS

The BETA fragment analyzer is developed by Mjolner Informatics as part of the Mjolner System.

Questions, bug−reports, etc. may be directed to support@mjolner.com if the local support organization cannot find solutions to the problems.

```
For more information of the Mjolner System,  please  contact
Mjolner  Informatics,  Helsingforsgade 27, DK−8200 Aarhus N,
Aarhus C, Denmark, phone: +45 70 27 43 43, fax: +45 70 27 43
44, e−mail: info@mjolner.com, web: http://www.mjolner.com.
```

6 Manpage for psbrowser

6.1 NAME

psbrowser − Mjolner BETA Persistent Store Browser

6.2 SYNOPSIS

psbrowser

6.3 AVAILABILITY

The Mjolner BETA Persistent Store Browser is available as part of the Mjolner System from Mjolner Informatics.

6.4 DESCRIPTION

psbrowser is a generic object browser for browsing in the object structures stored in a persistent store.

The psbrowser is specialized to browse objects found in a persistent store (~beta/persistentstore/). Psbrowser is able to browse objects whose code is not linked int o the browser executable.

Having launched the psbrowser, your next move is to open a persistent store. Do so by selecting "Open..." from the "File" menu. In the resulting dialog, enter the name of the persistent store and click the "Ok" button. A nested rootlist listing the names of the persistent roots will appear. The name of the rootlist window corresponds to the name of the persistent store. It is allowed to open multiple persistent stores at a time.

To open a window showing a persistent root object, double−click the name of that root in the rootlist. An ObjectView showing the root object will appear.

The objectview uses abstract presentation of the objects presented. This means that nested partobjects are initially shown contracted, i.e. as three dots. By double−clicking a line of the object view ending in '...', the hidden details will be shown. By double−clicking the same line again, the details are hidden.

Each line in the objectview corresponds to some attribute of the object. Simple attributes (@Char, @Integer, ...) cannot be further detailed, whereas other kinds of attributes can.

It is not possible to browse stores created on architectures with an endian different from that of the machine runnning the psbrowser (e.g. between Intel on the one hand and either

Motorola, HP−PA, SGI−MIPS or SPARC on the other hand).

However, it is a requirement that the persistent store is created, using the so−called 'full name patch'.

See the file ~beta/objectbrowser/psbrowser/README for more information on how to use psbrowser.

6.5 OPTIONS

(none)

6.6 SEE ALSO

mjolner(1) − Mjolner Integrated Development Environment

beta(1) − BETA Compiler

betawc(1) − BETA fragment analyzer

betatar(1) − BETA **tar(1)** front−end

betafs(1) − Mjolner BETA Fragment Structure Lister

6.7 AUTHORS

Psbrowser is developed by Mjolner Informatics as part of the Mjolner System.

Questions, bug−reports, etc. may be directed to support@mjolner.com if the local support organization cannot find solutions to the problems.

For more information of the Mjolner System, please contact Mjolner Informatics, Helsingforsgade 27, DK−8200 Aarhus N, Denmark, phone: +45 70 27 43 43, fax: +45 70 27 43 44, e−mail: info@mjolner.com, web: http://www.mjolner.com.