Support Toolbox Manual

Chapter 3

Documentation Utilities

Copyright © 1993, 1994 Khoral Research, Inc. All rights reserved.

Chapter 3 - Documentation Utilities

A. Documentation Utilities

This chapter describes the documentation utilities included in the support toolbox:

- kdoc2html, which generates html from documentation written with the Khoros macros.
- kdocdepend, for generating Makefile dependencies for a manual chapter, by recursing through the manual.
- kgentbindex, which generates an index for a toolbox.
- kgenman, which generates an explicit manpage.
- , which is used to generate a table of information about the objects within one or more toolboxes.
- kmkcontents, which generate a table of contents for a manual, based on the contents of each chapter in the manual.
- kwordfreq, which generates a list of word frequencies for a manual chapter, which can be useful when indexing your documentation.
- kgenwhatis, which updates the *whatis* database, which is used when kman is invoked with the **-k** switch.

A.1. kdoc2html — *convert khoros documentation to HTML*

Object Information

Category:	*Command Line Tools*
Subcategory:	*Documentation Tools*
Operator:	*Doc To HTML*
Object Type:	*script*
In Cantata:	*No*

Description

kdoc2html converts a manual chapter written in the khoros documentation macros, and generates equivalent (or near equivalent) HTML (the HyperText Markup Language used by the World-Wide Web).

Having written a manual chapter, the imake system support an *html* make target, which will invoke *kdoc2html* on the chapter's *.man* file:

```
% make html
```

Examples

The following command generates an html version of the first chapter of the SUPPORT toolbox manual:

```
% kdoc2html -i S.1.introduction.man -o S.1.introduction.html
```

Restrictions

At the moment, *kdoc2html* generates a single html file. In the future it will generate a structured html document, with options to control the hierarchy generated.

A.2. kdocdepend — *generate Makefile dependencies for a manual chapter*

Object Information

Category:	*Command Line Tools*
Subcategory:	*Documentation Tools*
Operator:	*Generate Doc Dependencies*
Object Type:	*script*
In Cantata:	*No*

Description

kdocdepend is used to generate make(1) style dependencies, which are then appended to the Makefile for a manual chapter. *kdocdepend* recurses through the chapter, following all .so and .include links.

The make system uses *kdocdepend* for the *depend* target, which is generated for all manual chapters automatically. So, you shouldn't have to invoke *kdocdepend* by hand, just run the following command from the manual's directory:

```
% make depend
```

Examples

The following will generate dependencies for the first chapter of the SUPPORT toolbox manual:

```
% kdocdepend -target manual.ps -toplevel manual.man \
        -s '# DEPENDENCIES: do not delete'
```

A.3. kgentbindex — *generate index for a toolbox manual*

Object Information

Category: *Command Line Tools*
Subcategory: *Documentation Tools*
Operator: *Generate Index*
Object Type: *kroutine*
In Cantata: *No*

Description

kgentbindex is used to generate the index chapter of a manual, by combining the index information from each chapter within the manual.

Command Line Arguments

```
Usage for kgentbindex: generate index for a toolbox manual
% kgentbindex

  Required Mutually Exclusive Group - Specify exactly one of the 2 options:
    [-alltb] (flag)   Generates Index for all toolbox manuals
    [-tb]    (string) Generate a manual index for a single toolbox
             (default = null)
```

A.4. kgenman — *generate man page(s) for a software object*

Object Information

Category: *Command Line Tools*
Subcategory: *Documentation Tools*
Operator: *Generate Manpage*
Object Type: *kroutine*
In Cantata: *No*

Description

This program parses a khoros public routine header for information, and generates a man 3 page for the public library routine. This routine takes a as input the library object the user wants man 3 pages for, and creates all the man 3 files for every public routine in a single file, or in the whole library. It then installs the man page in the directory of choice, or the final installation directory in $TOOLBOX/man/object_name/man.

Note, all the information stored between the lines ".SH ADDITIONAL INFO" and ".SH SIDE EFFECTS" is saved and propagated to the new man page. Thus, the user can put extra information in this section of the man page, such as an example section, if desired.

This program will detect when headers have missing fields, and recover gracefully. If a field is missing, it will indicate which field, routine, and filename the error occurred in, and continue on with that field left empty. Therefore, it is the programmers responsibility to fix the header, and re-run kgenman on that file, so that the man page will be complete.

The example field below has an example of the public header along with an explaination of each field that goes in it.

Command Line Arguments

```
Usage for kgenman: generate man page(s) for a software object
% kgenman
  -tb        (string) toolbox the library belongs to
  -lib       (string) library to generate man pages for

  Required Mutually Exclusive Group - Specify exactly one of the 3 options:
    [-dir]      (string) generate and install in the directory specified
                (default = .)
    [-install]  (flag)   Install the man pages in $TOOLBOX/man/man3
    [-list]     (flag)   List the man pages rather than generate them

  Required Mutually Exclusive Group - Specify exactly one of the 2 options:
    [-all]  (flag)   create man pages for all src in the library object
    [-i]    (string) input source code file
            (default = null)

  [-section] (string) Man section to put the man page into
             (default = null)
```

A.5. kobjtable — *generate table of toolbox objects information*

Object Information

Category: *Command Line Tools*
Subcategory: *Documentation Tools*
Operator: *Generate Object Table*
Object Type: *script*
In Cantata: *No*

Description

kobjtable is used to generate a table which contains information about the objects in one or more toolboxes. The manual for a toolbox should include such as table, since it provides a concise summary of the toolbox's contents.

kobjtable can generate two styles of tables: one is oriented towards command-line tools (requested with the -clui switch), and the other is oriented towards **cantata** (requested with the -gui switch), with objects grouped according to their *category* and *subcategory*, the same organization used for the operator menus in **cantata**. The cantata-oriented table is the default presentation format.

Examples

The following command generates an object information table for the SUPPORT toolbox, which can be seen in the first chapter of the SUPPORT toolbox manual:

```
% kobjtable -tb support -o manifest.tbl \
        -title "SUPPORT toolbox Object Manifest"
```

A.6. kmkcontents — *generate table of contents for a manual*

Object Information

Category:	*Command Line Tools*
Subcategory:	*Documentation Tools*
Operator:	*Generate Table of Contents*
Object Type:	*script*
In Cantata:	*No*

Description

kmkcontents generates the table of contents page(s) for a manual, based on the chapters contained within the manual.

Restrictions

This man page is a little too succinct.

A.7. kwordfreq — *generate word frequencies for documentation*

Object Information

Category:	*Command Line Tools*
Subcategory:	*Documentation Tools*
Operator:	*Generate Word Frequencies*
Object Type:	*script*
In Cantata:	*No*

Description

kwordfreq reads one or more documentation files, and produces a table of work frequencies, sorted by number of occurences. This can be useful when trying to determine helpful index words. The output table is generated in the file "kwordfreq.log".

Examples

To generate word frequencies for all *.sec* files in a given manual, you would use the following command:

% kwordfreq *.sec

A.8. kgenwhatis — *generate database of available manpages (whatis)*

Object Information

Category:	*Command Line Tools*
Subcategory:	*Documentation Tools*
Operator:	*Generate whatis*
Object Type:	*kroutine*
In Cantata:	*No*

Description

kgenwhatis is used to regenerate the 'whatis' database, used by *kman*. This database holds information about the manpages available within a toolbox, and is updated automatically by the code generation routines, but this program is provided in case the database becomes corrupted or needs updating manually for some other reason.

Examples

The following command will update the manpage database for the *design* toolbox:

% kgenwhatis -tb design

See Also

vman(1), intro(1)

Command Line Arguments

```
Usage for kgenwhatis: generate database of available manpages (whatis)
% kgenwhatis
  -tb        (string) toolbox name
```

Table of Contents

This page left intentionally blank