Support Toolbox Manual

Chapter 2

DBM Tools

Viewing and Editing
KDBM Databases

Chapter 2 - DBM Tools

A. Introduction

This chapters describes two tools which are useful when working with **kdbm** databases. A description of the **kdbm** database and API can be found in chapter 2 of the manual *Programming Services volume 1*. There are two major places where **kdbm** databases are used in the foundation toolboxes:

1. The Visualization and Image File Format (VIFF), is stored as a **kdbm** database.

2. The software services store configuration and other information in **kdbm** databases. For example, each toolbox has a database (in `repos/db/cms`) which holds information about the organization and contents of the toolbox. Each object also has a database, in the object's `db/` directory.

The tools described in this chapter are currently only useful for the second category, since the tools assume ascii contents, which is not true for VIFF databases.

B. kdbmcat

kdbmcat is a simple kroutine which prints out the contents of a KDBM database, in the format:

`'key':` *data stored under that key*

The following example output is from the toolbox database for a toolbox called **kperl**, containing one software object, **klsobj**, of type script.

```
'TB_NAME': KPERL
'CONFIG_FILE': $KPERL/repos/config/imake_config/kperl.def
'AUTHOR': Khoros Development Account
'CMS_VERSION': 1
'STATUS': 2
'SOFTWARE_OBJECTS': klsobj:script

'AUTHOR_EMAIL': neilb@khoros.unm.edu
```

There are three ways to use kdbmcat:

1. To print the contents of a toolbox database, use the **-tb** switch:

   ```
   % kdbmcat -tb design
   ```

2. To print the contents of a software object database, use the **-tb** and **-oname** switches:

   ```
   % kdbmcat -tb design -oname cantata
   ```

3. You can provide an explicit path using the **-dbm** switch:

   ```
   % kdbmcat -dbm ~/toolbox/repos/db/cms
   ```

C. kdbmedit

kdbmedit is used to edit KDBM databases, the operations available being:

- Edit the value stored under a key, using a text editor.
- Add a new key to the database.
- Delete an exiting key from the database.

C.1. Command Line Arguments

In addition to the standard Khoros command-line arguments, kdbmedit understands two arguments:

-dbm *filename*
which specifies the database file to be edited.

-view
which specifies that kdbmedit should run in read-only (or view) mode. In view mode the Edit, Add, and Delete buttons are unmapped.

C.2. The User Interface

The user interface for kdbmedit is shown in figure 1.

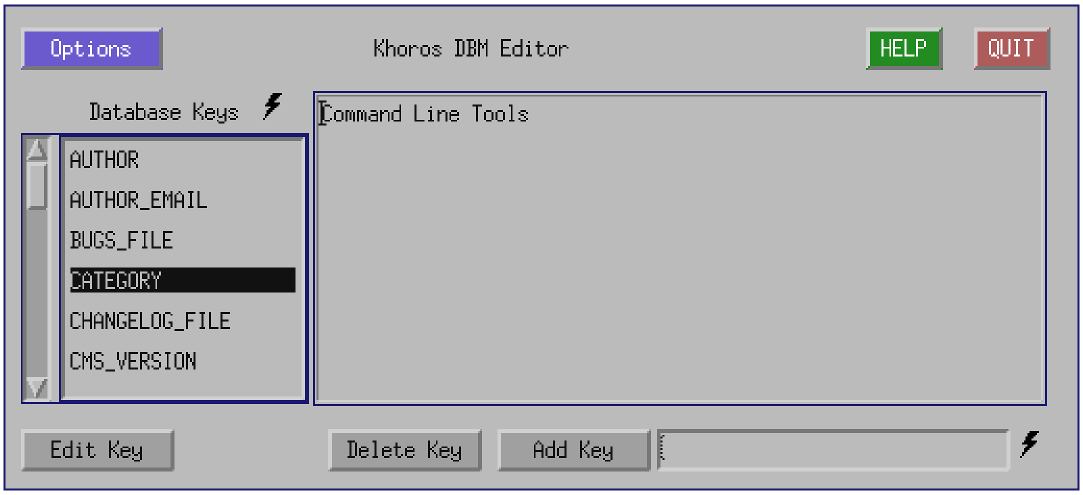

Figure 1: The user interface for kdbmedit, being used to edit the database for a software object.

C.2.1. Database Keys

The keys in the database are presented in the scrolled list widget at the left of the interface. The keys are sorted alphabetically. When you select a key, the contents stored under that key are displayed in the text widget to the right of the scrolled list.

C.2.2. Editing the selected key

Having selected a key in the scrolled list, you can edit the contents of that key. When you hit the *Edit Key* button, kdbmedit will spawn a text editor on a temporary file which contains the text stored under the selected key. If you change the contents of the key, and write out the file, then when you quit the editor kdbmedit will update the database.

The text editor used can be specified using the `KHOROS_EDITOR` environment variable, as described in Chapter 2 of the *Getting Started* manual. If this variable is not set, then kdbmedit will also check the `EDITOR` and `VISUAL` environment variables.

C.2.3. Deleting the selected key

Having selected a key in the scrolled list, hitting the *Delete Key* button will remove that key (and associated data) from the database. You will not be prompted!

C.2.4. Adding a key to the database

To add a new key to the database, you can either type in the new key, and press *carriage return*, or you can use the pulldown menu on the *Add Key* button, which contains the standard keys used for toolbox and software object databases.

This page left intentionally blank

Table of Contents

This page left intentionally blank