

Searching for Answers with Kdict

DIGITAL DICTIONARY



www.sxc.hu

Don't worry if your memory lets you down. All you need is a place to look for answers. If you want definitions, explanations, and vocabulary, Kdict, KDE's dictionary utility, is the tool you need.

BY STEFANIE TEUFEL

Are you sometimes stumped by arcane technical terms? No need to worry – that's what dictionaries are for. And if you have the *Kdict* program, you will also have a virtual translation tool to help you search for tricky terms in any language. The installation is no big deal, since the application has been part of the *kdenetwork* package for

quite a while now. Suse users will find the program under the pseudonym *kdenetwork3-query*.

Kdict uses the DICT protocol [1] to provide dictionary services; DICT searches databases located on Internet servers for definitions of the key words you enter. If you are hiding behind a firewall, or if you do not have a permanent

Internet connection – or if an Internet-based search approach is too expensive for your taste – you can set up a server locally instead. In addition to improved speed and reduced expense, a local server also allows you to install additional databases.

Installing the Server

Installing the server involves a few simple steps. First, download the package with the DICT daemon *dictd* from [2]. The current version is 1.9.7. Suse users can leave out this step, as the daemon is installed by default.

Users with other distributions who are unable to locate an appropriate RPM package at [3] will need to download the sources and build the daemon themselves. To do so, first unpack the sources, and then follow the usual three steps: *./configure*, *make*, and *make install*.

A server does not make much sense without reference material, so you should go hunting for dictionary databases before you launch the program.

Word Sources

Some useful dictionary resources on the Internet include:

- Webster: "Webster's Revised Unabridged Dictionary, 1913 Edition" is a well-known English language dictionary with a neat search facility (including wildcards) that now has over 100,000 words.
- Wordnet: Work on WordNet started in 1985 and just keeps on going. The project is mainly based on work by psychologists and linguists at Princeton University. The dictionary structure is not based on alphabetic order, but on psycholinguistic theories on human memory. Thus, WordNet will distinguish between the word form

- and word meaning for every term. The latter resolves to a large number of synonymous word forms. WordNet is not restricted to any specific subject and now has over 95,600 English word forms. This thesaurus is a useful tool when you are looking for a different way of expressing something.
- Jargon File: Eric Raymond's "The New Hacker's Dictionary" (this is the official name) is probably the best-known reference for Internet and hacking slang. It defines and explains networking and computer terms, often in a very humorous way.

You'll find these and other dictionary resources at <http://www.dict.org>.

You'll find a large selection of resources, including Webster (dict-web1913), Wordnet (dict-wn), and the Jargon File (dict-jargon) on the DICT project's FTP server at [4].

If this selection is not enough for you, you can check out [5] and [6] for a collection of preformatted databases. Be careful with the dict.org [1] link list. Many of the data repositories are in formats that *dictd* does not understand.

After selecting a few databases, unpack the files in a directory of your choice (for example `/usr/local/db`). Then configure local DICT access via the *dictd.conf* file. The *access* command allows you to specify who will have access to the server. Your best bet is to opt for something along the lines of *access {allow localhost *}*. This restricts access to the local machine.

The database command specifies the location of the index and the database data records:

```
database web1913 {
    data "/usr/local/db/web1913.dict.dz"
    index "/usr/local/db/web1913.index" }
```

If you will be accessing multiple database files, you will need to add a line for each database. When you have completed the setup stage, you can type *dictd* & in a command line to launch your very own dictionary server.

Kdict in Action

Time to launch Kdict. The program menu entry on Red Hat is hidden in *Tools | Online dictionary (Kdict)*. Of course you can still launch the application via the command line by typing *kdict* &. In both cases, the welcome window shown in Figure 1 appears.

Before you start searching for missing words in Kdict, you might like to check the settings to determine which server the tool will be querying. To do so, select the *Settings | Configure Kdict...* | *Server* entry. If you prefer to keep the online variant, you can leave

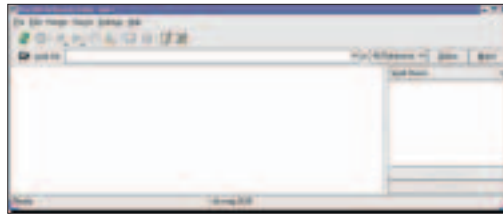


Figure 1: Kdict looks fairly sparse shortly after first launch.

everything as it is. The preselected server is dict.org, a public server.

If you use a local server, you will need to change the *dict.org* default to *localhost*, and then click on *Apply*. The other settings in this tab are only of interest to online users. For example, you can set a *Timeout* for the amount of time that Kdict should wait for a server response. The *Server | Server information* page (Figure 2) gives you more information on the server. Don't forget to select *Server | Get Capabilities* first, though, or there will be no information to view.

After querying the server data, you can select *Server | Strategy Information* to view the available search strategies. You can select *prefix* (this assumes that the search string is at the start of the entry), *regex* (this uses a regular expression for the search), or even *soundex*, which uses the SOUNDEX algorithm to search for words that sound similar to the one you are looking for.

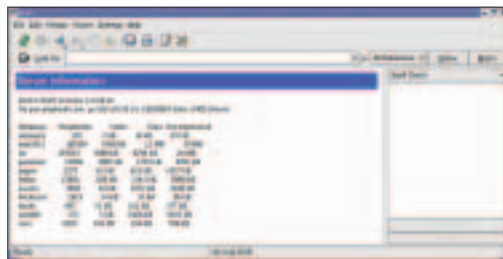


Figure 2: Kdict provides detailed status information on your current server.

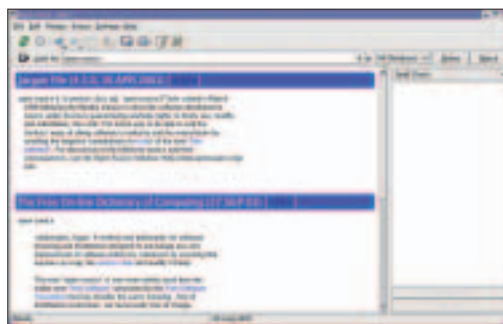


Figure 3: Kdict displays the search results in the left-hand panel.

Kdict searches for the entry in the *Search for* field and provides a list of results in the left-hand panel (Figure 3). You can tell Kdict which database to search in the pull-down menu next to the search box.

You can even define groups of available databases. These virtual databases are then displayed as peers in the pull-down menu. You can also set up customized groups using the *Server | Edit Database Sets ...* menu entry.

Cooperative

Two features that underline the seamless interaction between KDE programs should help Kdict win a position on your



Figure 4: The applet gives you easy access to Kdict.

list of favorite tools. If you stumble across a word you don't know while surfing the Internet, you can simply highlight

the word with your mouse in Konqueror and tell Kdict to search for the word by selecting *Edit | Match Clipboard Content*

Another practical feature is the ability to add Kdict to the panel as a panel applet. Right-click on the Kicker bar, and select *Configure Panel... | Add | Applet | Dictionary*. The panel applet gives you a tiny input field like the one shown in Figure 4, allowing you to type a query directly. The applet also has three additional buttons. *C* Searches for the the current content of the clipboard, *D* searches for the current content of the input field, and *M* searches for appropriate definitions.

The program even has a command-line mode: *kdict searchkey* will search for the term you enter. If your entry includes multiple words, put the search key in quotes. *kdict -c* or *kdict --clipboard* launches a query for the current clipboard content. ■

INFO

- [1] DICT Development Group: <http://www.dict.org/>
- [2] DICT daemon: <ftp://ftp.dict.org/pub/dict/>
- [3] Rpmfind.net: <http://rpmfind.net/>
- [4] DICT project dictionaries: <ftp://ftp.dict.org/pub/dict/pre/>
- [5] FreeDict project: www.freedict.de
- [6] Various dictionaries: <http://www.wh9.tu-dresden.de/~heinrich/dict>