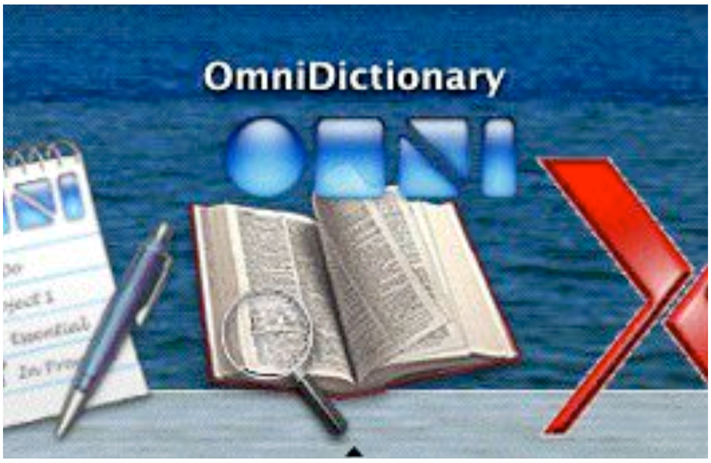
Mac OS X DICT Howto BETA



How to get a superior & free dictionary up and running on your Mac

a step-by-step walkthrough by Jens Ropers (www.ropersonline.com)

-> Please jump to Part II if you don't care for chatty introductions.
-> Please note that this document is in BETA stage. See the appendix for more info.

Part I: Introduction

Let me own up: I, too, am a switcher. And back and forth, too - but here I am again, writing on my iBook. ;-) Aaanyways, on Windows I got to know a nice program named WordWeb. A free dictionary. And none too shabby! (http://wordweb.info/) I couldn't find one like that for the Mac. I emailed the author, would he consider porting to Mac ? He wouldn't. (The good man had probably never touched a Mac, and was needlessly scared.) I then did some research on my own and found out that WordWeb was actually using a database from Princeton University called WordNet.
(http://www.cogsci.princeton.edu/~wn/)
Quite some time later, I came across the OmniDictionary program from the Omni Group. (http://www.omnigroup.com/) It could access public dictionary servers from the DICT Development Group. (http://www.dict.org/) To my delight I discovered that the WordNet database (among others) was being used there as well. To my dismay I discovered that this only worked as long as I was connected to the Internet.
What to do? It was mentioned in various places that it should be possible to get programs like OmniDictionary to work offline, i.e. without an Internet connection. Reportedly I could have a 'DICTD server' - basically the same software that runs on those servers at dict.org - to run locally on my own Macintosh, and then use OmniDictionary or another DICT client to access that.
(DICT, I learned, is actually a pretty open protocol, allowing you to access so-called DICTD servers with interfacing programs (DICT clients); so any compatible program can be used to connect to any compatible database, hosted on any

computer capable of running a DICTD server. Read all up on it at http://www.dict.org/.)

Part II: Decisions, decisions...

This document tells you how to set up an offline (or locally hosted; i.e. it's running on your own Mac) DICTD server for use with OmniDictionary under Mac OS X. This setup can then be used with a number of freely available dictionary databases, including the raved about WordNet.

Since I did that myself, I have found out about TWO WordNet frontend programs which are now available for Mac OS X. So if it's just WordNet you want, then following these instructions here is not required. Like the WordNet database itself, I find these programs quite brilliant and shamefully underreported. One is called WordNet.app, or just WordNet and can be got at http://www.mulle-kybernetik.com/software/WordNet/. The other is also called WordNet. That's right: both programs AND the database they use are called WordNet. Confused? Me too. Anyways, the second program can be gotten at: http://www2.ur.cmu.edu/~wtaysom/wordnet.html Despite using versions of the same database, they work quite different (at least in terms of their user interface), so see for youself which you prefer.

However, here's why you might STILL want to set up your DICTD server:

- You can get a number of very high quality free databases (dictionaries and other fact/info databases) for use with a DICTD server.
- A DICTD server allows you to use those multiple dictionaries concurrently. For instance I just looked up the word 'the' in OmniDictionary and got relevant results from 3 different dictionaries. And depending on the Dictionaries you use with your server, such results would often nicely supplement each other. (Plus WordNet doesn't even include words such as 'the'.)
- As I can access the WordNet database with my DICTD server, it's obviously nicer to just look something up in one program and get WordNet and other results.
- That said, the DICT system and the WordNet applications serve different purposes and supplement each other. The WordNet programs excel when you need a thesaurus, owing to WordNet's unique design and unequalled cross-searching capabilities. I mainly use OmniDictionary (together with my DICTD server and various databases), but I use one of the WordNet applications as an additional, more flexible thesaurus. (I'm still undecided with which one of the two WordNet applications I'll stick.)

For the record: I haven't seen any commercial dictionary/thesaurus program yet that would even come close to what DICT and WordNet offer.

Well then - let's install the DICT/OmniDictionary combo!
(Please see Appendix IV for a word on the installation of the WordNet applications.)

Part III: Getting Stuff

What comes before installing stuff? Right. Getting the files. Here's what to get, where and why:
What we need first is a DICTD server program. That program will run on our Mac, hosting whichever DICT databases we choose, and it will thus allow us to use DICT client programs such as OmniDictionary to access those databases to look up definitions. Sounds complicated? Let's do it step by step.
There is a wealth of info about DICT related stuff, including DICTD servers, on the dict.org resources page. (http://www.dict.org/links.html)
I decided on the JDictd DICTD server program, which is written in Java. The advantage to that? Mac OS X has a fully implemented Java environment built right in. So we can run the JDictd program pretty much out of the box. Maybe some of the other programs allow that, too; maybe they are even better, but I haven't tried all of them (only some and they didn't work as expected). There is another Java DICTD server listed on the dict.org site, but getting it to work would require more work as it expects the Bourne-Again Shell to be installed under /bin/bash and on a default Mac OS X installation that's not there. Was that waffle? Apologies. We can ignore this. Let's download JDictd at:
http://www.informatik.uni-leipzig.de/~duc/Java/JDictd/.
We've now gotten ourselves our server!

Next, we need some DICT databases. (There is only one included with JDict, which is not an actual dictionary but rather a chemical elements database - another example to the versatility of the DICT protocol.) We can get pre-formatted databases at: ftp://ftp.dict.org/pub/dict/pre/ (It's important our databases are preformatted; otherwise we can't use them until we get them formatted for use with the DICT protocol.) As of this writing, the following files are at that location:

devils-dict-pre.tar.gz	- The Devil's Dictionary
dict-foldoc-20020209.tar.gz	- The Free Online Dictionary Of Computing
dict-gazetteer-1.2-pre.tar.gz	- The Gazetteer
dict-gcide-0.44.tar.gz	- The GNU Collaborative Int'l. Dict. of English
dict-jargon_4.3.0.tar.gz	- The Jargon File, apparently just an older version
dict-jargon_4.3.1.tgz	- The Jargon File
dict-misc-1.5b-pre.tar.gz	- Two Bible DICTs. and the '95 CIA World Factbook
dict-vera_1.8.tar.gz	- The Virtual Entity of Relevant Acronyms
dict-web1913-1.4-pre.tar.gz	- The 1913 Webster's Unabridged Dictionary
dict-wn_1.7.tar.gz	- Princeton's WordNet
elements-20001107-pre.tar.gz	- A chemical elements database
foldoc-20010313.tar.gz	- An older FOLDOC version? (see above)

A quick word on each of them:

The Devil's Dictionary:
Relax, small town America! This one is not about satanism or anything like that. It's about good-humoured cynicism (which I assert exists) and moreover it's good for a few laughs. This is an electronic version of the book 'The Devil's Dictionary' by the writer Ambrose Beirce. It's basically a list of ironical or cynical definitions. An example: 'AMBITION, n. An overmastering desire to be vilified by enemies while living and made ridiculous by friends when dead.' You get the picture. Not a

serious dictionary.

The Free Online Dictionary Of Computing:
A serious and quite comprehensive dictionary of all things IT. Very useful. Incorporates some of the more objective definitions from the Jargon File.

The Gazetteer:
A database of ZIP codes, place names, their location, population and other info for the United States.

The GNU Collaborative International Dictionary of English:
This is the big one. A huge dictionary. Largely based on and incorporating the 1913 Webster (see below), this dictionary has been amended with a lot of more contemporary definitions by countless volunteers to the project. This very comprehensive dictionary alone is worth the hassle of installing a DICTD server.

The Jargon File:
An informal, often subjective dictionary of IT slang and/or hacker jargon. Very useful to the interested individual as it delivers a lot of background info. (Some of the definitions are almost short essays.) Not useful if you don't care for IT stuff. Again, most of the more objective and/or straightforward definitions from the Jargon File have been imported into the FOLDOC (see above), however that's sometimes minus the background info. So it's still useful to have both the FOLDOC and the Jargon File installed, though many definitions will be identical in both.

Easton's 1897 Bible Dictionary (part of the 'misc' file):
Almost a small encyclopedia on the Bible. Contains extensive references and definitions. Very useful to anyone with an interest in the Bible.

Hitchcock's Bible Names Dictionary (part of the 'misc' file):
Mostly very short and snappy definitions of biblical names. Good for quick reference.

The CIA 1995 World Factbook (part of the 'misc' file):
A wealth of information on most (if not all) of the world's countries. Type in a country's name and get a lot of facts from its geography up to its defence expenditures.

The Virtual Entity of Relevant Acronyms:
Well, it does exactly what it says on the tin. Ok, it is fairly IT centred as well and in my subjective impression relatively limited. Still quite useful.

The 1913 Webster's Unabridged Dictionary:
This is the full text of Webster's Revised Unabridged Dictionary from the year 1913. It is huge and very good quality. It is old however and unless you must have the original unadulterated Webster, there would be no reason not to go for the GCIDE (above) instead, which again includes the Webster text (together with its more contemporary amendments). Still, IMHO that's not much of a reason, as in the GCIDE, Webster definitions are clearly marked as such. Unlike the Jargon file and

the FOLDOC (which each have merits of their own), in case of the Webster and GCIDE it would be pretty pointless to install both.

Princeton's WordNet dictionary database:
Cannot be praised enough. It delivers extremely good and concise definitions and synonyms. However, unlike when used with its very own WordNet applications (see above and Appendix IV), e.g. searching for plural forms sometimes returns no results (you have to search for the singular or word root). That limitation apparently goes to some extent for most of these DICT databases, but I only noticed it with WordNet. Actually, this is not a big problem in practice. On my own Mac, I have configured WordNet definitions to come first (see below for how), so I first get a concise definition from WordNet and then all these comprehensive entries from the GCIDE.

The chemical elements database:
Enter the name of an element and get the symbol, atom weight and number and a definition/description. Useful to anyone interested in chemistry. Already included in the JDictd download.

I trust these explanations are reasonable clear. You obviously need to decide for yourself which dictionaries you want. I highly recommend getting WordNet and the GCIDE. They should cover about every common word. Then add any of the other dictionaries/databases as required - e.g. to supplement some specialist knowledge.

Watch out for more dictionaries from other sources as well - but make sure you are downloading preformatted DICT database files! I overlooked the appropriate link on dict.org and sucked megabytes over megabytes of useless files over a small analogue telephone line before I figured that out. Again, the right link is: ftp://ftp.dict.org/pub/dict/pre/ (It currently is not very prominent on the dict.org site, unfortunately.)

Right. All our desired DICT databases are downloaded? Grand. Let's go get our DICT client. OmniDictionary can be got at:
http://omnigroup.com/community/freeware/omnidictionary/
There are actually a number of DICT client programs we could use and I've tried most of them. I recommend OmniDictionary. Refer to Appendix VII for why that is and for my opinion on how other Macintosh DICT clients shape up.

We now have all required parts: A DICTD server program, one or more DICT databases, and a DICT client program. Let's install them!

Part IV: Setting me up

The installation locations I give here are really just recommendations. Even though Mac OS X is more restrictive than Mac OS 9 about what can be put where, it is still pretty much at your liberty where you actually prefer to put stuff. If you go for a different install location however, you obviously need to stick to that path throughout. Also, the following instructions tell you how to install JDictd for your own use only. If you want JDictd to be available to whoever uses your Mac, please refer to Appendix V. That said, here is what I recommend:

Create a JDictd folder in the Applications folder in your Home folder. (That's ~/Applications/JDictd. The symbol ~ stands for your Home folder - so please go into that and then into the Applications folder inside it and create a folder named JDict in there.) Copy the contents of the extracted jdictd.zip file into that newly created folder. (All of the downloaded files should have gotten unpacked automatically into own subfolders upon download completion. If they didn't, you should be able to manually unpack them using e.g. StuffIt Expander.)

Next, we need to create a file. Instead of me providing that file for you, I am giving instructions how to create it: That way you won't be in the stupid situation of missing that one crucial file if this very document should reach you without it. Also, this gives everyone the opportunity to understand what is going on and to make changes if desired.

Select and copy the following lines (no need to understand them, just copy them): (You may want to use that text selection tool in Acrobat. If you're just looking at a printout, you could just type things instead.)

```
#! /bin/tcsh
echo JDictd is starting up...
cd ~/Applications/JDictd
java -cp jdictd.jar org.dict.server.JDictd data/dict.ini
```

Then open a Terminal window.
Type the following:

```
cd ~/Applications/JDictd [press Return]
pico startdictd.sh [press Return]
```

This will have opened Pico with a new and empty file named startdictd.sh. Pico is a basic text editor from the UNIX world and it runs inside the Terminal. The reason we are using Pico and not Mac OS X' own TextEdit is that it's potentially trickier to tell TextEdit to create our file in the primitive text format we need than to use Pico. That goes even if you've never used Pico before. And why is that? Because even basic TextEdit tries to lay out its text nicely - which is great most of times, but right now TextEdit would save our text differently than we need it. Now all we do is pasting our text into Pico/into the Terminal. Then we press Ctrl+O. It prompts us for the filename, but our filename startdictd.sh is already in there, so we just press Return. It will tell us that it has saved something. After that, we press Ctrl+X to close Pico. We're back to our Terminal input prompt now. Now let's please type in:

```
chmod u=rwx startdictd.sh [press Return]
```

This is to allow ourselves to run the startdictd.sh file - because what we've just created here is really an executable UNIX shell script file. And UNIX is a pretty controlled or at least controllable environment. Hence appropriate permissions are required for every action in UNIX (well, almost). Ever wondered why the heck you need to click the lock and enter a password to perform installations on Mac OS X? It's basically for that very same reason. (UNIX experts would protest this to be a sweeping generalisation, but hey... I trust this explanation will do for the rest of us ;-) Anyways, let's close Terminal for now.

Now use Finder to go into that ~/Applications/JDictd folder. Select the startdictd.sh file and go for Show Info on that. Change the 'Open with application' setting to Terminal. You have to go to 'Other...' and change 'Recommended Applications' to 'All Applications' to do that. (Terminal, just to remind, should be under /Applications/Utilities.) Close that startdictd.sh Info window. Note: If you use TextEdit to edit and re-save startdictd.sh after this (e.g. by going TextEdit - File - Open...), the file will loose it's setting to get executed by Terminal, unless you go for 'Change all...' (not recommended).

Our startdictd.sh file is now finished. Give it a go and double-click on it in Finder. You should see something like this:

```
Welcome to Darwin!
[localhost:~] ropers% /Users/ropers/Applications/JDictd/startdictd.sh; exit
JDictd is starting up...
Mon Aug 05 01:02:37 IST 2002: JDictd started at 2628
Mon Aug 05 01:02:37 IST 2002: HTTP server started at 2626
```

That is our DICT server running. Just to show that it works (and this is a bit of a digression): open your favourite web browser. Note: on my system, this did not appear to work in MS Internet Explorer. Any other browser seemed fine (tried with Mozilla, Netscape, OmniWeb and Opera). Just enter:

```
http://localhost:2626/
```

in the address bar. That actually already brings up an interface where we can look stuff up. Ooops. It turns out or DICTD server is its own DICT client as well! Well, that's right: This is an additional functionality provided by the JDictd server program. Some might actually prefer this over a proper DICT client program, but I don't. We currently already have the aforementioned elements database installed with our DICTD server, because it came included with JDictd. So go on then: look up the word carbon! Not too bad, eh?

Of course looking up chemical elements is of limited use for language and writing purposes. It's time to install the downloaded databases! Please exit the JDictd server for now (Ctrl+C); that is if you haven't already done so. Close that Terminal window.

Each of these preformatted databases, in their extracted form, consists of two files:

xyz.index and
xyz.dict.dz

where *xyz* is to be substituted by each dictionary's name (or an abbreviation thereof). It's these files that we need to copy into the ~/Applications/JDict/data folder. Once that's done, we need to tell JDictd that those new databases are there. To do this, we need to edit another file. This time we'll be fine to use TextEdit. (Because the file already exists in the correct format and TextEdit will respect that.) So in the ~/Applications/JDict/data folder, double-click on the dict.ini file. It should look like this:

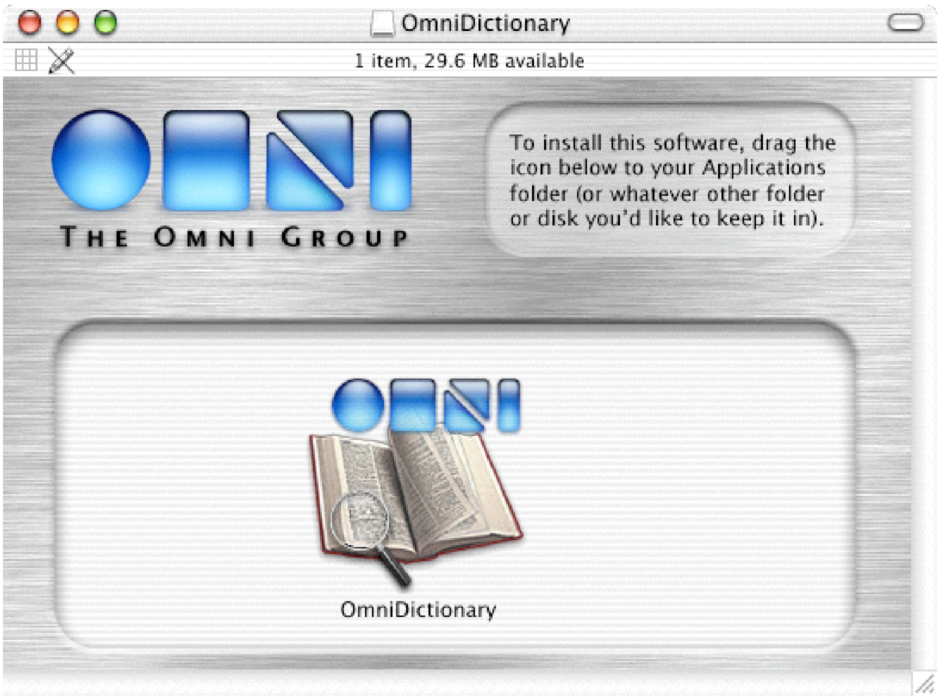
```
# The Elements Database
elements.data=elements.dict.dz
elements.index=elements.index

# Free Online Dictionary of Computing
foldoc.data=foldoc.dict.dz
foldoc.index=foldoc.index
foldoc.name=Dictionary of Computing
foldoc.use=false

# English-Russian Dictionary
er.data=eng-rus.dict.dz
er.index=eng-rus.index
er.morph=org.dict.kernel.EnglishMorphAnalyzer
er.encoding=utf-8
er.use=false
```

The two last paragraphs are there more or less as tutorials about what extra options we can specify. (And we won't bother with those now.) Their last lines, *xyz*`.use=false`, tell JDictd to ignore them. Of course, if you have just put the files for the FOLDOC and/or the English-Russian Dictionary (which I didn't mention above) into that data folder, you can simply delete these lines, respectively - or change them to *xyz*`.use=true`.
The first paragraph is important to us. This is how you tell JDictd to use databases. The first line is not strictly required - lines starting in the # sign are ignored - but it's a good idea to write in there which dictionary/database this is about, to avoid later confusion. The general syntax for the other two lines is:

```
xyz.data=xyz.dict.dz
xyz.index=xyz.index
```

Again, *xyz* stands for the database's name or abbreviation - this has to be identical to the respective filenames. In theory the dictionary/database files could be named something completely different. It would still work as long as the entries next to *xyz*`.data` and *xyz*`.index` reflect that. But why complicate things? Following this syntax, just enter all the files you've copied into the data folder and then save and close the dict.ini file. Note that the order in which you enter the various dictionaries/databases into dict.ini here decides in which order the results of your searches will be returned to your DICT client program if you have multiple dictionaries installed. You can thus get any results from a particular dictionary/database to display above results from other databases. I for one have found it useful to put the WordNet DICT database first (for concise definitions) and the GCIDE below that (for more details). If you now start your JDictd server again (by double-clicking on startdictd.sh in Finder, in ~/Applications/JDictd, see above), it should run with all the databases you just configured it for. Feel free to go to `localhost:2626` in your web browser and check.

We're almost there. Let's set up that OmniDictionary program! OmniDictionary actually comes as a disk image (.dmg) file. So just double-click on it and it will mount (if it hasn't already done that automatically when the download finished). Do I really need to give instructions if it's as simple as this? :-)


OmniDictionary
1 item, 29.6 MB available
THE OMNI GROUP
To install this software, drag the icon below to your Applications folder (or whatever other folder or disk you'd like to keep it in).
OmniDictionary

Actually, I DO need to give some instructions, but, again, this is very straightforward. We need to tell OmniDictionary that it should not look for servers on the Internet, but for our own local server that we so bravely set up. Now then: Once you've installed and opened the application, go to its Preferences. Next to 'Dictionary Server:', simply enter '`localhost`' (w/o speech marks). Confirm the Port is 2628. Attention: there seems to be a bug in OmniDictionary. You need to actually press Return after changing entries in its Preferences. Otherwise it *might* forget you made changes either when you close Preferences or when you close the application. That should however be easy to do. Close Preferences and OmniDictionary. Go back into it and verify that the changes you made persist in Preferences.

Now do a search in OmniDictionary. (Your results will obviously depend on the kind off dictionaries and/or databases you installed.) Great. Read the Help. - Don't miss out on all the great ways to use OmniDictionary. We're sorted.

No, wait! Something's missing. We don't want to manually start our server each

time before we open OmniDictionary! We want our server to launch when we boot up Mac OS X so we're never left without! And after all that hard work we did so far, this better be simple! There you go: It *is* very simple. Just go to System Preferences, go into Login and to the Login Items tab. Add startdictd.sh to the list and tick the Hide box. Close System Preferences.
The next time you reboot now, a Terminal session will automatically start with our server running in it. The Terminal window will only flash up briefly and then get hidden. Should you actually want to use Terminal, you can of course just open another Terminal window (by pressing Command+N).

N.B.: I have noticed that JDictd (in this setup) often produces lots of messages like this one:

```
java.io.InterruptedIOException: Read timed out
```

They seem to be harmless. Changing the timeout setting in OmniDictionary seems to make these less frequent (but it's really not required).

Is that it? Yes - if you're happy with what you've got. What you may notice however, is that you will get prompted to manually quit JDictd (by pressing Ctrl+C) when you are shutting down the system. Being a perfectionist, I found that quite annoying. I want JDictd to safely quit automatically when I shut down Mac OS X. Can that be done? Yes: by running JDictd in the background. To do that, we need to go back to editing our startdictd.sh file.
True, I could have told you to make the entries we're making now the first time around. But I hope you agree that it's a good idea to make things transparent. Plus, after we've done what we'll be doing now, you will no longer be able to simply go Ctrl+C to stop JDictd - or to double-click on startdictd.sh to it. You will either need to reboot in order to get JDictd to restart or (if you are comfortable enough at the Terminal prompt) end the JDictd program using the UNIX command `kill` (see Appendix VI). Alternatively, you could e.g. use a third party product like Catchy Software's Quitling (http://www.catchysoftware.com/products/quitling/). Again, if you don't want to do any of this, you're perfectly fine to leave it at this. I however felt that running JDictd in the background is the more elegant solution, especially for the less experienced user, as it makes the whole thing totally transparent.

Quit JDictd if it's still running (Ctrl+C). Open a new Terminal session and type the following:

`cd ~/Applications/JDictd` [press Return]

Now type:

`pico startdictd.sh` [press Return]

We're back in Pico now.
You should see the text you put in originally:

```
#! /bin/tcsh
echo JDictd is starting up...
cd ~/Applications/JDictd
java -cp jdictd.jar org.dict.server.JDictd data/dict.ini
```

All we need to do is to make a small change on the last line: add an ampersand ("&") to it. You may have to use the arrow keys to get to the end of that line - the mouse may not necessarily work (this can be changed in Terminal Preferences). The contents of that file should now look like this:

```
#! /bin/tcsh
echo JDictd is starting up...
cd ~/Applications/JDictd
java -cp jdictd.jar org.dict.server.JDictd data/dict.ini &
```

Make sure there is a space between the ampersand and the letters before it. Again, press Ctrl+O. We're prompted for the filename again - which is already in there, so we're pressing Return again. After pressing Ctrl+X we'll be back to our Terminal prompt. Now double-click on startdictd.sh. Unlike previously, you should now see something similar to this:

```
Welcome to Darwin!
[localhost:~] ropers%
/Users/ropers/Applications/JDictd/startdictd.sh; exit
JDictd is starting up...
[1] 907
logout

[Process completed]
```

The number given to us (here: 907) is the process ID number. Simply put, we can think of process ID's as numbers for every running program. The process ID will likely be different each time you start JDictd. Again, see Appendix VI for info on how to use that process ID in conjunction with `kill` to quit the JDictd server program when it is running in the background. If you open a new Terminal window now, you may notice that you cannot simply quit Terminal via Command+Q etc. This is normal and due to JDictd running in the background. You will be able to quit Terminal by entering exit or logout first, then quitting the Terminal application. We have just been able to start JDictd with a double-click because it was not running. If however you double-click while it is running, JDictd will NOT restart. It will only seem like it was. It will give a process ID and everything. In reality, the script is being run and the JDictd program is being *started* in the background (that's why it's saying it's starting and giving that number), but it will then immediately exit, realising that it's already running (actually, this results in an address conflict). It won't tell you that it's exiting, since it had been told to run in the background. So unless you want to worry with `kill` or the like, you'll need to reboot for JDictd to restart.

And that's it.

JDictd is now set up to start when your Mac starts and to keep running in the background. And it will not interfere with shutting down the system.

:-) Happy word hunting! :-)

If you have further questions:

- You're welcome to email me at dicthowto@ropersonline.com - I can however make no guarantees as to when or whether I'll be able to answer as I just don't know how much feedback I am likely to get from this. Also, my knowledge is limited of course. (Feel free to give feedback on my technical writing, btw.)
- Try asking your local UNIX geek if you know one. Chances are he knows more than me and can help you even if he's never touched a Mac. (Warning: He might come to like Mac OS X.)
- For questions directly pertaining to the products/programs mentioned here, contact the respective individuals and/or organisations. (E.g. the Omni Group for questions about the actual OmniDictionary application, etc.)

Appendix I: Shameless self-promotion

I am currently looking for a new job. If you might consider hiring an IT professional with great cross-platform skills, please kindly refer to my web site at www.ropersonline.com and have a look at my CV - you will find full contact details there.

Appendix II: Notes

- This current version of this document is released as BETA. Feel free to follow these instructions and please give feedback if it worked well or not. Having only had the opportunity to try this on one system (my own), I would feel it to be somewhat inappropriate to claim 'finished' state for this document. Again, please tell me how you got on (dicthowto@ropersonline.com).
- I did experience a weird freeze/crash once, which could have been related to either OmniDictionary or the DICTD server, but I could not ever reproduce it since. So maybe it was something completely different.
- The most recent version should remain available from www.ropersonline.com, however that web site is currently still under construction and may temporarily go down.
- My iBook (on which I tried all this) has 640 MB RAM. I have no experience with low RAM situations and frankly no idea how much RAM JDictd/OmniDictionary really require. Any feedback on this is highly welcome.
- The jdict.sh command line DICT client tool seems to just hang when JDictd is running and I'm connected to the Internet. This has no impact on or relation to the above instructions/configuration.
- Web links such as given on this site unfortunately often tend to become outdated

quickly. They are correct as of this writing. Feedback on dead links is welcome.

Appendix III: Legalese

I (Jens Ropers) am the author of this document. I hate legalese. It is my intention to sort of release this document into the public domain, to contribute something to the Mac community. (I would have liked a document like this when I first configured this on my Mac.) But just in case of any potential abuse/litigation, I officially retain a copyright on it. I am trying to keep this short but again just in case of any potential abuse/litigation, I reserve the right to have the GNU Public license 2 or higher or the library GPL apply to this (also). If in doubt, ask (dicthowto@ropersonline.com) and if you're not proposing anything unreasonable (like selling these instructions for a massive profit to yourself or saying that you wrote this), chances are I will say yes.
I currently have no relation with any of the companies/organisations mentioned above. To my knowledge all of the said products are offered for free, but I'm making no representations towards that. Some of these products have copyrights (possibly permitting free use under their terms); it is your responsibility to familiarise yourself with them. All trademarks etc. mentioned are the property of their respective owners.
In particular, I am making no representations towards the validity of these instructions. They worked on my system. If they don't work on and/or wreck yours, it's your fault. Don't blame me. I would however be grateful for feedback.

Appendix IV: Installing the WordNet applications

WordNet.app, by Marcus Muller of Mulle Kybernetik, is a Mac OS X program running as a server *and* client for the WordNet database. It is entirely separate and different from the DICT servers and/or clients. For that reason you need to download another copy (actually a different version) of the WordNet database. The required file is: `wn16.unix.tar.gz`; it can be gotten from:
ftp://ftp.cogsci.princeton.edu/pub/wordnet/
The application itself is at:
ftp://ftp.mulle-kybernetik.com/pub/software/WordNet/WordNetApp-4.MOSX.b.ppc.tar.gz
The database will extract as a folder named wordnet-1.6. It's a good idea to put that into the Documents folder (either ~/Documents or /Documents) or into the Library. At first run, the application needs to know the location of that database. Select that and you're sorted.

Another WordNet application has recently been released by William Taysom. Again, it's a client *and* server and it's seperate from the DICT system. Both of these WordNet applications have been designed as stand alone programs; that's why you can't share their databases, not even between these two programs. Actually, William Taysom's application comes as one download that includes the database: http://www2.ur.cmu.edu/~wtaysom/wordnet.html No setup is required.

Appendix V: Installing JDictd to be available for every user

To do this, install JDictd under /Applications/JDictd. Make sure you adjust the path appropriately with every part of these instructions. In addition, you need to adjust the permissions for startdictd.sh, so other users can run that file. For that, instead of (see above):

```
chmod u=rwx startdictd.sh
```

you would rather type:

```
chmod 777 startdictd.sh
```

That should make JDict available to everyone. (OmniDictionary should already be under /Applications.) To have JDictd start automatically, you would want to log in as each user and configure it as per above user Login Items in System Preferences. I'm sure there is a way to automatically start JDictd for every user independent (and regardless) of any Login Items setting, but I just haven't figured that out.
If you're on a LAN of course, you should be able to set up JDictd on one box and have all the other Macs (and even PC's if you have them) connect to it. I don't currently have a LAN, so I haven't tried that.

Appendix VI: Pretty lame kill and process ID instructions

Like most things in UNIX, `kill` comes with a number of options. Enter `man kill` in Terminal to see a reasonably technical explanation. (You probably already know you can use man to see documentation on pretty much every UNIX command.) By default, `kill` is actually less brutal than it sounds: It only sends the process a signal telling it to stop running. (It can be more forceful, depending on the options used.) To our ends, simply take note of the process ID given to you when JDictd is started in the background. Let's remember what we saw there:

```
JDictd is starting up...
[1] 564
logout

[Process completed]
```

I highlighted 564 here. This is our process ID (sometimes also called job number). Again, it's different each time. Every program that is started on our Mac gets assigned one, though if we want to go into detail, there can be processes that aren't really programs and a program can have a number of active processes. This can be useful to have applications or the OS work better. Our JDictd server runs as a single process, so all we need to do is to kill that one. Open another Terminal window and type `kill 564` [Return]. The JDictd process has now safely ended. If you've forgotten the number, you can look it up: type `top` [Return] at a command prompt. This will give you a list of the most active processes "top talkers" (which is a good way to memorise that command). Just press `q` to exit top. In top, you should see a java process. This will be the JDictd server program (unless you have separately started a different java process, in which case there should be two). So why is it java and not JDictd? Remember, the line in our startdictd.sh file that we modified? It was:

```
java -cp jdictd.jar org.dict.server.JDictd data/dict.ini &
```

Interpreting that line, you may be able to recognise that JDictd is actually running in java. But no worries: having java running JDictd doesn't prevent you from running other java programs and killing java doesn't kill other java programs. Coming back to our main point: in top, the number left to the word java is the process ID. Type kill followed by that process ID number. Sorted. If Jaguar (Mac OS X 10.2) is going to include Logout Items, it might be an interesting question if they could be used to to tell JDictd to quit. If this worked right it might obviate the need to run JDictd in the background in order to quit automatically. But let's not overdo it here! ;-)

Appendix VII: There can be only one

I looked at the following DICT clients during my travails:

- Ho Ngoc Duc's jdict.sh:
The jdictd.zip file includes this command line DICT client, which can be run inside Terminal. That may be handy if you're an avid UNIX command line interface aficionado, however I found that jdict.sh hangs if I'm using it while connected to the Internet with my locally hosted JDictd running.

- David Caldwell's Dict Client:
It seems to be great, but it's for Classic Mac OS only.

- Bryan Lee's DictX:
A nice client, but it needs a live Internet connection and cannot be set up to work with a locally hosted server such as JDictd.

- Navdeep Bains' MacDict:
Definitely not a bad client. I found however that it doesn't seem to display the text particularly well formatted, and if I look something up then clarity is the game. N.B: There are two more DICT databases available from the MacDict site. See the link below.

- The Omni Group's OmniDictionary:
My top contender. It doesn't hang, runs under Mac OS X, can be configured to work with our locally hosted server, layouts the dictionary entries nicely and offers a Service with hot key (accessible from compatible, Service aware Mac OS X applications), so you can simply go to an application's Services menu or press Command+= to look up a highlighted word. Unfortunately this doesn't currently work from Apple Works (Apple Works' fault, really). The feature to click on underlined words (hyperlinks) to look up synonyms, etc. is also very useful. Actually, I did manage to get OmniDictionary to hang on one occasion: by starting it *while JDictd was still starting up*. Ok. Don't do that then.

Appendix VIII: Links and Credits

Please note that I did not produce any of the programs/files this entire document is about. Let’s thank the respective people for all their hard work. Please find below links to their web sites among this list of relevant links:

- http://www.dict.org/ (The **DICT** Development Group)
  webmaster@dict.org
- ftp://ftp.dict.org/pub/dict/pre/
  (download site for the preformatted DICT databases)

- http://www.porkrind.org/dict/ (David Caldwell’s **Dict Client**)
  david@porkrind.org

- http://bryanlee.com/dictx/ (Bryan Lee’s **DictX**)
  dictx@bryanlee.com

- http://www.ccel.org/e/easton/ebd/ebd.html (**Easton’s** Bible Dictionary)

- http://foldoc.doc.ic.ac.uk/foldoc/index.html
  (The Free Online Dictionary Of Computing; **FOLDOC**)

- http://www.census.gov/cgi-bin/gazetteer/ (The US **Gazetteer**)

- The GNU Collaborative International Dictionary of English; **GCIDE**:
  There appear to be many sites, most of which seem just to mirror/host the dictionary data. I couldn’t find a home page/main web site for this project.

- http://www.gnu.org/ or http://www.fsf.org/
  (The **GNU** Project and Free Software Foundation)

- http://www.ccel.org/bible_names/title.html
  (**Hitchcock’s** Bible Names Dictionary)

- http://www.tuxedo.org/~esr/jargon/ (The **Jargon File**)

- http://www.informatik.uni-leipzig.de/~duc/Java/JDictd/
  (Ho Ngoc Duc’s **JDictd**)
  duc@informatik.uni-leipzig.de

- http://www.bainsware.com/macdict/ (Navdeep Bains’ **MacDict**)
  support@bainsware.com

- http://www.apple.com/**macosx**/ (OS by the inventors of the iTools sting)

- http://www.omnigroup.com/applications/omnidictionary/
  (The Omni Group’s **OmniDictionary**)

omnidictionary@omnigroup.com

- http://www.catchysoftware.com/products/quitling/ (Catchy Software's **Quitling**)
  support@catchysoftware.com

- **Setting me up**: a Dire Straits song

- **V.E.R.A.**, the Virtual Entity of Relevant Acronyms:
  There appear to be sites, that mirror/host the dictionary data. I couldn't find a home page/main web site for this project.

- The 1913 **Webster**:
  There appear to be many sites, most of which seem just to mirror/host the dictionary data. I couldn't find a home page/main web site.

- http://www.microsoft.com/**windows**/ (some operating systems you can run in Connectix Virtual PC if you have to)

- http://www.cogsci.princeton.edu/~wn/ (Princeton's **WordNet**)
  wordnet@princeton.edu

- http://www2.ur.cmu.edu/~wtaysom/wordnet.html
  (William Taysom's **WordNet** application)
  wtaysom@andrew.cmu.edu

- http://www.mulle-kybernetik.com/software/WordNet/
  (Erik Dornenburg and Marcus Muller's **WordNet.app**)
  erik@x101.net and znek@mulle-kybernetik.com

- http://wordweb.info/ (Anthony Lewis' **WordWeb** (for MS Windows))
  wordweb@wordweb.info

- http://www.cia.gov/cia/publications/factbook/index.html
  (the most recent CIA **World Factbook**)

Appendix IX: Requirements

- Mac OS X (I only tested this with version 10.1.5)
- a Mac capable of running the said OS
- some hard drive space (ca. 10-35 MB or more, depending on data files installed)
- a way to get the files off the Internet (e.g. via a dial-up connection or have a friend burn them onto CD)
- after getting the files, an Internet connection is no longer required
- real life RAM requirements are unknown at the time of this writing (pls email me!)

Appendix X: Version History

This is the fourth release of this document.

21 Oct 2002; version number 0.6.5b
- created a HTML version and made minor changes reflecting this

24 Aug 2002; version number 0.6.1b:
- fixed a stooopid syntax error right on the first page "How get a superior and free dictionary up and running on your Mac". How embarrassing! Why didn't anyone tell me?!?

18 Aug 2002; version number 0.6b:
- added info on William Taysom's WordNet application

06 Aug 2002; version number 0.5b:
- initial release

Appendix XI: To do list

I dunno. Any suggestions?

The most recent version of this PDF document can be downloaded from www.ropersonline.com.

End of File. Thanks for your interest. :-)