SCHEME-DICT README

CONTENTS

1 Introduction 1
2 Authors 1
3 Documentation 1
4 News 1
5 Install 1
6 Copying 6
7 Project Web-site 6

1 INTRODUCTION

Scheme-dict is an implementation of the DICT dictionary protocol. It includes both a set of modules for dealing with MD5 checksums, destructuring of lists, and the DICT protocol itself, as well as a client program that can be used to connect to a DICT server and retrieve dictionary definitions for words using the methods available at the server. The client program provides additional information on how to use it.

2 AUTHORS

Scheme-dict was written by the following people:

- Nikolai Weibull <nikolai@bitwi.se>

3 DOCUMENTATION

To be written. We'd also like a manual page.

4 NEWS

4.1 *Version 0.1.0: Initial release.*

Nothing to report; initial release.

5 INSTALL

5.1 *Basic installation instructions.*

These are generic installation instructions.

The `configure` shell script attempts to guess correct values for various system-dependent variables used during compilation. It uses those values to create a `Makefile`

in each directory of the package. It may also create one or more `.h` files containing system-dependent definitions. Finally, it creates a shell script `config.status` that you can run in the future to recreate the current configuration, and a file `config.log` containing compiler output (useful mainly for debugging `configure`).

The `configure` shell script can also use an optional file – which is typically called `config.cache` and enabled with `-cache-file=config.cache` or simply `-C`) that saves the results of its tests to speed up reconfiguring. (Caching is disabled by default to prevent problems with accidental use of stale cache files.)

If you need to do unusual things to compile the package, please try to figure out how `configure` could check whether to do them, and mail diffs or instructions to the address given in `README.pdf` so they can be considered for the next release. If you're using the cache, and at some point `config.cache` contains results you don't want to keep, you may remove or edit it.

The file `configure.ac` is used to create `configure` by a program called `autoconf(1)`. You only need `configure.ac` if you want to change it or regenerate `configure` using a newer version of `autoconf(1)`.

The simplest way to compile this package is:

1 `cd` to the directory containing the package's source code and type `./configure` to configure the package for your system. If you're using `csh(1)` on an old version of System V, you might need to type `sh ./configure` instead to prevent `csh(1)` from trying to execute `configure` itself.

Running `configure` takes awhile. While running, it prints some messages telling which features it is checking for.

2 Type `make` to compile the package.

3 Optionally, type `make check` to run any self-tests that come with the package.

4 Type `make install` to install the programs and any data files and documentation.

5 You can remove the program binaries and object files from the source code directory by typing `make clean`. To also remove the files that `configure` created (so you can compile the package for a different kind of computer), type `make distclean`. There is also a `make maintainer-clean` target, but that is intended mainly for the package's developers. If you use it, you may have to get all sorts of other programs in order to regenerate files that came with the distribution.

5.2 *Compilers and options.*

Some systems require unusual options for compilation or linking that the `configure` script doesn't know about. Run `./configure -help` for details on some of the pertinent environment variables.

You can give `configure` initial values for configuration parameters by setting variables in the command line or in the environment. Here is an example:

```
$ ./configure CC=c89 CFLAGS=-O2 LIBS=-lposix
```

Also see Section 5.8.

5.3 *Compiling for multiple architectures.*

You can compile the package for more than one kind of computer at the same time, by placing the object files for each architecture in their own directory. To do this, you must use a version of `make(1)` that supports the `VPATH` variable, such as GNU `make(1)`. `cd` to the directory where you want the object files and executables to go and run the `configure` script. `configure` automatically checks for the source code in the directory that `configure` is in and in the parent directory (`..`).

If you have to use a `make(1)` that does not support the `VPATH` variable, you have to compile the package for one architecture at a time in the source code directory. After you have installed the package for one architecture, use `make distclean` before reconfiguring for another architecture.

5.4 *Installation names.*

By default, `make install` will install the package's files in `/usr/local`. You can specify an installation prefix other than `/usr/local` by giving `configure` the option `-prefix=PATH`.

You can specify separate installation prefixes for architecture-specific files and architecture-independent files. If you give `configure` the option `-exec-prefix=PATH`, the package will use `PATH` as the prefix for installing programs and libraries. Documentation and other data files will still use the regular prefix.

In addition, if you use an unusual directory layout you can give options like `-bindir=PATH` to specify different values for particular kinds of files. Run `configure -help` for a list of the directories you can set and what kinds of files go in them.

If the package supports it, you can cause programs to be installed with an extra prefix or suffix on their names by giving `configure` the option `-program-prefix=PREFIX` or `-program-suffix=SUFFIX`.

5.5 *Optional features.*

Some packages pay attention to `-enable-FEATURE` options to `configure`, where FEATURE indicates an optional part of the package. They may also pay attention to `-with-PACKAGE` options, where PACKAGE is something like `gnu-as` or `x` (for the X Window System). The `README` should mention any `-enable-` and `-with-` options that the package recognizes.

For packages that use the X Window System, `configure` can usually find the X include and library files automatically, but if it doesn't, you can use the `configure` options `-x-includes=DIR` and `-x-libraries=DIR` to specify their locations.

5.6 *Specifying the System Type*

There may be some features `configure` cannot figure out automatically, but needs to determine by the type of machine the package will run on. Usually, assuming the package is built to be run on the *same* architectures, `configure` can figure that out, but if it prints a message saying it cannot guess the machine type, give it the `-build=TYPE` option. `TYPE` can either be a short name for the system type, such as `sun4`, or a canonical name which has the form:

```
CPU-COMPANY-SYSTEM
```

where `SYSTEM` can have one of these forms:

```
OS KERNEL-OS
```

See the file `config.sub` for the possible values of each field. If `config.sub` isn't included in this package, then this package doesn't need to know the machine type.

If you're *building* compiler tools for cross-compiling, you should use the `-target=TYPE` option to select the type of system they will produce code for.

If you want to *use* a cross compiler, that generates code for a platform different from the build platform, you should specify the "host" platform (i.e., that on which the generated programs will eventually be run) with `-host=TYPE`.

5.7 *Sharing defaults.*

If you want to set default values for `configure` scripts to share, you can create a site shell script called `config.site` that gives default values for variables like `CC`, `cache_file`, and `prefix`. `configure` looks for `PREFIX/share/config.site` if it exists, then `PREFIX/etc/config.site` if it exists. Or, you can set the `CONFIG_SITE`

environment variable to the location of the site script. A warning: not all `configure` scripts look for a site script.

5.8 *Defining variables.*

Variables not defined in a site shell script can be set in the environment passed to `configure`. However, some packages may run configure again during the build, and the customized values of these variables may be lost. In order to avoid this problem, you should set them in the `configure` command line, using `VAR=value`. For example:

```
$ ./configure CC=/usr/local2/bin/gcc
```

will cause the specified `gcc(1)` to be used as the C compiler (unless it is overridden in the site shell script).

5.9 *Invoking* `configure`

`configure` recognizes the following options to control how it operates.

-h, –help Print a summary of the options to `configure`, and exit.

-V, –version Print the version of `Autoconf(1)` used to generate the `configure` script, and exit.

–cache-file=FILE Enable the cache: use and save the results of the tests in FILE, traditionally `config.cache`. FILE defaults to `/dev/null` to disable caching.

-C, –config-cache Alias for `-cache-file=config.cache`.

-q, –quiet, –silent Don't print messages saying which checks are being made. To suppress all normal output, redirect it to `/dev/null` (any error messages will still be shown).

–srcdir=DIR Look for the package's source code in directory `DIR`. Usually `configure` can determine that directory automatically.

`configure` also accepts some other, not widely useful, options. Run `configure -help` for more details.

6 COPYING

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the license, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but *without any warranty*; without even the implied warranty of *merchantability* or *fitness for a particular purpose*. See the GNU General Public License for more details.

The GNU General Public License can be found at http://www.gnu.org/licenses/gpl.html. It can also be requested by writing to the Free Software Foundation. *I* bear no responsibility for *you* finding this license, as all it does is grant *you* rights that *you* wouldn't have without it.

7 PROJECT WEB-SITE

To be created.