Quickie

Reference Manual

Peter Miller
millerp@canb.auug.org.au

.

This document describes Quickie version 1.1
and was prepared 21 May 2006.

NAME

Quickie − a small fast C++ Wiki Wiki

DESCRIPTION

The *Quickie* program is a small fast wiki engine written in C++, hence the name.

Quickie is a small footprint, fast C++ Wiki engine. I went looking for small fast wiki engines and didn't find one. Too many were written in interpretive languages with huge run-time footprints. Quickie is small and fast.

ARCHIVE SITE

The latest version of Quickie is available on the Web from:

URL:	http://quickie.sourceforge.net/	
File:	quickie.html	# the Quickie page
File:	quickie-1.1.README	# Description, from the tar file
File:	quickie-1.1.lsm	# Description, LSM format
File:	quickie-1.1.spec	# RedHat package specification
File:	quickie-1.1.tar.gz	# the complete source
File:	quickie-1.1.pdf	# Reference Manual

BUILDING QUICKIE

Full instructions for building Quickie may be found in the *BUILDING* file included in this distribution.

COPYRIGHT

Quickie version 1.1
Copyright © 2005, 2006 Peter Miller; All rights reserved.

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.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111, USA.

It should be in the *LICENSE* file included with this distribution.

AUTHOR

Peter Miller E-Mail: millerp@canb.auug.org.au
/\/* WWW: http://www.canb.auug.org.au/~millerp/

RELEASE NOTES

This section details the various features and bug fixes of the various releases. For excruciating and complete detail, and also credits for those of you who have generously sent me suggestions and bug reports, see the *etc/CHANGES.** files.

Version 1.0 (21-may-2006)

- Several build problems have been fixed.

- The "validate" program has been added to the test suite, to ensure html standards compliance.

- There is a new umask configuration parameter.

- Iy is now possible to ask quickie to chroot before doing anything else. This offers some protection should weaknesses be found in the code.

- Users passwords are now stored on disk encrypted. The passwords can’t be accessed via quickie, but this offers some defence against users who have read access to the files holding the passwords.

- It is now possible to login as a particular user.

- A bug has been fixed the the parsing of the HTTP_COOKIE environment variable.

- A bug has been fixed in the new user form. It did not include a name for the submit button, which made the form processing mis-cue.

- It is now possible to set the width of the sidebar in the configuration file via the sidebar-width field.

- The edit pages are now conditional, you need to be logged in and have edit permission, or the system needs anonymous edit permission.

Version 1.0 (Apr-2005)

First public release.

NAME

How to build Quickie

SPACE REQUIREMENTS

You will need about 8MB to unpack and build the Quickie package. Your milage may vary.

BEFORE YOU START

There are a few pieces of software you may want to fetch and install before you proceed with your installation of Quickie.

GNU Groff

The documentation for the Quickie package was prepared using the GNU Groff package (version 1.14 or later). This distribution includes full documentation, which may be processed into PostScript or DVI files at install time − if GNU Groff has been installed.

GCC You may also want to consider fetching and installing the GNU C Compiler if you have not done so already. This is not essential. Quickie was developed using the GNU C++ compiler, and the GNU C++ libraries.

The GNU FTP archives may be found at `ftp.gnu.org`, and are mirrored around the world.

SITE CONFIGURATION

The Quickie package is configured using the *configure* program included in this distribution.

The *configure* shell script attempts to guess correct values for various system-dependent variables used during compilation, and creates the *Makefile* and *lib/config.h* files. It also creates a shell script *config.status* that you can run in the future to recreate the current configuration.

Normally, you just *cd* to the directory containing Quickie's source code and then type

```
% ./configure
...lots of output...
%
```

If you're using *csh* on an old version of System V, you might need to type

```
% sh configure
...lots of output...
%
```

instead to prevent *csh* from trying to execute *configure* itself.

Running *configure* takes a minute or two. While it is running, it prints some messages that tell what it is doing. If you don't want to see the messages, run *configure* using the quiet option; for example,

```
% ./configure --quiet
%
```

To compile the Quickie package in a different directory from the one containing the source code, you must use a version of *make* that supports the *VPATH* variable, such as GNU Make. Change directory (cd) to the directory where you want the object files and executables to go and run the *configure* script. The *configure* script automatically checks for the source code in the directory that *configure* is in. and also in `..` (the parent directory). If for some reason *configure* is not in the source code directory that you are configuring, then it will report that it can't find the source code. In that case, run *configure* with the option `--srcdir=`*DIR*, where *DIR* is the directory that contains the source code.

By default, *configure* will arrange for the *make install* command to install the Quickie package's files in */usr/local/bin*, and */usr/local/man*. There are options which allow you to control the placement of these files.

`--prefix=`*PATH*

This specifies the path prefix to be used in the installation. Defaults to */usr/local* unless otherwise specified.

`--exec-prefix=`*PATH*

You can specify separate installation prefixes for architecture-specific files files. Defaults to *${prefix}* unless otherwise specified.

`--bindir=`*PATH*
: This directory contains executable programs. On a network, this directory may be shared between machines with identical hardware and operating systems; it may be mounted read-only. Defaults to *${exec_prefix}/bin* unless otherwise specified.

`--mandir=`*PATH*
: This directory contains the on-line manual entries. On a network, this directory may be shared between all machines; it may be mounted read-only. Defaults to *${prefix}/man* unless otherwise specified.

`--sysconfdir=`*PATH*
: This directory contain the system configuration files. The commonest use of this is to set `/etc` so that the Quickie configuration file is `/etc/quickie.conf`

`--localstatedir=`*PATH*
: This directory contain the local state files. The commonest use of this is to set `/var` so that the Quickie directory `/var/quickie` contains the wiki files.

The *configure* script ignores most other arguments that you give it; use the `--help` option for a complete list.

On systems that require unusual options for compilation or linking that the Quickie package's *configure* script does not know about, you can give *configure* initial values for variables by setting them in the environment. In Bourne-compatible shells, you can do that on the command line like this:

```
$ CXX='g++ -traditional' LIBS=-lposix ./configure
...lots of output...
$
```

Here are the *make* variables that you might want to override with environment variables when running *configure*.

Variable: CXX
: C++ compiler program. The default is *c++*.

Variable: CPPFLAGS
: Preprocessor flags, commonly defines and include search paths. Defaults to empty. It is common to use `CPPFLAGS=-I/usr/local/include` to access other installed packages.

Variable: INSTALL
: Program to use to install files. The default is *install* if you have it, *cp* otherwise.

Variable: LIBS
: Libraries to link with, in the form `-l`*foo* `-l`*bar*. The *configure* script will append to this, rather than replace it. It is common to use `LIBS=-L/usr/local/lib` to access other installed packages.

If you need to do unusual things to compile the package, the author encourages you to figure out how *configure* could check whether to do them, and mail diffs or instructions to the author so that they can be included in the next release.

BUILDING Quickie

All you should need to do is use the

```
% make
...lots of output...
%
```

command and wait. When this finishes you should see a directory called *bin* containing three files: *Quickie*, *Quickie* and *Quickie*.

quickie The *quickie*(1) program is a common gateway interface (CGI) program used to manipulate an maintain a wiki site. For more information, see *quickie*(1).

quickie_svt

The *quickie_svt*(1) program can be used to examine and manipulate the "simple version tool" history files used by *quickie*(1) to stoke wiki page histories. For more information, see *quickie_svt*(1).

quickie_prime

The *quickie_prime*(1) program is used to pre-cache the html corresponding to wiki files in a wiki site. For more information, see *quickie*(1).

If you have GNU Groff installed, the build will also create a *etc/reference.ps* file. This contains the README file, this BUILDING file, and all of the man pages.

You can remove the program binaries and object files from the source directory by using the

```
% make clean
...lots of output...
%
```

command. To remove all of the above files, and also remove the *Makefile* and *lib/config.h* and *config.status* files, use the

```
% make distclean
...lots of output...
%
```

command.

The file *etc/configure.in* is used to create *configure* by a GNU program called *1autoconf*. You only need to know this if you want to regenerate *configure* using a newer version of *autoconf*.

TESTING QUICKIE

The Quickie package comes with a test suite. To run this test suite, use the command

```
% make sure
...lots of output...
Passed All Tests
%
```

The tests take a few seconds each, with a few very fast, and a couple very slow, but it varies greatly depending on your CPU.

If all went well, the message

```
Passed All Tests
```

should appear at the end of the make.

INSTALLING QUICKIE

As explained in the *SITE CONFIGURATION* section, above, the Quickie package is installed under the */usr/local* tree by default. Use the `--prefix=`*PATH* option to *configure* if you want some other path. More specific installation locations are assignable, use the `./configure --help` option for details.

All that is required to install the Quickie package is to use the

```
% make install
...lots of output...
%
```

command. Control of the directories used may be found in the first few lines of the *Makefile* file and the other files written by the *configure* script; it is best to reconfigure using the *configure* script, rather than attempting to do this by hand.

Web Server `cgi-bin` Directory

It is necessary to copy the `quickie` executable into your web server's `cgi-bin` directory. This isn't done automagically by *make install* because it tends to move around a great deal from one web server to another.

You could also use a symbolic link, provided your web server is configured to allow symbolic links in the `cgi-bin` directory (the is the Apache default).

Quickie `/var` Directory

The wiki pages are stored in the *$(prefix)/var/quickie* directory, unless you overrode this by using the *./configure --localstatedir* command line option.

This directory must be readable and writable by the UID Apache (or any other web server) runs as. For example, RedHat tends to use UID "`apache`" and Debian tends to use UID "`www-data`".

You need to change the file owner to match the web server. On RedHat the following command

```
chmod -R apache.apache ${prefix}/var/quickie
```

will perform this task. Make sure you set the UID, GID and directory name appropriate for your situation.

For the Apache web server, the **User** and **Group** directives in the *apache.conf* file are the ones to look for.

Automated Post-Install

You can use the *quickie_post_install*(1) program to see if the above web server configuration data can be automatically determined; the information will be printed on the standard output.

If you are satisfied with the results (they are not always correct for every site) you can ask for them to be performed for you by using the *quickie_post_install −auto* command line option.

GETTING HELP

If you need assistance with the *Quickie* package, please do not hesitate to contact the author at

```
Peter Miller <millerp@canb.auug.org.au>
```

Any and all feedback is welcome.

When reporting problems, please include the version number given by the

```
% Quickie -version
Quickie version 1.1.D001
...warranty disclaimer...
%
```

command. Please do not send this example; run the program for the exact version number.

COPYRIGHT

Quickie version 1.1

The *Quickie* package 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.

It should be in the *LICENSE* file included with this distribution.

AUTHOR

Peter Miller	E-Mail:	millerp@canb.auug.org.au
`/\/\*`	WWW:	http://www.canb.auug.org.au/˜millerp/

NAME

quickie − a small fast C++ Wiki engine

SYNOPSIS

quickie

DESCRIPTION

The *quickie* program is used to serve wiki pages when invoked from a HTTP web server, such as Apache.

The *quickie* program does not accept any command line options, as these could be abused by web clients.

Web Server `cgi-bin` Directory

It is necessary to copy the `quickie` executable into your web server's `cgi-bin` directory. This isn't done automagically by *make install* because it tends to move around a great deal from one web server to another.

You could also use a symbolic link, provided your web server is configured to allow symbolic links in the `cgi-bin` directory (the is the Apache default).

Quickie `/var` Directory

The wiki pages are stored in the *$(prefix)/var/quickie* directory, unless you overrode this by using the *./configure --localstatedir* command line option.

This directory must be readable and writable by the UID Apache (or any other web server) runs as. For example, RedHat tends to use UID "`apache`" and Debian tends to use UID "`www-data`".

You need to change the file owner to match the web server. On RedHat the following command

```
chmod -R apache.apache ${prefix}/var/quickie
```

will perform this task. Make sure you set the UID, GID and directory name appropriate for your situation.

For the Apache web server, the **User** and **Group** directives in the *apache.conf* file are the ones to look for.

Automated Post-Install

You can use the *quickie_post_install*(1) program to see if the above web server configuration data can be automatically determined; the information will be printed on the standard output.

If you are satisfied with the results (they are not always correct for every site) you can ask for them to be performed for you by using the *quickie_post_install −auto* command line option.

ACCESSING THE WIKI PAGES

Once the *quickie*(1) program has been installed into your web server's `cgi-bin` directory, you will be able to access through your web browser. The URL of the wiki pages will be

http://*server.domain.name*/cgi-bin/quickie

however you need to substitute the correct domain name (Internet name) for the web server. If the cgi-bin directory has an unusual name, you will have to take that into account as well.

It is common to make a link from your web site's static pages into the Quickie wiki pages, to simplify access for your site's visitors.

EXIT STATUS

The *quickie* command will exit with a status of 1 on any error. The *quickie* command will only exit with a status of 0 if there are no errors.

COPYRIGHT

quickie version 1.1
Copyright © 2005, 2006 Peter Miller;
All rights reserved.

The *quickie* program comes with ABSOLUTELY NO WARRANTY; for details use the '*quickie -VERSion License*' command. This is free software and you are welcome to redistribute it under certain conditions; for details use the '*quickie -VERSion License*' command.

AUTHOR

Peter Miller	E-Mail:	millerp@canb.auug.org.au
/\/*	WWW:	http://www.canb.auug.org.au/˜millerp/

NAME

quickie-config − print configuration parameters

SYNOPSIS

quickie-config --etc
quickie-config --var
quickie-config --version

DESCRIPTION

The *quickie-config* program is used to print various configuration details, to aid system administration and script writing.

OPTIONS

The following options are understood:

−−etc This option is used to determine the directory in which system configuration files (e.g. quickie.conf) are kept.

−−var This option is used to determine the directory in which the wiki data files are kept.

−−version
Print the version of the *quickie-config* program being executed.

All other options will produce a diagnostic error.

EXIT STATUS

The *quickie-config* command will exit with a status of 1 on any error. The *quickie-config* command will only exit with a status of 0 if there are no errors.

COPYRIGHT

quickie-config version 1.1

AUTHOR

Peter Miller E-Mail: millerp@canb.auug.org.au
/\/* WWW: http://www.canb.auug.org.au/˜millerp/

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

> *one line to give the program’s name and a brief idea of what it does.*
> Copyright (C) 19yy *name of author*
>
> 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.
>
> You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

> Gnomovision version 69, Copyright (C) 19yy name of author
> Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’. This is free software, and you are welcome to redistribute it under certain conditions; type ‘show c’ for details.

The hypothetical commands ‘show w’ and ‘show c’ should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

> Yoyodyne, Inc., hereby disclaims all copyright interest in the program ‘Gnomovision’ (which makes passes at compilers) written by James Hacker.
>
> *signature of Ty Coon,* 1 April 1989
> Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

NAME

quickie_post_install − supplementary installation operations

SYNOPSIS

quickie_post_install [**--auto** | **--cgi** | **--var**]
quickie_post_install -Help
quickie_post_install -VERSion

DESCRIPTION

The *quickie_post_install* program is used to perform some operations after the Quickie package has been installed in order for your web server to be able to access the *quickie*(1) CGI program and it's data.

If no options are specified, a descriptions of the necessary operations will be printed, but none of those operations will be performed.

If you want the operations to be performed, you must secify one of the options below. Note that in most cases you will have to be root to perform these operations.

OPTIONS

The following options are understood:

−Cgi-bin
Perform the operations necessary to locate the appropriate `cgi-bin` directory, and install the *quickie*(1) program in that directory.

−Local-state-dir
Perform the operations necessary to change the ownership of the files in the *quickie*(1) data directory to be that of the user which executes the HTTP server.

−AUTOmatic
Perform both of the above operations.

−Help Provide some help with using the *quickie_post_install*(1) program.

−VERSion
Print the version of the *quickie_post_install*(1) program being executed.

All other options will produce a diagnostic error.

All options may be abbreviated; the abbreviation is documented as the upper case letters, all lower case letters and underscores (_) are optional. You must use consecutive sequences of optional letters.

All options are case insensitive, you may type them in upper case or lower case or a combination of both, case is not important.

For example: the arguments "-project, "-PROJ" and "-p" are all interpreted to mean the **-Project** option. The argument "-prj" will not be understood, because consecutive optional characters were not supplied.

Options and other command line arguments may be mixed arbitrarily on the command line, after the function selectors.

The GNU long option names are understood. Since all option names for *quickie_post_install* are long, this means ignoring the extra leading '-'. The "**--***option***=***value*" convention is also understood.

EXIT STATUS

The *quickie_post_install* command will exit with a status of 1 on any error. The *quickie_post_install* command will only exit with a status of 0 if there are no errors.

COPYRIGHT

quickie_post_install version 1.1
Copyright © 2005, 2006 Peter Miller;
All rights reserved.

The *quickie_post_install* program comes with ABSOLUTELY NO WARRANTY; for details use the '*quickie_post_install -VERSion License*' command. This is free software and you are welcome to redistribute it under certain conditions; for details use the '*quickie_post_install -VERSion License*' command.

AUTHOR

Peter Miller	E-Mail:	millerp@canb.auug.org.au
/\/*	WWW:	http://www.canb.auug.org.au/˜millerp/

NAME

quickie_prime − pre-cache html from wiki files

SYNOPSIS

quickie_prime [*option*...]

DESCRIPTION

The *quickie_prime* program is used to bring a Quickie directory tree up-to-date. The directory is walked, and each `.wiki` file is examined.

If the corresponding `.html` file is not up-to-date (or does not exist) it will be regenerated in the same way *quickie*(1) does on-the-fly for edited pages.

If the corresponding `,svt` file is not up-to-date (or does not exist) the `.wiki` file will be checked in, in the same way that *quickie*(1) does on-the-fly for edited pages.

By using the *quickie_prime* program you can pre-cache the .wiki pages for better performance. It is also used by the Quickie install to bring the initial tree of wiki files up-to-date, including all of the back links.

OPTIONS

The following options are understood:

−D *directory-path*
This option mant be used to specify the root of the wiki tree to be regenerated. You can’t do it for a subtree; you must specify the root of whe whole wiki tree.

−v This option causes *quickie_prime* to be verbose about what it is doing. Specifying it once tells you about each directory walked (is is supposed to make at least two passes). Specifying this option twice adds information about each file it processes.

All other options will produce a diagnostic error.

EXIT STATUS

The *quickie_prime* command will exit with a status of 1 on any error. The *quickie_prime* command will only exit with a status of 0 if there are no errors.

COPYRIGHT

quickie_prime version 1.1
Copyright © 2005, 2006 Peter Miller;
All rights reserved.

The *quickie_prime* program comes with ABSOLUTELY NO WARRANTY; for details use the ’*quickie_prime -VERSion License*’ command. This is free software and you are welcome to redistribute it under certain conditions; for details use the ’*quickie_prime -VERSion License*’ command.

AUTHOR

Peter Miller	E-Mail:	millerp@canb.auug.org.au
/\/*	WWW:	http://www.canb.auug.org.au/˜millerp/

NAME

quickie_svt − simple version tool

SYNOPSIS

quickie_svt −Check_Output −History *file* **−File** *output-file* [**−e** *edit*]
quickie_svt −Check_In −History *file* **−File** *input-file*
quickie_svt −List −History *file*
quickie_svt −Query −History *file*
quickie_svt −Version

DESCRIPTION

The *quickie_svt* program is used to manage Quickie's history version files. Normally the *quickie*(1) CGI progran does this by itself, but there may be times when a maintainer needs to access the history directly. (It's also useful for testing.)

OPTIONS

The following options are understood:

−History *history-file*
: This option is used to specify the name of the history file. The usual suffix is "`,svt`".

−File *file-name*
: This option is used to specify the name of the input or output file. On checkout, the file name "−" is understood to mean the standard output. There is **no** equivalent for checkin.

−Edit *edit-number*
: This option is used to specify the edit number (version number). On checkout, if no version number is specified, the most recent version is given.

−Check_In
: This option is used to check a file into the history.

−Check_Out
: This option is used to checkout a file from the history.

−List This option is used to list the file's history.

−Query This option is used to query edit number of most recent checkin.

−Version
: This option is used to print version number.

All other options will produce a diagnostic error.

All options may be abbreviated; the abbreviation is documented as the upper case letters, all lower case letters and underscores (_) are optional. You must use consecutive sequences of optional letters.

All options are case insensitive, you may type them in upper case or lower case or a combination of both, case is not important.

For example: the arguments "-project, "-PROJ" and "-p" are all interpreted to mean the **-Project** option. The argument "-prj" will not be understood, because consecutive optional characters were not supplied.

Options and other command line arguments may be mixed arbitrarily on the command line, after the function selectors.

The GNU long option names are understood. Since all option names for *quickie_svt* are long, this means ignoring the extra leading '-'. The "**--***option***=***value*" convention is also understood.

FILE FORMAT

Each version in the history file consists of an RFC822 header, plus the file contents. The header includes (at least) the Content-Length, used to remember the length of the file data in bytes; the Checksum, used to remember the Adler32 checksum of the file data; and Version, used to remember the version number. The file data can be text or binary, because its length is determined by the header. There is no quoting mechanism of any kind for the data. Except for the mandatory fields, additional user-defined us-ascii metadata may also be stored in the header. There is no diff or delta of any kind for any version.

This combination of header and data has good end-to-end behaviour, because there is a checksum to validate the file data against. Bad blocks in the data will be detected then next time a checkin or checkout is attempted.

The format of the history file consists of one or more file versions with the above layout, joined head-to-tail with no separators or boundary indicators of any kind. The versions are in descending order, from most recent (greatest edit number) to least recent (version number one). To determine where one version stops and the next version starts, use the Content-Length field in the header. The entire history file is then compressed using the GNU Zip algorithm (via zlib). There is no diff or delta of any kind in the history file.

The advantage of compressing the file is that there is usually a very high redundancy between file versions. For example, if two identical versions are checked in (not necessarily sequentially) the second copy will compress to only a few bytes. Unlike *diff*(1) style deltas, this also copes very will with moving blocks of data within the file. The use of GNU Zip formatting means there is also a checksum for the whole history file, which allows you to detect bad blocks in the header portions; it also means there is a simple way to extract the data from a history file even without the *quickie_svt* program, or for testing, or because you are curious.

This style of history file was inspired by RFC 3284 − The VCDIFF Generic Differencing and Compression Data Format. While the *quickie_svt* format does not use RFC3284 internally, the arguments for compression across file versions in still relevant.

EXIT STATUS

The *quickie_svt* command will exit with a status of 1 on any error. The *quickie_svt* command will only exit with a status of 0 if there are no errors.

COPYRIGHT

quickie_svt version 1.1

AUTHOR

Peter Miller	E-Mail:	millerp@canb.auug.org.au
/\/*	WWW:	http://www.canb.auug.org.au/˜millerp/

NAME

Welcome to Quickie

DESCRIPTION

Welcome to Quickie Wiki, the C++ Wiki Wiki.

You are seeing this message because you have recently installed Quickie, and have yet to customise your pages. This first page can be replaced, like all other wiki pages, by using the "Edit" link you see at the bottom of this page.

Quickie was written because the author had a significantly under-powered ancient 386 as a web server, and any sort of CPU load at all was going to bring it to its knees. Nor did it have much memory, so huge runtime footprints were out of the question as well. These constraints ruled out interpretive languages such as Perl and Python, and even Java.

There was a distinct lack of compiled wiki engines written in C or C++, so in the true "how hard can it be?" open source spirit, the Quickie project was born.

The earliest insight for Quickie was that most wiki pages are only read by clients, and writes are much less likely. By caching the HTML, it became possible to place a very low burden on the web server. The wiki pages only had to be converted into HTML when they were modified.

How hard can it be? Not very, but it takes an order of magnitude longer than anyone would guess.

Thank you for choosing Quickie. Suggestions are feedback are always welcome. Send your email to Peter Miller and he will endeavour to answer them. Source code patches are, of course, preferable to simple bug reports.

The Quickie Manual

The Quickie Manual contains documentation about all aspects of using and managing your Quickie installation.

NAME

Configuration Guide

DESCRIPTION

The configuration file is used to control various aspects of wiki pages, including their location and their appearance.

The configuration file

The Quickie configuration file is called "`quickie.conf`", but which directory the configuration file is located in is controlled by two factors.

If the `QUICKIE_SYSCONFDIR` environment variable is set, the `quickie.conf` file is looked for in this directory.

Otherwise, the compiled-in default is used. This default is set by using the `./configure --sysconfdir=`*path* command line option when the program was built. This often defaults to `${prefix}/etc`, and given that `$prefix` defaults to `/usr/local`, if you changed nothing, the file you are looking for is called `/usr/local/etc/quickie.conf`, but most Linux distributions (and many sites) override this using the `./configure --sysconfdir=/etc` option, making the path `/etc/quickie.conf` quite likely, as well.

Viewing Current Settings

You can see the current values of all the configuration settings via the Settings page. Unlike regular wiki pages, this page is generated on-the-fly each time, and represents the current state of the wiki engine.

The configuration fields

The `quickie.conf` file is composed of a number of fields. The field name is on the left, then a colon and white space, and then the value. The field name ares **not** case sensitive, but the values may be.

Comments are whole lines which start with a hash (#) character. Blank lines are ignored.

In general, numeric fields can have K or M suffixes to indicate 2**10 and 2**20 multipliers.

Each configuration field is described below.

Anonymous User Edit

The *anonymous-user-edit* field is used to control whether or non anonymous users (users who are not logged in) may edit wiki pages. Defaults to false if not set, *i.e.* users must be logged in.

Chroot

The *chroot* field is a boolean, and is used to control whether or not the server changes it's root path into the *localstatedir* directory before processing requests. This offers a small measure of protection against wily hackers who exploit bugs in Quickie (no security bugs are known, at this time). This field defaults to **false** if not set.

Unfortunately, modern systems now make *chroot*(2) a privileged operation (because it can lead to root escalation) and thus not universally available, which is a what a default of *true* assumes.

Please not that, as a result of the *chroot*(2) system call the *localstatedir* will be displayed on the Current Settings page as "`/`".

You will also need to duplicate the *file*(1) magic number database inside the chroot jail, otherwise it's going to give useless **Content-Type** headers for all non-wiki pages.

Default User Edit

The *default-user-edit* field is used to populate the user permission to edit wiki pages when a new user is created. Defaults to true if not set.

Local State Dir

The *localstatedir* field can be set from three places, whichever is found first: the `QUICKIE_LOCALSTATEDIR` environment variable, or the *localstatedir* field of the `quickie.conf` file, or the `./configure --localstatedir=`*path* command line option at build time.

The line in the `quickie.conf` file could look like this:

```
localstatedir: /var/quickie
```

The usual place is `/var/quickie` by convention, but suit yourself.

Post-Limit

The *post-limit* field can be used to limit the amount of data fed through POST requests to the Quickie server. This can be used to prevent your server from becoming a free file server. Defaults to 100K if not set. A value of zero means *unlimited*.

Recent-Limit

The *recent-limit* field is used to control the number of pages shown in the Recent Changes page. It defaults to 100 if not set.

Show Mod Times

The *show-mod-times* field is use to control the addition of descriptive text showing the time-last-modified (when the wiki text of the page was last modified) and time-last-updated (when the HMTL text of the wiki page was last cached) at the bottom of each page, after the Edit link.

```
Show-Mod-Times: true
```

Defaults to `true` if not set.

Sidebar-Layout

See the Sidebar Guide page for more information.

Style Sheet File

The *style-sheet-name* field controls the name of the file in the `/admin/` directory which contains the CSS (Cascading Style Sheet) for this wiki. defaults to `default.css` if not set. See the Text Formatting Guide for more information.

Umask

The *umask* field controls the file creation mask. It defaults to 022. See *umask*(2) for more information.

NAME

The Quickie Manual

DESCRIPTION

This page links you to the various pages of documentation about all aspects of configuring and running Quickie, the C++ Wiki engine. It is meant to be small, compact, quick and easy to use, but every now and then we all have to resort to reading the destructions.

Creating Your Content

The Text layout and formatting guide will help you to write your wiki pages.

The editing hints page is placed at the bottom of each page you edit, it is a brief version of the complete layout of formatting guide.

The Sidebar Guide

The Sidebar Guide describes how to configure and use the column of text which is usually on the left of the main text on each page.

Managing User Accounts

The Managing User Accounts describes how to create and manager user accounts.

The Configuration Guide

The Configuration Guide describes how to configure Quickie, both file locations and appearances.

Markup Hints

Paragraph breaks are inserted at blank lines.

Using double underscore characters you can use Bold text. For example, `_ _ bold _ _` is rendered in **bold** face.

Using double single quote characters you can use Italic text. For example, `' ' italic ' '` is rendered in *italic* face.

Using double curly brace characters you can use mono-spaced text. For example, `{ { fixed } }` is rendered as the `mono-space` font.

Using square brackets, will give you a hyper-link to another wiki page. For example,

```
[index]
```

would be rendered as index , a link back to the front page of this wiki.

If you want different highlighted text in the hyper-link, separate the wiki page name from the text with a vertical bar (`|`) character. For example,

```
[index|the front page]
```

would be rendered as the front page , a sightly more informative link.

External links place the URL on the left-hand side. Again, the right hand side is optional. For example,

```
[http://quickie.sourceforge.net/|Quickie Home Page]
```

would be rendered as Quickie Home Page .

See the Text Layout and Formatting Guide for detailed and more complete information.

NAME

Text Layout and Formatting Guide

DESCRIPTION

The input text for wiki pages is meant to be very simple. In most cases, will you need to do is paste in the text you want in the page, with a blank line between paragraphs, and you are done.

This page came with the default Quickie installation. Like all wiki pages, this page can be edited. If you find something which needed more explaining please email your improvements to Peter Miller so that they can be included in the next release.

There is also a brief summary of this page made available on all edit pages.

No HTML Markup

Please don’t try to use HTML markup in your wiki pages, it simply will not work. This is intentional; trying to mix wiki markup and HTML markup on the same page is unreasonably difficult, considering what a wiki is supposed to be.

If you know enough HTML to be tempted to use it, maybe you should consider dumping the wiki concept and simply write HTML pages instead.

Comments

You can put comments in your wiki files that will not show in the final page displayed to readers.

```
# This is an example comment.
```

Lines with a hash (#) character at the start of the line are treated as comments.

Headings

To get headings in your text, you need a line which starts with an exclamation mark. It will look something like this:

```
! My Heading
```

The number of exclamation marks controls the style of the heading. Top level headings get one exclamation mark, the next level get two, *etc*.

If your wiki page has a level one heading as its first line, that heading will be used for the page heading, rather than a "pretty version" of the page’s name.

Emphasis

Your can add emphasis to your text with **bold** and *italic* markup.

You get italics by surrounding your text with pairs of single quote (’) characters. For example

```
’’hello’’
```

becomes *hello*.

You get bold by surrounding your text with pairs of underscore (_) characters. For example

```
__hello__
```

becomes **hello**.

Mono-spaced Text

There are two ways to get `mono-spaced text`, sometimes called fixed-width.

Inline text as you see in this paragraph is produced by having two opening braces in a row `{ {` to start the mono-spaced text, and then two closing braces in a row `} }` to finish. Text like this is still subject to the usual **bold `markup` and** *italic `markup` sequences.*

The second way is to have sets of triple braces. Text of the form { { { blah blah } } } appears like this:

```
blah blah
```

within this text, there is no additional markup interpretation. This is useful for inserting software code examples.

Links

You can put links to other wiki pages using `}} square brackets {{` around the name of the wiki page to be linked. For example,

```
[index]
```

is the markup needed to produce this link: index . This link will take you back to the default front page of this wiki.

If you create a link to a wiki page which does not exist (perhaps because you are about to create it) a following question mark will be highlighted. If you click on it, you will be taken to the edit page for the new wiki page.

If you want to have different text than the wiki page title as the text highlighted by the link, you specify the text after the page name, separated by a vertical bar (`|`) symbol. For example,

```
[index|the front page]
```

is the markup needed to produce this link: the front page .

All of the usual markup, such as **bold** and *italic* text, may be used in the right-hand half of this kind of link markup. However, you can't nest links.

Categories

You can divide your wiki pages into categories by using slash `/` characters between the wiki words. For example, the Quickie manual is a category called "manual", so a link to this manual page would look like

```
[manual/markup]
```

and other manual pages can be linked in a similar way. When you do not provide a default heading, the slash turns into a space when the default page heading is calculated.

External Links

You can link to ordinary web pages by using a very similar notation to internal links. For example,

```
[http://quickie.sourceforge.net/]
```

is the markup needed to produce this link: http://quickie.sourceforge.net/ . The text which is highlighted by link is the same as the link itself. You will notice that external links have a little arrow next to them, to tell you that the link is to an web page outside this wiki.

If you would like to change the highlighted text, use the same style of link as for internal links. For example,

```
[http://quickie.sourceforge.net/|Quickie Home Page]
```

is the markup needed to produce this link: Quickie Home Page .

Just as for internal links, you can have additional markup in the right-hand half of this kind of link.

NAME

Style Sheets

DESCRIPTION

You can use CSS (Cascading Style Sheets) to customise the appearance of your wiki pages. Best of all, CSS changes take effect immediately, there is no need to regenerate the HTML from the wiki text.

The default style sheet may be found in the admin/default.css file of your wiki tree. Even though you can view it here, you will have to edit it with a text editor from outside Quickie.

The name of the style sheet file may be altered using the *style-sheet-name* field of the configuration settings.

More information about CSS, including tutorials, can be found on the learning CSS page .

NAME

Sidebar Guide

DESCRIPTION

The sidebar is the column of text which (by default) appears on the left hand side of each wiki page.

You can use the sidebar to provide context for visitors, or direct links to interesting places in your wiki or web site. You can also provide links to external web sites.

Editing

You can go directly to the sidebar here . You can edit the sidebar with the usual edit link at the bottom of the page. If there is a sidebar on the page you are reading (there is one, by default) it will also have a sidebar edit link on it, for direct access.

Additional text

The sidebar includes two links, one to the sidebar as a wiki page, and one to the *edit* link for the sidebar.

At the bottom of the sidebar there is also a list of wiki pages which reference the page text opposite the sidebar. This text does not appear for wiki pages not referenced by any other wiki page. There is no other way to prevent this reference text from appearing.

Configuration

The sidebar location is controlled by the *Sidebar-Layout* field in the configuration file .

```
Sidebar-Layout: left
```

Valid settings are `left`, `right` and `none`. If not set, it defaults to `left`.

The `left` and `right` settings indicate which side of the page text to place the sidebar. The `none` setting means that no sidebar will appear on each page.

The sidebar width is controlled by the *Sidebar-Width* field in the configuration file .

```
Sidebar-Width: 100px
```

Valid settings are an integer followed by "`px`" to indicate a width in pixels, or a number followed by a percent (`%`) sign, to indicate a relative width wide fraction. If not set, it defaults to `100` pixels. The width must be a valid CSS width specification.

Delayed Application

Please note that changes to the configuration file will only take effect when the each wiki page is next saved, because the HTML of each page is cached until it is next edited.

NAME

Managing User Accounts

DESCRIPTION

The Quickie wiki engine includes to concept of *user accounts* to make it possible to manage various permissions.

The most significant permission is the ability to create and edit wiki pages.

The System Configuration File

There are several fields in the System Configuration file which control anonymous user permissions, and also the default user permissions when an account is forst created.

Creating New User Accounts

Users create their own accounts through the admin/new_user wiki page. They will usually do this when prompted by actions which require being logged in.

The permissions granted to new users come from the *default-user-** fields of the System Configuration file.

Login

Users can login to get permissions specific to them rather than the more limited anonymous permissions through the admin/login wiki page.

The permissions granted to anonymous users come from the *anonymous-user-** fields of the System Configuration file.

This is the sidebar. You can put links to useful places here, such as to the Quickie web site .

Back to the first page.

There are several system pages, including things like the current configuration settings.

There is also the Quickie Manual available as a series of wiki pages.

Like all wiki pages, you can edit this text, just follow the "Edit" link you see below, and customise the sidebar for your own uses.