Scratchbox’s Documentation Tools

Janne Langi

janne.langi@movial.fi

Scratchbox's Documentation Tools

by Janne Langi

Revision history

Version:	Author:	Description:
2004-04-29	Langi	Initial version

Table of Contents

Chapter 1. Introduction

Opensource software source packages are usually distributed with documentation which is in some markup language since it can then be easily converted to many other formats (html, pdf, man-pages etc).

The documentation is usually converted at the same time the software is built. Since Scratchbox aims to be a complete development environment where any software can be built without needing to change anything in the original distribution package we need various documentation tools to handle the converting of documentation. Another reason is that installing target and architecture specific tools might not be the best way since they would have to be installed separately for all targets and, for example, when using ARM target the tools are run using emulation or on an ARM device and that might be very slow. Instead, we have the tools in Scratchbox and they are run on the host which is a lot faster.

Chapter 2. Documentation tools

2.1. Obtaining the tools

The documentation tools are available as a prebuilt package (in tar.gz, deb or rpm format) or in a GAR[1] build tree from CVS on Scratchbox website*.

To download the documentation tools as a prebuilt package go to http://www.scratchbox.org/download/.

To download the current CVS version in one package go to http://viewcvs.scratchbox.org/cgi-bin/viewcvs.cgi/ and click on the ’Download tarball’ link at the bottom of the page.

To checkout the GAR tree from CVS you need to have your *static* IP address added to the list of IP addresses from where ssh connections are allowed. When that is done you can checkout the GAR tree with the following command:

```
CVSROOT=:ext:scratchbox.org:/cvsroot/scratchbox CVS_RSH=ssh cvs checkout -P scratchbox
```

> **Note:** Access to the CVS is allowed for Scratchbox developers *ONLY*!

2.2. Installing the tools

Installing is rather easy using the prebuilt packages. If you want you can also build the tools yourself and install those.

> **Note:** You most likely must be root to install the documentation tools because you need write permissions to `/scratchbox/` directory.

You can install the tools from a prebuilt package as follows:

1. Tarball:

   ```
   tar xvf scratchbox-doctools-<version>.tar.gz -C /
   ```

2. Debian package:

   dpkg -i scratchbox-doctools_<version>_i386.deb

3. RPM package:

   rpm -Uvh scratchbox-doctools-<version>.i386.rpm

If you want to use the CVS version of the documentation tools package:

1. Go to the documentation tools directory in the GAR tree

   cd scratchbox/doc_tools/

2. Build and install the tools

   make && make install

The documentation tools are installed into `/scratchbox/doc_tools/` directory. It's added to users' PATH inside scratchbox so one can run the tools by just typing their name. Users usually don't need to run the tools explicitly since the build scripts of software packages do it automatically when needed.

2.3. Included tools

The following is a list of the documentation tools currently in Scratchbox along with short descriptions.

groff

The groff (GNU Troff) software is a typesetting package which reads plain text mixed with formatting commands and produces formatted output. Output can be produced in a number of formats including plain ASCII, HTML and PostScript.

The package contains the traditional UN*X text formatting tools troff, nroff, tbl, eqn, and pic. These utilities, together with the man package, are essential for displaying the online manual pages. A number of other utilities are also included together with several fonts.

Location in GAR: `scratchbox/doc_tools/groff/`

For more info see: groff homepage*

html2text

A command line utility that converts HTML documents into plain text. The program is able to preserve the original positions of table fields, allows you to set the screen width (to a given number of output characters), and accepts also syntactically incorrect input (attempting to interpret it "reasonably").

Location in GAR: `scratchbox/doc_tools/html2text`

For more info see: html2text homepage*

libxml2

libxml2 is the XML C parser and toolkit originally developed for the Gnome project but is usable as a standalone as well.

Location in GAR: `scratchbox/doc_tools/libxml2`

For more info see: libXML2 homepage*

libxslt

Libxslt is the XSLT C library developed for the Gnome project but is usable as a standalone as well. Based on libxml2.

Location in GAR: `scratchbox/doc_tools/libxslt`

For more info see: libXSLT homepage*

openjade

OpenJade is a suite of tools for validating, processing, and applying DSSSL (Document Style Semantics and Specification Language) stylesheets to SGML and XML documents. DSSSL is an ISO standard for formatting SGML (and XML) documents.

Location in GAR: `scratchbox/doc_tools/openjade`

For more info see: OpenJade homepage*

opensp

OpenSP is a library and a set of tools for validating, parsing, and manipulating SGML and XML documents. OpenSP is a part of OpenJade project.

Location in GAR: `scratchbox/doc_tools/opensp`

For more info see: OpenSP homepage*

pfaedit

An outline font editor that lets you create your own postscript, truetype, opentype, cid-keyed, multi-master, cff, svg and bitmap (bdf) fonts, or edit existing ones. Also lets you convert one format to another.

Location in GAR: `scratchbox/doc_tools/pfaedit`

For more info see: PfaEdit homepage*

sgml-base

Utilities to maintain SGML catalog files.

Location in GAR: `scratchbox/doc_tools/sgml-base`

For more info see: SGML-base homepage*

sgmltools-lite

Consists of the easy-to-use front-end, a large number of processing backends, and some custom stylesheets.

Location in GAR: `scratchbox/doc_tools/sgmltools-lite`

For more info see: SGMLtools-lite homepage*

tetex

teTeX is a complete TeX distribution for UNIX compatible systems. TeX is a typesetting system.

Location in GAR: `scratchbox/doc_tools/tetex`

For more info see: teTeX homepage*

Chapter 3. Adding documentation tools

Scratchbox uses the GAR build system [1]. It's a mechanism for automating the compilation and installation of third-party source code. It appears in the form of a tree of directories containing Makefiles and other ancillary bookkeeping files (such as installation manifests and checksum lists).

> **Note:** Only Scratchbox developers are allowed to access the CVS repository. The following instructions apply for adding new tools to a local GAR tree as well, just ignore the adding to CVS and copying files to Scratchbox webserver.

3.1. Obtaining the GAR tree

To download the current CVS version in one package go to http://viewcvs.scratchbox.org/cgi-bin/viewcvs.cgi/ and click on the 'Download tarball' link at the bottom of the page. Extract the tarball to a appropriate directory.

To checkout the GAR tree from CVS you need to have your *static* IP address added to the list of IP addresses from where ssh connections are allowed. Once that is done you can checkout the GAR tree with the following command:

CVSROOT=:ext:scratchbox.org:/cvsroot/scratchbox CVS_RSH=ssh cvs checkout -P scratchbox

> **Note:** Access to the CVS repository is allowed for Scratchbox developers *ONLY*!

You should now have the whole Scratchbox GAR tree in `scratchbox/` directory. Documentation tool directory is `scratchbox/doc_tools/`.

3.2. Adding new tools

Adding new documentation tools to Scratchbox is fairly straightforward:

1. Obtain source package for the software.
2. Create a directory for the new tool in `scratchbox/doc_tools/` directory.
3. Copy the source package to `scratchbox/doc_tools/<newtool>/files/` directory.

4. Create a new GAR makefile `scratchbox/doc_tools/<newtool>/Makefile`. The GAR makefile must have some specific variables for making it work correctly. At least the following variables should be there.
   - GARNAME defines the package's name.
   - GARVERSION defines the package's version.
   - CATEGORIES defines the package's category. In this case it is 'doc_tools'.
   - DISTFILES defines the source package that contains the source. This is usually combination of GARNAME, GARVERSION and source package's extension (usually tar.gz or tar.bz2).
   - LIBDEPS tells which libraries this package depends. These packages are compiled before compiling this package.
   - DEPENDS defines packages that must be compiled before this package can be compiled properly. This might contain tools that are required during building process of this package.
   - DESCRIPTION contains description of this package.
   - CONFIGURE_ENV defines some Scratchbox specific variables. For example Scratchboxs perl and python should be defined here if package uses them.
   - CONFIGURE_ARGS defines options that are passed to package's configure script. This should contain at least the install prefix.
   - BUILD_ARGS defines arguments that are used to build this package. This should contain at least Scratchbox specific LDFLAGS, CC and PERL or PYTHON paths if they are used.
   - INSTALL_ARGS contains arguments that are used when package is installed. Usually they are the same as BUILD_ARGS.
   - CONFIGURE_SCRIPTS defines the configuration script. Usually the package's own configure script.
   - BUILD_SCRIPT defines the build script. Usually the package's own makefile.
   - INSTALL_SCRIPT defines the install script. Usually the package's own makefile.
   - In addition to package specific options packages GAR makefile should include correct GAR category file. In Scratchbox this is done by including '../category.mk' file after variable definitions. See the Makefiles in Scratchbox GAR tree for examples.
5. Type **make checksums** in packages directory to generate checksum information for package.
6. Package is now ready to be compiled. Compiling and installing package can be done with **make install** command.

   Note: If the package does not compile properly check that you have necessary library dependencies etc. in the Makefile. You might also have to add some Scratchbox specific configuration and build options.

7. Once the package builds cleanly and you have verified that it works you can add it to Scratchbox CVS repository. You must also copy the source package to `/work/files/sbox-files/` directory on Scratchbox webserver so that it will be availble for others. Make sure the file permissions allow a read access for others as well!

Note: This step applies to Scratchbox developers only!

References

[1] *GAR build system (http://www.lnx-bbc.org/garchitecture.html)* .

[2] *Concurrent Versions Systems website (http://www.cvshome.org)* .

[Scratchbox website] *http://www.scratchbox.org/* .

[groff homepage] *http://www.gnu.org/software/groff/groff.html* .

[html2text homepage] *http://userpage.fu-berlin.de/~mbayer/tools/html2text.html* .

[libXML2 homepage] *http://xmlsoft.org* .

[libXSLT homepage] *http://xmlsoft.org/XSLT* .

[OpenJade homepage] *http://openjade.sourceforge.net* .

[OpenSP homepage] *http://openjade.sourceforge.net* .

[PfaEdit homepage] *http://fontforge.sourceforge.net/* .

[SGML-base homepage] *http://packages.debian.org/unstable/text/sgml-base* .

[SGMLtools-lite homepage] *http://sgmltools-lite.sourceforge.net* .

[teTeX homepage] *http://www.tug.org/teTeX* .