Shell Script Reference

tpt-script-crf

Generated by Doxygen 1.2.11.1

Jan 13, 2003

Contents

Chapter 1

Voyant TechPubs Tools Main Page

This is a Voyant internal module.

Chapter 2

Voyant TechPubs Tools Module Index

2.1 Voyant TechPubs Tools Modules

Here is a list of all modules:

Chapter 3

Voyant TechPubs Tools File Index

3.1 Voyant TechPubs Tools File List

Here is a list of all files with brief descriptions:

Chapter 4

Voyant TechPubs Tools Module Documentation

4.1 Low-Level Drivers for CMS

Modules

- XPT Driver for CMS

  Defines the low-level driver code for the XPT chips in the CMS platform.

4.1.1 Detailed Description

Warning:

Do NOT use these drivers if you are programming at the application level or at a level that uses BAPI/SAPI.

4.2 XPT Driver for CMS

Defines the low-level driver code for the XPT chips in the CMS platform.

Warning:

You should NOT be calling this driver directly if you are programming at the application level or at a level that uses BAPI/SAPI.

4.3 TechPubs Tools

Tools for technical publications departments.

Files

- file 00_build_tp_tools.b

  The control script for all tp_tools.

- file 20_cp_com_files.b

  Copies commonly used files to the directory where this was called.

- file 30_tp_tools.b

  Calls a couple of other shell scripts which do CVS checkout and then eventually a doxygen build.

- file 31_perl.b

  Checks out of CVS the appropriate code files.

- file 31_script.b

  Checks out of CVS the appropriate code files.

- file 32_perl.b

  Runs a wrapper function for Doxygen on the selected project file.

- file 32_script.b

  Runs a wrapper function for Doxygen on the selected project file.

- file 35_gen_dox.b

  Calls doxygen with the specified input directory and then copies generated files where they need to go for later processing.

- file 35_gen_tree.b

  Calls tree_js_2_script.pl for the specified project.

- file 40_latex_build.b

  Prepares the Latex template files and then generates ultimately PDF files from the Doxygen Latex output.

- file 45_latex_gen.b

  Calls the requisite programs for ultimately generating a PDF file from Latex that came from doxygen.

- file 50_nav_update.b

  Recreates the entire master TOC and index for the system.

- file 55_nav_cp.b

  Copies previously generated mini-index and mini-TOC files to a known location.

- file 55_nav_gen.b

  Creates the mini-TOC and mini-index navigation files for the specified project using information extracted from two input files.

- file 56_nav_index.b

  Generates master index pages for all found mini-index files in the specified directory.

- file 56_nav_script.b

  Generates a master TOC script file appropriate for all found mini-TOC files in the specified directory.

Modules

- Unix Shell Scripts

  Script files that control building and generating the system.

- Latex Tool

  Provides minor updating to LaTex files.

- Table of Contents Tool

  Generates master tree files from previously generated tree files to get master table of contents.

- Doxygen Filter Tools

  Provides input filter to "fake-out" Doxygen into thinking it has C code.

- Navigation Tools

  Swaps out navigation, creates temporary TOC files and index_ files.

- Indexer Tools

  Creates a comprehensive index from previously generated index_ files.

4.3.1 Detailed Description

Tools for technical publications departments.

4.4 Indexer Tools

Creates a comprehensive index from previously generated index_ files.

4.5 Navigation Tools

Swaps out navigation, creates temporary TOC files and index_ files.

4.6 Doxygen Filter Tools

Provides input filter to ”fake-out” Doxygen into thinking it has C code.

4.7 Table of Contents Tool

Generates master tree files from previously generated tree files to get master table of contents.

4.8 Latex Tool

Provides minor updating to LaTex files.

4.9 Unix Shell Scripts

Script files that control building and generating the system.

Files

- file 00_build_tp_tools.b

  The control script for all tp_tools.

- file 20_cp_com_files.b

  Copies commonly used files to the directory where this was called.

- file 30_tp_tools.b

  Calls a couple of other shell scripts which do CVS checkout and then eventually a doxygen build.

- file 31_perl.b

  Checks out of CVS the appropriate code files.

- file 31_script.b

  Checks out of CVS the appropriate code files.

- file 32_perl.b

  Runs a wrapper function for Doxygen on the selected project file.

- file 32_script.b

  Runs a wrapper function for Doxygen on the selected project file.

- file 35_gen_dox.b

  Calls doxygen with the specified input directory and then copies generated files where they need to go for later processing.

- file 35_gen_tree.b

  Calls tree_js_2_script.pl for the specified project.

- file 40_latex_build.b

  Prepares the Latex template files and then generates ultimately PDF files from the Doxygen Latex output.

- file 45_latex_gen.b

  Calls the requisite programs for ultimately generating a PDF file from Latex that came from doxygen.

- file 50_nav_update.b

  Recreates the entire master TOC and index for the system.

- file 55_nav_cp.b

  Copies previously generated mini-index and mini-TOC files to a known location.

- file 55_nav_gen.b

  Creates the mini-TOC and mini-index navigation files for the specified project using information extracted from two input files.

- file 56_nav_index.b

  Generates master index pages for all found mini-index files in the specified directory.

- file 56_nav_script.b

  Generates a master TOC script file appropriate for all found mini-TOC files in the specified directory.

4.9.1 Detailed Description

Script files that control building and generating the system.

Chapter 5

Voyant TechPubs Tools File Documentation

5.1 00_build_tp_tools.b File Reference

The control script for all tp_tools.

5.1.1 Detailed Description

The control script for all tp_tools.

This is used when building from scratch. It checks out all other scripts and tools needed from CVS, and then proceeds to call those scripts in the proper order.

This is typically only run the first time. Thereafter, the 00_short.b can be more efficient or even calling individual scripts.

Note:
This file needs to be updated to support other directories.

Author:
Glenn C. Maxey

Definition in file 00_build_tp_tools.b.

5.2 20_cp_com_files.b File Reference

Copies commonly used files to the directory where this was called.

5.2.1 Detailed Description

Copies commonly used files to the directory where this was called.

This is mostly used to make sure the CSS is up to date. However, other common doxygen files could also be updated.

Limitations and Caveats:

This needs to be updated to reflect the directory structure of the person who implements it (e.g., non-Voyant, non-techpubs).

Author:

Glenn C. Maxey

Definition in file 20_cp_com_files.b.

5.3 30_tp_tools.b File Reference

Calls a couple of other shell scripts which do CVS checkout and then eventually a doxygen build.

5.3.1 Detailed Description

Calls a couple of other shell scripts which do CVS checkout and then eventually a doxygen build.

True, in this simple example, nearly all of the functionality of the shell scripts (eventually) called by this one (e.g., 31∗.b and 32∗.b, and deeper down 35∗.b and 55∗.b) could be placed into this file.

However, this is only a simple example. In my real world, a documentation suite includes several doxygen projects. Plus, I have many different documentation suites covering different aspects of our code and aimed at different (mostly internal) audiences.

In my environment, a given documentation suite has multiple 30∗.b files, which call associated 31∗.b and 32∗.b files. (Lately, I've been wrapping yet another 30_build_all.b around all of the 30∗.b files which can be hooked into the software build process.)

With respect to the 31∗.b and 31∗.b separation, I found it beneficial to compartmentalize the CVS checkout procedures (31∗.b) from the actual doxygen calls (32∗.b), because it facilitated the maintenance/tweak mode better. I could focus on one project and one area of a project to get it correct when doing partial system builds.

Moreover, I found many operations that either were repeated frequently within the build process or were being forgotten whenever I tried to do a partial build. These were pulled out into 35∗.b and 55∗.b files. Yes, in some cases, certain perl programs are called multiple times on the same set of information. The run-time inefficiencies and delays (seconds) in repeating operations was deemed acceptable in comparison to the alternative, which was a compromise in the integrity of the system (readily evident in the TOC or index) if I forgot a step in the partial build process.

Note:

This file can be used as a template for other code directories that Doxygen is being run against. Make sure than any new 30 files are added to the 00 script files, in addition to appropriate entries in the other script files.

Author:

Glenn C. Maxey

Definition in file 30_tp_tools.b.

5.4 31_perl.b File Reference

Checks out of CVS the appropriate code files.

5.4.1 Detailed Description

Checks out of CVS the appropriate code files.

For the tp_tools project where I eat-my-own-dog-food, the CVS checkout captured in this file comes either a bit late or is needlessly repeated.

Why? Because I need to have the b script files and their associated perl tools already checked out and up to date before I run them elsewhere.

If this were a true code project, however, the 31∗.b file would contain a CVS checkout of code files that the tp_tools would then process. Moreover, sometimes there are additional operations that my tools perform immediately after a checkout, some of them related to the nature of the code and others related to finding and extracting code elements in order to create temporary files that the associated 32∗.b would process.

Note:

This file can be used as a template for other code directories that Doxygen is being run against. Make sure than any new 30 files are added to the 00 script files, in addition to appropriate entries in the other script files.

Author:

Glenn C. Maxey

Definition in file 31_perl.b.

5.5 31_script.b File Reference

Checks out of CVS the appropriate code files.

5.5.1 Detailed Description

Checks out of CVS the appropriate code files.

For the tp_tools project where I eat-my-own-dog-food, the CVS checkout captured in this file comes either a bit late or is needlessly repeated.

Why? Because I need to have the b script files and their associated perl tools already checked out and up to date before I run them elsewhere.

If this were a true code project, however, the 31∗.b file would contain a CVS checkout of code files that the tp_tools would then process. Moreover, sometimes there are additional operations that my tools perform immediately after a checkout, some of them related to the nature of the code and others related to finding and extracting code elements in order to create temporary files that the associated 32∗.b would process.

Note:
This file can be used as a template for other code directories that Doxygen is being run against. Make sure than any new 30 files are added to the 00 script files, in addition to appropriate entries in the other script files.

Author:
Glenn C. Maxey

Definition in file 31_script.b.

5.6 32_perl.b File Reference

Runs a wrapper function for Doxygen on the selected project file.

5.6.1 Detailed Description

Runs a wrapper function for Doxygen on the selected project file.

This calls the 35_gen_dox.b function which compartmentalizes several steps that are repeated for every single (doxygen) project. When they are placed in their own file, it makes the 32∗.b files easier to maintain across projects, because they are project specific.

Note:

This file can be used as a template for other code directories that Doxygen is being run against. Make sure than any new 30 files are added to the 00 script files, in addition to appropriate entries in the other script files.

Author:

Glenn C. Maxey

Definition in file 32_perl.b.

5.7 32_script.b File Reference

Runs a wrapper function for Doxygen on the selected project file.

5.7.1 Detailed Description

Runs a wrapper function for Doxygen on the selected project file.

This calls the 35_gen_dox.b function which compartmentalizes several steps that are repeated for every single (doxygen) project. When they are placed in their own file, it makes the 32∗.b files easier to maintain across projects, because they are project specific.

Note:
This file can be used as a template for other code directories that Doxygen is being run against. Make sure than any new 30 files are added to the 00 script files, in addition to appropriate entries in the other script files.

Author:
Glenn C. Maxey

Definition in file 32_script.b.

5.8 35_gen_dox.b File Reference

Calls doxygen with the specified input directory and then copies generated files where they need to go for later processing.

5.8.1 Detailed Description

Calls doxygen with the specified input directory and then copies generated files where they need to go for later processing.

Parameters:

dox_path A portion of the path used to distinguish the project. The final destination is assumed to be cref_<dox_path>. In addition, it is assumed that a related <dox_path>.dox configuration file for doxygen exists in the directory from where this was called.

master_nav The name of an master HTML file that has common HTML fragments which are to be inserted in all generated HTML pages.

master_proj The name of the project_toc.txt file that defines all directories and their associated names, pdf files, and document numbers.

Limitations and Caveats:

This calls both 55_nav_gen.b and 56_nav_script.b with the information passed into this script. These were two steps that were often forgotten. Doxygen does not generate HTML based on the master navigation file, so 55_nav_gen.b forces the insertion of the standard tags.

Likewise, this script by itself does not make sure that the master TOC script file is updated. As such, after copying over the mini-index and mini-TOC files to a known location, this goes ahead regenerates the master TOC.

It does not regenerate the master index, because that is much more time-consuming.

Author:

Glenn C. Maxey

Definition in file 35_gen_dox.b.

5.9 35_gen_tree.b File Reference

Calls tree_js_2_script.pl for the specified project.

5.9.1 Detailed Description

Calls tree_js_2_script.pl for the specified project.

Parameters:

dox_path A portion of the path used to distinguish the project. The final destination is assumed to be cref_<dox_path>.

master_nav NOT USED. The name of an master HTML file that has common HTML fragments which are to be inserted in all generated HTML pages.

This is intended for doxygen projects that were previously generated. This functionality has since been built into 35_gen_dox.b, so this script doesn't have to be called as often.

It generates the mini-TOC script file which is later included in the master TOC file.

Limitations and Caveats:

This does not use the master_nav file, but by keeping the same input parameters as the 35_gen_dox.b, it is easier to call in scripts.

In addition, this only works on doxygen directories and assumes that they begin with cref_<dox_path>

Author:

Glenn C. Maxey

Definition in file 35_gen_tree.b.

5.10 40_latex_build.b File Reference

Prepares the Latex template files and then generates ultimately PDF files from the Doxygen Latex output.

5.10.1 Detailed Description

Prepares the Latex template files and then generates ultimately PDF files from the Doxygen Latex output.

The 40∗.b file is project specific and can contain multiple calls to the 45∗.b file. The 45∗.b file contains commands specific to doxygen's implementation of generating Latex and PDF files. Its steps were compartmentalized into its own files to facilitate maintenance and to make the project specific 40∗.b files easier to understand.

Note:
This file needs to be updated to support other code directories.

Author:
Glenn C. Maxey

Definition in file 40_latex_build.b.

5.11 45_latex_gen.b File Reference

Calls the requisite programs for ultimately generating a PDF file from Latex that came from doxygen.

5.11.1 Detailed Description

Calls the requisite programs for ultimately generating a PDF file from Latex that came from doxygen.

Parameters:

lat_path The shortened version of the directory name where the latex files can be found. It is assumed that zlx_<lat_path> exists in a subdirectory from where this was called.

In order to get a PDF from a doxygen project, you have to make sure that doxygen generates latex files. Then you have to call various commands to get latex to generate a postscript and then PDF file. The steps are the same for all projects.

Rather than polluting the higher-level scripts that are essentially the same for all doxygen projects, this separates the details from the application-specific projects. Thus, if doxygen changes how PDF files are generated, this can be modified in one place to handle it.

Limitations and Caveats:

This is intended to be called from a 40_latex_gen.b file that determines the directory name.

Author:

Glenn C. Maxey

Definition in file 45_latex_gen.b.

5.12 50_nav_update.b File Reference

Recreates the entire master TOC and index for the system.

5.12.1 Detailed Description

Recreates the entire master TOC and index for the system.

This calls either 55_nav_gen.b or 55_nav_cp.b to make sure that all mini-TOC and mini-index files have been generated for all specified sub-projects and have been copied into the proper location for subsequent process by 56_master_index.b and 56_master_-script.b.

In addition, the 55_nav_gen.b calls make sure that all HTML pages are consistent with respect to their top/bottom navigation, copyright notices, and other HTML internals (e.g., CSS callout, java applet).

Limitations and Caveats:

This file need to be maintained for each bigger project whenever sub-projects are added or removed. Also, this script determines which master files are used for the navigation.

Author:

Glenn C. Maxey

Definition in file 50_nav_update.b.

5.13 55_nav_cp.b File Reference

Copies previously generated mini-index and mini-TOC files to a known location.

5.13.1 Detailed Description

Copies previously generated mini-index and mini-TOC files to a known location.

Parameters:

nav_path The starting location to look for subdirectories. Generally, this is "doc_-publish/". It was broken out of the full path, because the sim_name parameter was needed for other purposes.

sim_name The simplified name of the destination project. This parameter is used to locate both the source HTML files (as in <nav_path><sim_name> but also to name generated mini-index and mini-TOC files appropriately for the project.

master_nav The name of an master HTML file that has common HTML fragments which are to be inserted in all generated HTML pages.

master_proj The name of the project_toc.txt file that defines all directories and their associated names, pdf files, and document numbers.

This does NOT call the voyant_nav.pl perl program. However, it should use the same input parameters as 55_nav_gen.b so that 55_nav_cp.b can be interchanged with it in a calling 50_nav_update.b file.

This script needs to copy the previously generated mini files to a known location for subsequent processing. It must name them uniquely for the project, which is why the <sim_name> is required to be broken out.

In addition, getting individual project directories updated with the latest CSS was error prone. This handles it while processing, so that it doesn't get forgotten.

Author:

Glenn C. Maxey

Definition in file 55_nav_cp.b.

5.14 55_nav_gen.b File Reference

Creates the mini-TOC and mini-index navigation files for the specified project using information extracted from two input files.

5.14.1 Detailed Description

Creates the mini-TOC and mini-index navigation files for the specified project using information extracted from two input files.

Parameters:

nav_path The starting location to look for subdirectories. Generally, this is "doc_-publish/". It was broken out of the full path, because the sim_name parameter was needed for other purposes.

sim_name The simplified name of the destination project. This parameter is used to locate both the source HTML files (as in <nav_path><sim_name> but also to name generated mini-index and mini-TOC files appropriately for the project.

master_nav The name of an master HTML file that has common HTML fragments which are to be inserted in all generated HTML pages.

master_proj The name of the project_toc.txt file that defines all directories and their associated names, pdf files, and document numbers.

This calls the voyant_nav.pl perl program. It processes the files specified in the <nav_-path><sim_name> directory. It requires the master_nav file and the master_proj file.

The master_nav file is an HTML file that contain tags which are to be placed in every single HTML file in the directory.

The voyant_nav.pl perl program generates a mini-index and mini-TOC file. When finished, this script needs to copy them to a known location for subsequent processing. It must name them uniquely for the project, which is why the <sim_name> is required to be broken out.

In addition, getting individual project directories updated with the latest CSS was error prone. This handles it while processing, so that it doesn't get forgotten.

Author:

Glenn C. Maxey

Definition in file 55_nav_gen.b.

5.15 56_nav_index.b File Reference

Generates master index pages for all found mini-index files in the specified directory.

5.15.1 Detailed Description

Generates master index pages for all found mini-index files in the specified directory.

This makes some assumptions about where template files are found and where to process them.

The template file for an individual index page should be found in /rtfm/techpubs/web_files/voyant_master_index.html If this is not the directory or file you want to use, you can change it here in one place. Generally, all of my projects can get by with only one template file for the index.

This script copies the index file into all possible letters. Then it calls the voyant_indexer.pl with the same template file and the destination directory. The voyant_indexer.pl will overwrite any default index letter files with what it generates. However, you need to have all default files for all letters in order for the navigation in the index to work.

Depending upon the number of mini-index files that are in the specified directory, voyant_indexer.pl can churn away for more than just a few minutes.

When finished, this script copies the generated index files into the destination for publishing.

Author:
Glenn C. Maxey

Definition in file 56_nav_index.b.

5.16 56_nav_script.b File Reference

Generates a master TOC script file appropriate for all found mini-TOC files in the specified directory.

5.16.1 Detailed Description

Generates a master TOC script file appropriate for all found mini-TOC files in the specified directory.

This script calls voyant_mt_app.pl (master tree application). This looks at all tree_-<...>.script files in the specified project file. If a tree file doesn't exist, a warning is output. It creates a master tree script file that references the mini-TOC tree script files.

In addition, it creates master_tree script files for each of the mini-TOC files, whereby the paths to the individual topics are resolved for where they actually reside with respect to the top-level publication directory.

In other words, the mini-TOC tree files are intended to work from where they were generated. When plugged into a bigger system, the master tree navigation files typically reside at a lever higher.

This two-stage process has benefits, most of them along the lines of how a modular system can be generated from different TOC files. A given directory can be symbolically linked in several different places and be referenced in several different project files. As such, information can be re-used efficient with accurate master TOCs for the project.

Author:
Glenn C. Maxey

Definition in file 56_nav_script.b.

5.17 voyant_common.h File Reference

5.18 voyant_tp_tools.h File Reference

5.19 voyant_tp_tools.h File Reference

Index